

# Índice

1 Introducción.....	2
1.1 Objetivo.....	2
1.2 Planteamiento del problema.....	2
1.3 Hipótesis.....	3
1.4 Motivación.....	3
1.5 Descripción del documento.....	3
2 Marco teórico.....	4
2.1 Medición de la corriente por medio del sensor.....	4
2.1.1 Física acerca del sensor.....	4
2.1.2 Protecciones y manejo del sensor.....	8
2.1.3 Explicación de uso sensor e implementación.....	11
2.2 Dispositivos de IoT.....	11
2.2.1 Microcomputadoras (Raspberry Pi 3 B).....	11
2.2.2 Microcontroladores (Arduino nano).....	14
2.2.3 Manejo de señales digitales y análogicas (ADC).....	18
2.2.4 Regulador de voltaje.....	19
2.3 Arquitectura de los sistemas IoT.....	20
2.3.1 Dispositivo embebido con internet de las cosas (IoT).....	20
2.3.2 Web Service (RESTful).....	24
2.3.3 Uso de base de datos.....	29
2.3.4 Sistema de visualización de datos.....	31
3 Desarrollo.....	32
3.1 Dispositivo embebido.....	32
3.1.1 Sensores comerciales.....	34
3.1.2 Sensor SCT-013-03.....	36
3.1.3 Reguladores de voltaje.....	37
3.1.4 Circuito del sistema.....	39
3.1.5 Cálculos de la corriente eléctrica.....	42
3.1.6 Interconexión Arduino-Raspberry.....	43
3.1.7 Software del sistema.....	45
3.1.8 Comunicación externa para envío de datos.....	46
3.2 Sistema de adquisición de datos.....	47
3.2.1 Base de datos con MongoDB.....	47
3.2.2 Algoritmos de búsqueda.....	48
3.2.3 Protocolos de respuesta.....	49
3.3 Sistema de visualización.....	49
3.3.1 Sistema gráfico de los datos.....	49
3.3.2 Búsqueda y filtrado de datos.....	54
3.3.3 Robustez de la página.....	59
4 Pruebas y resultados.....	60
4.1 Comparación con sistema comerciales.....	60
4.1.1 Comparación con un medidor de corriente de gancho comercial.....	60
4.1.2 Comparación con el medidor puesto por CFE.....	60
4.2 Pruebas con el sistema en ambiente real.....	60
4.2.1 Prueba en un hogar.....	60
4.2.2 Prueba en una estación de carga.....	60

4.3 Análisis de consumo durante un periodo.....	60
5 Conclusiones y trabajo futuro.....	60
5.1 Conclusiones.....	60
5.2 Trabajo futuro.....	60
6 Bibliografia.....	60

# 1 Introducción

parrafos de introducción y describir lo que se va a hacer

Dar a conocer las aplicaciones de IoT y la versatilidad que se le puede dar en la sociedad.

## 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, el consumo eléctrico a través de una computadora o un dispositivo móvil por medio de una interfaz gráfica.

Para lograr este objetivo se realizaron las siguientes actividades:

- Desarrollar un dispositivo embebido para medir el consumo de energía eléctrica.
- Enviar 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 la problemática del consumo eléctrico en hogares o en establecimientos radica en el uso excesivo de la energía eléctrica que provee la comisión federal de electricidad, ya que personas que utilizan este servicio realizan muchas quejas acerca de los precios elevados que aparecen en los recibos bimestrales que se les envía por parte de la compañía, asegurando que su consumo eléctrico esta entre los rangos establecidos por la comisión federal de electricidad, pero sin tener pruebas reales que

lo comprueben.

En la actualidad, el consumo eléctrico excesivo ha representado un problema dentro de la sociedad, entre ellos dos problemas en específico que se relacionan entre sí y que su impacto hacia las personas es muy considerable. Estos problemas son los relacionados a temas económicos y ambientales.

En México, los problemas en cuestión de economía se debe a que la compañía encargada del suministro eléctrico realiza el cobro bajo un subsidio. En la comisión federal de electricidad, se encarga actualmente del suministro de energía eléctrica en el país y mantiene unas tarifas que se rigen dependiendo de la zona de México en la que habites, por una de las dos tarifas que existen en el país: la primera la tarifa subsidiada que se aplica dentro de la temperatura media mensual mínima en verano, y la segunda que rebasa el promedio de la media mensual.

Los problemas ambientales son un de gran impacto en el medio en el que vivimos, pues en algunos estados de la república mexicana se tienen plantas termoeléctricas generan gases como lo son CO<sub>2</sub>, el dióxido de azufre y óxido nitroso. Estos gases dañan a grandes rasgos la capa de ozono. El abuso de la generación de estos gases se debe a la demanda y uso excesivo de la energía eléctrica en zonas residenciales e industriales

## 1.3 Hipótesis

Se puede desarrollar un dispositivo embebido con el Internet de las cosas de fácil uso, que le permita a los usuarios consultar por medio una interfaz gráfica desde la computadora o un dispositivo móvil el consumo eléctrico dentro de sus residencias para que los usuarios tomen acciones preventivas en caso de consumo excesivo.

## 1.4 Motivación

## 1.5 Descripción del documento

Este trabajo se encuentra organizado de la siguiente manera: en el capítulo 2 se presentan los aspectos básicos que constituyen al sistema completo, desde como se encuentra constituido el dispositivo para llevar acabo las lecturas y sean enviadas al sistema de consulta, y este a su vez que procesos debe llevar acabo para que la página web realice la visualización de los datos a consultar.

## 2 Marco teórico

### 2.1 Medición de la corriente por medio del sensor

Para poder medir la corriente se hace uso de un 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 varios tipos de sensores, entre ellos sensores que muestran su resultado según la cantidad de electricidad, los sensores que dependiendo de la presión cambian su resultado, sensores que miden la cantidad de fuerza ejercida, o sensores que leen el comportamiento de un fenómeno químico y transformarlos en un valor numérico que representativo.

#### 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 detecta son de tipo digital y de tipo analógico:

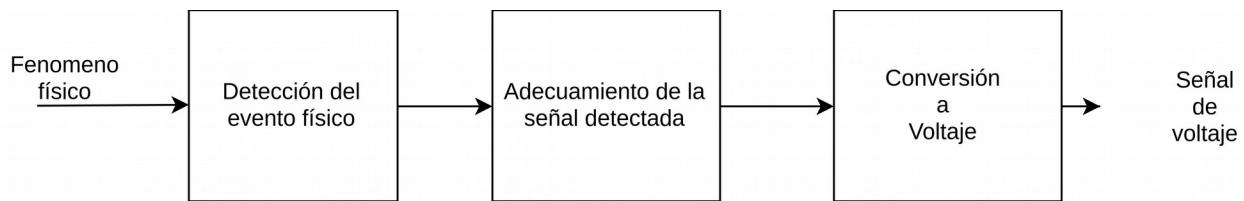
- La señal digital es aquella que solo tiene un número finito de valores posibles (Torrente, 2013). Estos valores se les conoce como variables discretas porque toma solo un valor que pertenecen al rango de voltaje que un microcontrolador puede leer.
- La señal analógica es aquella que tiene infinitos 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 debe ser proporcional al valor físico.
- Acondicionamiento: la salida eléctrica que genera el sensor es acondicionada por un sistema para obtener el voltaje de la señal

obtenida.

- 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*

El uso de los sensores eléctricos se aplican comúnmente para el monitoreo de sistemas en el exterior, y de esa forma obtener información del medio. Por ejemplo medir la distancia, fuerza, corriente, luminosidad. Para ello 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.

Ejemplos de algunos descriptores estáticos son:

- Rango: valores mínimos y los valores máximos que pueden tener la entrada y la salida del sensores.
- 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 en llegar a la mitad de la medición final.
- Tiempo de subida: tiempo en el que el sensor alcanza a llegar a la medición final.
- 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)<sup>1</sup> 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*

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

- Dentro del sensor de encuentran el transformador (Avelino, 2001), que es un embobinado de cable de cobre que generan un campo magnético al colocar un flujo de electricidad.
- El transformador de corriente genera una carga eléctrica en el devanado

---

<sup>1</sup> Imagen tomada de <http://www.directindustry.es/prod/honeywell-safety-and-productivity-solutions/product-12365-306124>.

secundario, el cual es proporcional a la carga que se genera en el devanado primario (Figura 2.3).

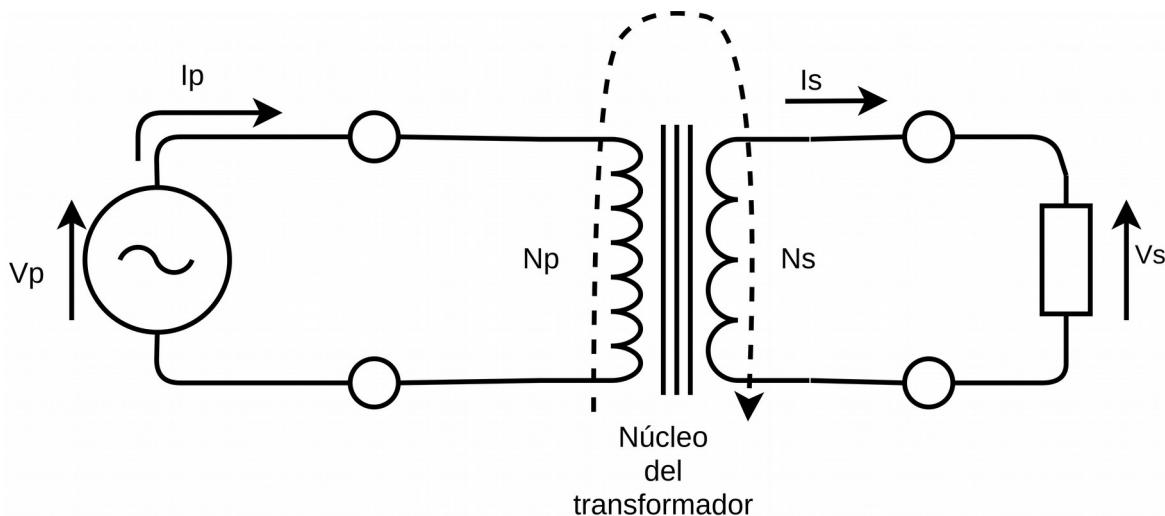
- El número de espiras que tiene cada devanado tiene que ajustarse de tal manera que cuando se genere un campo magnético por el flujo de corriente en el primario, en el secundario se genere un flujo de corriente muy aproximado al valor primero.
- La fórmula con la que se relaciona el número de espiras con el voltaje y el flujo de corriente se muestra en la fórmula 2.1<sup>2</sup>:

$$\frac{Is}{Ip} = \frac{Vp}{Vs} = \frac{Np}{Ns}$$

*Fórmula 2.1 Relación de intensidad con el número de espiras*

En donde:

- $Is$  es la intensidad del secundario
- $Ip$  es la intensidad del primario
- $Vp$  es la diferencia de potencial del primario
- $Vs$  es la diferencia de potencial del secundario
- $Np$  es el número de espiras en el primario
- $Ns$  es el número de espiras en el secundario

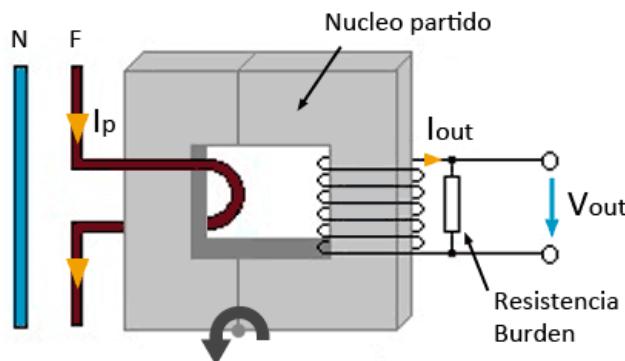


*Figura 2.3 Diagrama funcional de un transformador*

---

<sup>2</sup> Avelino Pérez, Pedro (2001). Transformadores de distribución: teoría, cálculo, construcción y pruebas. Editorial Reverte

- 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.
- La bobina que tiene colocada el sensor esta el imán que funciona como núcleo para detectar el flujo de corriente convirtiéndolo en campo magnético.
- Si no existe un flujo de corriente no puede generarse un cambio en el campo magnético y la bobina que mide ese cambio 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<sup>3</sup>, pues en el transformador que tiene dentro, en la parte del secundario no se debe quedar abierto, ya que las corrientes que se generan podrían dañar todo el circuito.



*Figura 2.4 Conexión de la resistencia de carga*

### 2.1.2 Protecciones y manejo del sensor

El sensor necesita estar acondicionado para al momento de utilizarlo en conjunto de un microcontrolador. Para ello se tiene que adaptar para su conexión de manera correcta y obtener una lectura. Esto por medio de su

<sup>3</sup> Recuperado de <https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>

conversión de campo magnético a corriente por medio del proceso de inducción del transformador.

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

La salida en intensidad de corriente utiliza una resistencia de carga, que es básicamente una resistencia de carga de 33 ohm para tener el valor máximo que acepta un transformador de corriente sin superar su tope o que tenga el valor mínimo.

Un microcontrolador no detecta tal cual tensiones negativas, por lo cual se utiliza un amplificador operacional. El operador 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 sumadores analógicos (Figura 2.5)<sup>4</sup>.

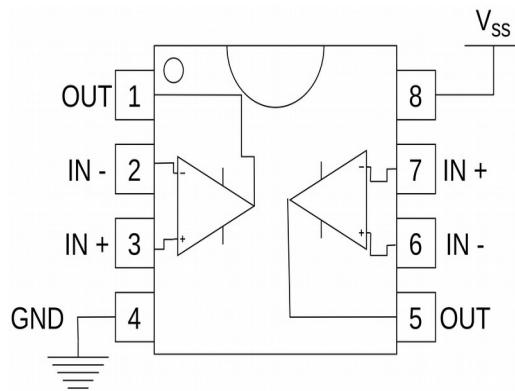


Figura 2.5: Integrado LM358

El rango que genera de su salida es 0V a 1V, que es un valor máximo del sensor pues 1V equivale a 30A, además lo accesible para el microcontrolador que se utilice. A esta variación de voltaje se le conoce como pico, que es el valor máximo del voltaje, tanto positivo como negativo (Jin, Wang, Xu and Zhou, 2009). Un microcontrolador obtiene las lecturas de 0 a 5 V como lo es un PIC o un arduino.

Cuando la medición es de voltaje en alterna se tiene que es un voltaje RMS (root mean square) es el valor del voltaje en alterna que produce el mismo

4 Recuperado de <https://www.st.com/en/amplifiers-and-comparators/lm358.html>

efecto de disipación de valor que su equivalente de voltaje o corriente directa (Cartwright, 2007).

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

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

### *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
- $I_p$  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 la fórmula anterior, se puede obtener la fórmula 2.3 (Cartwright, 2007).

$$V_{pp} = 2V_p$$

### *Fórmula 2.3 Cálculo de voltaje pico a pico*

En donde:

- $V_{pico}$  es el voltaje pico
- $V_{pico\ a\ pico}$  es el voltaje

Una vez completa la onda completa 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 positivo y la región negativa. Pero como el lector analógico del microcontrolador tiene un rango de valores en bits de 1024 bits (del 0 al 1023), 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 circuito integrado LM358 conectado en una configuración de offset se puede analizar la señal.

### 2.1.3 Explicación de uso sensor e implementación

Para obtener la señal

## 2.2 Dispositivos de IoT

Los dispositivos IoT nos han ayudado para facilitar trabajos para diferentes áreas. Con las herramientas que cuentan nos pueden informar y controlar sistemas para un propósito en general. Y esto se debe a que se constituyen con varios componentes electrónicos para llevar acabo la comunicación y el control. Principalmente del uso de microcontroladores y microcomputadoras en conjunto para tener un sistema.

### 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 que cuenta una microcomputadora son las siguientes:

- Velocidad de procesamiento
- Interfaz de un solo usuarios
- De uso personal
- Tamaño pequeño
- Fácil uso
- Bajo costo
- Peso mínimo

La Raspberry Pi (Figura 2.6)<sup>5</sup> es una tarjeta que hace la función de una microcomputadora a un bajo costo, con el propósito de llevar a cabo proyectos que necesitan un procesamiento de computo para realizar algunas tareas en específico. La tarjeta tiene en todas sus versiones incluye un procesador Broadcom, una memoria RAM, una GPU, puertos USB, HDMI, Ethernet , 40 pines GPIO y un conector para cámara (Upton, 2018).

---

<sup>5</sup> Tomada de <http://salonurody.info/?y=The+New+Raspberry+Pi+3++With+WiFi+and+Bluetooth++Make>

Utiliza una tarjeta SD y en ediciones posteriores una tarjeta MicroSD la cual tiene grabada 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 a una velocidad en la que trabaja de 1,2GHz.

Cuenta también con el añadido de un modulo Wi-Fi BCM43438 BUILT-IN, ademas de un modulo Bluetooth low energy para evitar el uso de antenas externas para la comunicación inalámbrica de la Pi. Son muy importantes para el dispositivo para la conectividad en una red WLAN (wireless local area network) y lleve a cabo el envío de datos hacia un servidor.

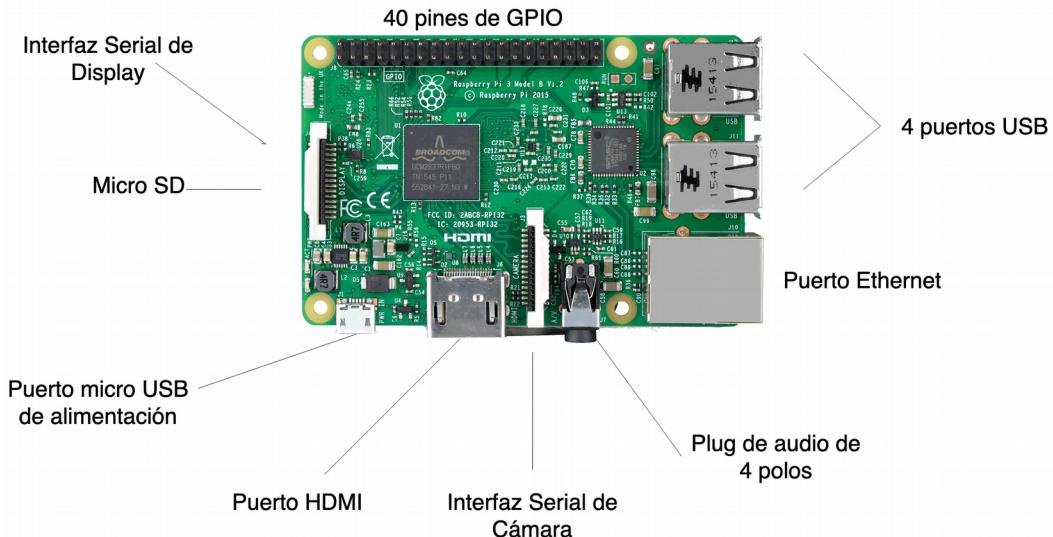


Figura 2.6: Diagrama de una Raspberry PI 3

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 que utiliza la placa son para poder enviar 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 2.7<sup>6</sup>.

6 Tomado de <https://es.pinout.xyz/resources/raspberry-pi-pinout.png>

Raspberry Pi GPIO BCM numbering



Figura 2.7: Pinout de la Raspberry Pi 3B

Las nomenclaturas se asocian (figura 2.7) del pinout son tanto 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

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. Algunos programadores y desarrolladores comúnmente se inclinan a el uso de Linux. Esto es porque con el uso de linux ocupa menos recursos de memoria. Uno de los más usados o conocidos por el mundo es el sistema operativo basado en linux es Raspian.

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

su 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 agradable para una fácil instalación en su Raspberry Pi.

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 (figura 2.8)<sup>7</sup>.

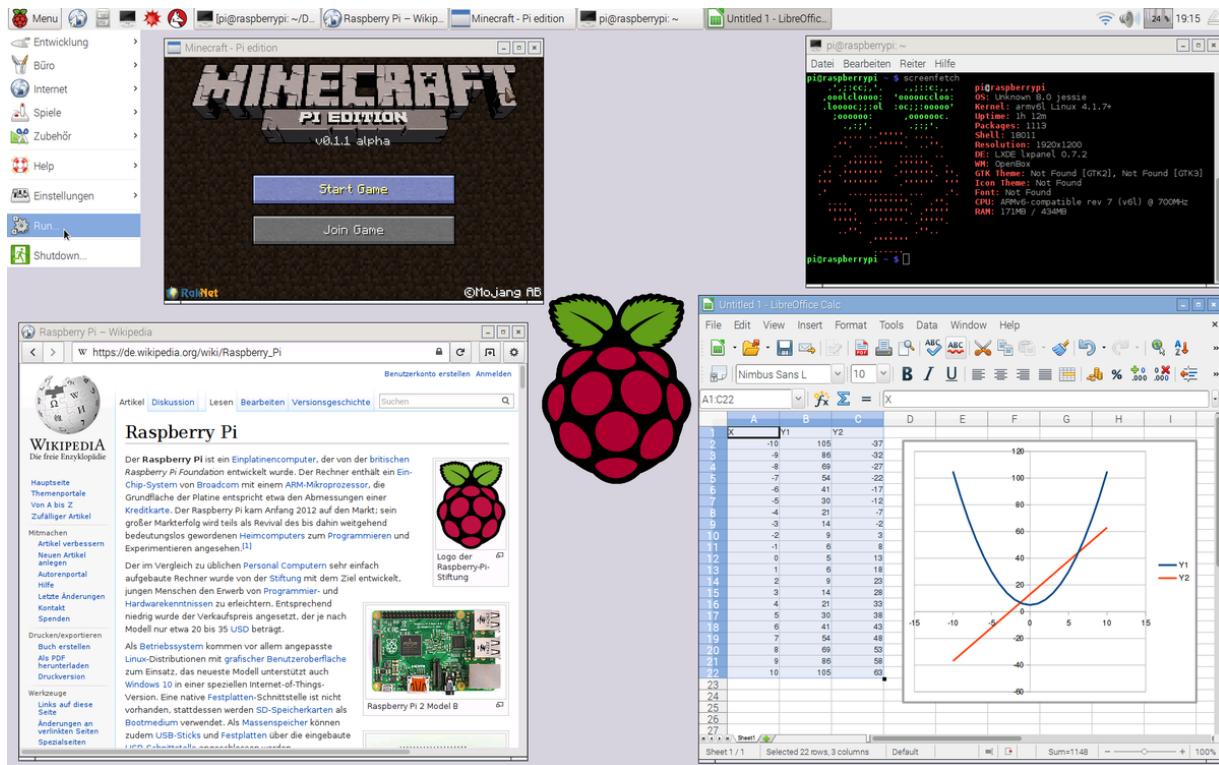


Figura 2.8 Vista previa del sistema operativo Raspbian

## 2.2.2 Microcontroladores (Arduino nano)

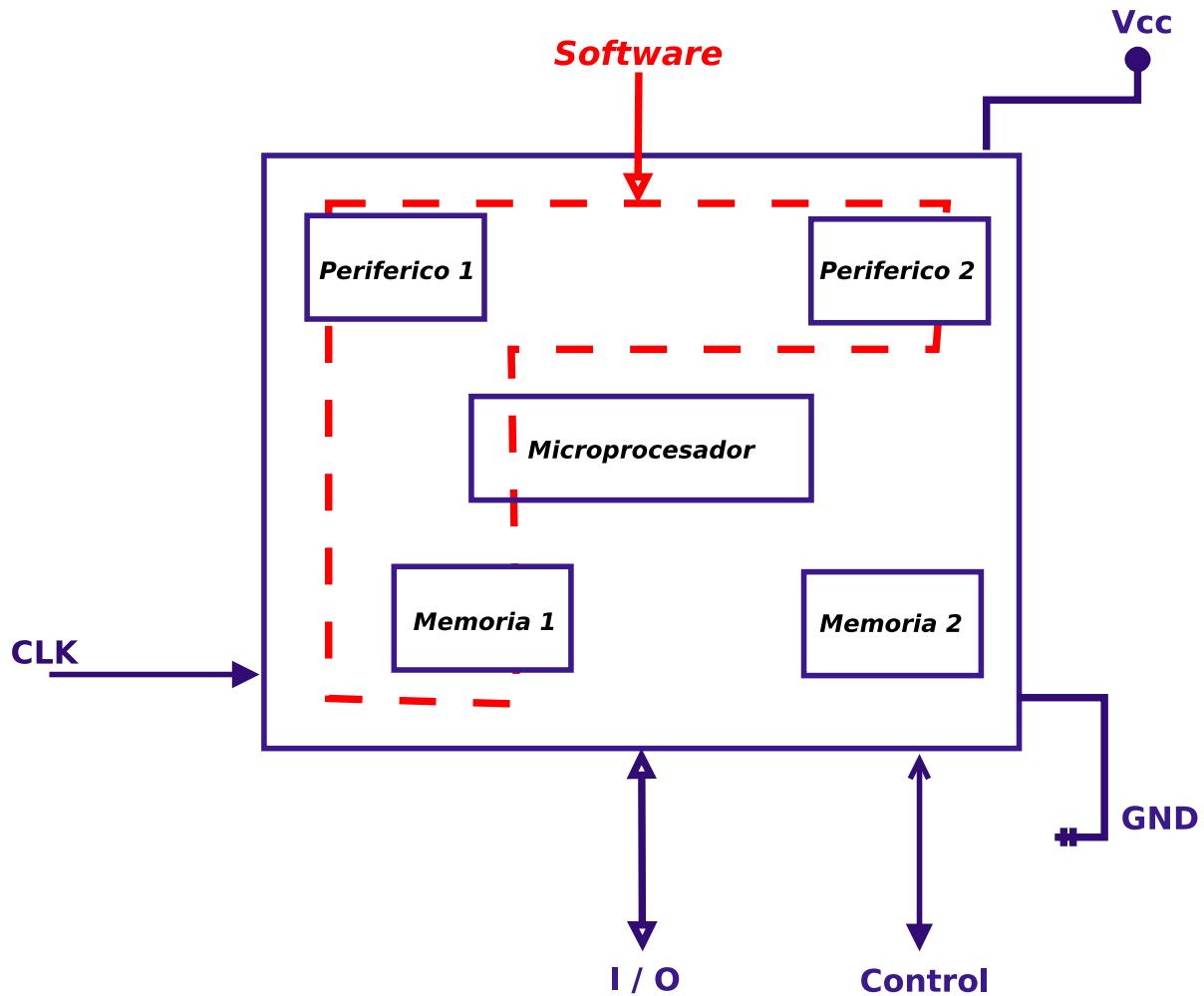
Un microcontrolador (Torrente, 2013) es un circuito integrado o chip, es decir, es un dispositivo electrónico que integra en un solo encapsulado un gran número de componentes que tiene la característica de ser programable. El circuito integrado tiene la función de ejecutar de forma autónoma una serie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

<sup>7</sup> Tomado de <https://hu.wikipedia.org/wiki/Raspbian>

Todos los microcontroladores (figura 2.9) 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 con el que trabajan con un regulador a 5V y un puente de diodos interno para la protección interna del microcontrolador.

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



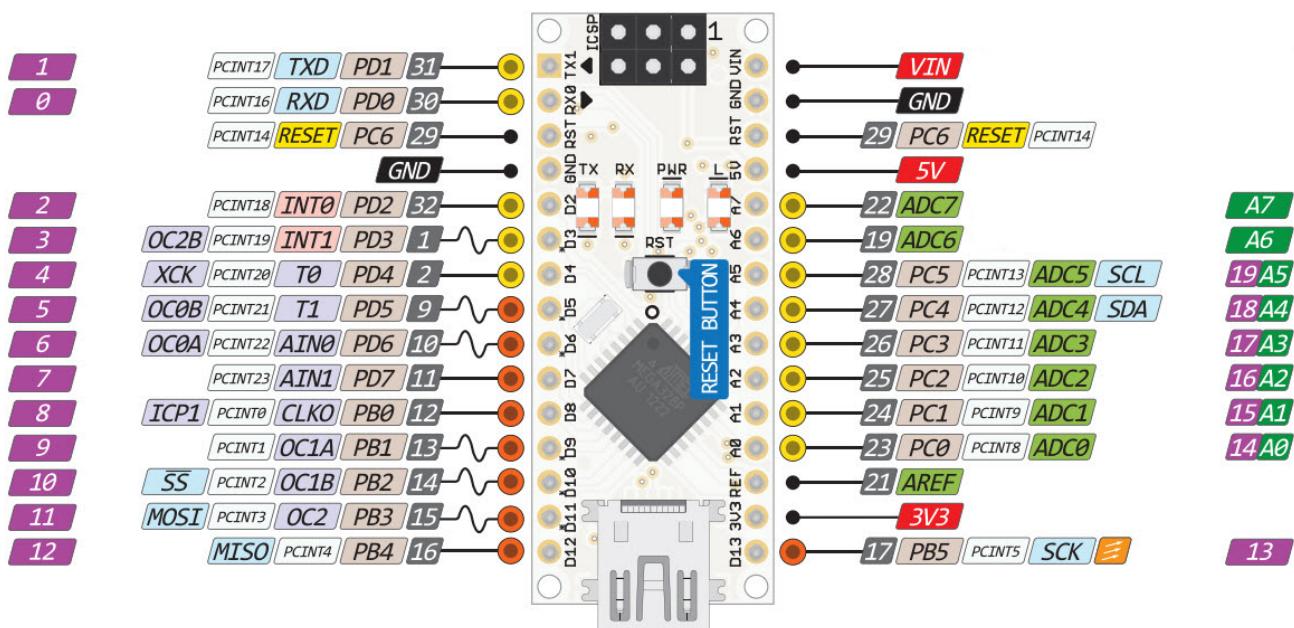
*Figura 2.9 Diagrama básico de un microcontrolador*

Arduino es una placa de desarrollo basado en 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 sketchbook con programación mucho más sencilla esta basado sentencias como las que se encuentran en C para el programador.

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 2.10) cuenta con las características tales como:

- 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 para alimentar algún otro sensor que necesite alimentación 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 2.10 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 perifericos de entradas y salidas de señales digitales. Cuenta con 14 pines (figura 2.10) que son determinados como pines de propósito general (GPIO), pues depende de la configuración que se le de pueden manejar como vías en las que se realiza lecturas o se dan salidas de voltaje, en otras palabras se puede leer voltajes de 0 o 5 volts o responder con los mismos voltajes de 0 o 5 volts.

El manejo de las entradas analógicas en el arduino nano cuenta con seis pines dedicados 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 2.11.

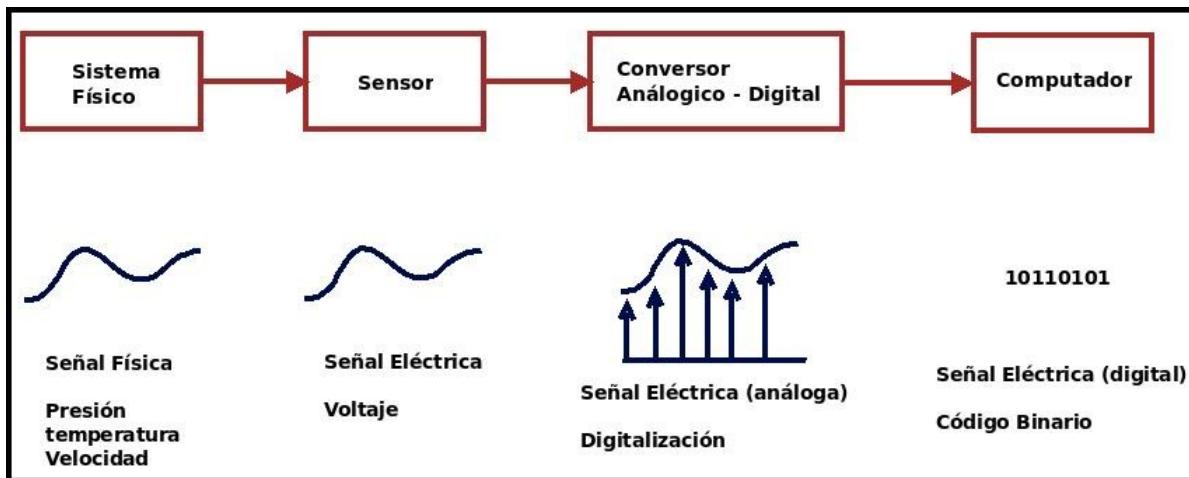


Figura 2.11 Diagrama de un proceso de conversión analógico/digital

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

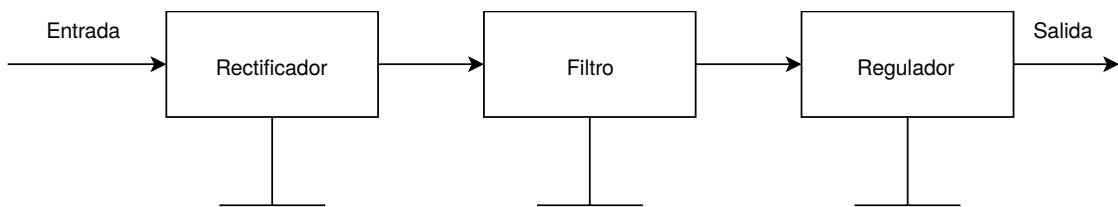
- 1) A partir de una señal física, ya sea mediciones de temperatura, presión atmosférica, velocidad se obtiene la gráfica de unidad de medición contra intervalo de unidades de tiempo para ver los cambios conforme transcurre el tiempo.

- 2) Una vez tenida 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  $2^{10}$  dígitos binarios que van desde el 0 hasta el 1023 en binario.
  
- 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. Para mantener el voltaje continuo lo hacemos con ayuda de una batería o una fuente de alimentación con voltaje alterno. 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 rizado<sup>8</sup> 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 la fuente de alimentación se regulan por medio de un dispositivo de control y de una salida de régimen continuo.



*Figura 2.12 Diagrama de una fuente regulada*

---

<sup>8</sup> Una oscilación de pequeña amplitud impuesta sobre un valor constante

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 2.4. Esto en la práctica,  $V_L$  es siempre menor. A esta variación entre los voltajes se le conoce como regulación de carga.

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

*Fórmula 2.4 Regulación de carga*

Para obtener el porcentaje al que el regulador actúa se obtiene de la fórmula 2.5

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

*Fórmula 2.5 Porcentaje de regulación*

## 2.3 Arquitectura de los sistemas IoT

### 2.3.1 Dispositivo embebidos con internet de las cosas (IoT)

Es común encontrar un dispositivo embebido para utilizar internet de las cosas para administrar su uso. Un sistema embebido (Departamento de Ingeniería Eléctrica, Electrónica y de Control de UNED, 2019) 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 tal 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 ejecute sobre éste. Sin embargo este software necesitará sin duda un lugar donde poder guardarse para luego ser ejecutado por el procesador. Esto podría tomar la forma de memoria RAM o ROM.

Todo sistema embebido (figura 21) necesita de una cierta cantidad de memoria, la cual puede incluso encontrarse dentro del mismo chip del procesador. También contará con una serie de salidas y entradas necesarias para comunicarse con el mundo exterior (Departamento de Ingeniería Eléctrica, Electrónica y de Control de UNED, 2019).

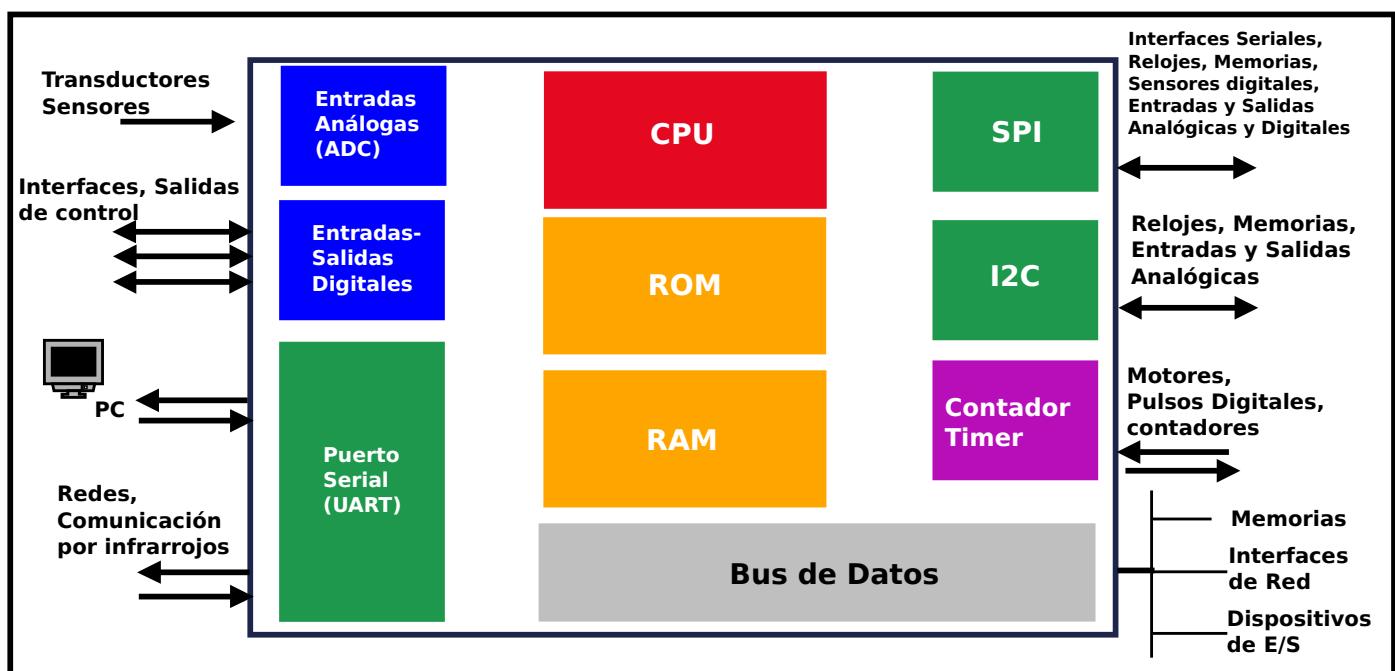


Figura 2.13 Diagrama físico de un dispositivo embebido

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 unos de los pines de microprocesador.

- 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 (contadores 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 de las cosas se define como (Vermesan & Backquest 2017) una infraestructura global para la sociedad 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 como los educativos, negocios, comerciales y sociales.
- Los datos, la parte necesaria 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 consulto y se le sean entregados en tiempo y forma.

Internet de las cosas trabaja bajo las 5 capas de la figura 2.14:

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. Aquí es donde los programadores que realizan aplicaciones de IoT para trabajar con los recursos de las API sin utilizar una base física.
4. En la capa 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.

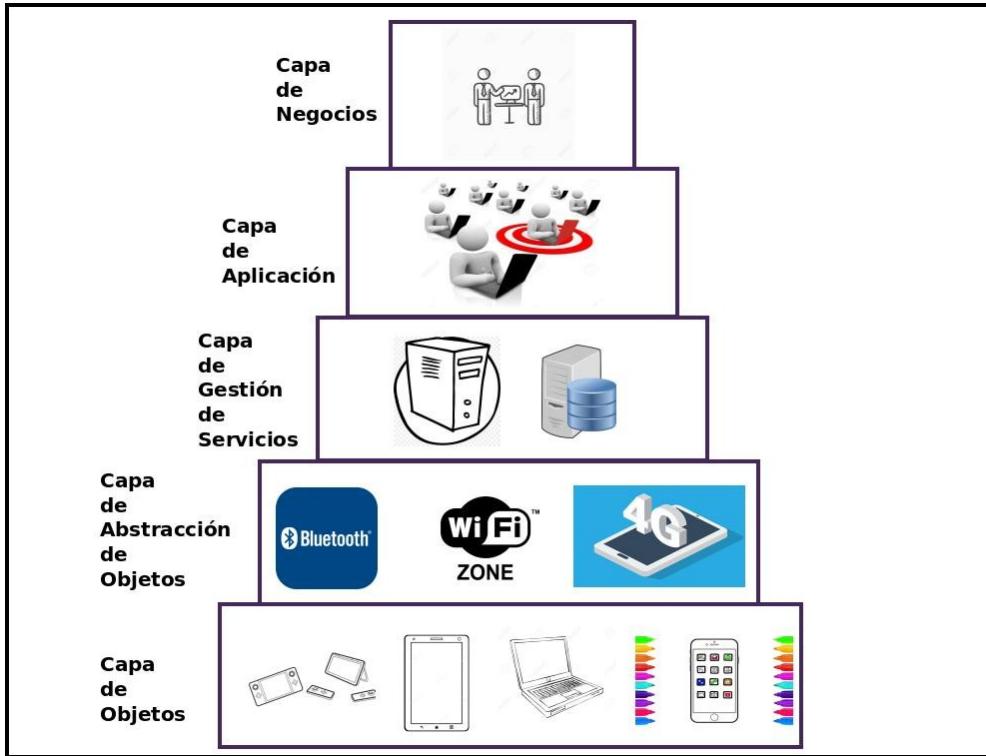


Figura 2.14 Diagrama de Capas de internet de las cosas

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 puedes consultar 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 solicito.

### 2.3.2 Web Service (RESTful)

Un Web service, según una definición del Dept. Ciencia de la Computación e IA de la universidad de Alicante (2012-2013) como una colección de procedimientos o métodos a los que podemos llamar desde cualquier lugar de Internet o de nuestra intranet, siendo este mecanismo de invocación totalmente independiente de la plataforma que utilicemos y del lenguaje de programación

en el que se haya implementado internamente el servicio.

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

- El protocolo SOAP (Simple Object Access Protocol) está 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.

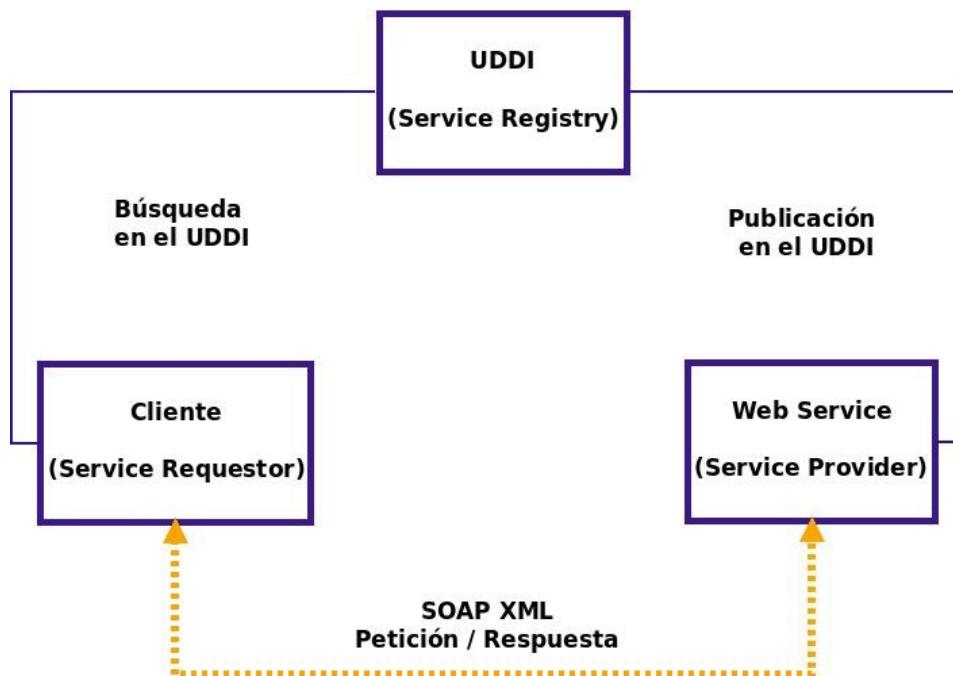


Figura 2.15 Diagrama de funcionamiento de un Web Service

La forma por la cual un Web Service funciona es de la siguiente manera (figura 2.15):

- Existe un servicio proveedor el cual se encarga de generar el WSDL que hace una descripción del Web Service y hacer su registro en 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 realizará 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.

Todo web service está basado en una misma arquitectura, la cual consiste en cuatro partes las 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 como 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 se transmiten entre cliente y el servidor. Utiliza el protocolo de HTTP (HyperText Transfer Protocol) puesto que es el más utilizado actualmente por navegadores y páginas web.

Sabiendo como es el funcionamiento de un web service, podemos de hablar de un servicio en específico que funciona de diferente forma pero hace el mismo propósito de enviar información en internet, llamado 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.
- Si se habla de un método GET es para llevar a cabo una consulta. 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 lo que realiza es dentro de un sitio web, es ser un código para que dos programas puedan comunicarse entre si de manera más fácil sin afectar la información dentro de la aplicación. En este caso la unión de ambos, una API RESTful es un código de comunicación basados en la metodología REST antes mencionada para utilizar las cuatro operaciones de peticiones web para la comunicación.

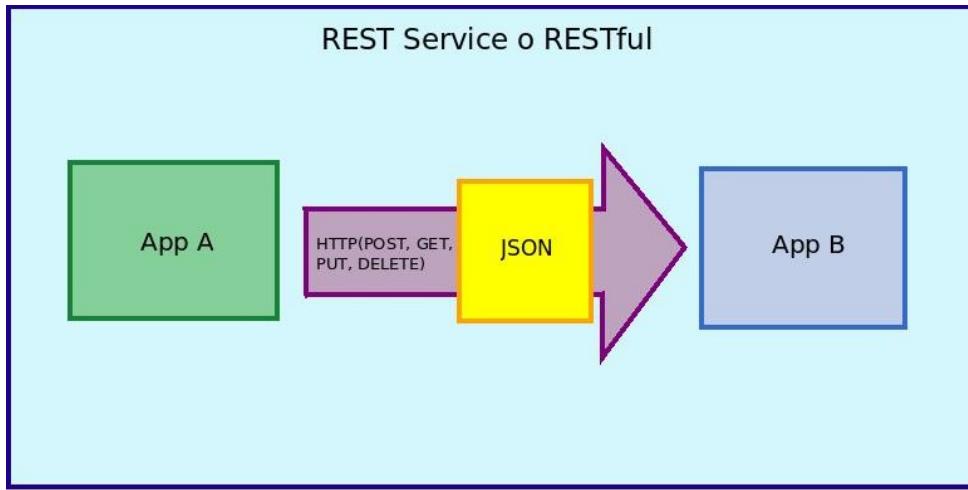
Una API RESTful hace un análisis de los datos que se piden desde un cliente en algún dispositivo por medio de internet y buscarlos desde el dispositivo servidor con la información; esto es de gran utilidad puesto que no es necesario tener la base de datos en el mismo lugar que el servicio RESTful, sino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

REStful es tan versátil de enviar cualquier tipo de información, puesto que el tipo de dato es definido dentro del Header Content-Type, y de sea manera poder manejar 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. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es 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.

Una explicación más sencilla de la comunicación por RESTful es por medio de la figura 2.16. 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 B.

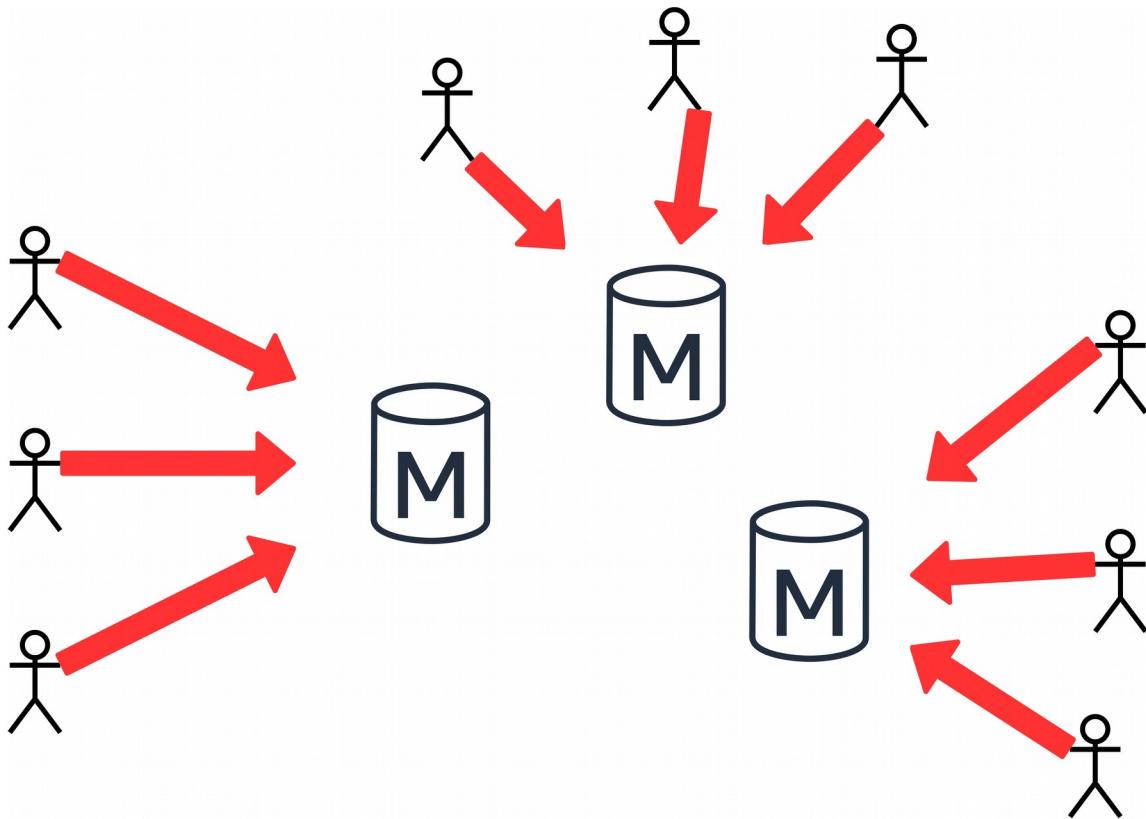


*Figura 2.16 Diagrama de funcionamiento de un servicio RESTful*

### 2.3.3 Uso de base de datos

Una base de datos es una colección de información 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 integra ante fallas en el sistema o de intentos no autorizados a la base de datos (Figura 2.17).

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 2.17 Diagrama de funcionamiento de una base de datos*

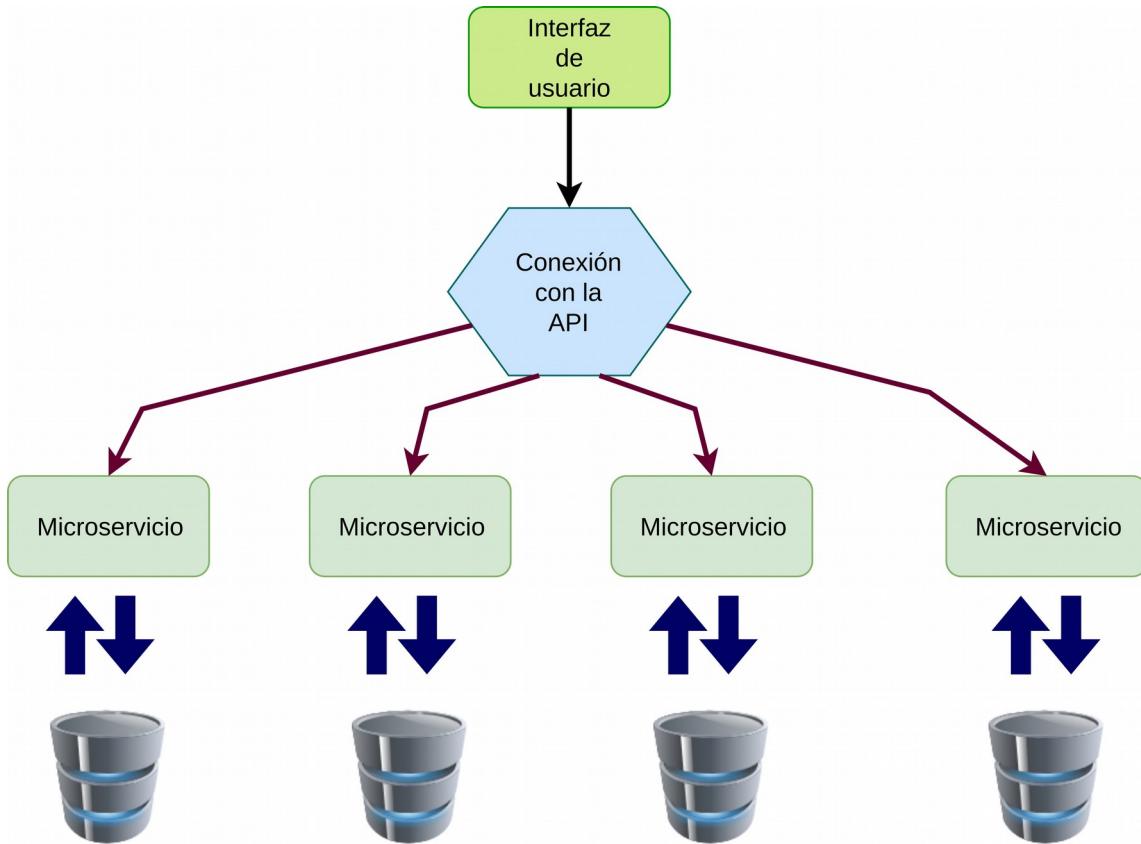
Una de las formas más fáciles para realizar la acción de almacenamiento de datos es con ayuda de una API. En una base de datos tiene una interfaz para poder llevar a cabo tareas a través de los clientes y de los servidores de bases de datos, es decir, que un cliente interactúa con peticiones hacia el servidor por medio de la interfaz.

Para que se pueda realizar el acceso y manipulación de la base de datos por medio de la API se debe hacer una conectividad de los datos externos, y de esa manera vincularlos hacia la API para que puedan ser consultados sin importar donde se encuentre la base de datos.

La manera en la que lleva a cabo el almacenamiento de los datos por medio de una API que se representa en la Figura 2.18 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 interfaz del usuario que dicha acción se realizó exitosamente.



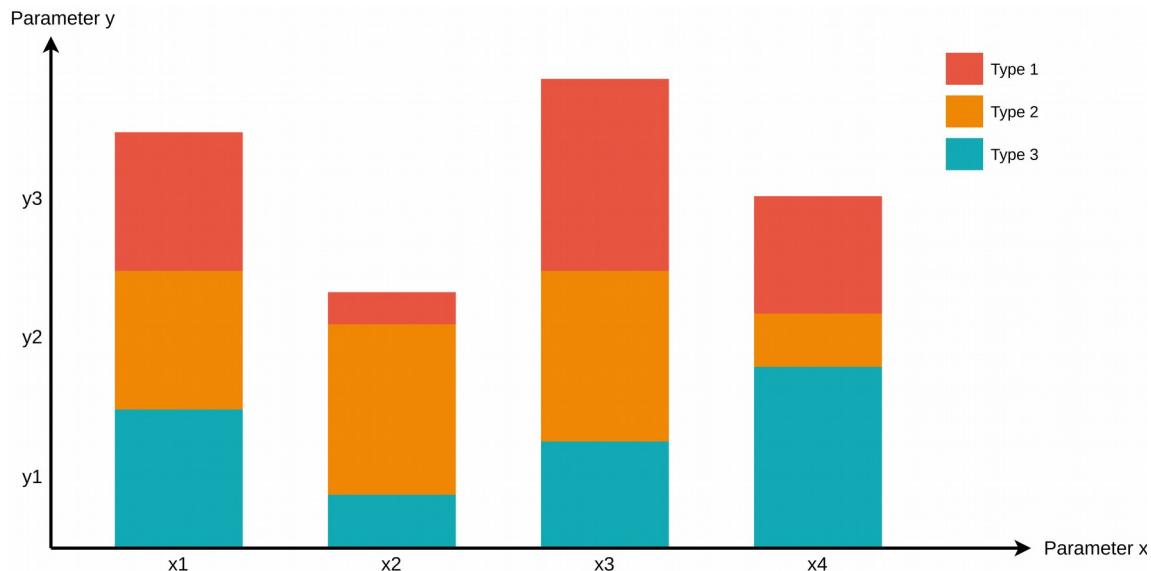
*Figura 2.18 Diagrama de funcionamiento de una base de datos junto a un API*

#### 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 nos ayuda a comprender mejor la información de como se comportan los datos. Los datos se pueden representar en formas de puntos, de barras o de líneas. Estos elementos de 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 gráfica de datos, en la que los datos se representan mediante símbolos, como barras en un gráfico de barras como en la Figura 2.19 o como líneas en un gráfico de líneas o como divisiones en un gráfico circular.



*Figura 2.19 Representación de un gráfico de barras*

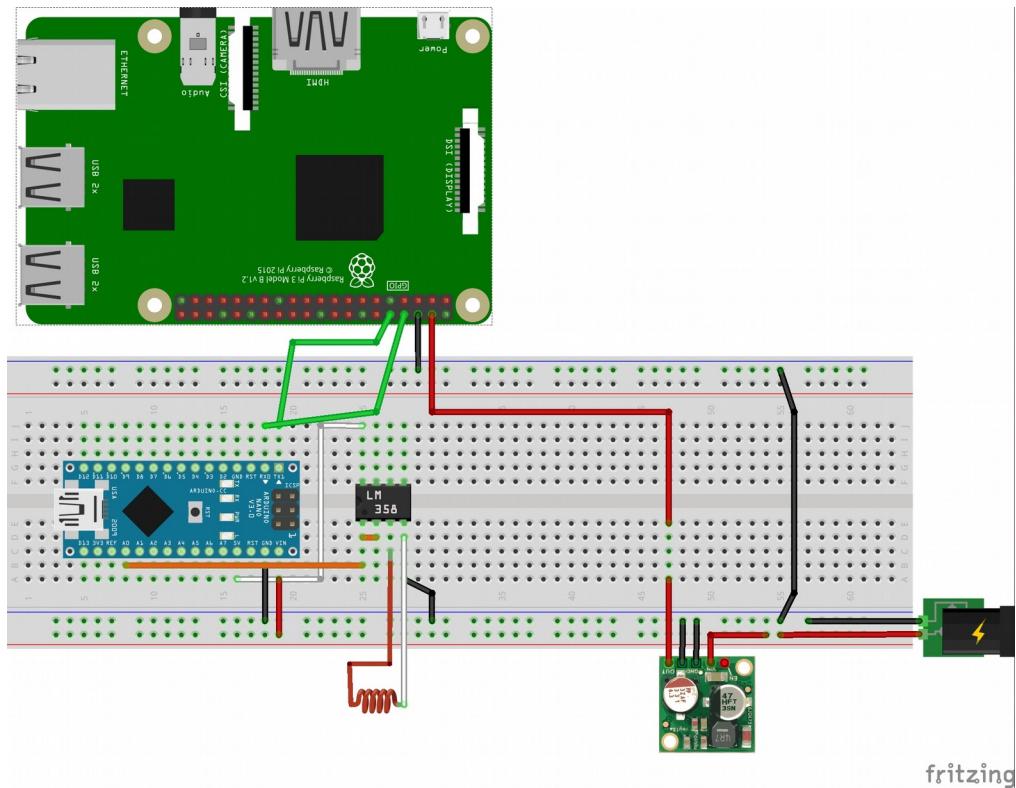
### 3 Desarrollo

En este capítulo se describe el diseño y el funcionamiento del diseño del sistema de consulta del consumo eléctrico. En la sección 3.1 se describe 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 3.2 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 3.3 se explica el uso de la página web para visualizar desde cualquier lugar con conexión a internet desde una computadora, celular o tableta puedes monitorear el consumo eléctrico medido, así como obtener un precio aproximado según la tarifa de CFE.

### 3.1 Dispositivo embebido

El dispositivo embebido con el cual se realizan las lecturas esta constituido por los siguientes elementos (figura 3.1):

- El sensor SCT-013-030 con el cual se obtienen los valores del consumo de corriente en residencias. Y para poder acondicionar las señales para lecturas hacia el arduino un integrado LM358 que es un amplificador operacional que rectifica la señal a un cierto voltaje y este pueda ser leído por el microcontrolador.
- El microcontrolador arduino nano que nos va ayudar 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 que funciona como una mini computadora que por medio de código se toman los valores de corriente que llegan por su serial, se colocan en archivos JSON con una cierta cantidad de lecturas en un intervalo de tiempo junto con el tiempo en el que se generó la lectura, y cada JSON se envían hacia un servidor fijo.
- El servidor dentro de el contiene la base de datos de MongoDB que toma cada JSON y almacena el valor de corriente dentro de su base de datos, esto para que la aplicación final pueda consultar el consumo energético en tiempo real. A su vez el servidor tiene montada la aplicación de AdminLTE que es la parte final en la que el usuario puede consultar de manera gráfica el historial o en tiempo real de su consumo energético.



*Figura 3.1 Diagrama completo del dispositivo embebido*

La alimentación para el circuito tiene que ser regulada a 5V con 2.5A para que el funcionamiento sea el adecuado. Se utiliza una batería de polímero litio para energizar todo el circuito con ciertas especificaciones para que funcione a cierto tiempo. Una opción viable de batería fue con una que trabaja a 11.1V a 2.5A la cual da un funcionamiento en un tiempo prolongado. Pero aun así hay que regular el voltaje a 5 V para evitar quemar todo el circuito.

### 3.1.1 Sensores comerciales

En el mercado existen variedad de sensores de corriente no invasivos, y cada uno varia según la magnitud de corriente a medir. Estos van de magnitudes mínimas de 3A hasta corrientes de 100A dependiendo de la información de construcción del fabricante de cada uno de los modelos.

Para los propósitos de consumo eléctrico se consideraron varios 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 más óptimos para desarrollar el sistema se encuentran en la Tabla 1 con todas sus características de medición y límites de lectura.

Entre todas las opciones consideradas se busco que el sensor sea lo más

sencillo de conectar, puesto que es dirigido a cualquier persona lo pueda utilizar en su casa, y con el propósito de 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 se tiene 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 3.2.

Aunque también existen sensores de corriente que se colocan dentro del cableado eléctrico para poder medir el consumo eléctrico, se prefiere usar este tipo de conexión no invasivo para consulta de consumo eléctrico en sistemas que funcionan en alterna, puesto que por su facilidad de uso es ser colocado en una de las terminales sin tener que tener que invadir el cableado eléctrico que se desea consultar.

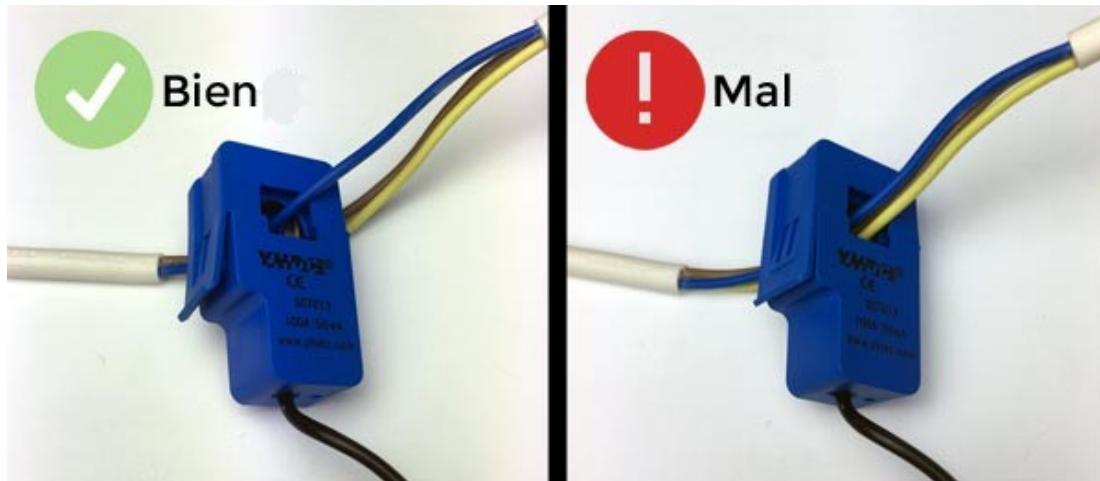


Figura 3.2 Conexión del sensor de corriente no invasivo

Nombre del Sensor	Imagen	Precio	Características
Sensor Corriente Invasivo 30A de No		\$350.00 MXN	<ul style="list-style-type: none"> <li>• 30 A</li> <li>• 250 Ohm internos</li> <li>• 300 mV de lectura</li> </ul>
Mini Transformador De Corriente Sensor Transductor 5a / 5ma		\$60.00 MXN	<ul style="list-style-type: none"> <li>• Corriente de sensado 0-5A</li> <li>• Corriente de salida 0-50mA</li> <li>• Relación 1:100</li> <li>• Aislamiento eléctrico: 3000V</li> <li>• Temperatura de trabajo: -40 a +70°C</li> </ul>
Sensor De Corriente Analógico 5a Inductivo Arduino Avr Pic		\$170.00 MXN	<ul style="list-style-type: none"> <li>• Coeficiente de transformación: 1000:1</li> <li>• Corriente de entrada: 5A.</li> <li>• Corriente de salida: 5mA.</li> <li>• Resistencia de muestreo: 200 Ohms.</li> <li>• Voltaje de muestreo: 1V.</li> <li>• Frecuencia de trabajo: 20 Hz a 20 KHz.</li> <li>• Temperatura de operación: -55 a 85°C.</li> <li>• Fuerza dieléctrica: 6KAC/1min.</li> </ul>

Sensor de Corriente Invasivo No 30A , SCT013		\$320.00 MXN	<ul style="list-style-type: none"> <li>• Frecuencia: 50HZ-150KHZ</li> <li>• Conmutación tipo transformador SCT-013</li> <li>• corriente nominal de entrada 30A</li> <li>• Tipo: SCT-013</li> <li>• Corriente de entrada: 60A/100A</li> </ul>
--	---	--------------	--

Tabla 1 Tabla comparativa de precios de modelos de sensores de corriente

### 3.1.2 Sensor SCT-013-03

Los sensores SCT-013-03 (figura 3.3)<sup>9</sup> son transformadores de corriente que se ocupan como dispositivos de instrumentación. Estos proporcionan una medición proporcional a la intensidad 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.

---

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



Figura 3.3 Sensor SCT-013-030

### 3.1.3 Reguladores de voltaje

Una de las opciones para regular la fuente de alimentación del dispositivo fue con el uso de el regulador de la marca pololu. Este regulador de (figura 3.4<sup>10</sup>) funciona como un commutador 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 que maneja van de 85% a 95% para poder alimentar cargas moderadas como sensores o motores pequeños.

Algunas de las especificaciones generales dentro 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

---

10 Recuperado de <https://www.pololu.com/product/2850/specs>

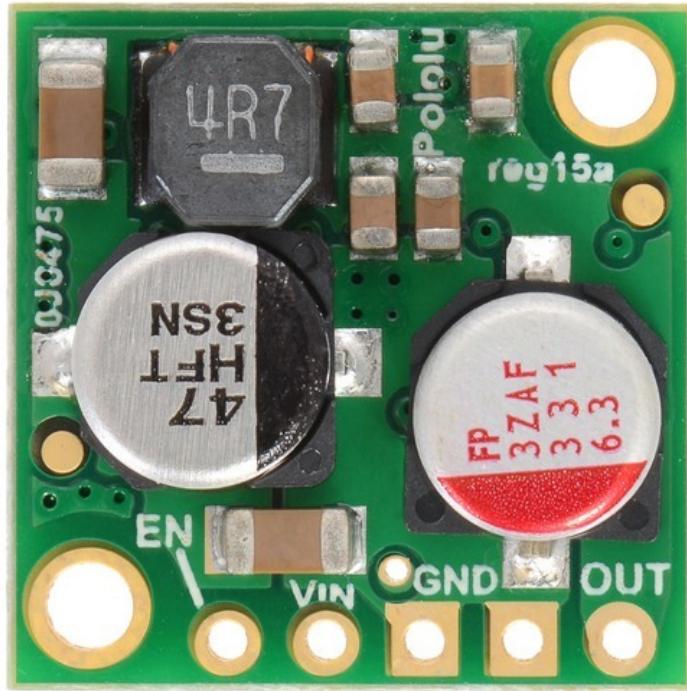


Figura 3.4 Foto de regulador marca pololu reg15a

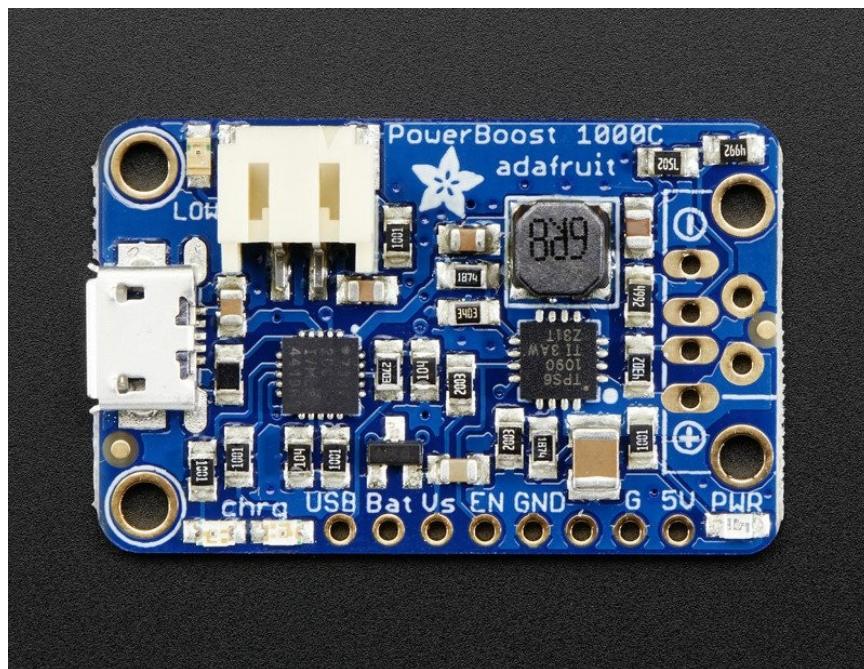
Las desventajas de utilizar la batería en conjunto con el regulador es tener que recargar la batería siendo desconectada para poder recargarse y perder las mediciones que pasen mientras la batería se encuentra cargando.

Otra opción es utilizar es con otro tipo de regulador con la opción de cargar la batería al mismo tiempo. El Modulo PowerBoost 1000C (Figura 3.5) es un módulo convertidor DC / DC boost puede ser alimentado por cualquier batería de 3.7V Lilon / LiPoly, y convierte la salida de la batería a 5.2V DC para ejecutar sus proyectos de 5V. 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:

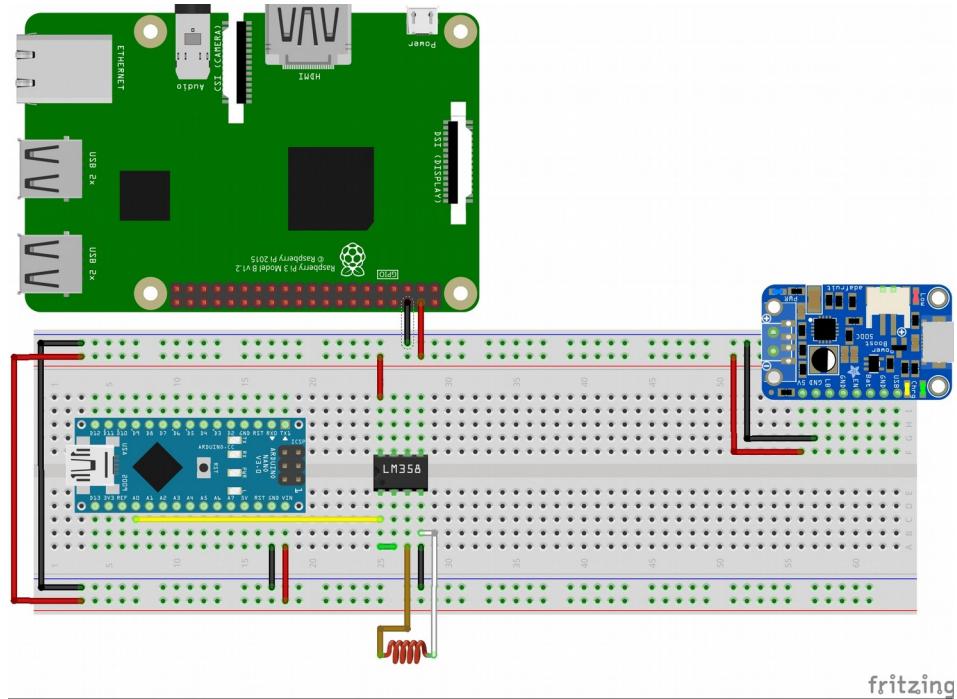
- La operación síncrona significa que puede desconectar la salida completamente conectando el pin de enable a tierra. Esto apagará completamente la salida
- El interruptor interno de 2A (~ 2.5A de limitación de pico) significa que puede obtener 1000mA + de una batería de 3.7V LiPoly / Lilon.
- El indicador LED de batería baja se ilumina en rojo cuando el voltaje cae por debajo de 3.2V, optimizado para el uso de la batería LiPo / Lilon

- Desglose completo para la entrada de la batería, los pines de control y la salida de alimentación
- 90% más de eficiencia operativa en la mayoría de los casos y baja corriente de reposo: 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)



*Figura 3.5 Módulo de carga Adafruit Powerboost 1000C*

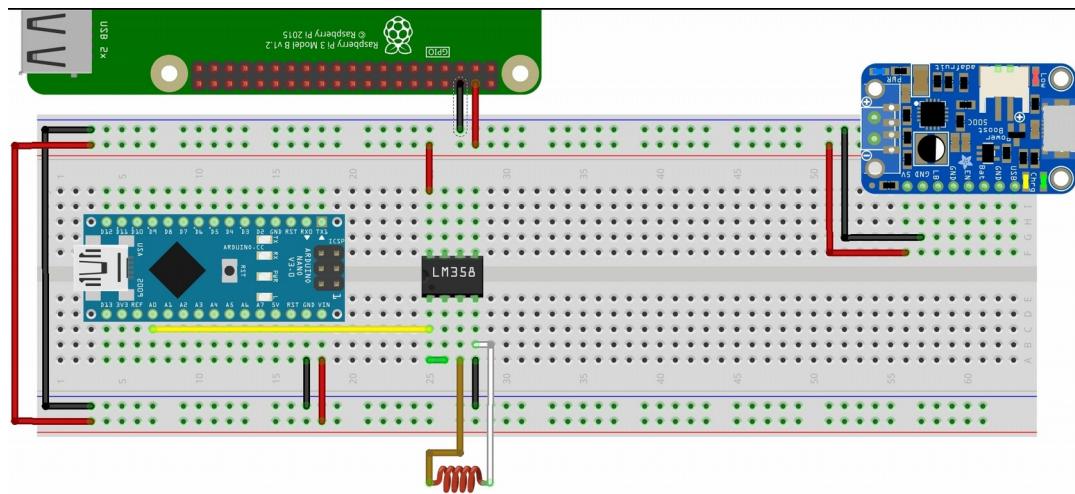
En conjunto al uso del módulo, se utiliza una batería de polímero de litio de 3.7V a 4.2A para tener un mayor tiempo del funcionamiento del sistema embebido, quedando la configuración como se muestra en la figura 3.6.



*Fígura 3.6 Conexión del sistema embedido con el uso del PowerBoost1000C*

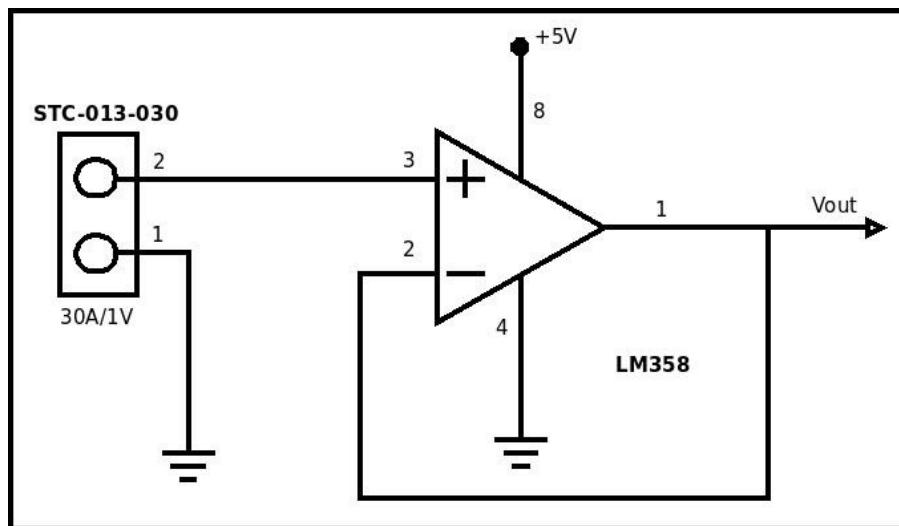
### 3.1.4 Circuito del sistema

El circuito básico para la lectura se basa en el uso del sensor SCT-013 conectado en el encapsulado LM358, esto para que la señal pueda ser acondicionada puesto que el sensor brinda una salida de -50mA a 50mA entonces para llevarla a una lectura para el arduino de 0V a 5V haciendo una conversión de corriente a voltaje con relación -1V a 1V, amplificarlo hasta llegar a un rango de -2.5V a 2.5V y con ayuda de un sumador quitar la parte negativa para que llegue al rango deseado.



*Figura 3.7 Sección del circuito para la lectura de corriente y alimentación de la Raspberry*

Otra forma es rectificando la entrada del sensor y manejar la parte positiva con el LM358, pues este integrado trabaja la parte positiva de la señal que tenga como entrada y hacer una rectificación de media onda.



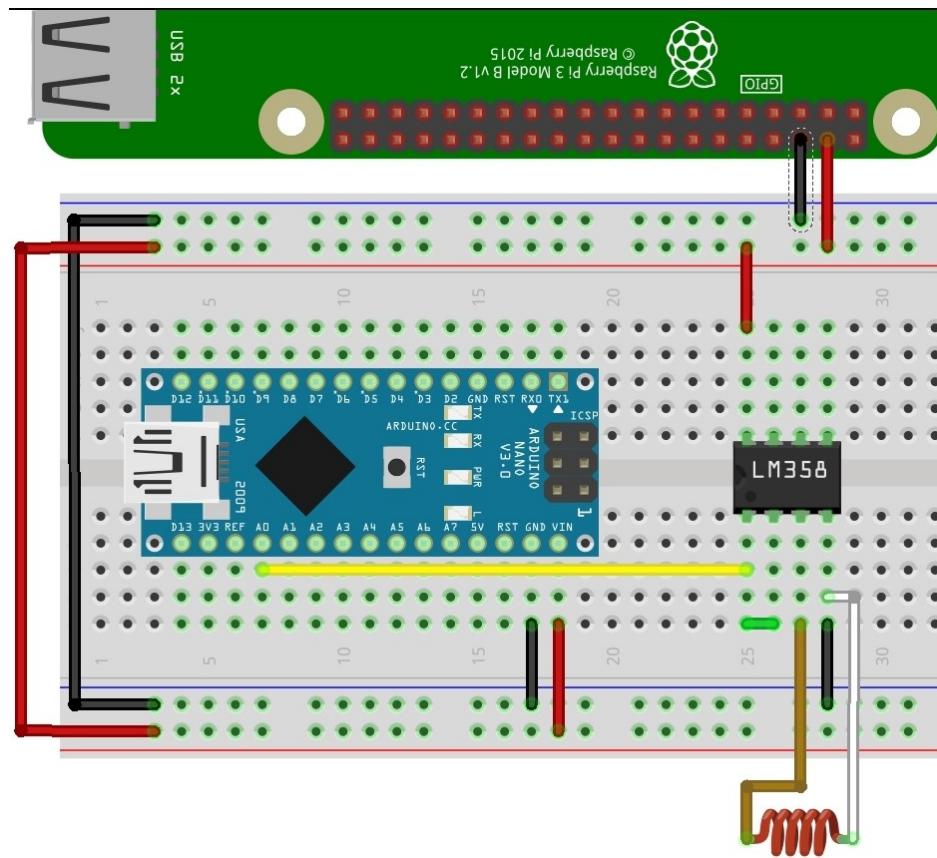
*Figura 3.8 Diagrama de conexión simbólico del LM358 con el sensor SCT-013-030*

El LM358 se alimenta con 5V pero se satura con 3.5V, entonces con ayuda del arduino se configura 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.

Todo el circuito esta alimentado de una fuente externa que puede ir de los 9V a los 12V para que el funcionamiento del arduino sea el adecuado, por lo cual el regulador de 5V se utiliza especialmente para el funcionamiento de la Raspberry, pues necesita tener una alimentación constante de los 5V a 2A para su funcionamiento correcto. Al alimentar el regulador con 12V el mismo regulador baja el voltaje de 5V y entregando una corriente que va de 2A a 2.5A sea la demanda del dispositivo a usar.

El arduino está alimentado en la fuente externa por su pin de Vin (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 3.9.



*Figura 3.9 Parte del diagrama de la conexión del sensor SCT-013-030 al arduino*

### 3.1.5 Cálculos de la corriente eléctrica

El ADC que maneja arduino es de 1024 números en bits, 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 A/D, convirtiendo el binario a decimal, para que entonces se tiene que hacer la conversión a voltaje con una relación de valor voltaje de la Fórmula 3.1.

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

*Fórmula 3.1 Cálculo del voltaje a medir por el arduino*

Donde 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. Pero lo que necesitamos es una medición de corriente, por lo cual en sensor maneja una relación de 1V/30A, que cada 1V nos da 30A en la medición. Por lo cual la conversión que se obtiene esta dada por la Fórmula 3.2.

Obteniendo 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.

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

*Fórmula 3.2 Conversión a corriente*

Una vez teniendo la corriente en cada instante se puede obtener la corriente eficaz o corriente RMS, que es la corriente que se usa en los hogares. La corriente eficaz es la corriente que es capaz de producir el mismo trabajo que el valor en corriente directa o alterna con la Fórmula 3.3.

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

### *Fórmula 3.3 Corriente RMS en el dominio continuo*

En donde T es el periodo en el que se muestrea la corriente y la variable i es la Corriente medida con el sensor anteriormente. Se tienen muestras cada cierto tiempo por el sensor, entonces la formula se tiene que llevar al dominio discreto para tener en cuenta cada muestra obtenida, 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*

Una vez teniendo la corriente eficaz es posible obtener la potencia con la cual esta trabajando los aparatos eléctricos dentro de un hogar con la fórmula de potencia (Fórmula 3.5). El valor de 127V es el voltaje con el que nominalmente se encuentra en todo México.

$$Potencia = I_{rms} * 127V$$

### *Fórmula 3.5 Fórmula de potencia*

#### 3.1.6 Interconexión Arduino-Raspberry

Para la comunicación entre la raspberry y el arduino es por medio del puerto serial que ambos tienen en común. El arduino una vez teniendo la corriente eficaz envía los datos por su puerto serial por el comando de “println()” que básicamente envía el dato seguido de un salto de linea “\n”, esto para que la Raspberry pueda detectar un valor nuevo para almacenarlo en el JSON.

Una vez ya alimentada la Raspberry con el voltaje de 5V del regulador en los pines de alimentación (pin 2 para +5V y pin 4 para GND) empieza a correr el programa que detecta cada salto de linea que le llega por su puerto serial y

siempre lo mantiene abierto para recibir constantemente las lecturas del arduino. La conexión se puede llevar acabo 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. Entonces 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 claro esta en la Figura 3.10.

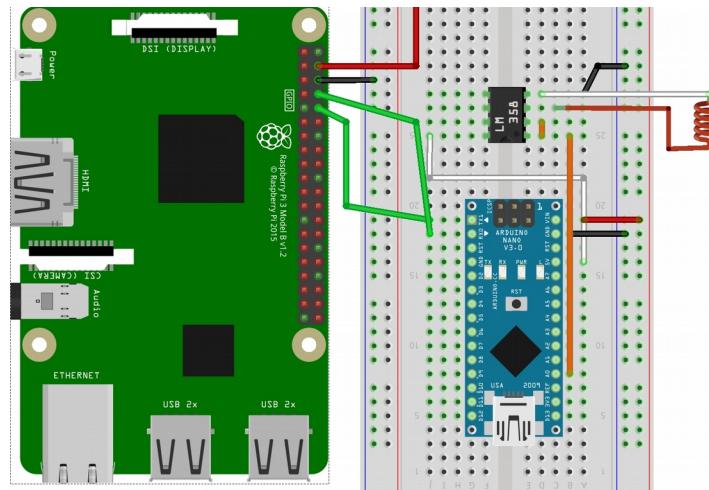
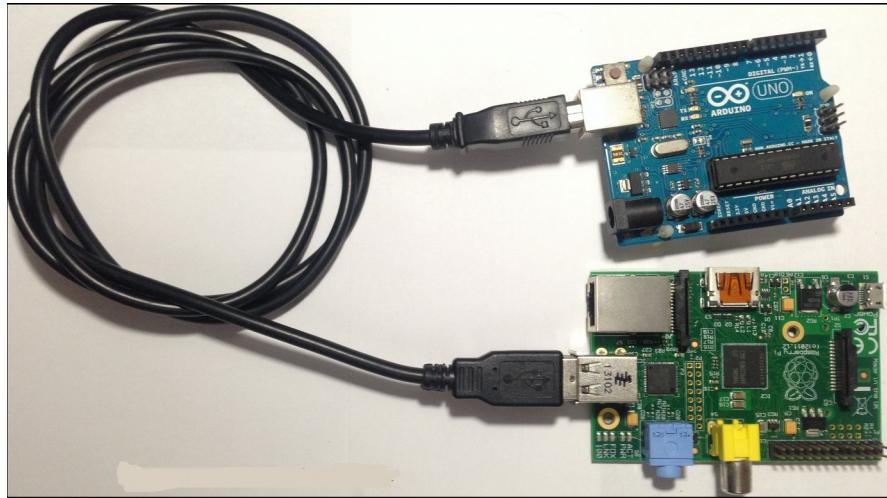


Figura 3.10 Conexión por pines serial

- Mientras la segunda y la más fácil de hacer la conexión por puertos serial entre el Arduino y la Raspberry es usando los puertos USB serial que contiene cada una de las placas por medio de un cable USB como lo muestra en Figura 3.11.



*Figura 3.11 Conexión por USB serial*

### 3.1.7 Software del sistema

Con el sistema operativo de Raspian se implementa el programa que hace todo el manejo de los datos desde el recolectarlos desde el puerto serial de donde llegan por medio del arduino, hasta el envío de los archivos JSON hacia el servidor donde el Web service estará recibiendo los archivos para pasar la información a la base de datos.

El programa esta hecho en Python y esta constituido por multiprocesamiento, es decir que cada tarea en un proceso diferente llamado 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 espacio de un proceso principal. Y para este caso existen tres hilos y un hilo principal donde se encuentran dentro, cada hilo tiene su propia tarea:

- El primer hilo interno se encarga de la recepción de los datos provenientes del bus serial para almacenarlos junto a la fecha en la que se adquirió para que la raspberry 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 obtuvo 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 JSON para enviarlos hacia el web service y una vez recibida la confirmación de que se entregaron borra los archivos que se enviaron.

Todos los hilos se manejan 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.

### 3.1.8 Comunicación dispositivo-api con Flask y Gevent

El web service esta desarrollado 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 su visualización de apoya de jinja, que es un motor de plantillas para python.

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 la función de la API es gevent. Gevent es una herramienta que realiza una rutina para proporcionar a una API síncrona de alto nivel en la parte superior del ciclo de eventos (Gevent, 2019).

Gevent cuenta con una herramienta llamada pywsgi que tiene como función crear servidores WSGI y 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 parámetros bajo peticiones http, mientras que con

Gevent levanta el API para hacer la comunicación entre la raspberry y el web service.

El web service programado con Flask y Gevent se encuentra instalado en el mismo servidor en donde se encuentra la base de datos. Todo el manejo de los datos se lleva acabo por medio de la dirección [www.labmovilidad.unam.mx](http://www.labmovilidad.unam.mx):5000. La dirección http maneja dos apartados: un apartado para la adquisición de datos y uno para búsqueda de datos para poder visualizarlos en la gráfica.

Para la recepción de datos por vía internet con protocolo http hacia la dirección fija que se encuentra dentro del servidor. En este caso la dirección destino es [www.labmovilidad.unam.mx](http://www.labmovilidad.unam.mx):5000/api/v1/data 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](http://www.labmovilidad.unam.mx):5000/api/consult que es el apartado global donde se realiza las búsquedas. Las búsquedas son de manera específica, ya que se necesita buscar los datos por medio de rangos de fechas; esto para que el sistema pueda visualizar por períodos el consumo eléctrico. Los parámetros de búsqueda por medio de la dirección son id, fmin y 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 dispositivo. 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. Ejemplo fmin=2018-10-13%2019:00:00, donde %20 es un espacio para protocolos http.

De tal manera que para que el buscador haga una búsqueda completa desde el dispositivo cuya dirección MAC es b8:27:eb:1f:34:8b con la fecha del 13 de octubre de las 19 a las 21 hrs es de la siguiente manera: [www.labmovilidad.unam.mx](http://www.labmovilidad.unam.mx):5000/api/consult?id=b8:27:eb:1f:34:8b&fmin=2018-10-13%2019:00:00&fmax= 2018-10-13%2021:00:00

## 3.2 Sistema de adquisición de datos

El web service está directamente conectado a la base de datos por la misma dirección web de [www.labmovilidad.unam.mx](http://www.labmovilidad.unam.mx) solo que en este caso es por un puerto específico de comunicación hacia la base de datos de mongoDB. Cualquier procedimiento de insertar o extraer información a la base de datos son por su puerto 27017 que corre de manera local en el servidor.

Para conectarse a la base de datos se necesita encontrar la sección destinada para almacenar los datos de corriente. En este caso el cliente que es el web service accede a la sección llamada raspi.

### 3.2.1 Base de datos con MongoDB

Una de las bases de datos comúnmente usada 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.
- Tiene disponibilidad de entregar a las aplicaciones que pueden recuperarse frente a una 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

El manejo de MongoDB es muy fácil y versátil de usar, 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. PyMongo es una distribución de Python que contiene herramientas para trabajar con MongoDB , y es la forma recomendada de trabajar con MongoDB de Python (MongoDB, 2008-2019).

Una vez que los datos llegan al web service recorre cada archivo que le llega y obtiene los datos de cada uno para agregarlos a la base de datos. Para esto se utiliza la herramienta de pymongo, que es una biblioteca en python para el manejo de bases de datos de mongoDB en un código de programación.

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 uno a uno dentro de la base de datos. La ventaja de que sean datos en formato JSON es que mongoDB acepta tal cual la estructura que y lo añade como un objeto con atributos.

MongoDB hace inserciones a su base de datos con forme entran los datos, sin un formato en específico. Pero cada uno de los datos que están dentro de la base de datos de mongo tiene un id propio para diferenciar entre sí para ayudar al manejo de cada objeto.

### 3.2.2 Algoritmos de búsqueda

Las búsquedas 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.

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.

Después de la búsqueda, se crea una copia de cada JSON que se encontraron en la base de datos. Se crea una lista donde se retornan todos sus datos a excepción del id propio que mongoDB asignó.

### 3.2.3 Protocolos de respuesta

Cada acción que realiza el webservice va acompañado por las peticiones web en http. Para las peticiones de añadir datos a la base de datos son por método POST para recibir el archivo que se envío 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 el código es de 500 (error al procesar la información en http).

Para solicitar los datos es con un método GET para que la búsqueda retorne a la dirección todos los JSON resultantes. De igual manera se maneja el código 200 para envío correcto y 500 si se generó 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 los correctos se bosqueja la gráfica con los datos solicitados.

### 3.3 Sistema de visualización

El sistema requiere una parte gráfica para visualizar los datos que se registraron mediante las lecturas, y de esa manera tener un control de nuestro consumo eléctrico. Por ello se realizó una página web donde el usuario pueda consultar tanto en tiempo real como en lapsos de tiempo el consumo eléctrico de su hogar, así como consultar un aproximado de la tarifa mensual que CFE brinda a sus usuarios dependiendo de su consumo en kiloWatt hora o kWh. Una interfaz que sea de fácil consulta y que cualquier persona lo pueda utilizar.

#### 3.3.1 Sistema gráfico de los datos

La página web maneja XML lo que la hace más fácil de comunicar con la API, pues comparten los mismos tipos de comandos de peticiones web para llevar a cabo las consultas. Para ello nos apoyamos de el manejo de Bootstrap.

Bootstrap (figura 3.12) 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 CSS 3, tiene compatibilidad con sitios web o con aplicaciones dedicados a dispositivos y de navegadores, el diseño gráfico de la pagina 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.
- 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 Boostrap 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.

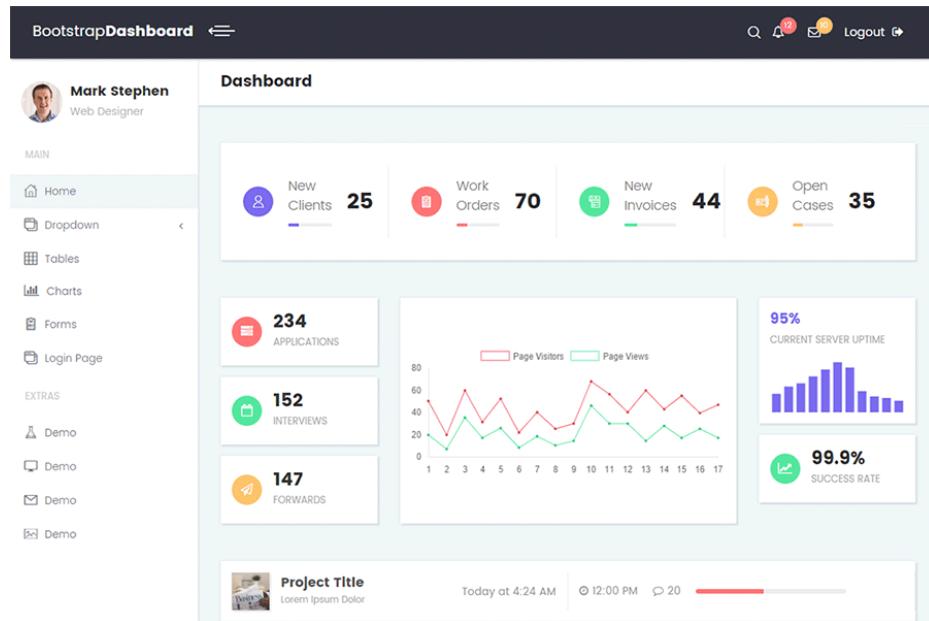


Figura 3.12 Vista previa de Bootstrap

AdminLTE (figura 3.13<sup>11</sup>) 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.

11 AdminLTE (2014-2017). *AdminLTE Control Panel Template*. Recuperado de <https://adminlte.io/>

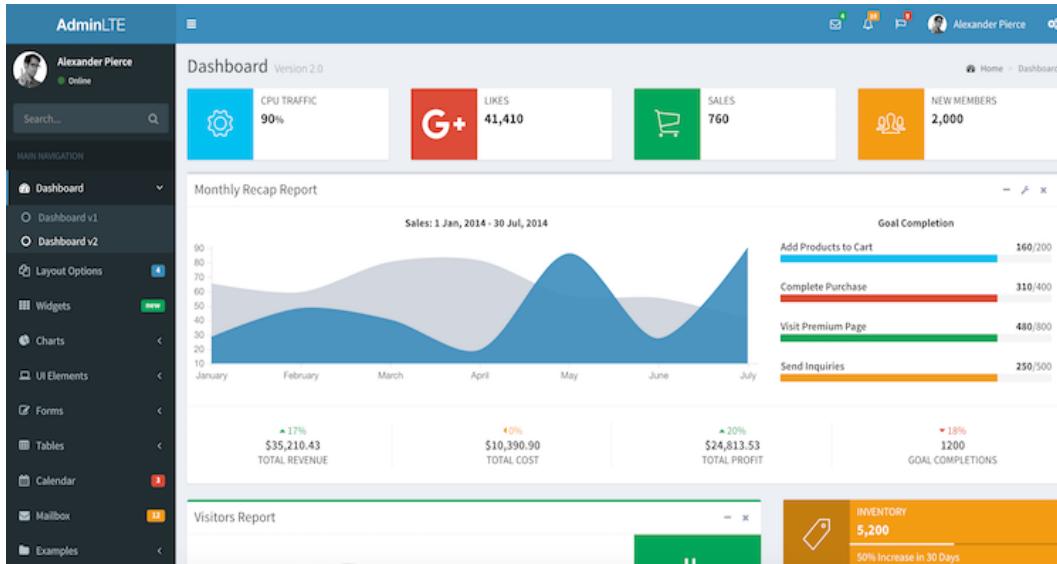


Figura 3.13 Vista previa de la interfaz AdminLTE

El entorno gráfico para la consulta de los datos consiste en una página para llevar acabo el acceso al sistema y otra página dedicada para la visualización de los datos. Esto es para poder llevar un control y protección al momento de acceder a consulta de los datos y de su visualización.

Una vez accedido al sistema de la manera requerida, el sistema redirige a la página de consulta y visualización de los datos. La interfaz está constituida por lo siguiente:

- Un bloque para visualizar de manera gráfica el historial de consumo (Figura 3.14) 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 (Grafica Historial) la cual tiene una relación intensidad de corriente contra tiempo.

TesterLTE

Resumen de consumo

Corriente IRMS Registrado [I]

Get History Get Send

Introduzca los datos que se le solicitan:

<b>Id de dispositivo :</b>	Pi001
<b>Fecha inicial :</b>	mm/dd/yyyy
<b>Hora inicial (Formato 24 Hrs) :</b>	HH:mm
<b>Fecha final :</b>	mm/dd/yyyy
<b>Hora final (Formato 24 Hrs) :</b>	HH:mm

Grafica Historial

Consumo de Corriente [A]

10.0  
0.0

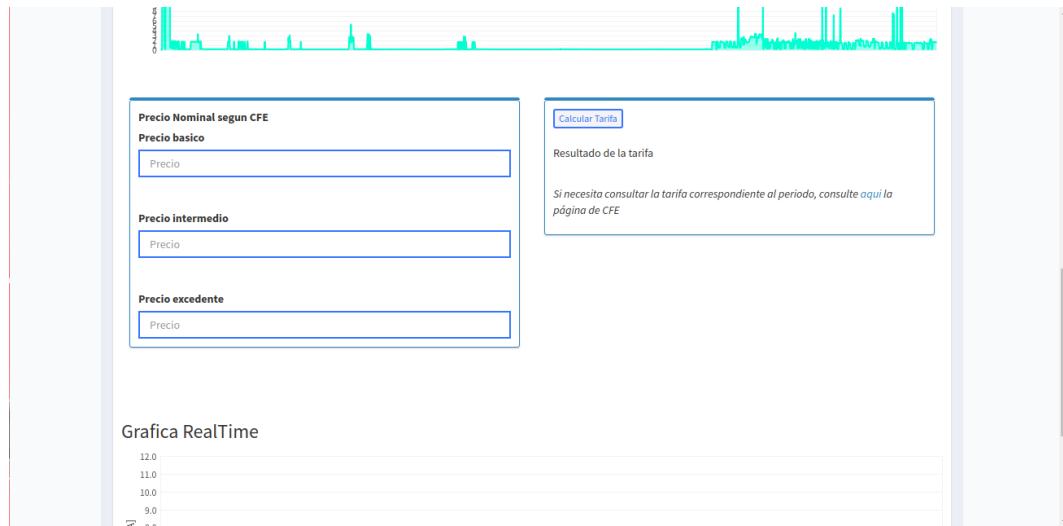
Figura 3.14 Consulta del consumo eléctrico histórico

- Otro bloque 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 3.15). Esta gráfica se actualiza cada 10 segundos con los últimos datos recolectados por el dispositivo en intervalos de una hora.



*Figura 3.15 Gráfica en tiempo real*

- El último bloque se dedica a obtener un precio estimado por parte de CFE (Figura 3.16) 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 y 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 kW (kiloWatts) y a cuanto equivale aproximadamente en tarifa a pagar.



*Figura 3.16 Cálculo de la tarifa por parte de CFE*

### 3.3.2 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 JavaScript asíncrono y 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.
- XML como formato usado para la transferencia de datos solicitados al servidor.

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.

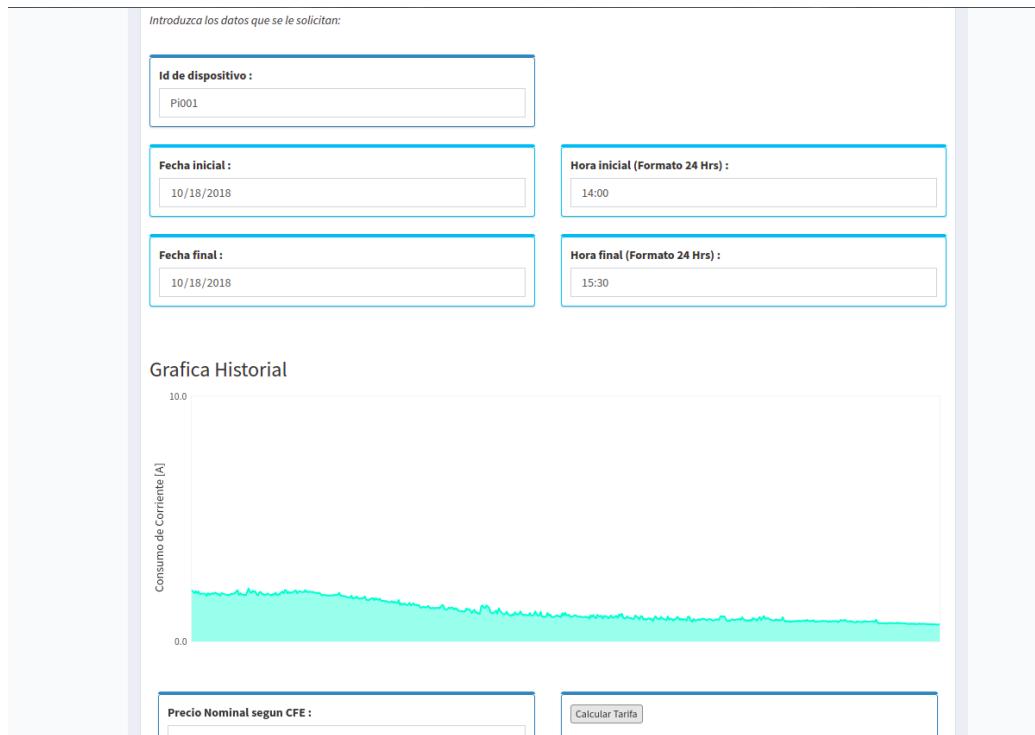
Para ambas funciones el sistema de solicitud de los datos hacia la base de datos se realiza con ayuda de AJAX para realizar la petición hacia la base datos. Se tienen ya predeterminado la url '<https://www.labmovilidad.unam.mx/tesismonitor/api/consult?>' a la cual se le añaden los parámetros id de dispositivo, 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 3.17) 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.

- 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 3.17 Vista previa de la gráfica de historial*

Para que la consulta en tiempo real y la gráfica en tiempo real lleven a cabo su función es 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 que corresponde a un valor máximo, y a partir de esa misma fecha se obtiene la fecha mínima restando cinco minutos a la fecha máxima.
- 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 funcione es la siguiente:

- 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 por 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 cada lectura de  $I_{rms}$  y hace el cálculo de la potencia en ese instante. La fórmula para obtener la potencia esta dada por la Fórmula 3.5. El valor del voltaje en este caso esta dado como VoltajeMedido = 127V, como se había mencionado anteriormente.
- Teniendo la potencia de cada uno de los intervalos de tiempo, se procede a sumar todas las potencias de los intervalos como lo muestra la Fórmula 3.6. La división entre 1000 es para convertir los watts a kiloWatts. Los cálculos obtiene la potencia total por mes, por lo cual si se desea consultar un mes diferente, se debe realizar una nueva consulta con diferente tarifa.

$$PotenciaTotal = \frac{\left( \sum_{n=fmin}^{fmax} PotenciasLeidas \right)}{1000}$$

*Fórmula 3.6 Suma de la potencia*

- Para calcular el consumo eléctrico de los kiloWatt/hora se requiere de la suma de la potencia obtenida, y con el tiempo que pasa entre la fecha inicial y final obtener el número de horas que hubo en ese periodo. Para calcular el tiempo en horas se utiliza una conversión de milisegundos a

horas, pues la diferencia entre las fechas las muestra en milisegundos. Todo el cálculo se realiza por medio de la Fórmula 3.7.

$$HorasTotales = (TiempoFinal - TiempoInicial) * \left( \frac{1 \text{ Segundo}}{1000 \text{ Milisegundos}} \right) * \left( \frac{1 \text{ Minuto}}{60 \text{ Segundo}} \right) * \left( \frac{1 \text{ Hora}}{60 \text{ Minutos}} \right)$$

Fórmula 3.7 Cálculo de horas totales

- Por último, con los datos obtenidos anteriormente, se puede calcular el kiloWatt hora por medio de la Fórmula 3.8. Con esto podemos estimar en que rango de la tarifa entra el valor obtenido. Para eso lanza un mensaje con la cantidad obtenida y el usuario pueda consultar que tarifa le corresponde.

$$kWh = \frac{PotenciaTotal}{HorasTotales}$$

Fórmula 3.8 Cálculo de kiloWatt/Hora

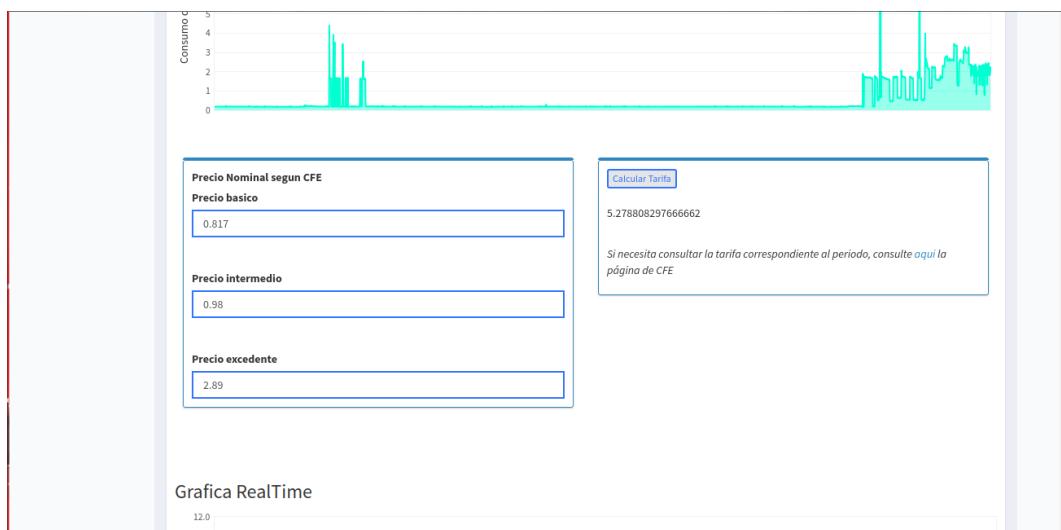
- Para obtener el precio total como lo hace CFE, se realiza conforme las tarifas dependiendo de cuantos kiloWatt/hora por medio de multiplicaciones con la cantidad de kiloWatt/hora previamente calculada y con los precios que CFE maneja. Se calcula por medio de una de las formulas (Fórmula 3.9) que se aplican dependiendo de el consumo mensual: Si se maneja un consumo menor o igual a los 75kWhrs se aplica costo total básico; si el rango esta entre los 75 y los 140kWhrs se maneja un costo total Intermedio; y si el rango sobrepasa los 140kWhrs 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 3.18.

$$CostoTotalBasico = kWh * TarifaBasica$$

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

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

**Fórmula 3.9 Costo total por kiloWatt Hora**



**Figura 3.18 Vista previa del cálculo de la tarifa según lineamientos CFE**

### 3.3.3 Robustez de la página

La página tiene funciones en caso de no dar 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, marca un error de id vacío puesto que necesita el número del dispositivo para realizar la búsqueda. En el caso de los campos de fecha mínima y máxima se puede presentar el caso que estén vacíos mientras exista un id de búsqueda, dando así un resultado de todas las lecturas del dispositivo hasta el momento. En el campo de precio tarifa, también verifica si se encuentra un precio escrito.

- La misma api cuenta con una función para limitar los datos. 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 año, 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.

## 4 Pruebas y resultados

En este capítulo se detalla la implementación del sistema de consumo eléctrico en ambientes reales y que tan eficiente es su funcionamiento comparado con sistemas que se encuentran en el mercado, o herramientas que hacen la misma función como lo son 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 Comparación con sistema comerciales

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 de dos dispositivos: uno de ellos es un multímetro con gancho para medir la corriente en uno de los cables de las terminales eléctricas, mientras que el otro es el medidor de corriente de CFE que se encuentra en cada casa para llevar la medición del consumo de corriente en el hogar.

#### 4.1.1 Comparación con un medidor de corriente de gancho comercial

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 600 A. Al ser un multímetro con gancho es de fácil uso para conectarlo a una terminal de la alimentación como se muestra en la Figura 4.1.

Este gancho se colocó en dos diferentes puntos de un hogar. En el primero de ellos se colocó el gancho para medir un cuarto dedicado 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 conecto cargadores de celular a 1A de consumo además que de en ciertos periodos de conectó el cargador que recarga la batería del dispositivo.

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 que tan parecidas eran las lecturas entre sí.



*Figura 4.1 Multímetro profesional de gancho RMS con auto rango marca Steren*

Los días que se tomaron en cuenta fueron el 31 de agosto de 2019 y el 01 de septiembre de 2019 desde las 10 horas hasta las 20 horas dando como resultado los valores que se encuentran en la Tabla 2. Como resultado de los dos dispositivos colocados en el cuarto de costura se pudo apreciar que entre las mediciones del sistema embebido y el gancho comercial de steren las variaciones son de 0.1 A, haciendo que el sistema tenga poco margen de error.

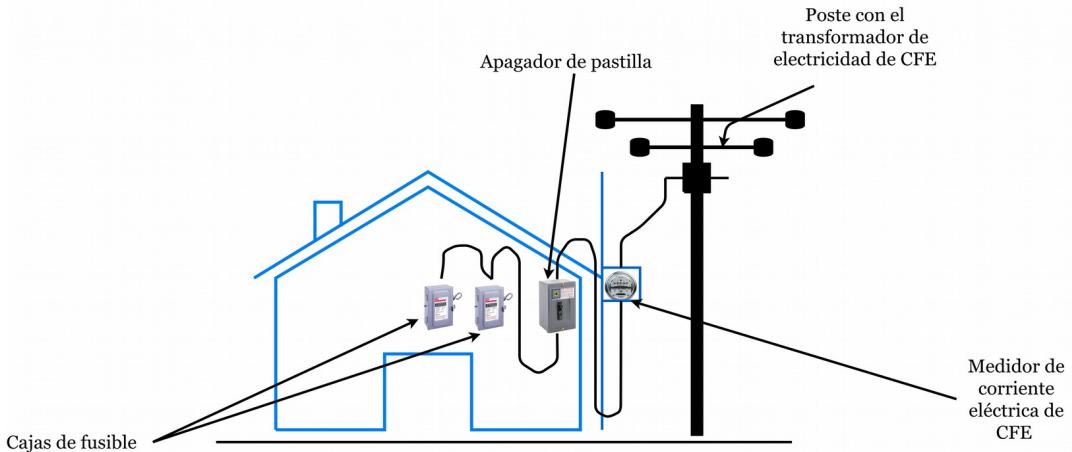
Día	Hora	Medición Corriente del gancho comercial [A]	Medición Corriente del sistema [A]	Día	Hora	Medición Corriente del gancho comercial [A]	Medición Corriente del sistema [A]
31 de Agosto de 2019	10:00 hrs		0.186A	01 de Septiembre de 2019	10:00 hrs		0.183A
	11:00 hrs		3.231A		11:00 hrs		0.222A
	12:00 hrs		0.201A		12:00 hrs		0.207A
	13:00 hrs		0.193A		13:00 hrs		0.194A
	14:00 hrs		0.183A		14:00 hrs		1.696A
	15:00 hrs		1.742A		15:00 hrs		0.189A
	16:00 hrs		0.188A		16:00 hrs		0.191A
	17:00 hrs		0.191A		17:00 hrs		0.187A
	18:00 hrs		0.188A		18:00 hrs		1.635A
	19:00 hrs		0.19A		19:00 hrs		0.207A
	20:00 hrs		0.19A		20:00 hrs		0.195A

Tabla 2 Tabla comparativa en el cuarto de costura

#### 4.1.2 Comparación con el medidor puesto por CFE

Una vez comparado con el gancho comercial, el dispositivo se colocó dentro de una casa, en la parte del medidor de luz de la compañía CFE. Esta instalación se encuentra cableada como indica la Figura 4.2: 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 4.2 Cableado eléctrico de la casa*

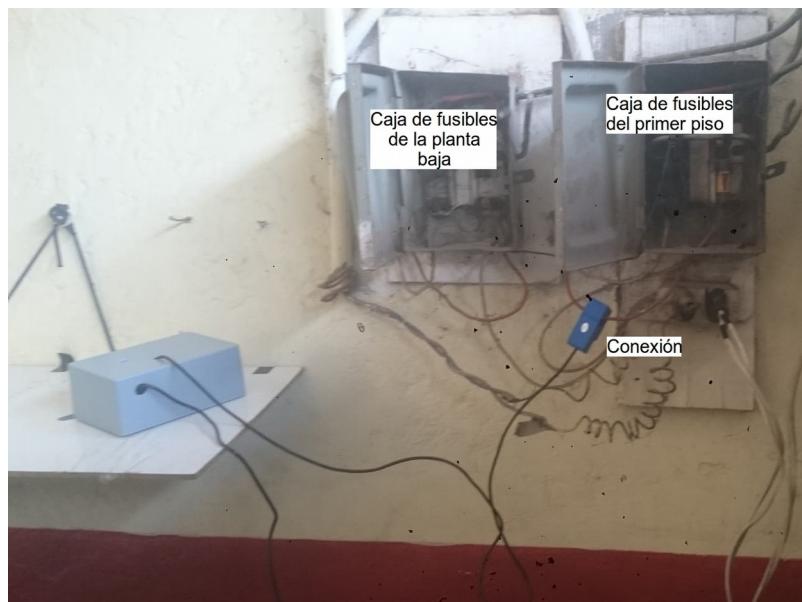
Se tiene en cuenta que como es un solo medidor se encarga de las mediciones de la planta baja y el primer piso, se puede obtener el valor a partir del valor del medidor de CFE y con ayuda del sensor de gancho antes utilizado tener la medición de la planta baja. Podemos visualizar el valor por medio de la Fórmula 4.1, donde la medición CFE es el valor que nos arroja el medidor de CFE en ese instante de tiempo, mientras que la medición de planta baja es la lectura que se obtiene del medidor de gancho conectado en la caja de fusibles correspondiente a la planta baja.

$$\text{MediciónPrimerPiso} = \text{MediciónCFE} - \text{MediciónPlantaBaja}$$

*Fórmula 4.1 Lectura de consumo eléctrico para el primer piso*

Una vez identificado el cableado de la casa, el dispositivo se colocó en una de las terminales de la caja de fusibles que sube hacia la instalación eléctrica que se encuentra en el primer piso de la casa. El primer piso cuenta con una cocina, una sala y tres habitaciones, por lo cual se cuenta con algunos electrodomésticos de uso cotidiano como lo son refrigerador, microondas, plancha eléctrica, televisión, lavadora, licuadora, y radio; por lo cual el dispositivo al conectarlo en una terminal del cableado del primer piso registra el consumo eléctrico de lo que hayan consumido los electrodomésticos en dos períodos de un día cada uno.

La conexión se lleva acabo como se muestra en la Figura 4.3, colocando el gancho medidor en una de las terminales de la caja de fusibles, en este caso en la terminal de la caja del primer piso y pueda llevar acabo las lecturas de cada día. Los valores registrados que se encuentran en la Tabla 3 son las comparaciones que se obtuvieron de realizar la operación con la Fórmula 4.1 y los valores que el dispositivo registro en un rango de las horas de más alto consumo eléctrico.



*Figura 4.3 Instalación del dispositivo*

Al momento de comparar los valores obtenidos como del dispositivo, así como de la lectura del medidor, es mínima la diferencia entre cada valor obtenido

Día	Hora	Medición Corriente con la Fórmula 4.1 [A]	Medición Corriente del sistema [A]	Día	Hora	Medición Corriente con la Fórmula 4.1 [A]	Medición Corriente del sistema [A]
13 de Septiembre de 2019	09:00 hrs			14 de Septiembre de 2019	09:00 hrs		
	10:00 hrs				10:00 hrs		
	11:00 hrs				11:00 hrs		
	12:00 hrs				12:00 hrs		
	13:00 hrs				13:00 hrs		
	14:00 hrs				14:00 hrs		
	15:00 hrs				15:00 hrs		
	16:00 hrs				16:00 hrs		
	17:00 hrs				17:00 hrs		
	18:00 hrs				18:00 hrs		
	19:00 hrs				19:00 hrs		

Tabla 3 Resultados de la medición con el medidor CFE

## 4.2 Pruebas con el sistema en ambiente real

4.2.1 Prueba en un hogar

4.2.2 Prueba en una estación de carga

## 4.3 Análisis de consumo durante un periodo

# 5 Conclusiones y trabajo futuro

## 5.1 Conclusiones

## 5.2 Trabajo futuro

# 6 Bibliografia

<https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>  
<http://www.isa.cie.uva.es/~maria/sensores.pdf>  
<https://www.st.com/en/amplifiers-and-comparators/lm358.html>  
<https://www.adafruit.com/product/3055>  
<https://store.arduino.cc/usa/arduino-nano>  
<https://diego.com.es/introduccion-a-los-web-services>  
<http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/wholesite.pdf>

Avelino Peréz, 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.

CFE (2019) CFE. Comisión Federal de Electricidad.

<https://app.cfe.mx/Applicaciones/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\\_ISE\\_3\\_1.pdf](http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_ISE_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](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%A9nica-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 la materia de Bases de Datos [PDF file]. Recuperado de [http://fcasua.contad.unam.mx/apuntes/internos/docs/98/6/bases\\_datos.pdf](http://fcasua.contad.unam.mx/apuntes/internos/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](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 SENORES INALÁMBRICOS: REVIEW [PDF File]. Recuperado de [https://www.researchgate.net/publication/316438631\\_INTERNET\\_DE\\_LAS\\_COSAS\\_Y\\_RED\\_ES\\_DE\\_SENORES\\_INALAMBRICOS REVIEW](https://www.researchgate.net/publication/316438631_INTERNET_DE_LAS_COSAS_Y_RED_ES_DE_SENORES_INALAMBRICOS REVIEW)

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*. [https://www.steren.com.mx/multimetro-profesional-de-gancho-rms-con-auto-rango.html?gclid=CjwKCAjwkqPrBRA3EiwAKdtwk9EQ9WiinB\\_laoxVn3A0S9EHFAr0NSZzjPrSYBu9JHZg2Vag5rMzmRoC7I4QAvD\\_BwE](https://www.steren.com.mx/multimetro-profesional-de-gancho-rms-con-auto-rango.html?gclid=CjwKCAjwkqPrBRA3EiwAKdtwk9EQ9WiinB_laoxVn3A0S9EHFAr0NSZzjPrSYBu9JHZg2Vag5rMzmRoC7I4QAvD_BwE)

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

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.

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