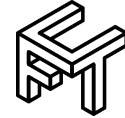




Universidad de Matanzas
Facultad de Ciencias Técnicas



SISTEMA PARA EL MONITOREO DEL CONSUMO DE ENERGÍA ELÉCTRICA EN TIEMPO REAL

Tesis Presentada como Requisito Parcial
para la Obtención del Título de
Máster en Ingeniería Asistida por Computadora

Autor: Lic. Ramón de Jesús López Suárez

Tutor: Dr. C. Ramón Quiza Sardiñas

Matanzas, 2020

DECLARACIÓN DE AUTORIDAD Y NOTA LEGAL

Yo, Ramón de Jesús López Suárez, declaro que soy el único autor de la siguiente tesis, titulada “*Sistema para el monitoreo del consumo de energía eléctrica en tiempo real*” y, en virtud de tal, cedo el derecho de copia de la misma a la Universidad de Matanzas, bajo la licencia *Creative Commons* de tipo *Reconocimiento No Comercial Sin Obra Derivada*, con lo cual se permite su copia y distribución por cualquier medio siempre que mantenga el reconocimiento de sus autores, no haga uso comercial de la obra y no realice ninguna modificación de ella.

Matanzas, 23 de septiembre de 2020.

Ramón de Jesús López Suárez

RESUMEN

La revolución industrial 4.0 y la evolución progresiva del internet de las cosas incrementaron la demanda del uso de sensores y medios automatizados. El ahorro de energía eléctrica es un tema de actualidad por su importancia desde el punto de vista económico y medioambiental. Este trabajo se propuso elaborar un sistema para la medición del consumo de energía eléctrica en tiempo real, que permita medir, registrar, visualizar y analizar su comportamiento, así como determinar su costo. Se diseñó un divisor de tensión que se montó sobre una placa y se conectó a un Arduino Nano para acoplar un transformador de corriente HWCT004 con una relación de transformación de 50A / 50 mA. Se desarrolló un programa con el empleo de Programming/Wiring para el Arduino y C# en el entorno de desarrollo Visual Studio Community 2019 para producir una aplicación de escritorio con una interfaz de usuario intuitiva, ligera, portable y aplicable, tanto a computadoras personales, como a portátiles en el sistema operativo Windows. Se obtuvo un sistema que permite monitorear (medir, registrar, visualizar) el consumo de energía eléctrica y luego analizar tales mediciones para valorar el comportamiento del proceso en cada instante de tiempo medido. La validación del sistema en los casos estudiados demostró que erradica los errores e imprecisiones provocadas por las mediciones manuales, facilita que se obtengan recomendaciones apropiadas para la optimización y el ahorro de energía eléctrica, emite el gasto del proceso medido y contribuye a su sostenibilidad.

Palabras claves: Industria 4.0, Arduino, automatización, control, IoT, sostenibilidad.

ABSTRACT

The industrial revolution 4.0 and the progressive evolution of the internet of things increased the demand for the use of sensors and automated media. Saving electrical energy is a current topic due to its importance from an economic and environmental point of view. This work was proposed to develop a system for the measurement of electrical energy consumption in real time, which allows to measure, record, visualize and analyze its behavior, as well as determine its cost. A voltage divider was designed that was mounted on a board and connected to an Arduino Nano to couple a HWCT004 current transformer with a 50A / 50 mA turn ratio. A program was developed with the use of Programming / Wiring for the Arduino and C # in the Visual Studio Community 2019 development environment to produce a desktop application with an intuitive, lightweight, portable user interface, applicable to both personal computers and computers. to notebooks on Windows operating system. A system was obtained that allows to monitor (measure, record, visualize) the consumption of electrical energy and then analyze such measurements to assess the behavior of the process at each moment of time measured. The validation of the system in the cases studied showed that it eradicates errors and inaccuracies caused by manual measurements, makes it easier to obtain appropriate recommendations for optimization and saving of electrical energy, emits the expense of the measured process and contributes to its sustainability.

Keywords: Industry 4.0, Arduino, automation, control, IoT, sustainability.

TABLA DE CONTENIDO

| | |
|--|----|
| Introducción | 1 |
| Capítulo 1. Estado del Arte..... | 7 |
| 1.1 Antecedentes..... | 7 |
| 1.2 Sistemas SCADA..... | 10 |
| 1.3 Sistema de adquisición de datos de señal de corriente eléctrica..... | 11 |
| 1.4 Soluciones integradas..... | 12 |
| 1.5 Tecnologías dominantes..... | 12 |
| 1.6 Tecnologías Integradas a la Revolución Industrial 4.0..... | 15 |
| 1.7 ROFLEXIN/LC..... | 18 |
| 1.8 Conclusiones Parciales del Capítulo..... | 19 |
| Capítulo 2. Materiales y Métodos..... | 20 |
| 2.1 Sensor de corriente HWCT004..... | 20 |
| 2.2 Circuito acondicionador..... | 21 |
| 2.3 Arduino NANO..... | 22 |
| 2.4 Herramientas y tecnologías utilizadas en la investigación | 24 |
| 2.4.1 Lenguajes de Programación..... | 24 |
| 2.4.2 Aplicaciones de Escritorio | 27 |
| 2.5 Redes Neuronales Artificiales (RNA) | 34 |
| 2.5.1 Concepto de Red Neuronal | 35 |
| 2.5.2 Composición de las redes neuronales | 36 |
| 2.5.3 Tipos de Redes Neuronales..... | 37 |
| 2.5.4 Clasificación de las RNA según el aprendizaje | 38 |

| | | |
|-------------|---|----|
| 2.5.5 | Uso de algoritmos de aprendizaje no supervisado en el análisis de datos ... | 39 |
| 2.5.6 | Ventajas y desventajas de las redes neuronales | 42 |
| 2.6 | Metodología SCRUM..... | 43 |
| 2.6.1 | Requisitos del <i>software</i> | 43 |
| 2.6.2 | Requisitos funcionales | 44 |
| 2.6.3 | Requisitos no funcionales | 46 |
| 2.6.4 | Etapa de planificación..... | 49 |
| 2.6.5 | Definición del Equipo | 49 |
| 2.6.6 | Historias de usuarios | 50 |
| 2.6.7 | Pila del Producto (<i>Product Backlog</i>) | 51 |
| 2.6.8 | Planeación de las Entregas (<i>Sprints Backlog</i>)..... | 52 |
| 2.7 | Acondicionamiento de la señal | 55 |
| 2.7.1 | Empleo de la raíz media cuadrática (RMS) | 56 |
| 2.7.2 | Ajuste de la curva mediante el método spline cúbico..... | 57 |
| 2.7.3 | Filtro de Chebyshev | 59 |
| 2.7.4 | Mapa Auto-Organizado | 60 |
| 2.8 | Estimación del costo | 64 |
| 2.8.1 | Coste de personal | 65 |
| 2.8.2 | Coste de <i>hardware</i> | 65 |
| 2.8.3 | Coste de <i>software</i> | 66 |
| 2.8.4 | Coste total | 66 |
| 2.9 | Conclusiones parciales del capítulo | 67 |
| Capítulo 3. | Análisis de los Resultados..... | 68 |

| | |
|---|----|
| 3.1 Multímetro FLUKE | 68 |
| 3.2 Caso de estudio 1: Monitoreo y medición del consumo de una impresora 3D durante la impresión de una pieza..... | 69 |
| 3.3 Caso de estudio 2: Monitoreo y medición del consumo de un aire acondicionado tipo Split de 1 tonelada de capacidad..... | 74 |
| 3.4 Conclusiones parciales del capítulo | 76 |
| Conclusiones | 78 |
| Recomendaciones | 79 |
| Referencias Bibliográficas | 80 |
| Anexo I..... | 86 |
| Multímetros Fluke. Especificaciones eléctricas..... | 86 |

GLOSARIO DE TÉRMINOS

| | |
|-------------|--|
| .NET | <i>Framework</i> de Microsoft. |
| API | <i>Application Programming Interface</i> o, Interfaz de programación de aplicaciones. |
| BMU | <i>Best matching unit</i> o, Unidad mejor correspondida |
| COVID-19 | Corona Virus <i>Desease</i> 2019 |
| ECMA | ECMA <i>International</i> es una organización internacional basada en membresías de estándares para la comunicación y la información. |
| EEPROM | <i>Electrically Erasable Programmable Read Only Memory</i> . Memoria que se puede programar y borrar múltiples veces |
| GDI+ | <i>Graphics Device Interface</i> , es uno de los tres componentes o subsistemas de la interfaz de usuario de <i>Microsoft Windows</i> . Trabaja junto con el núcleo y la <i>API de Windows</i> . |
| IDE | <i>Integrated Development Enviroment</i> o, Entorno de Desarrollo Integrado |
| IoT | Internet de las cosas, <i>Internet of Things</i> |
| ISO | Organización Internacional de Normalización |
| LED | <i>Light-emitting diode</i> o, Diodo emisor de luz. |
| PWM | <i>Pulse-width modulation</i> o, Modulación por ancho de pulso |
| RNA | Redes Neuronales Artificiales |
| ROFLEXIN/LC | Sistema Robusto Flexible e Inteligente de Bajo Costo |
| SCADA | <i>Supervisory Control and Data Acquisition</i> o, Adquisición de datos y supervisión de Control |
| SIEM | <i>Security Information and Event Management</i> . Gestión de eventos e información de seguridad. |
| SOM | <i>Self Organizing Maps</i> o, Mapas auto-organizados |
| SRAM | <i>Shared Random Access Memory</i> . Memoria compartida de acceso aleatorio |

INTRODUCCIÓN

Existe un amplio consenso entre especialistas y expertos acerca de que el conjunto de tecnologías que se conoce como Internet de las Cosas (IoT por sus siglas en inglés), va a producir un enorme impacto en la sociedad, al transformar la vida cotidiana, el mundo laboral, los negocios y prácticamente todos los ámbitos de la sociedad (Cruz Vega *et al.*, 2015). Este trabajo es un paso en ese sentido.

La energía eléctrica desde su descubrimiento e implementación jugó un papel fundamental en el desarrollo de la humanidad, debido a su gran versatilidad y posibilidad de control, a la inmediatez en su utilización y a la limpieza en el punto de consumo. En la vida diaria de las personas se emplea para proporcionar toda clase de servicios, desde los más básicos hasta los asociados al ocio y a un sinnúmero de comodidades (luz, refrigeración de alimentos, climatización, radio, televisión, ordenadores, entre otros) (López Suárez, 2019). También el uso de la electricidad está igualmente extendido en los ámbitos comercial e industrial: alumbrado, climatización, motores eléctricos con multitud de aplicaciones de industria, con una utilización específica o intensiva. Es muy difícil reemplazarla en la mayoría de sus usos y aplicaciones, por lo que puede afirmarse que la calidad de vida y el propio funcionamiento de la sociedad depende de forma significativa de la disponibilidad de energía eléctrica (Abur *et al.*, 2002).

En el caso de Cuba, no puede olvidarse que cualquier análisis de la economía cubana pasa por la realidad del bloqueo imperialista, que funciona como una barrera adicional al desarrollo (adicional a las barreras económicas que los países ricos construyen contra el desarrollo de todos los países del sur). El bloqueo funciona como un elemento de disuasión

o riesgo en casi todas las negociaciones. Solamente se puede perforar con importantes innovaciones; no con pequeñas mejoras de productos, las cuales pueden tener cierta importancia después (Lage Dávila, 2007).

Los principales consumidores de energía eléctrica en Cuba se concentran en el sector doméstico con 36,3 %, seguido por la industria con 25,4 % y en menores proporciones, el comercio, la agricultura, el transporte y la construcción, con alrededor del 5,3 %, el resto son consumidores misceláneos con el 18,7 % (Suárez Rodríguez *et al.*, 2011).

Es imprescindible el ahorro de energía eléctrica, debido a que en el país se utiliza principalmente combustible importado para producir electricidad. Se aspira a un aumento en el uso de energía de fuentes renovables como la energía eólica y la energía solar (Suárez Rodríguez y Beatón Soler, 2007). El petróleo que se extrae de los pozos disponibles en el territorio nacional, presenta un alto contenido de azufre, motivo por el cual no se puede quemar directamente en las termoeléctricas (García y Toledo, 2004). Por ello es necesario importar combustible desde el exterior y rentar barcos que lo transporten, luego se almacena y se traslada hasta las termoeléctricas; incluso, antes de producir energía ya representa un costo elevado (Núñez Barreto, 2020).

Otro elemento a tener en cuenta es que el ahorro de energía eléctrica es una medida de mitigación al cambio climático (SPA, 2015), debido a que por cada kW/h que se genera en una planta térmica donde se quema petróleo, se envían a la atmósfera como promedio 700 gramos de CO₂, el gas acompañante que junto a otros como el metano, es causante del calentamiento global y del cambio climático que ocurre a escala planetaria y cuyas

consecuencias se dejarán sentir en todos los países de un modo o de otro (Núñez Barreto, 2020).

Con el análisis del desempeño energético, se puede determinar sus usos significativos y sus principales potenciales de perfeccionamiento, para así establecer objetivos de mejora (Carretero Peña y García Sánchez, 2012). Debido a esto, en Cuba se desarrollan muchas estrategias de ahorro de energía eléctrica, estas se reflejan en la actualización de los lineamientos de la política económica y social del Partido y la Revolución Cubana (Núñez Jover y Montalvo Arriete, 2015).

En la actualidad, en Cuba como en el resto del mundo, tanto en el sector estatal como en el residencial, se cuenta con un sistema de monitoreo de consumo de energía eléctrica llamado “metro contador”, a través del que se realiza la medición del consumo para el cobro de la energía eléctrica. Estos dispositivos indican numéricamente el consumo en hectovatios y kilovatios (hW y kW), la recopilación de la información suele ser manual y se realiza por una persona en el lugar mensualmente, lo que provoca que se pierda información valiosa sobre el comportamiento de las variables energéticas en función del tiempo, las que pudieran utilizarse en el análisis y la toma de decisiones para hacer más eficiente el uso de la energía eléctrica.

Para análisis más específicos, existen otros instrumentos de medida, como es el caso de los vatímetros y los analizadores de red, estos últimos, ofrecen la posibilidad de almacenar mediciones, aunque son muy costosos y por lo general no rentables en pequeños procesos productivos.

Por todo lo anterior, la medición del consumo eléctrico y su análisis para la determinación de medidas de ahorro y optimización de procesos productivos es un tema de gran actualidad. No obstante, existe un gran número de ellos en los que el desconcierto y la falta de información sobre el consumo de energía que se genera, imposibilitan el desarrollo de medidas de eficiencia energética (Manqueros Avilés *et al.*, 2019).

Para obtener este conocimiento del comportamiento, no solo de la evolución del consumo eléctrico, sino también de cualquier otro tipo de variable energética, se requiere de un sistema de supervisión básico que permita medir, registrar, visualizar y analizar estas variables (Pérez Barreto, 2019). Además de que reducir tiempo y aumentar el grado de confiabilidad, hace que el factor tecnológico y el recurso información, tomen otro valor más significativo para las empresas o instituciones e incluso, en el sector doméstico.

En el proceso de informatización de la sociedad se involucra la industria cubana. Si se tiene en cuenta que las condiciones del bloqueo que impone los Estados Unidos a Cuba, impiden la adquisición de tecnologías de punta y que como estrategia gubernamental se procura el desarrollo de tecnologías propias que garanticen independencia tecnológica (López Suárez, 2019), el desarrollo de un “*Sistema para el monitoreo del consumo de energía eléctrica en tiempo real*” que satisfaga las condiciones antes mencionadas y que pueda adecuarse para el proyecto ROFLEXIN/LC, es el **problema científico** a resolver. Su novedad radica en que no se trata de un sistema específico para un proceso particular, sino que se puede utilizar universalmente en todos los procesos que requieran análisis en el rango de corriente y fases admitidas por el sensor, lo que permite que se emplee indistintamente, tanto en el sector doméstico, como en el industrial y obtener un conjunto de datos almacenables para su posterior utilización.

De ahí surge la **hipótesis** de que, si se desarrolla un sistema para el monitoreo del consumo de energía eléctrica en tiempo real, que permita medir, registrar, visualizar y analizar dicho consumo con exactitud, posibilitará el desarrollo de medidas de eficiencia energética y sostenibilidad, además de ser adecuado para incorporarse al proyecto ROFLEXIN/LC.

En concordancia con lo anterior se propone como **objetivo general** el siguiente: Elaborar un sistema para la medición del consumo de energía eléctrica en tiempo real que permita la medición, el registro, visualización y el análisis de su comportamiento en el tiempo.

Como **objetivos específicos**, a su vez, se plantean:

1. Diseñar el esquema electrónico del dispositivo para el acondicionamiento y procesamiento de las señales provenientes del sensor de corriente.
2. Fabricar el dispositivo electrónico que conforma la interfaz para el acondicionamiento de señales a partir del diseño realizado.
3. Desarrollar un programa que permita el monitoreo y control de la interfaz respecto a los sensores conectados a la misma.
4. Desarrollar una red neuronal multicapa para la creación del mapa autoorganizado de patrones de consumo cuando estos se definan.
5. Calibrar el sistema obtenido mediante el empleo de un instrumento de precisión demostrada.
6. Validar el funcionamiento de la interfaz con casos de estudio que permitan la demostración de su utilidad.

Esta investigación se realizó en el marco del proyecto “ROFLEXIN/LC: Sistema robusto, flexible e inteligente, de bajo costo, para el monitoreo de sistemas y procesos mecánicos”, asociado al Programa Nacional de Automatización de los Procesos Tecnológicos.

CAPÍTULO 1. ESTADO DEL ARTE

Este capítulo se propone fundamentar teóricamente cada uno de los elementos que condujeron a esta investigación. Se realiza un análisis histórico evolutivo de las tecnologías que dan origen al mismo y sustentan las herramientas con que se cuenta para su desarrollo. Toma como punto de partida el estado actual de la temática a nivel mundial, donde se valoran los trabajos realizados, herramientas utilizadas y posibles aplicaciones en este proyecto.

1.1 Antecedentes

Trabajos anteriores diseñaron e implementaron proyectos con propósitos similares al de esta investigación. Acosta *et al.* (2017) diseñaron un sistema de monitoreo de consumo eléctrico inalámbrico y analizador de la calidad de la energía para instalaciones que usan electricidad monofásica, el mismo se compone de dos partes fundamentales: la primera parte se enfoca en la medición de las variables eléctricas y hace uso de sensores a través de la placa controladora Arduino y la segunda parte, consta de la interfaz de usuario con la cual se graban los datos y se analizan las variables eléctricas. Para ello utilizan el software LabVIEW y el recibo de consumo eléctrico se genera mediante una aplicación *Visual Basic for Applications* (VBA, por sus siglas en inglés) en Microsoft Excel.

La información recopilada del programa se almacena y clasifica dentro de una plantilla del paquete computacional Microsoft Access 2016, todos son *software* propietario y en la manera que se conciben hacen compleja y específica su aplicación a su caso de estudio y emite un recibo que no se adapta a los objetivos que se persiguen, lo que hace que la

solución dada no sea la adecuada tanto por su costo, así como, por las condiciones del bloqueo de los Estados Unidos contra Cuba, que impide el uso de tecnologías propietarias dependientes de empresas americanas.

Martin (2018) desarrolló una investigación de cómo utilizar la arquitectura de *software* SIEM para el monitoreo en línea del consumo de energía eléctrica en un ambiente residencial y optimizar el mismo. Si bien esta es una solución que permite generar interfaces *web* en las que se despliegan los cambios del consumo y las variables energéticas en tiempo real, se plantea que se puede lograr implementar un prototipo funcional que permita mediante una arquitectura segura, interoperable, escalable y sostenible, optimizar el consumo energía eléctrica. Esta solución depende de una conexión libre a internet y hace uso de plataformas para desarrollo de IoT, a las que Cuba no tiene acceso, como es el caso de *Particle Cloud* (Poderosa pieza central de la plataforma *Particle*). Esta no se adecúa a las condiciones objetivas del país ni a los objetivos del sistema que se necesita, luego es imposible su utilización por el momento.

Ruiz Ariza (2019) desarrolló un sistema de control y monitoreo para sistemas de climatización. Este proyecto se enfoca en los equipos de climatización y permite monitorear de manera efectiva variables energéticas como es el caso de la corriente nominal y la corriente máxima, así como, la potencia consumida y la temperatura. El procesamiento de los datos en este proyecto se realiza a partir de un software externo, en este caso Matlab r2018b, y cuenta con una plataforma en línea para la visualización de las variables en tiempo real. Este proyecto demuestra la relación que existe entre la temperatura externa y el consumo de energía eléctrica, sin embargo, como es una solución

específica y depende de *software* propietario, no permite su aplicación en procesos del proyecto ROFLEXIN/LC.

Según Novillo Vicuña *et al.* (2018), el problema a resolver es la necesidad de monitorear inalámbricamente y a bajo costo, señales eléctricas de voltaje, para controlar la calidad de la energía y de esta manera evitar problemas económicos, de rendimiento o mal funcionamiento de los equipos electrónicos. El objetivo que se planteó fue diseñar una aplicación móvil mediante la utilización de los dispositivos Arduino Uno y Wifi Shield, transformadores de voltaje 110/220V, acondicionador de señal compuesto por un circuito divisor de voltaje - circuito offset, y Android Studio para la monitorización inalámbrica de señales eléctricas con voltajes de 110/220V. Este trabajo tiene aspectos muy interesantes, pero se utiliza Wifi Shield no disponible a causa de la COVID-19 y su objetivo dista del propuesto por la presente investigación.

Otros autores también propusieron sistemas para la medición del consumo eléctrico con el empleo de dispositivos Arduino, pero con objetivos limitados, como es el caso del trabajo cuyo objetivo fue desarrollar una aplicación móvil con la utilización de la plataforma Arduino y Android Studio para la medición y monitoreo de señales de corriente en un tablero de distribución eléctrica (Rivas Cárdenas, 2017). En este caso no se hace mención explícita al *software* y sus funcionalidades, por lo que no satisface los requerimientos mínimos para su empleo en el proyecto ROFLEXIN/LC.

La tendencia básica en la industria de las pruebas automatizadas es un cambio muy marcado hacia sistemas de pruebas basados en *software*. Por ejemplo, el Departamento de Defensa de los EE.UU. especificó, que en los futuros sistemas ATE (Equipo para Pruebas

Automatizadas), deben seguir una estructura modular de *hardware* y *software* reconfigurable, llamada instrumentación sintética. Este tipo de instrumentación refleja el claro cambio que deja a un lado el *hardware* como pieza fundamental del sistema y se centra más en el *software* (Montoro, 2016).

1.2 Sistemas SCADA

El sistema SCADA (*Supervisory Control and Data Acquisition* o Adquisición de datos y supervisión de Control), o conocido como adquisición de señales análogas o digitales de un sistema físico o de un proceso determinado como el de un sensor. Su función principal es recolectar información y digitalizarla para su posterior análisis y manipulación por una computadora.

Un sistema de adquisición contiene diferentes elementos que realizan labores específicas dentro de su proceso (Neuta Chiguasuque, 2013). El mismo se diseñó para automatizar el monitoreo y control de procesos industriales, se comunica con los dispositivos de campo y se controla de forma automática desde un máster o computadora principal.

Los sistemas SCADA (Figura 1.1) eliminan la necesidad de vigilar y ajustar físicamente los componentes del proceso, una red de sensores transmite información del estado de los componentes a una sala de operadores, que deciden si hay que realizar alguna modificación sobre el proceso. Muchas veces esta toma de decisiones se apoya por una unidad central que descarga al operario de tareas repetitivas, dejándole actuar sobre el sistema a muy alto nivel. La comunicación se realiza mediante buses especiales o redes de área local. Todo

esto se ejecuta normalmente en tiempo real, y se diseñan para dar al operador de planta la posibilidad de supervisar y controlar dichos procesos (Salazar Serna y Correa Ortiz, 2011).

La funcionalidad de un sistema SCADA (Herrera *et al.*, 2014) está dividida en tres bloques principales:

1. *Software* de adquisición de datos y control (SCADA).
2. Sistema de adquisición y mando (Sensores y actuadores).
3. Sistemas de interconexión (comunicaciones).

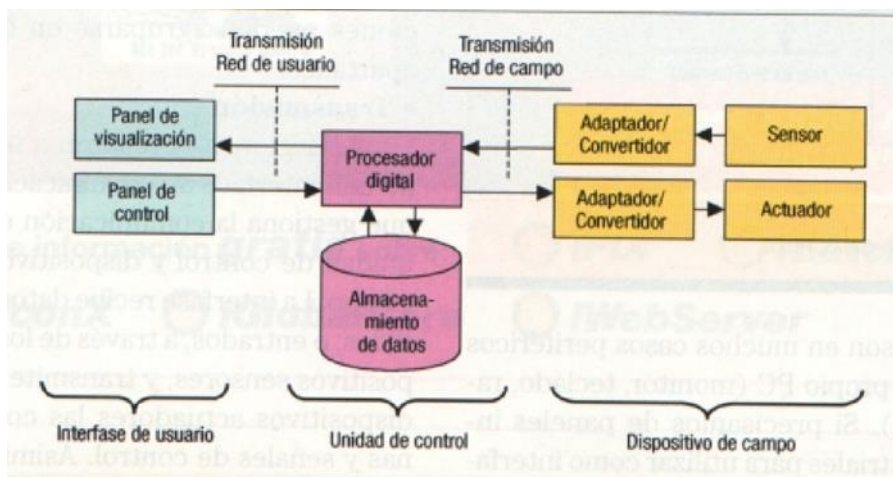


Figura 1.1. Esquema básico de un SCADA (Albán Guerrero y Paguay Llamuca, 2017).

1.3 Sistema de adquisición de datos de señal de corriente eléctrica

La eficiencia energética es la reducción de la potencia y energía eléctrica demandada en la red sin que afecte a las actividades normales realizadas en un edificio, industria o proceso de transformación (Núñez Barreto, 2020). El sistema de adquisición de datos se compone de dos elementos, el primero de ellos es el sensor, que es el encargado de convertir el valor

de la corriente eléctrica que circula por el conductor eléctrico, en este caso las líneas eléctricas en las que se requiera realizar la medición del consumo de energía y el segundo componente es el circuito acondicionador.

1.4 Soluciones integradas

Con el avance de la tecnología, se desarrollaron dispositivos que permitieron dar soluciones a problemas cotidianos o complejos en diversas áreas como: las comunicaciones, servicio militar, seguridad gubernamental, medicina, educación, industrias, empresas, entre otros (Vega *et al.*, 2014).

La estandarización es otro de los grandes retos que trae consigo la implementación de la Industria 4.0, convirtiéndose en el tema objeto de interés de los gobiernos, empresas y la comunidad científica para el desarrollo de soluciones, modelos o aplicaciones predictivas basadas en IoT (Civerchia *et al.*, 2017). En opinión de algunos autores existe una oportunidad para los investigadores en el contexto de la cuarta revolución industrial, porque si bien es cierto, que las redes de sensores no son un tema reciente, aún son un desafío cuando se trata de encontrar soluciones para el problema de la fusión automática de los datos, el procesamiento y la integración del gran volumen de información que generan estas fuentes heterogéneas (Cruz Vega *et al.*, 2015).

1.5 Tecnologías dominantes

En los últimos 60 años, la evolución del control de procesos sufrió fuertes cambios debido al aumento de las necesidades de los mismos procesos. La tecnología dominante (Figura 1.2) en la década de los cincuenta fue el control neumático con sistemas mecánicos.

Estos sistemas, que todavía pueden encontrarse en algunos procesos, se reemplazaron en los sesenta por tecnología analógica, mediante sistemas electrónicos (Tapia, 2017). En aquel momento, fue la velocidad de cálculo, el menor costo de instalación y bajo mantenimiento, junto con la popularización de la electrónica, lo que empujó a este gran cambio de tecnología. A partir de entonces, la medición y acciones correctoras se realizaban directamente sin amplificadores mecánicos, lo que simplificó y abarató el control. Y así se entró en la era analógica, que en gran medida todavía está vigente (Vásquez, 2008).

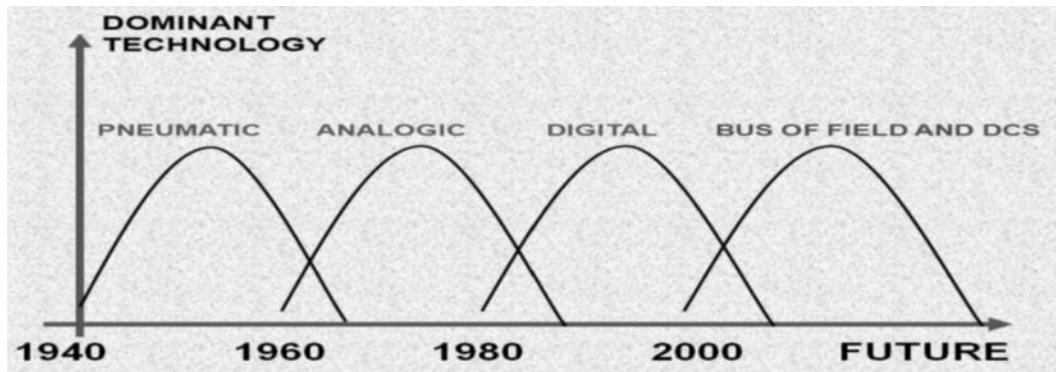


Figura 1.2. Evolución del control de procesos (Vásquez, 2008).

Vehículos, edificios con sus elementos, tales como electrodomésticos, puertas, ventanas, bombillas, etc., semáforos, farolas, cámaras, boyas en altamar y personas (a través de los teléfonos inteligentes y de los wearables “vestibles”) son algunas de las “cosas” que ya se conectan (Vega *et al.*, 2014). Pero, sobre todo, concurrirán sensores por todas partes, que medirán temperatura, humedad, viento, luz, tráfico de vehículos y de personas, polen y un sinnúmero de variables más. Todo esto interconectado para tomar decisiones sobre salud, tráfico, seguridad, educación, etc. Realmente las aplicaciones son innumerables. La información que se podrá disponer a partir de estas conexiones permitirá la prestación de

servicios de maneras que hoy en día son imposibles y, en algunos casos, inimaginables (Cruz Vega *et al.*, 2015).

1.6 Tecnologías Integradas a la Revolución Industrial 4.0

La expresión Industria 4.0, fue acuñada en Alemania en el año 2011 para describir a la fábrica inteligente, una perspectiva de la fabricación informatizada con todos los procesos interconectados por medio del Internet de las Cosas. Es lo que se conoce como Internet industrial de las cosas y es un proceso que apunta al siguiente nivel de la revolución industrial, con la capacidad de impulsar cambios fundamentales a la altura de la primera revolución industrial a través del vapor y el agua, la producción en masa de la segunda revolución y de la tercera caracterizada por el crecimiento de la electrónica y la proliferación de las tecnologías de la información (Val Román, 2016). La Industria 4.0 representa una etapa trascendental en la evolución de la industria, donde la clave es la fusión de la fábrica con el Internet a través del diseño y la implantación de componentes inteligentes dotados de identidades digitales propias, con miras a facilitar su manejo y reparación a distancia (Tapia, 2017).

La primera revolución industrial, que ocurrió en 1784 (Figura 1.3), se distingue por la integración de los sistemas de potencia hidráulicos y térmicos a los sistemas de manufactura para el aumento en la producción industrial y ocasiona la creación y desarrollo de los programas de Ingeniería Mecánica. La segunda revolución industrial (1870) implanta las líneas de producción en la planta con reducción en los tiempos de producción y se organiza la producción masiva de productos con características idénticas para su producción por lotes. Y con la generación de la electricidad se diseñan los motores eléctricos como unidades de potencia en los sistemas de manufactura y estimula la creación y desarrollo de programas de Ingeniería Electricista. La tercera revolución industrial (1989) se caracteriza por el uso intensivo y extensivo de la automatización y robotización de base

electrónica con circuitos electrónicos integrados en dispositivos como los microcontroladores, los controladores lógicos programables (PLC Programmables Logics Controllers) y los computadores digitales para ejercer la función de control de procesos (Rojas y Humberto, 2017).

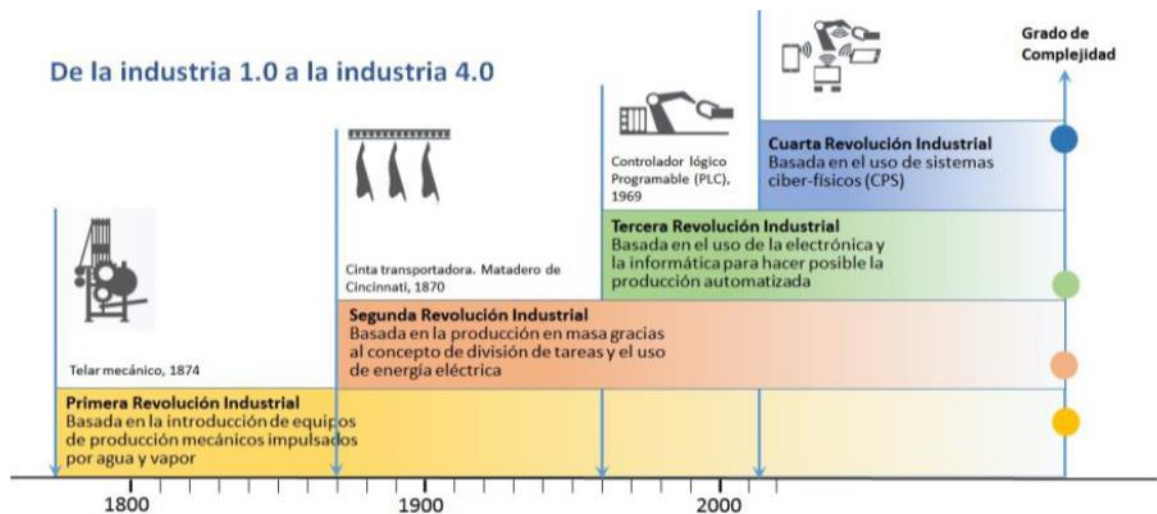


Figura 1.3. Evolución de las revoluciones industriales (Val Román, 2016).

La adopción de tecnología a pesar de parecer un gasto inicial, podría aumentar considerablemente la productividad y competitividad de una organización, hace más eficientes los procesos, genera nuevos negocios para las empresas y favorece la creación de nuevos puestos de trabajo. Cada día más, los gobiernos diseñan e implementan instrumentos e incentivos para potenciar la economía del conocimiento y acompañar a las empresas en la adopción de nuevas tecnologías (Candia *et al.*, 2018).

En Industria 4.0 se unifican al menos nueve tecnologías que integradas requieren nuevas competencias profesionales (Figura 1.4):

- **Big Data:** Capacidad de coleccionar, almacenar y analizar grandes cantidades de datos para identificar ineficiencias y cuellos de botella en la producción.
- **Autonomous Robots:** Habilidad para interacción Humano – Robot en su espacio de trabajo.
- **Simulation:** Capacidad de concebir, modelar, implementar, operar y optimizar productos y procesos en ambientes virtuales.
- **Universal System Integration:** Capacidad de Integración física – virtual y horizontal – vertical de todos los sistemas productivos en la Fabrica Digital.
- **Industrial IoT:** Habilidad de Conexión industrial de Internet en tiempo real de dispositivos, plantas, oficinas, y compañías para compartir información.
- **Cybersecurity:** Habilidad en técnicas de seguridad en Sistemas de Información (SI) y en Sistemas de Telecomunicación (ST).
- **Cloud Computing:** Capacidad de computación en la nube de IoT y de Big Data.
- **Additive Manufacturing:** Capacidad en Diseño e Impresión en 3D para lotes pequeños y cambios rápidos en los diseños, reducción apilamiento de materiales y bajos costos de transportación.
- **Augmented Reality:** Capacidad de Integración de elementos físicos con elementos virtuales para crear una realidad aumentada en tiempo real en la Fabrica Digital (Rojas y Humberto, 2017).

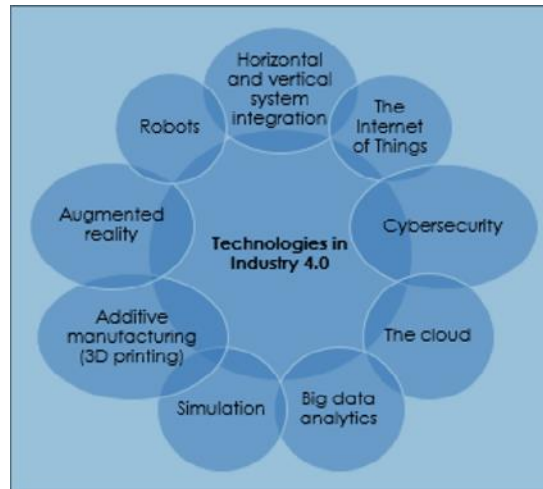


Figura 1.4. Tecnologías Integradas en Industria 4.0 (Rojas y Humberto, 2017).

1.7 ROFLEXIN/LC

En procesos industriales, el uso de equipos de medición facilita la obtención de datos eléctricos pero su principal desventaja es que su capacidad de lectura se reduce a pocos parámetros, razón por la que el hombre con ayuda de la ciencia y tecnología desarrolló equipos que permiten obtener una innumerable gama de datos eléctricos. Este hecho unido a la utilización de programas de control y adquisición de datos (SCADA) permitió llevar a ejecución grandes y eficaces sistemas de uso racional y ahorro de energía (Hernández Cevallos y Ledesma Marcalia, 2010).

El proyecto ROFLEXIN/LC se enfoca a la obtención de un sistema de monitoreo de bajo costo para procesos y sistemas mecánicos. El mismo se basa en la utilización de tecnologías de *hardware* más baratas que las convencionales, con un *software* que se desarrolla con herramientas libres, de código abierto, bajo una arquitectura que solo use aquellos módulos que sean necesarios en dependencia de las exigencias de clientes (Valido, 2019).

1.8 Conclusiones Parciales del Capítulo

Como resultado del análisis crítico de la bibliografía sobre la temática tratada, se llegó a las siguientes conclusiones:

1. Se pudo constatar la tendencia internacional al desarrollo de proyectos de automatización y control con vista a mejorar la sostenibilidad a través del uso optimizado de la energía eléctrica.
2. No existe un trabajo en la bibliografía consultada que sea un sistema para la medición en tiempo real del consumo de energía eléctrica que permita medir, registrar, visualizar y analizar su comportamiento, así como determinar su costo que satisfaga los requisitos y condiciones del proyecto ROFLEXIN/LC.
3. Se comprobó que se dispone de la base teórica, las principales herramientas y las tecnologías para el desarrollo del proyecto.

CAPÍTULO 2. MATERIALES Y MÉTODOS

En el presente capítulo se realiza la descripción de la propuesta de solución basada en los principios y reglas de la metodología de desarrollo SCRUM para el *software* y el diseño electrónico clásico para la interfaz de adaptación, así como los materiales y métodos empleados en el diseño y desarrollo del sistema de medición.

2.1 Sensor de corriente HWCT004

Los sensores son dispositivos capaces de transformar magnitudes físicas a señales eléctricas que puedan interpretarse por un microcontrolador. La salida de los sensores se tiene que conectar a una de las entradas del microcontrolador. Estos dispositivos pueden ser analógicos o digitales, dependiendo del tipo de señal que devuelvan (Martín Cantalejo, 2015).

Estos dispositivos de *hardware* producen una respuesta medible a un cambio en una condición física como corriente, temperatura o presión. Los sensores miden datos físicos del parámetro para ser controlado. El flujo de señal analógica producida por los sensores es digitalizado por un convertidor analógico-digital y enviada a controladores para procesamiento adicional (Ashlock y Warren, 2015).

Un nodo de sensor tendría que ser pequeño en medida, con un consumo extremadamente bajo de energía, operar en densidades volumétricas altas, ser autónomos y operar desatendido, así como ser adaptativo al entorno. Como los nodos de sensor inalámbrico son típicamente muy pequeños dispositivos electrónicos, sólo pueden ser equipados con una fuente de poder limitada de menos de 0.5 a 2 amperio-hora y 1.2 a 3.7 voltios (Curay

Aldana, 2017). Para la investigación como sensor se utiliza un HWCT004 que se puede observar en la Figura 2.1 con una relación de transformación de corriente de 50 A / 50 mA.



Figura 2.1. HWCT004 Sensor de corriente (MIKROELECTRON, 2020).

El HWCT004 es un sensor de corriente AC inductivo, de dimensiones de 48 x 30 x 40 mm además de un rango de medición máximo de 50 A, trabaja en frecuencias que oscilan desde 50 a 60 Hz y por último pero no menos importante, es compatible con microcontroladores con canales analógicos como es el caso de Arduino, lo que lo convierte en el sensor idóneo para nuestro proyecto ya que es pequeño, barato, permite leer señales analógicas, tiene incorporado un conversor analógico-digital, gran capacidad de efectuar prototipos rápidamente y con buena precisión (Pessanha Santos, 2008).

2.2 Circuito acondicionador

La señal de salida del sensor de un sistema de medición en general se debe procesar de una forma adecuada para la siguiente etapa de la operación. La señal puede ser, por ejemplo, demasiado pequeña, y sería necesario amplificarla; podría contener interferencias que eliminar; ser no lineal y requerir su linealización; ser analógica y requerir su digitalización; ser digital y convertirla en analógica; ser un cambio en el valor de la resistencia, y convertirla a un cambio en corriente; consistir en un cambio de voltaje y convertirla en un

cambio de corriente de magnitud adecuada, etcétera. A todas estas modificaciones se les designa en general con el término acondicionamiento de señal (Lara Cortez, 2012).

El circuito acondicionador (Figura 2.2) como su nombre lo indica, es un circuito de adaptación para acondicionar la señal del sensor de forma tal que pueda ser leída por la placa Arduino NANO, que no es más que una placa de desarrollo de tamaño compacto, de bajo costo, utilizada en muchas aplicaciones en el campo de la automatización. La primera parte del circuito consta de un divisor de voltaje el cual arroja un valor de 2.5V (voltaje de referencia). En la segunda parte se tiene una resistencia en paralelo con la bobina secundaria la cual es llamada resistencia *burden* (Figura 6, resistencia R3 33 Ω) y cumple la función de arrojar un voltaje proporcional a la corriente que circula por la bobina primaria.

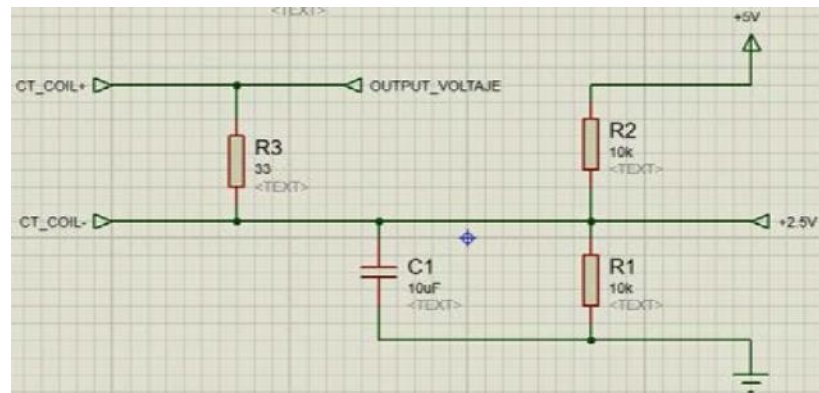


Figura 2.2. Circuito acondicionador (Acosta et al., 2017).

2.3 Arduino NANO

Arduino NANO (Figura 2.3) es un sistema que se basa en el microcontrolador de 8 bits ATmega328, de Atmel©, un chip sencillo y de bajo coste. Arduino NANO, tiene 14 pines (0 a 13), cada uno de los cuales se puede configurar como entrada o salida digital. Los

pinos configurados como salida pueden proporcionar o absorber una corriente de hasta 40 mA, suficiente para excitar multitud de circuitos, sensores, etc., aunque insuficiente para otros, tales como algunos relés, solenoides, motores, etc., para los que es preciso utilizar algún circuito excitador. Seis de estos pines se pueden configurar también como salidas PWM (*Pulse Width Modulation* o, Modulación por ancho de pulso), lo que permite variar el ciclo de trabajo de la señal cuadrada generada en el pin, obtener una tensión cuyo valor medio puede variar entre 0 y 5 V y así simular una salida analógica sobre una salida digital. La placa incluye un LED conectado al pin 13, lo que permite entre otras cosas utilizarlo como dispositivo de salida en la verificación y depuración de programas. Los pines 0 y 1 de E/S digital se pueden configurar para utilizarse como un puerto serie (Pessanha Santos, 2008).

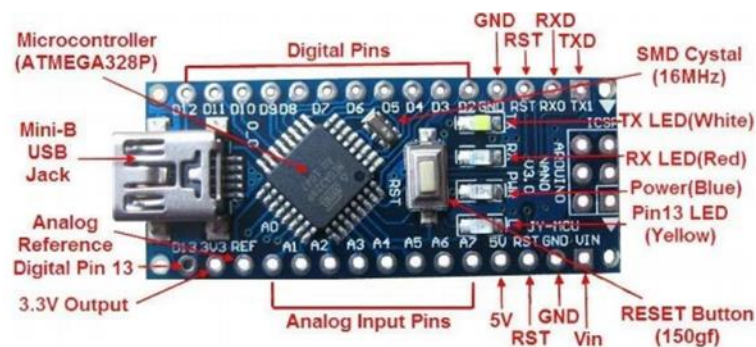


Figura 2.3. Arduino NANO. Fuente: Tomada de (Herranz & Allende, 2015)

El microcontrolador Atmega328 que usa Arduino NANO, incluye un conversor analógico-digital (A/D) de 6 canales, con una resolución de 10 bits cada uno (2^{10}) y retorna un valor entero entre 0 y 1023, es decir, 1024 valores. El uso principal de estos pines es para la lectura de sensores analógicos. Estos pines tienen también toda la funcionalidad de los pines de entrada-salida digitales, como los pines 0-13. En consecuencia, si para alguna

aplicación se precisan más pines de entrada-salida digital, y no se usa ningún pin analógico, pueden usarse estos pines para ello. La placa contiene todo lo necesario para utilizar el microcontrolador, cristal del oscilador de 16 MHz, regulador de tensión línea I, entre otros. Es suficiente con conectarlo al ordenador a través del cable USB o alimentarlo desde una fuente de alimentación o una batería externa, que se recomienda un voltaje de 7 a 9 V, para empezar a trabajar con él. Tiene 32 KB de memoria Flash, 2 KB de memoria SRAM y 1 KB de memoria EEPROM (Herrero Herranz y Sánchez Allende, 2015).

2.4 Herramientas y tecnologías utilizadas en la investigación

Con el objetivo de desarrollar una solución eficiente fue necesario realizar un estudio previo de las tecnologías existentes, a fin de seleccionar las más adecuadas para la tarea en cuestión. Sobre esta base se seleccionan las herramientas tecnológicas que se muestran.

2.4.1 Lenguajes de Programación

La selección del lenguaje de programación es un paso importante porque constituye el soporte sobre el cual se verifica el algoritmo y su codificación (Díaz Pérez y Serra López, 2015). Otros elementos a tener en cuenta son el ambiente y la facilidad para la escritura de los códigos (González Hernández *et al.*, 2004).

Después que se realizó un estudio de los lenguajes de programación más utilizados tales como Python, Java, C++, C, C# entre otros, el autor decide utilizar como lenguaje principal C# para el desarrollo de la solución propuesta porque se diseñó con el objetivo de aglutinar algunas de las capacidades más provechosas presentes en otros lenguajes lo que lo hace muy potente y amigable al programador, cuenta con una extensa documentación y es el

más conocido por el autor. Como lenguaje de desarrollo adicional para desarrollar el driver que se instalará en la placa de desarrollo Arduino NANO se seleccionó Processing/Wiring (Barragán, 2019) debido a que es el lenguaje que se emplea en el entorno de trabajo o desarrollo integrado llamado **IDE Arduino** el cual es el *software* que más se usa para programar estos dispositivos y cuenta con una gran documentación.

2.4.1.1 Lenguaje de programación C#

C# es un lenguaje de programación orientado a objetos que se desarrolló y estandarizó por Microsoft como parte de su plataforma .NET, que después se aprobó como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común (Kovacs, 2007). Las bibliotecas de clases fueron escritas originalmente mediante el uso de un sistema de código gestionado llamado *Simple Managed C* (SMC). En enero de 1999, Anders Hejlsberg formó un equipo con la misión de desarrollar un nuevo lenguaje de programación llamado Cool (Lenguaje C orientado a objetos). Este nombre se cambió debido a problemas de marca y pasó a llamarse C#.

La biblioteca de clases de la plataforma .NET fue migrada entonces al nuevo lenguaje (Microsoft Corp., 2020). Hejlsberg lideró el proyecto de desarrollo de C#. Anteriormente, ya había participado en el desarrollo de otros lenguajes como Turbo Pascal, Delphi y J++. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes. El nombre C Sharp fue inspirado por la notación musical, donde '#' (sostenido, en inglés *sharp*) indica que la

nota (C es la nota do en inglés) es un semitono más alto, sugiriendo que C# es superior a C/C++. Además, el signo '#' se compone de cuatro signos '+' pegados.

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente, diseñado para generar programas sobre dicha plataforma (Charte Ojeda, 2002). Ya existe un compilador implementado que provee el marco Mono - DotGNU, el cual genera programas para distintas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux (Peña Daniel, 2019).

2.4.1.2 Lenguaje de programación Processing/Wiring

El lenguaje con el que trabaja la plataforma Arduino, es el lenguaje de programación Arduino que se basa en *Wiring* y el entorno de desarrollo Arduino que se basa en *Processing* (Barragán, 2019). Este no es solamente un lenguaje de programación orientado a la creación visual, si no toda una plataforma de creación de aplicaciones computacionales altamente interactivas que no solo se alimentan de los eventos producidos por los usuarios, si no por una gran cantidad de librerías que permiten conectarse a un sin número de plataformas y aplicativos que enriquecen las aplicaciones y le permiten producir una gran cantidad de acciones sobre el mundo virtual de la computadora.

En cuanto a *Wiring*, se centra en ofrecer un marco de programación especialmente para microcontroladores, proyecto abierto que fue iniciado por Barragán (2019) con el ideal de que estudiantes, diseñadores y expertos fomentaran su aprendizaje y se motivaran en la creación de nuevas ideas. “*Wiring* es un entorno de programación de entradas/salidas de código abierto para explorar las artes electrónicas, los medios materiales, la enseñanza y el

aprendizaje de la programación informática y creación de prototipos con electrónica” (Astudillo y Cadena, 2016).

Wiring permite escribir *software* para manipular dispositivos conectados a la plataforma electrónica para crear objetos y espacios interactivos que pueden tener la capacidad de sentir y modificar el ambiente. La idea es escribir unas líneas de código, conectar unos componentes electrónicos a la plataforma electrónica y ver cómo una luz se enciende o un motor se mueve. Agregar más líneas de código y algunos sensores a la plataforma electrónica y ver cómo la luz se enciende cuando una persona se acerca o el sol se oculta. Este proceso se denomina bocetar con *hardware* (*sketching with hardware*) y es un punto de partida positivo en la forma en la que la electrónica y la programación son enseñadas tradicionalmente: se enfocan desde el principio en los fenómenos físicos interactivos, ideas o conceptos, más que en el código y la electrónica (Barragán, 2019).

2.4.2 Aplicaciones de Escritorio

Los formularios *Windows Forms* se utilizan para desarrollar aplicaciones en las que se espera que el cliente maneje una parte significativa de la carga de trabajo de la aplicación (Butow y Ryan, 2002). Entre ellas, se incluyen las aplicaciones clásicas de escritorio Win32 que solían desarrollarse en versiones anteriores de Visual Basic y Visual C++. Ejemplo de ello son las aplicaciones gráficas o de diseño, los sistemas de entrada de datos, los sistemas de punto de venta y los juegos. Todas estas aplicaciones tienen en común el hecho de que dependen de la potencia del equipo de escritorio para el procesamiento y la presentación de contenidos de alto rendimiento. Algunas aplicaciones de formularios Windows Forms pueden estar completamente auto contenidas y ejecutar todo el proceso

de la aplicación en el equipo del usuario. A menudo, los juegos se escriben de este modo (Charte Ojeda, 2002).

Otras pueden ser parte de un sistema mayor, fundamentalmente, utilizan el equipo de escritorio para procesar los datos proporcionados por el usuario. Por ejemplo, un sistema de punto de venta normalmente necesita una interfaz de usuario rápido y sofisticado que se crea en el equipo de escritorio, pero está enlazado a otros componentes que ejecutan el procesamiento en segundo plano. Dado que una aplicación *Windows* que utilice formularios *Windows Forms* se genera alrededor de un marco de trabajo *Windows*, tendrá acceso a los recursos del sistema del equipo cliente, que incluyen los archivos locales, el Registro de *Windows*, la impresora, entre otros. Se puede restringir este nivel de acceso para eliminar cualquier riesgo para la seguridad o problema potencial que pueda provocar un acceso no deseado (Peña Daniel, 2019).

Los formularios *Windows Forms* pueden aprovechar las clases gráficas .NET GDI+ para crear una interfaz gráficamente rica, lo que suele ser un requisito para las aplicaciones de extracción de datos o de juegos. Cuando se diseñan aplicaciones que incluyan una interfaz de usuario, habrá dos opciones: formularios *Windows Forms* y *Web Forms*. Ambos disponen de compatibilidad total en tiempo de diseño dentro del entorno de desarrollo y pueden proporcionar una interfaz de usuario completa y funciones de aplicaciones avanzadas para solucionar problemas de negocios.

¿Cómo decidir, entonces, qué tecnología es la adecuada para una aplicación dada? La opción puede ser evidente si se tiene en cuenta ciertas consideraciones relativas a la aplicación; por ejemplo, si se va a crear un sitio Web de comercio electrónico accesible

para el público a través de Internet, es obvio que la aplicación se deberá desarrollar mediante el uso páginas de formularios *Web Forms*. Si se crea una aplicación de alto rendimiento, que haga un uso intensivo de la capacidad de proceso y necesite aprovechar toda la capacidad del equipo cliente, como puede ser una aplicación de productividad para la oficina, se utilizará formularios *Windows Forms*. Sin embargo, en otros casos es posible que la elección no esté tan clara.

Según Peña Daniel (2019), las características principales de las aplicaciones de escritorio son:

- **Implementación:** Permiten su implementación “sin contacto”: las aplicaciones se pueden descargar, instalar y ejecutar directamente en los equipos de los usuarios, sin ninguna alteración del Registro.
- **Gráficos:** Incluyen GDI+, que permite utilizar gráficos sofisticados para juegos y otros entornos gráficos extremadamente ricos.
- **Capacidad de respuesta:** Pueden ejecutarse por completo en el equipo cliente; pueden proporcionar la respuesta más rápida para aquellas aplicaciones que necesiten un alto grado de interactividad.
- **Formularios y control de flujo de texto:** Las funciones de posición de cuadrícula de formularios *Windows Forms* proporcionan un preciso control bidimensional (coordenadas x e y) para la colocación de controles. Para mostrar texto en *Windows Forms*, es necesario insertarlo primero en algún control como *Textbox* o *RichTextBox* (las posibilidades de formato son limitadas).
- **Plataforma:** Los formularios *Windows Forms* requieren que .NET Framework se ejecute en el equipo cliente.

- **Acceso a recursos locales (sistema de archivos, Registro de Windows):** Las aplicaciones, si se les permite, pueden tener acceso total a los recursos del equipo local. Si es preciso, se pueden restringir los permisos de la aplicación para que no utilice recursos específicos.
- **Modelo de programación:** Los formularios Windows Forms se basan en un modo de suministro de mensajes Win32 en el cliente, donde el desarrollador crea, utiliza y descarta instancias de componentes.
- **Seguridad:** Los formularios Windows Forms utilizan permisos granulares en su implementación de seguridad de acceso a código a fin de proteger los recursos del equipo y la información reservada. Esto permite exponer cuidadosamente la funcionalidad al mismo tiempo que se conserva la seguridad. Por ejemplo, el permiso de impresión, que en un nivel sólo permitiría imprimir en la impresora predeterminada, en otro nivel podría permitir la impresión en cualquier impresora.

2.4.2.1 Visual Studio 2019 Community Edition

Visual Studio Community Edition, no es una versión recortada de un producto comercial y es gratis no solamente para el sector educativo, sino para desarrolladores individuales o en equipos de hasta cinco personas. Solamente una versión, para Empresas (*Enterprise*), se mantiene con costo. El entorno de desarrollo integrado (IDE) de esta versión de *Visual Studio* soporta múltiples tipos de proyectos en un solo archivo solución dentro del IDE y tiene todas las características de productividad y extensibilidad del IDE, lo que significa que se puede usar Xamarin, ReSharper, VsVim y otras extensiones VSIX, que los desarrolladores aman de Visual Studio, a decir del propio Microsoft.

Las condiciones de su gratuidad son: que todo desarrollador individual trabajando en un proyecto comercial o no comercial; cualquier desarrollador contribuyendo a un proyecto de código abierto; cualquiera en una investigación académica o curso escolar (ejemplo, estudiantes, profesores, cursos en línea incluso); cualquier Empresa con 5 o menos desarrolladores trabajando juntos en un proyecto comercial o no (Microsoft Corp., 2020).

2.4.2.2 Características Visual Studio 2019

Permite crear aplicaciones más inteligentes, rápido con mejoras a las funciones populares, como la navegación de código, refactorización, y los arreglos de código, le ahorran tiempo y esfuerzo, sin importar el idioma o la plataforma. Toda la experiencia de depuración y prueba incluye mejoras para ayudarle a detectar y abordar problemas lo antes posible. Características como *Live Unit Testing*, *Exception Helpers* y *Run to Click* refuerzan el bucle de desarrollo de operaciones al reducir los riesgos de regresión y exponer inmediatamente la causa principal de los problemas nuevos.

Las herramientas incorporadas proporcionan una integración completa con la totalidad de su *Core .NET*, aplicaciones Azure, los servicios y los contenedores *Docker*. Administra de forma directa los proyectos de los equipos alojados por cualquier proveedor, incluyendo *Visual Studio Team Services*, *Team Foundation Server*. Con las herramientas avanzadas de depuración y generación de perfiles y las características de generación de pruebas unitarias de Xamarin, se aceleran y simplifican más que nunca los procesos de compilación, conexión y ajuste de las aplicaciones móviles nativas para Android, iOS y Windows. También puede optar por desarrollar aplicaciones móviles con Apache Cordova o compilar bibliotecas multiplataforma de C++ (Microsoft Corp., 2020).

2.4.2.3 Software para programar Arduino

El *software* que se eligió para programar el microcontrolador es Arduino IDE, se basa en el lenguaje C. Todo programa en Arduino IDE está compuesto de dos funciones principales. Una función inicial llamada *void setup*, donde se configuran detalles como la velocidad de transmisión de la información (o baudios), la inicialización de los puertos serie que se vayan a emplear o el modo de empleo de un pin digital concreto (es decir, si dicho pin será empleado como entrada o salida de datos). El *void setup* solamente se ejecutará una vez al inicio del programa (Ruiz Gutiérrez, 2007).

La segunda función del programa es el *loop ()*, que es un bucle que ejecuta el código en su interior una y otra vez. En él se programan las instrucciones que se quieren realizar periódicamente, como la lectura o escritura de datos sobre los pines digitales y analógicos deseados. Puesto que es un bucle, se debe controlar el periodo de ejecución del mismo con alguna instrucción. Esta es *delay (miliseg)* siendo *miliseg* el tiempo en milisegundos que el programa esperará hasta pasar a la siguiente línea (o volver a repetir el bucle si esa instrucción era la última línea del *void loop*). Estas funciones se pueden observar señaladas en rojo en la Figura 2.4.


```
Proyecto_de_tesis Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
Proyecto_de_tesis §
int firstSensor = 0; // first analog sensor
int secondSensor = 0; // second analog sensor
int thirdSensor = 0; // digital sensor
int inByte = 0; // incoming serial byte

void setup() {
  // start serial port at 9600 bps:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  pinMode(2, INPUT); // digital sensor is on digital pin 2
  establishContact(); // send a byte to establish contact until receiver responds
}

void loop() {
  // if we get a valid byte, read analog ins:
  if (Serial.available() > 0) {
    // get incoming byte:
    ...
  }
}
```

Guardado.

El Sketch usa 444 bytes (1%) del espacio de almacenamiento de programa. El máximo es 30720 bytes. Las variables Globales usan 9 bytes (0%) de la memoria dinámica, dejando 2039 bytes para las variables locales.

42 Arduino Nano, ATmega328P en COM4

Figura 2.4. IDE de Arduino. Elaboración propia.

Cabe mencionar que en las primeras líneas del programa se pueden declarar librerías y variables que serán usadas tanto en el *void setup* como en el *void loop*.

Para programar en Arduino es necesario conocer cuatro funciones de control que son muy empleadas en todos los programas:

- **digitalWrite** (pin, estado): esta función escribe en el pin digital deseado (cualquier número distinto de 0 y 1 presente en la placa) el estado binario indicado (LOW, 0 Voltios o HIGH, 5 Voltios).
- `int dato = digitalRead (pin)`: procede a la lectura del pin digital elegido y devuelve en la variable dato un valor de 0 (voltaje del pin menor de 2.5V) o 1 (voltaje del pin mayor o igual a 2.5V) dependiendo del estado de dicho pin.

- `int dato = analogRead (pin)`: lee el pin analógico deseado (número entre 0 y 15) y devuelve en `dato` su tensión de 0 a 5 voltios convertido en un valor entre 0-1023. Esto es debido a un convertidor analógico digital de 10 bits, que segmenta los 5 voltios en 2^{10} divisiones. Posteriormente, se podrá obtener la tensión digital con una función de conversión: Tensión de entrada al pin analógico calculada como $(\text{dato} * 5) / 1023$.
- `analogWrite (pin, ciclo)`: esta función puede resultar confusa, pues, aunque su nombre haga referencia a analógico, lo que realmente hace es generar sobre el pin digital deseado una señal **PWM**. Es decir, escribirá un valor de tensión entre 0 y 5 V en dependencia del valor de ciclo toma un valor entre (0-255). La tensión de salida seguirá por tanto la siguiente función de conversión: Tensión de salida por pin digital [V] como $(\text{ciclo} * 5) / 255$

2.5 Redes Neuronales Artificiales (RNA)

Tratar de imitar la inteligencia de los seres vivos con el uso de circuitos y máquinas sin alma es actualmente uno de los objetivos más ambiciosos de los investigadores y aunque se lograron grandes avances e innovaciones, pensar que en un futuro próximo tendremos máquinas más inteligentes que nosotros es aún un sueño incierto. Uno de los paradigmas de la computación que trata de asemejar el funcionamiento del cerebro de los seres vivos son las redes neuronales artificiales (RNA), que avanza rápidamente gracias a las nuevas tecnologías, y al desarrollo de equipos de cómputo con mayores capacidades de procesamiento y almacenamiento (Martínez Contreras *et al.*, 2019).

Las Redes Neuronales Artificiales (RNA) se utilizan de forma satisfactoria en el ámbito de agrupamiento de datos, predicción, optimización; de tal forma que se convirtió en una herramienta muy importante para la resolución de problemas de clasificación de patrones (Martin del Rio y Sanz Molina, 2001).

2.5.1 Concepto de Red Neuronal

No existe un concepto general si nos referimos a las redes neuronales artificiales ya que existen diversos artículos en donde el autor da su punto de vista con respecto a la misma.

Se seleccionó la siguiente definición (Cervantes Suarez, 2017):

- Una red neuronal consiste en un modelo de la computación, en paralelo, que está compuesta por unidades de procesamiento adaptivas con una alta interconexión entre ellas.
- Son sistemas de procesamiento de información, que utilizan algunos principios fundamentales de la organización del cerebro humano.
- Emulación del cerebro humano, por medio de modelamientos matemáticos.
- Sistema de procesamiento de información, con características de funcionamiento comunes con las redes neuronales biológicas.

2.5.2 Composición de las redes neuronales

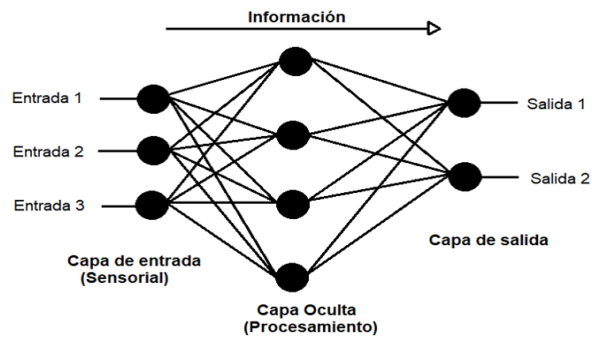


Figura 2.5. Red Neuronal. Elaboración propia

Por lo general, una red neuronal artificial se divide en tres partes, denominadas capas como se muestra en la Figura 2.5. de las cuales el autor (Nunes Silva *et al.*, 2017) ofrece las definiciones siguientes:

- **Capa de entrada.** Esta capa es responsable de recibir información (datos), señales, características o mediciones del entorno externo. Estas entradas (muestras o patrones) normalmente se normalizan dentro de los valores límite producidos por las funciones de activación. Esta normalización resulta en una mejor precisión numérica para las operaciones matemáticas realizadas por la red.
- **Capas ocultas, intermedias o invisibles.** Estas capas están compuestas de neuronas que son responsables de extraer patrones asociados con el proceso o sistema que se analiza. Estas capas realizan la mayor parte del procesamiento interno desde una red.
- **Capa de salida.** Esta capa también está compuesta de neuronas y, por lo tanto, es responsable de producir y presentar las salidas finales de la red, que resultan del procesamiento realizado por las neuronas en las capas anteriores.

2.5.3 Tipos de Redes Neuronales

Existe una serie de modelos que aparecen en la mayoría de estudios académicos y la bibliografía especializada, los autores (Andana Quijada y Duran Figueroa, 2015) y (Martin del Rio y Sanz Molina, 2001) mencionan los siguientes:

1. Perceptrón.
2. Adaline.
3. Perceptrón multicapa.
4. Memorias asociativas.
5. Máquina de Boltzmann.
6. Máquina de Cauchy.
7. Propagación hacia atrás.
8. Redes de Elman.
9. Redes de Hopfield.
10. Red de Contra propagación.
11. Redes de neuronas de base radial.
12. Redes de neuronas de aprendizaje competitivo.
13. Mapas Autoorganizados (SOM) (Redes de Kohonen).

14. Crecimiento dinámico de células.

15. Gas Neuronal Creciente.

16. Redes ART.

2.5.4 Clasificación de las RNA según el aprendizaje

Según la forma de aprendizaje, las RNA se pueden clasificar en:

- **Aprendizaje supervisado:** La estrategia de aprendizaje supervisado consiste en disponer de las salidas deseadas para un conjunto dado de señales de entrada; En otras palabras, cada muestra de entrenamiento está compuesta por las señales de entrada y sus correspondientes salidas. En lo sucesivo, se requiere una tabla con datos de entrada / salida, también llamada tabla de atributo / valor, que representa el proceso y su comportamiento. Es a partir de esta información que las estructuras neurales formularán "hipótesis" sobre el sistema que se está aprendiendo.
- **Aprendizaje no supervisado:** Al igual que el aprendizaje no supervisado, utiliza un conjunto de patrones de entrada; sin embargo, no se establece una salida. Por lo tanto, la red necesita organizarse cuando existan particularidades entre los elementos que componen el conjunto de muestras completo, así se identifican subconjuntos que presentan similitudes. El algoritmo de aprendizaje ajusta los pesos y umbrales sinápticos de la red para reflejar estos grupos dentro de la propia red. Alternativamente, el diseñador de red puede especificar (a priori) la cantidad máxima.

- **Aprendizaje por refuerzo:** Es considerado una variación de las técnicas de aprendizaje supervisado, ya que analizan continuamente la diferencia entre la respuesta producida por la red y la salida deseada correspondiente. Los algoritmos que utilizan el aprendizaje de refuerzo ajustan los parámetros neuronales internos basándose en cualquier información cualitativa o cuantitativa recibida a través de la interacción con el sistema (entorno) que se mapea, con el empleo esta información para evaluar el rendimiento de aprendizaje. El proceso de aprendizaje en red suele hacerse por ensayo y error porque la única respuesta disponible para una entrada dada es si fue satisfactoria o insatisfactoria. Si es satisfactorio, los pesos y umbrales sinápticos se incrementan gradualmente para reforzar (recompensar) esta condición de comportamiento involucrada con el sistema (Nunes Silva *et al.*, 2017).

2.5.5 Uso de algoritmos de aprendizaje no supervisado en el análisis de datos

Los algoritmos de aprendizaje no supervisado se pueden emplear para reducir las dimensiones de los datos, lo que facilita la visualización y la extracción de información implícita. Existe una gran cantidad de técnicas que permiten la estimación de los valores futuros de una determinada variable. Algunas de las técnicas más utilizadas están basadas en series temporales y tratan de hacer una estimación de la evolución de la variable a predecir, en función de la evolución pasada. Muchas de estas técnicas utilizan la información obtenida por medio de la minería de datos con el fin de obtener modelos de predicción que sean aplicados al problema de las series temporales.

En la bibliografía consultada existen multitud de técnicas que utilizan redes neuronales tales como MLP (*MultiLayer Perceptron*) y RBF (*Radial Basis Function*)(Leung et al., 2001), técnicas de regresión como ARIMA (*AutoRegressive Integrated Moving Average*), SVM(*Support Vector Machines*) (Cao, 2003) y los Mapas Auto-Organizados, que constituye una entidad propia gracias a la gran cantidad de modificaciones que son utilizadas para predicción (Barreto *et al.*, 2004), redes *fuzzy*, sistemas expertos, métodos híbridos que combinan varios métodos, después de analizar los más utilizados para el análisis.

La predicción de las variables de consumo energético permite estimar la evolución futura de estas variables con el fin de poder anticiparse a posibles picos y conocer el consumo a lo largo del día siguiente. En el caso de consumos eléctricos, la predicción es una herramienta imprescindible a la hora de programar la producción de energía por parte de las centrales eléctricas. La energía eléctrica sufre el problema de que no se puede almacenar de forma eficaz, por lo que toda la energía producida se tiene que consumir. Por esta razón, un conocimiento lo más exacto posible de la evolución del consumo permite reducir al máximo las pérdidas por un exceso de energía producida (Morán Álvarez, 2012).

2.5.5.1 Mapa Auto-Organizado de Kohonen

El mapa autoorganizado está formado por una matriz rectangular de neuronas, de modo que las relaciones entre los patrones de entrada son mucho más fácilmente visibles en forma de relaciones de vecindad. Cada neurona sintoniza o aprende por sí misma a reconocer un determinado tipo de patrón de entrada. En el espacio de salida la topología esencial del de entrada queda preservada, de manera que neuronas próximas en el mapa

aprenden a reconocer patrones de entrada similares, cuyas imágenes, por lo tanto, aparecerán cercanas en el mapa creado (Andana Quijada y Duran Figueroa, 2015).

Este espacio de salida se representa por una capa discreta de neuronas artificiales o procesadores elementales, generalmente ordenados que forman una matriz rectangular (Narváz Serrano, 2007).

2.5.5.2 Uso del SOM en el análisis de datos

El SOM no solo puede utilizarse como método de *clustering* (segmentación) para agrupar series temporales con la misma dinámica, sino que también se puede utilizar directamente para crear modelos de predicción a partir de las series temporales. Existen dos razones principales por las que se puede utilizar el SOM para predicción de series temporales (Barreto *et al.*, 2004).

- ***Naturaleza local de los modelos generados por el SOM.*** Los modelos que genera el SOM son modelos que realizan una aproximación por medio de funciones locales, entendiéndose por locales, aquellos modelos cuya salida queda determinada por operaciones matemáticas que son de aplicación solo en determinadas zonas del espacio de entrada. Modelos globales, como son los basados en redes MLP, utilizan representaciones muy distribuidas del espacio de entrada que dificultan la interpretación de los resultados. Los modelos locales permiten un mejor entendimiento de las dinámicas de los subprocesos generados por las series temporales, al menos en las regiones que se usan para calcular el modelo local. Esta propiedad del SOM, junto con su preservación de la topología,

son útiles para segmentar o visualizar las series temporales (Andana Quijada y Duran Figueroa, 2015).

- **Arquitecturas simples de crecimiento.** La segmentación (*clustering*) del espacio de entrada es un paso fundamental para la eficiencia de los modelos basados en el SOM. Por tanto, es necesario obtener una buena estimación del número de vectores prototipo. Una mala estimación de la cantidad de clúster puede llevar a generar modelos locales que contienen pocos datos para construir un modelo correcto o tienen vectores de otras zonas con dinámicas diferentes. Una de las principales ventajas del SOM es que, al ser un algoritmo no supervisado, no es necesario especificar el número de neuronas iniciales, pudiéndose utilizar técnicas de crecimiento competitivo. En cambio, otros métodos generan modelos a partir de las neuronas entrenadas (Morán Álvarez, 2012).

2.5.6 Ventajas y desventajas de las redes neuronales

Las RNA presentan un grupo importante de ventajas, dentro de las que se destacan (Trujillano *et al.*, 2004):

- Necesitan menos formalismo estadístico para su desarrollo.
- Detectan relaciones no lineales.
- Detectan interacciones entre variables predictoras.
- Tienen múltiples algoritmos de entrenamiento.

Por el contrario, sus principales limitaciones son:

- Son «cajas negras» para identificar las interacciones.

- Tienen dificultad de utilización.
- Necesitan grandes recursos computacionales.
- Implementación llena de procesos empíricos.

2.6 Metodología SCRUM

La solución propuesta está basada en los principios y reglas de la metodología de desarrollo SCRUM para el *software* (Figura 2.6). A continuación, se describen cada uno de sus componentes.

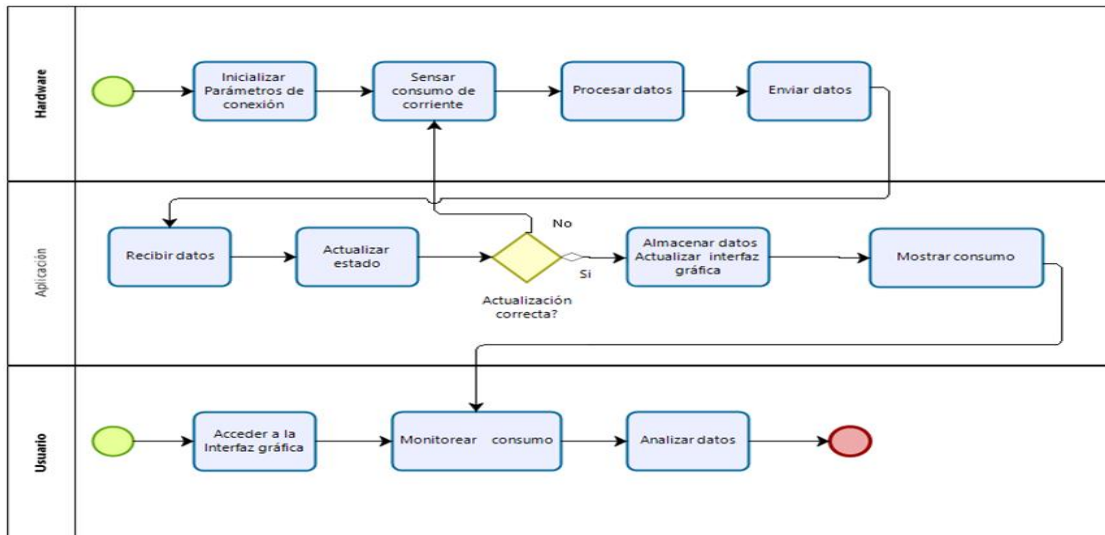


Figura 2.6. Diagrama de procesos. Elaboración propia.

2.6.1 Requisitos del *software*

La Ingeniería de Requisitos se debe estructurar en etapas, cuyas actividades buscan comprender, estructurar y documentar las necesidades que los usuarios desean satisfacer con el producto, y que con posterioridad se traducen en un conjunto de sentencias precisas,

no ambiguas, que se utilizan para desarrollar el sistema, conocidas como requisitos del sistema (Pressman, 2010).

2.6.2 Requisitos funcionales

Los requisitos funcionales identificados para el *software* son:

1. Desarrollo del driver para instalar en el *hardware*.

1.1 Analizar y definir los tipos de sensores.

1.2 Definir protocolo de comunicación.

1.3 Definir la velocidad de transmisión.

1.4 Definir velocidad de muestreo.

1.5 Definir cantidad de muestras a tomar.

1.6 Establecer conexión y realizar pruebas.

2. Almacenar datos de sensores en tiempo real.

2.1 Definir formato de documento.

2.2 Establecer comunicación con base de datos.

2.3 Realizar pruebas.

3. Implementar red neuronal.

3.1 Implementar Algoritmo.

- 3.2 Definir patrones de entrenamiento.
- 3.3 Definir función de activación.
- 3.4 Entrenar red.
- 3.5 Mostrar los resultados.
- 4. Graficado de reporte en tiempo real.
 - 4.1 Crear y maquetar la vista de monitoreo.
 - 4.2 Comprobar la correcta funcionabilidad de la vista.
 - 4.3 Comprobar que la información sea correcta.
 - 4.4 Graficar valores en tiempo real.
- 5. Normalizar datos provenientes del *hardware*.
 - 5.1 Implementar filtro digital.
 - 5.2 Realizar pruebas.
- 6. Análisis de mediciones.
 - 6.1 Crear y maquetar la vista de Análisis de datos.
 - 6.2 Comprobar la correcta funcionabilidad de la vista.
 - 6.3 Implementar algoritmo de detección de picos.
- 7 Establecer comunicación con dispositivo de *hardware*.

7.1 Definir protocolo de comunicación.

7.2 Sincronizar con el *hardware*.

7.3 Verificar la recepción de datos.

7.4 Implementar algoritmos de recepción.

- Realizar pruebas.

2.6.3 Requisitos no funcionales

Los requisitos no funcionales son requerimientos que imponen restricciones en el diseño o implementación de un sistema como estándares de calidad. Son propiedades o cualidades que el producto debe tener. Los siguientes requisitos no funcionales, están orientados a la seguridad, estandarización, y centralización de datos, para asegurar así el rendimiento y calidad del software (Lara Sobrino, 2019). Para el desarrollo del sistema se logró definir 29 requisitos, los que se clasifican en requisitos no funcionales (RNF) de confiabilidad, eficiencia, soporte, restricciones de diseño, de interfaz, seguridad, *hardware*, *software* y ayudas.

Ambiente:

RNF1. El sistema debe presentar una interfaz de usuario intuitiva, fácil de entender y usar.

RNF2. El tiempo de respuesta brindado por el sistema debe ser menor de 300 ms.

RNF3. El sistema debe contar con manuales de usuario estructurados de forma adecuada.

RNF4. El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.

RNF5. El idioma de todas las interfaces de la aplicación debe ser el español.

RNF6. El sistema debe contar con un menú que permite acceder a todas las funcionalidades.

RNF7. El sistema no debe requerir de más de tres interfaces para lograr una funcionalidad completa.

Confiabilidad:

RNF8. El sistema debe imponer campos obligatorios para garantizar la integridad de la información que se introduce por el usuario.

RNF9. El sistema no debe permitir la entrada de datos incorrectos.

RNF10. Ante el fallo de una funcionalidad del sistema, el resto de las funcionalidades que no dependen de esta deben seguir su funcionamiento regular.

Soporte:

RNF11. El uso del sistema requiere de un tiempo de preparación previa por los usuarios finales para su correcta explotación.

RNF12. Se debe recibir entrenamiento para el mantenimiento del sistema.

Restricciones de diseño:

RNF13. El *software* debe ser desarrollo para desktop por solicitud del Cliente.

RNF14. Las etiquetas de cada funcionalidad y los campos de cada interfaz deben tener títulos asociados a su función.

RNF15. El sistema debe presentar los términos capitalizados, es decir, la primera palabra debe tener su primera letra en mayúsculas.

Interfaz:

RNF16. El sistema debe presentar una interfaz legible, simple de usar e interactiva.

RNF17. La tipografía y colores deben ser estándares en toda la aplicación.

Seguridad:

RNF18. El sistema debe garantizar la protección ante acciones no autorizadas.

Hardware:

RNF19. Los requisitos mínimos de *hardware*:

1. Memoria RAM: 4 GB.
2. Micro procesador: Intel Core I3 2.2 GHz de 4 generación o superior.
3. Capacidad de Disco Duro: 100 GB.

Software:

RNF20. Sistema operativo: Windows 8 o superiores.

Requisitos de ayudas y documentación en línea

RNF29. El sistema debe contar con una ayuda integrada con el objetivo de facilitar al usuario su utilización.

2.6.4 Etapa de planificación

La etapa de planificación es la etapa inicial de todo el proyecto. Se realiza con el objetivo de lograr una eficiente organización del prototipo inicial del problema y proporcionar así un buen comienzo a una solución eficaz. Con este objetivo y según las ideas del cliente sobre el *software* se desarrollan las Historias de Usuario, mediante la cual se obtiene un punto de partida para el resto de la planificación del proyecto. De igual forma se debe realizar un estimado de cada una de las entregas del proyecto y del tiempo, que se basan en que la planificación inicial se podría afectar debido a cambios que pudiesen sufrir estos aspectos durante el desarrollo del proyecto (Scull Echeverría, 2018).

2.6.5 Definición del Equipo

Los roles principales en Scrum son el Scrum Master, que mantiene los procesos y trabaja de forma similar al director de proyecto, el *Product Owner*, que representa a los *stakeholders* (interesados externos o internos), y el *Team* que incluye a los desarrolladores (Díaz Ortiz y Romero Suárez, 2017). Los Equipos Scrum son autoorganizados y multifuncionales. Los equipos autoorganizados eligen la mejor forma de llevar a cabo su

trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad. Los Equipos Scrum entregan productos de forma iterativa e incremental, donde se maximizan las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre se dispone de una versión útil y funcional del producto (Pressman, 2010). En la Tabla 2.1 se define el equipo del proyecto.

Tabla 2.1. Equipo del proyecto.

| <i>Rol</i> | <i>Miembro</i> |
|---|----------------------------------|
| Dueño del producto (<i>Product Owner</i>) | Dr. C. Ramón Quiza Sardiñas |
| Equipo de desarrollo (<i>Scrum Team</i>) | Ing. Renier Fagundo Mesa |
| <i>Scrum master</i> | Lic. Ramón de Jesús López Suárez |

2.6.6 Historias de usuarios

Las historias de usuarios son desarrolladas por el equipo de trabajo durante el proceso de levantamiento de requisitos para la implementación del proyecto. Para la estimación de los datos se toman los siguientes criterios:

- **Prioridades en el Negocio (PN):** Se mide en función al rango de: Alta, Media y Baja, las cuales, se asignan por el *Product Owner*.
- **Importancia del Desarrollo (ID):** Se asignan valores con ponderaciones del 1 al 100 entre el *Product Owner* y los miembros del equipo Scrum, donde:

1. Todos los elementos con importancia ≥ 100 deben estar incluidos en el Sprint 1, por ser considerados de extrema importancia para el proyecto.
 2. Todos los elementos de importancia de 99-50 deben estar incluidos en el Sprint 2, pero eso depende de la velocidad del Sprint.
 3. Los elementos con importancias de 49-25 se pueden incluir en el último Sprint, según el avance del equipo ya que son requisitos que no alteran el desarrollo del mismo o funcionalidades del mismo.
- **Tiempo Estimado (TS):** Se asigna por medio de cartas con ponderaciones del 1 al 20 entre el Product Owner y los miembros del equipo Scrum.

2.6.7 Pila del Producto (*Product Backlog*)

La lista de objetivos / requisitos priorizada representa la visión y expectativas del cliente respecto a los objetivos y entregas del producto o proyecto. El cliente es el responsable de crear y gestionar la lista (con la ayuda del Facilitador y del equipo, quien proporciona el coste estimado de completar cada requisito). Dado que reflejar las expectativas del cliente en esta lista, permite involucrarle en la dirección de los resultados del producto o proyecto. Contiene los objetivos / requisitos de alto nivel del producto o proyecto, que se suelen expresar en forma de historias de usuario. Para cada objetivo / requisito se indica el valor que aporta al cliente y el coste estimado de completarlo. La lista se prioriza mediante el balanceo del valor que cada requisito aporta al negocio frente al coste estimado que tiene su desarrollo, es decir, basándose en el Retorno de la Inversión (ROI) (Díaz Ortiz y Romero Suárez, 2017).

En la Tabla 2.2 se muestra la pila de productos del proyecto.

Tabla 2.2: Pila de productos ordenada por prioridad e importancia.

| <i>Módulo</i> | <i>Historia de Usuario</i> | <i>Prioridad</i> | <i>Importancia</i> | <i>Tiempo estimado (días)</i> |
|-----------------|--|------------------|--------------------|-------------------------------|
| <i>Hardware</i> | Desarrollo del <i>driver</i> para instalar en el <i>hardware</i> . | Alta | 100 | 12 |
| Cliente | Implementar red neuronal. | Alta | 100 | 18 |
| Cliente | Graficado de reporte en tiempo real. | Alta | 98 | 10 |
| Cliente | Análisis de mediciones. | Alta | 89 | 10 |
| Cliente | Almacenar datos de sensores en tiempo real. | Media | 70 | 8 |
| Sistema | Establecer comunicación con dispositivo de <i>hardware</i> . | Media | 60 | 9 |
| Sistema | Normalizar datos provenientes del <i>hardware</i> . | Baja | 30 | 8 |

2.6.8 Planeación de las Entregas (*Sprints Backlog*)

Sprint Backlog es el conjunto de ítems del *Backlog* del Producto seleccionados para *Sprint* más un plan para entregar el producto, incrementar y realizar el objetivo de *Sprint*. *Sprint Backlog* es un pronóstico por el Equipo de Desarrollo sobre qué funcionalidad va a ser el próximo *incremento* y el trabajo necesario para entregar esa funcionalidad en un *incremento* "Hecho". Es la lista que descompone las funcionalidades de la pila del producto (historias de usuario) en las tareas necesarias para construir un incremento: una parte completa y operativa del producto (Villalva Castañeda, 2017).

El sprint es cada ciclo o iteración de trabajo que produce una parte del producto terminada y que funciona de forma operativa (Rey Guevara, 2017). Su velocidad de desarrollo se define según la importancia de las historias de usuario y el tiempo de trabajo del equipo de Scrum para el proyecto y la dedicación que se le da al mismo.

El tiempo del equipo de trabajo está dado dentro de las jornadas laborales de ocho horas a la semana de lunes a viernes y sábados cuatro horas durante tres meses (Tabla 2.3), de los cuales, se obtiene como resultado la cantidad de días de trabajo dedicados al proyecto por cada Sprint.

Tabla 2.3. Tiempo para cada Sprint.

| <i>Equipo Scrum</i> | <i>Jornada Laborar (h)</i> | <i>Horas de trabajo al proyecto por día (h)</i> | <i>Horas de trabajo al proyecto por semana (h)</i> | <i>Semanas de Trabajo por mes (u)</i> | <i>Total de horas (h)</i> | <i>Total de días laborables para el proyecto (días)</i> |
|---------------------|----------------------------|---|--|---------------------------------------|---------------------------|---|
| Renier Fagundo Mesa | 8 | 8 | 44 | 4 | 176 | 22 |

Debido al tiempo de dedicación que se le da al proyecto y las horas asignadas dentro de horario de trabajo, se esperan tener algunas distracciones e impedimentos pero que están dentro de las estimaciones para el proyecto, por lo cual, el *Product Owner* da un factor de dedicación del 90 % del tiempo comprendido para el mismo. Según lo indicado se procede a calcular la velocidad estimada para el desarrollo de los *Sprints*, la cual es: 19,8 días.

De acuerdo a la velocidad obtenida para la ejecución de cada Sprint y al tomar en cuenta el nivel de importancia definido por cada historia de usuario, se procede a agrupar las mismas y determinar la cantidad de *Sprints* para el proyecto. La planificación de los *Sprints* se detalla en las Tablas 2.4 a 2.7.

Tabla 2.4. Estimación del Sprint N° 1.

| <i>Módulo</i> | <i>Historia</i> | <i>Prioridad</i> | <i>Importancia</i> | <i>Tiempo estimado (días)</i> |
|---------------------------------|---|------------------|--------------------|-------------------------------|
| <i>Hardware</i> | Desarrollo del driver para instalar en el <i>hardware</i> . | Alta | 100 | 12 |
| Cliente | Almacenar datos de sensores en tiempo real. | Media | 70 | 8 |
| Total de días del Sprint | | | 20 días | |

Tabla 2.5. Estimación del Sprint N° 2.

| <i>Módulo</i> | <i>Historia</i> | <i>Prioridad</i> | <i>Importancia</i> | <i>Tiempo estimado (días)</i> |
|---------------------------------|---------------------------|------------------|--------------------|-------------------------------|
| Cliente | Implementar red neuronal. | Alta | 100 | 18 |
| Total de días del Sprint | | 18 días | | |

Tabla 2.6. Estimación del Sprint N° 3.

| <i>Módulo</i> | <i>Historia</i> | <i>Prioridad</i> | <i>Importancia</i> | <i>Tiempo estimado (días)</i> |
|---------------------------------|---|------------------|--------------------|-------------------------------|
| Cliente | Graficado de reporte en tiempo real. | Media | 98 | 10 |
| Sistema | Normalizar datos provenientes del <i>hardware</i> . | Baja | 30 | 8 |
| Total de días del Sprint | | 18 días | | |

Tabla 2.7 Estimación del Sprint N° 4.

| <i>Módulo</i> | <i>Historia</i> | <i>Prioridad</i> | <i>Importancia</i> | <i>Tiempo estimado (días)</i> |
|---------------------------------|--|------------------|--------------------|-------------------------------|
| Cliente | Análisis de mediciones. | Alta | 89 | 10 |
| Usuario | Establecer comunicación con dispositivo de <i>hardware</i> . | Media | 60 | 9 |
| Total de días del Sprint | | 19 días | | |

2.7 Acondicionamiento de la señal

El acondicionamiento de la señal se realiza al considerar que la corriente a medir es alterna, forma de onda sinusoidal, es decir, tiene hemiciclos negativos y positivos mientras que el Arduino solo puede medir magnitudes de voltaje positivas. El otro aspecto es que debe estar en el rango de 0 V a 5 V, por lo que la amplitud máxima no debe exceder este rango. Para lograrlo se implementó un divisor de voltaje para correr el offset de 0 V a 2,5 V ($5V/2$) y para ello se emplean dos resistencias iguales con valores de 10 k Ω cada una, que forman un divisor de tensión según se aprecia en la Figura 2.7. Como el HWCT004 no tiene una resistencia de carga incorporada (*burden resistor*) la señal de corriente necesita ser convertida a señal de voltaje y por eso se usa una resistencia de carga (*Burden*). El valor se calcula de manera que el rango de voltaje no sobrepase los 5 V para la corriente máxima de 50 A. Los cálculos de los valores de los componentes se basan en los realizados según (Energy, 2019) donde:

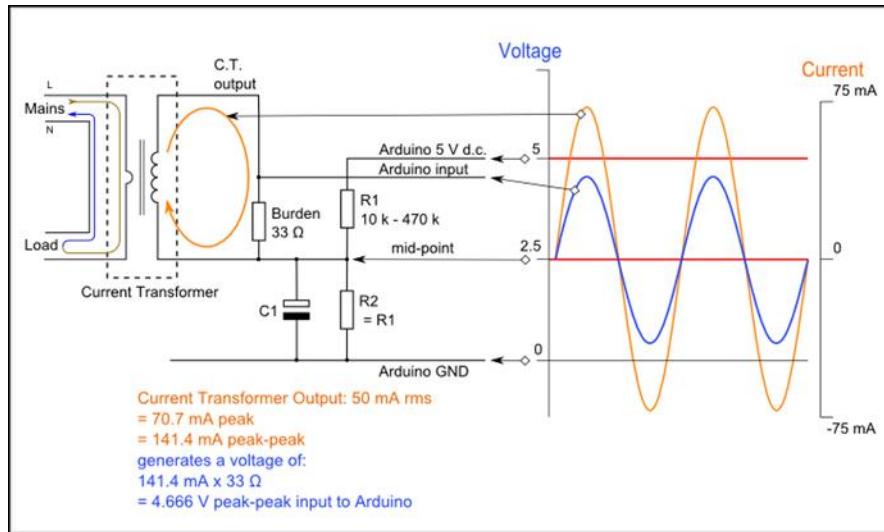


Figura 2.7. Esquema de la interfaz de adaptación. (Energy, 2019).

2.7.1 Empleo de la raíz media cuadrática (RMS)

En la interfaz principal (Figura 2.8). se puede observar la gráfica del muestreo en tiempo real de comportamiento del consumo de corriente y de potencia aparente. Estos valores son los valores medios cuadráticos resultados de la aplicación de las ecuaciones en 5 000 muestras por segundo que corresponde al muestreo del sensor por parte del dispositivo de *hardware*.

$$P_{pc} = \text{RMS_current} * \sqrt{2} = 50 \text{ A} \times 1,414 = 70,71 \text{ A} \quad (2.1)$$

$$S_{pc} = P_{pc} / \text{no_of_turns} = 70,71 \text{ A} / 1000 = 0,0707 \text{ A} \quad (2.2)$$

$$I_{br} = (\text{AREF}/2) / S_{pc} = 2,5 \text{ V} / 0,0707 \text{ A} = 35,4 \Omega \quad (2.3)$$

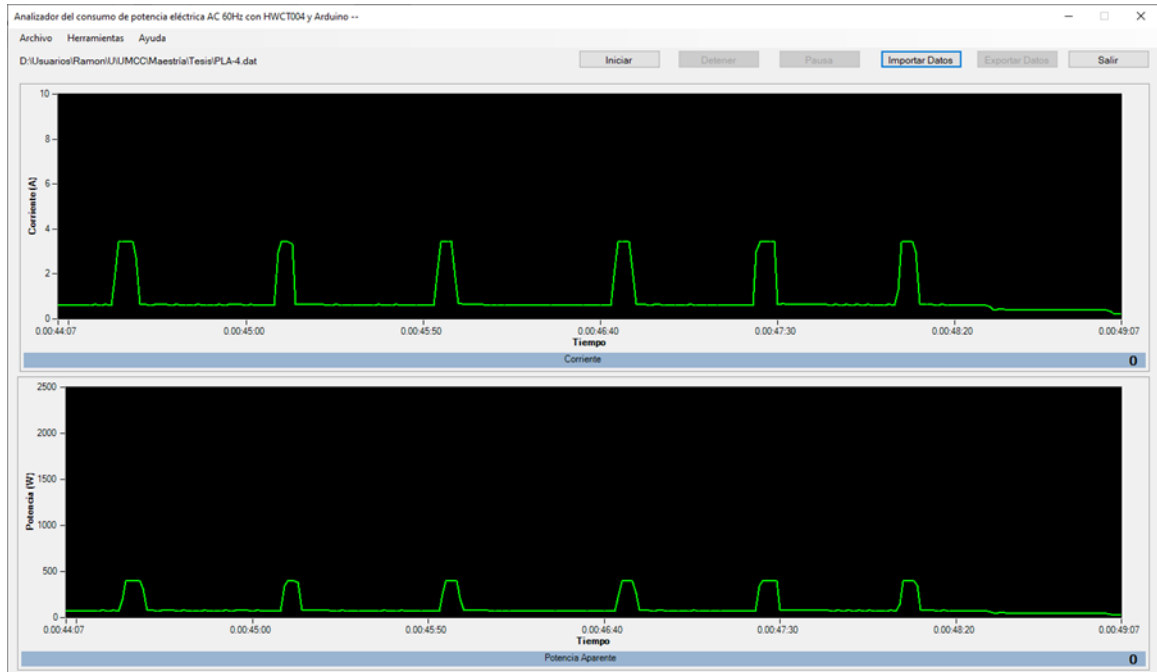


Figura 2.8. Interfaz principal. Fuente: Elaboración propia.

2.7.2 Ajuste de la curva mediante el método spline cúbico

En el subcampo matemático del análisis numérico, un *spline* es una curva diferenciable definida en porciones mediante polinomios (Figura 2.9). Para el ajuste de curvas, los *splines* se utilizan para aproximar formas complicadas. La simplicidad de la representación y la facilidad de cómputo de los *splines* los hacen populares para la representación de curvas en informática, particularmente en el terreno de los gráficos por ordenador (Figura 2.10).

Definición: La función $s(x)$ se llama cúbica a trozos en $[x_0, x_n]$ si existen polinomios cúbicos $q_0(x), q_1(x), \dots, q_{n-1}(x)$ tales que:

$$s(x) = q_k(x) \text{ en } [x_k, x_{k+1}], \text{ para } k = 0, 1, \dots, n-1 \quad (2.4)$$

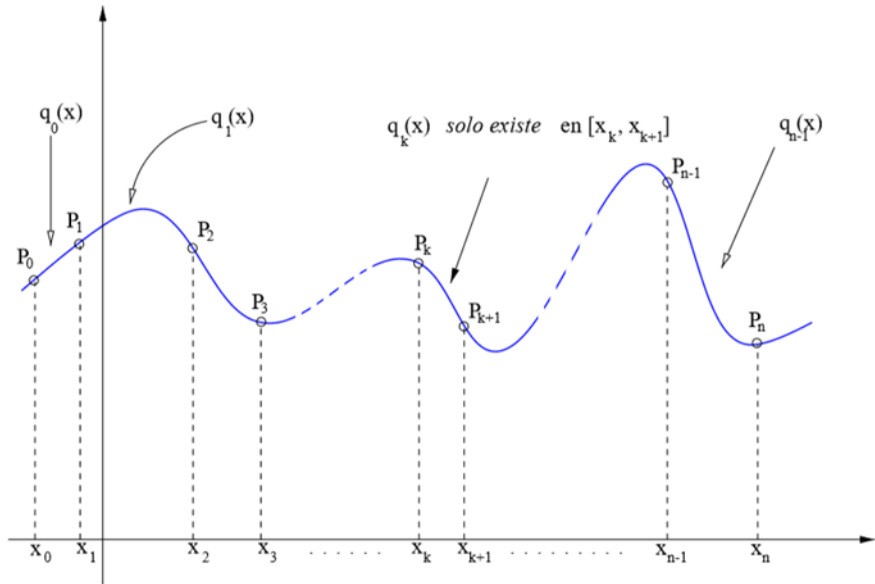


Figura 2.9. Ejemplo de curva ajustada (Quesada, 2000).

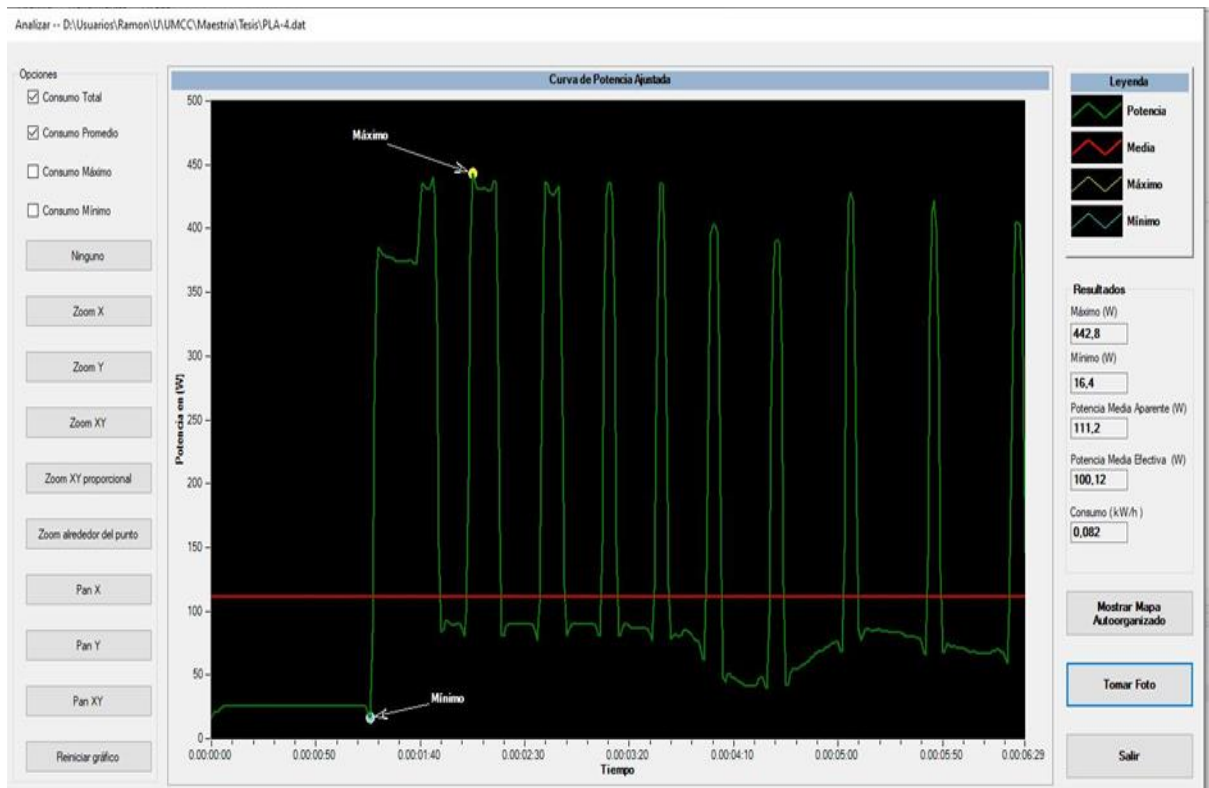


Figura 2.10. Interfaz de análisis. Fuente: Elaboración propia.

2.7.3 Filtro de Chebyshev

Nombrados en honor al matemático ruso Pafnuti Chebyshev, debido a que la función matemática de su respuesta en frecuencia utiliza los denominados polinomios de Chebyshev (Wikipedia, 2019).

Con los filtros de Chebyshev se consigue una caída de la respuesta en frecuencia más pronunciada en frecuencias bajas debido a que permiten rizado en alguna de sus bandas (paso o rechazo) (Rastogi *et al.*, 2013). A diferencia del Filtro de Butterworth donde los polos se distribuyen sobre una circunferencia, los polos del filtro Chebyshev lo hacen sobre una elipse; sus ceros se encuentran en el eje imaginario.

Se conocen dos tipos de filtros Chebyshev (Figura 2.11), dependiendo del rizado en alguna banda determinada (Liao *et al.*, 2007; Wikipedia, 2019):

- Filtros de Chebyshev de tipo I.
- Filtros de Chebyshev de tipo II.

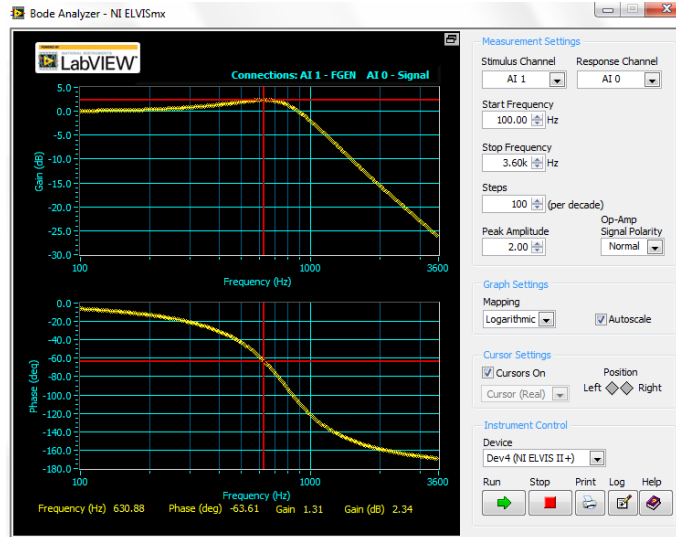


Figura 2.11. Gráfica de Bode para respuesta de Chebyshev (Pérez, 2017).

2.7.4 Mapa Auto-Organizado

La descripción del algoritmo expuesta por Andana Quijada y Duran Figueroa (2015) es la siguiente:

Sea N el número de neuronas de entrada y M el número de neuronas de salida.

1. Inicializar los pesos de la matriz de salida con valores aleatorios Normalmente, vectores en $[0,1] n$.
2. Presentar a la red una información de entrada en forma de vector.
 - $E_k = (e_1, \dots, e_N)$, cuyas componentes sean valores continuos.
3. Determinar la neurona vencedora o BMU (*Best Matching Unit*) de la capa de salida: será aquella cuyo vector de pesos, W_j , sea el más parecido a la información de entrada E_k . Se recuerda que las componentes de W_j son los

valores de los pesos de las conexiones entre esa neurona, j , y cada una de las neuronas de entrada. Para ello se calcula la distancia euclidiana entre los vectores E_k y W_j para cada neurona de salida. La expresión matemática sería la siguiente:

$$dj = \sum_{i=1}^n (e_i - w_{ij})^2 : 1 \leq j \leq M \quad (2.5)$$

Por motivos de eficiencia no será calculada la raíz cuadrada.

4. Se calcula el radio del entorno de BMU. Este radio comenzará siendo grande (como para cubrir la red completa) y se va reduciendo en cada iteración. La fórmula que establece el radio en función de la iteración (que hace que vaya disminuyendo, pero no linealmente) es:

$$r(t) = r_0 \exp(-t / \lambda) \quad (2.6)$$

Donde r_0 es el radio inicial (habitualmente, el radio de la red, es decir, uno suficiente para cubrir en el primer paso todos los nodos) y λ una constante que permite hacer que el radio sea muy pequeño cuando llegamos a la iteración máxima.

$$\lambda = \frac{t_T}{\ln r_0}; \quad (2.7)$$

donde t_T es el tiempo de entrenamiento.

La Figura 2.12 muestra el efecto de ir reduciendo paulatinamente el radio del entorno, donde se marcan los nodos que se verían afectados si el nodo BMU es el nodo amarillo.

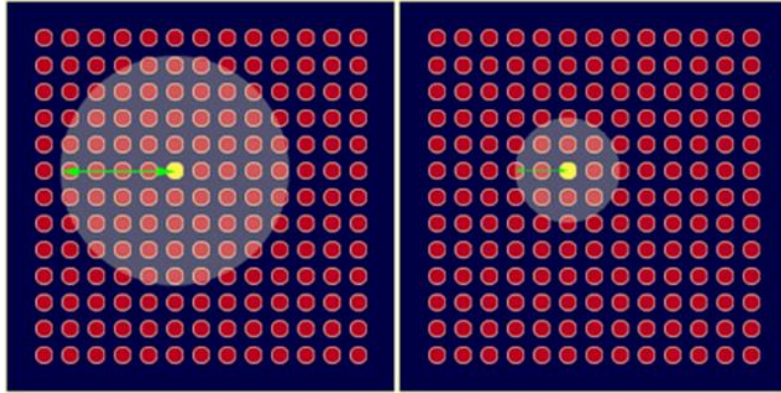


Figura 2.12. Reducción Radio de Entorno de la BMU. (Sancho Caparrini, 2014).

5. Una vez localizada la neurona vencedora, se actualizan los pesos de sus conexiones de entrada y también los de las neuronas vecinas (las que pertenecen a su zona de vecindad). Lo que se consigue es asociar la información de entrada con una cierta zona de la capa de salida.

Para aproximar los pesos (W) de los nodos de la matriz de aprendizaje al vector de entrenamiento (V) se puede usar la siguiente fórmula:

$$W(t+1) = W(t) + L(t)[V(t) - W(t)] \quad (2.8)$$

El factor $L(t)$ se denomina tasa de aprendizaje, y permite aproximar W a V con el paso del tiempo. Como se quiere que su valor también disminuya a medida que el tiempo pasa, se puede usar una expresión similar a la del radio:

$$L(t) = L_0 \exp(-t / \lambda). \quad (2.9)$$

El valor de L_0 se ajusta experimentalmente. Con el fin de que el efecto de aprendizaje sea más notable en las cercanías del BMU, se puede añadir un factor

más al producto anterior, que hace que los nodos más cercanos al BMU se vean más afectados:

$$W(t+1) = W(t) + D(t)L(t)[V(t) - W(t)] \quad (2.10)$$

donde, por ejemplo, se puede hacer que $D(t)$ siga una gaussiana de la forma:

$$D(t) = \exp\left(-\frac{d^2}{2r(t)^2}\right); \quad (2.11)$$

siendo d la distancia del nodo que se ajusta al BMU.

6. El proceso se debe repetir, vuelve a presentar todo el juego de patrones de aprendizaje E1, E2..., un mínimo de 500 veces ($t = 500$).
7. Después del aprendizaje se pasaría a la fase de funcionamiento de la red, durante la cual se pueden presentar nuevos puntos para la que la red clasifique de acuerdo a lo aprendido.

En la Figura 2.13 se muestra la interfaz de Clasificación de datos la cual nos permite clasificar tres tipos de clases en las que en un primer momento se clasifican las mediciones del monitoreo realizado estas clases son Alto_consumo que se representa en el mapa con el color negro, Consumo_medio representado por el color rojo y Bajo_consumo en color azul. El proceso de selección de las clases debe ser a partir de que se cuente con un alto volumen de información por lo que es un proceso a largo plazo, que se extiende fuera del límite del tiempo empleado para la investigación, por lo que hasta el momento solo se

realizó la implementación del algoritmo y de la vista cuyo objetivo será el futuro entrenamiento de la red neuronal.

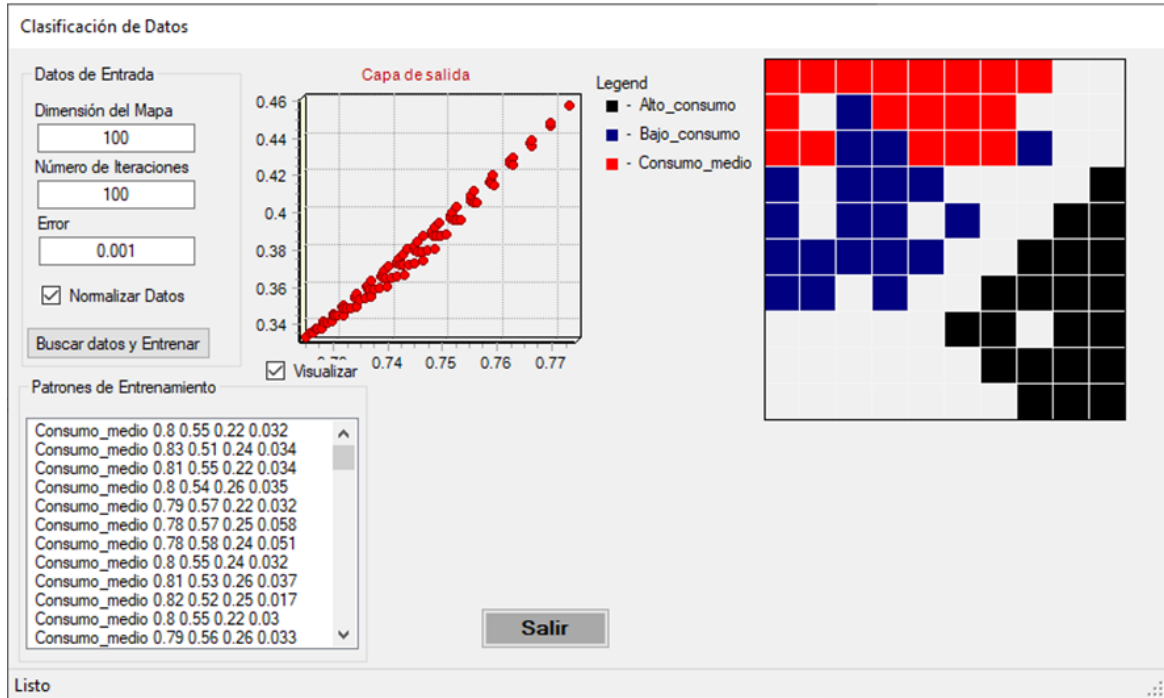


Figura 2.13. Interfaz de clasificación de datos. Fuente: Elaboración propia.

2.8 Estimación del costo

Uno de los aspectos a abordar dentro de este capítulo es el del análisis económico de la solución propuesta. A continuación, se realizará un desglose del coste de los elementos necesarios en esta investigación. Dichos elementos incluyen costes de personal, de *hardware* y de *software*.

La investigación se realizará entre el 1 de abril de 2020 al 31 de agosto de 2020 por lo tanto son 5 meses de trabajo. Teniendo en cuenta una jornada laboral de 8 horas tendremos un

total de 960 horas de trabajo, distribuidas entre diferentes tareas y diferentes roles profesionales que las llevan a cabo.

2.8.1 Coste de personal

La metodología de *software* escogida propone un equipo de desarrollo pequeño donde cada integrante tiene su rol y funciones bien definidas. Para determinar el coste del personal involucrado se va desglosar el equipo de acuerdo a la categoría de cada uno, así como en la fase donde participa y queda el desglose del coste como se aprecia en la siguiente tabla (Tabla 2.8).

Tabla 2.8. Coste de personal.

| <i>Fase</i> | <i>Categoría</i> | <i>Horas</i> | <i>Coste/hora</i> | <i>Coste</i> |
|----------------|-------------------|--------------|-------------------|-------------------|
| Documentación | Analista | 86 | 40 CUP | 3 440 CUP |
| Análisis | Analista | 105 | 40 CUP | 4 200 CUP |
| Diseño | Diseñador | 172 | 30 CUP | 5 160 CUP |
| Diseño gráfico | Diseñador gráfico | 66 | 30 CUP | 1 980 CUP |
| Implementación | Programador | 475 | 20 CUP | 9 500 CUP |
| Pruebas | Programador | 66 | 20 CUP | 1 320 CUP |
| Total | | | 1 344 CUP | 25 600 CUP |

2.8.2 Coste de *hardware*

Para el *hardware* calcularemos el coste según el período de amortización teniendo en cuenta una duración del proyecto de 5 meses. El equipo está formado por 2 miembros y 2 ordenadores para el desarrollo (Tabla 2.9).

Tabla 2.9. Coste del hardware.

| <i>Equipo</i> | <i>Coste</i> | <i>Coste de amortizado</i> |
|--------------------------|-------------------|----------------------------|
| CPU | 657,50 CUP | 219,17 CUP |
| Monitor | 154,24 CUP | 15,42 CUP |
| Teclado | 4,50 CUP | 0,65 CUP |
| Mouse | 3,46 CUP | 1,73 CUP |
| Computadora portátil | 500,00 CUP | 166,67 CUP |
| Arduino Nano | 37,00 CUP | 18,50 CUP |
| Componentes Electrónicos | 5,00 CUP | 2,50 CUP |
| Total | 1361,7 CUP | 424,64 CUP |

2.8,3 Coste de software

En la realización de esta investigación se utilizó un *software* propietario (NI Measurement Studio) cuyo costo es \$671 USD (NI Corporation, 2020), equivalente al cambio hoy (septiembre 2020) 16775,00 CUP.

2.8.4 Coste total

A partir del coste de cada uno de los elementos necesarios para la investigación se puede llegar al coste total, como se aprecia en la siguiente tabla (Tabla 2.10).

Tabla 2.10. Coste total.

| <i>Tipo de coste</i> | <i>Total</i> |
|--------------------------|---------------------|
| Coste de personal | 25 600,00 CUP |
| Coste de <i>hardware</i> | 424,64 CUP |
| Coste de <i>software</i> | 16775,00 CUP |
| Total | 42799,64 CUP |

Por tanto, el coste total para la presente investigación asciende a: 42799,64 CUP.

2.9 Conclusiones parciales del capítulo

Una vez finalizado el presente capítulo, se arribó a las siguientes conclusiones parciales:

1. El dispositivo acondicionador que se diseñó y fabricó, acondicionó apropiadamente las señales provenientes del sensor.
2. El programa que se desarrolló para el monitoreo y control de la interfaz respecto al sensor que se instaló, permitió medir, registrar, visualizar, analizar el comportamiento y determinar el consumo eléctrico total.
3. La red neuronal multicapa desarrollada para la creación del mapa autoorganizado de patrones, funcionó según lo esperado y quedó lista para la determinación de los patrones de consumo.

CAPÍTULO 3. ANÁLISIS DE LOS RESULTADOS

En este capítulo se describen los casos de estudio correspondientes a aplicaciones del sistema para el monitoreo del consumo de energía eléctrica de un proceso, donde se evidencia el modo de empleo y la utilidad del mismo para el análisis ulterior con el propósito de adoptar las medidas apropiadas para lograr la sostenibilidad del mismo. También se explica el método e instrumento utilizado para la verificación de la medición con el sistema para garantizar la confiabilidad del mismo.

3.1 Multímetro FLUKE

El multímetro Fluke 376 mide tensión y corriente de CA rms, tensión y corriente de CC, corriente de arranque y capacitancia. El multímetro 376 mide también frecuencia y milivoltios de CC. La sonda de corriente tipo pinza flexible iFlex extraíble, que se incluye con el 376 (y es opcional en el 374 y 375), amplía el rango de medida a 2.500 A CA. La sonda de corriente flexible ofrece mayor flexibilidad de visualización, permite realizar medidas en conductores de tamaño difícil y acceder de forma más fácil a los cables (Figura 3.1, se muestra Fluke 376) (Fluke Corp, 2010).

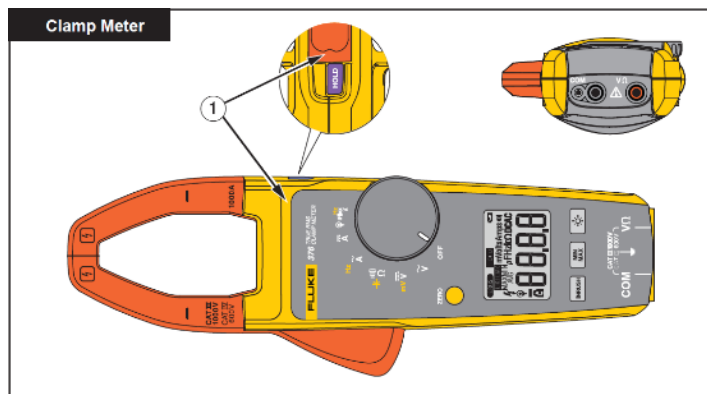


Figura 3.1: Multímetro con sonda tipo pinza (Fluke Corp, 2010).

Este instrumento se usó para calibrar y comprobar la precisión de las mediciones realizadas con el sistema de monitoreo que se desarrolló y arrojó como resultado una precisión similar a la del instrumento referencia Fluke 376, por lo que se constató la confiabilidad de las mediciones realizadas (Véase Anexo I). A partir de estos resultados se procedió con los casos de estudio siguientes.

3.2 Caso de estudio 1: Monitoreo y medición del consumo de una impresora 3D durante la impresión de una pieza

Durante el proceso de impresión de una pieza se hizo necesario el monitoreo y medición continuo del consumo de energía eléctrica para analizarlo posteriormente con el propósito de optimizar parámetros de fabricación, ya que para realizarlo, se podía apelar a la medición manual con instrumentos como un amperímetro digital, la cual es imprecisa y muy engorrosa, ya que se tienen que anotar todos los cambios perceptibles a lo largo del proceso y después realizar los cálculos correspondientes, o a la ficha técnica suministrada por el fabricante, la cual también es imprecisa, ya que abarca solo el caso de máximo consumo, valor que no se alcanza en todo tipo de piezas. Se tiene en cuenta el hecho de que la precisión de la medición es importante no solo desde el punto de vista del ahorro, sino también para la optimización del diseño y fabricación al menor costo y la mayor calidad posible, es decir, lograr la sostenibilidad.

En la Figura 3.2 se pueden apreciar a simple vista varias de las características distintivas de la herramienta de análisis:

1. El eje de tiempo con precisión de segundos, permite observar el instante de ocurrencia de eventos.

2. Es fácil calcular la duración de un evento.
3. Se puede mostrar máximo, mínimo y promedios.
4. Muchas de las características son opcionales y selectivas.
5. Herramientas para adecuar el gráfico al análisis.
6. Se muestran también los resultados numéricos.
7. Permite documentar con foto el análisis para su inclusión en trabajos científicos.

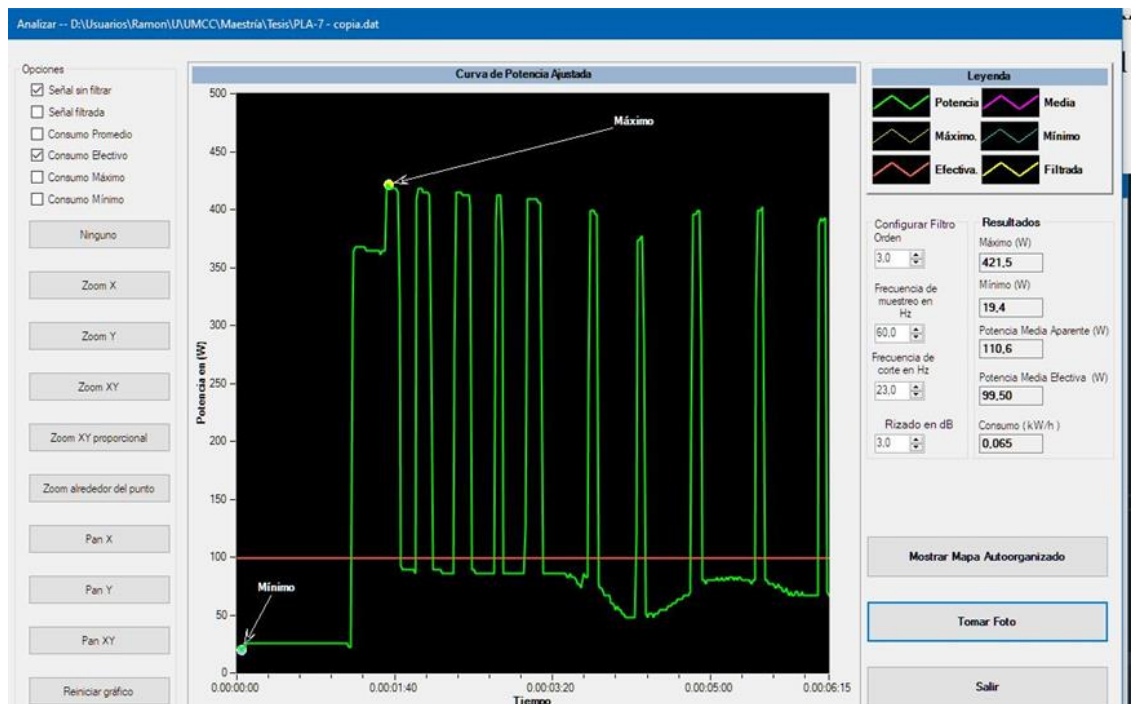


Figura 3.2. Detalles de la ventana de análisis. Elaboración propia.

Una de las piezas tenida en cuenta fue la denominada PLA1, de una familia de piezas muy similares de un mismo proyecto que están en estudio, la que se muestra en la Figura 3.3 y representa una de las variantes utilizadas. Aunque se tomó esta pieza, se imprimieron siete

variantes de la misma para la realización de un análisis comparativo que también incluyó otras variables. Cabe destacar que el consumo de energía eléctrica es una de las variables de alto impacto en el proceso de fabricación y en los indicadores de sostenibilidad, es por ello que fue necesario el estudio minucioso de su fabricación, especialmente, desde el punto de vista del consumo eléctrico.



Figura 3.3. Piezas denominadas PLA1 y PLA7 respectivamente.

Mediante el uso del sistema de monitoreo y medición se pudo hacer un análisis exhaustivo del comportamiento del consumo de energía eléctrica en cada instante durante el proceso de impresión como se puede observar en la Figura 3.4.

La ventana de análisis permite realizar varias acciones orientadas al análisis de las mediciones y su interpretación. Por ejemplo, es posible seleccionar porciones del gráfico en un intervalo de tiempo determinado con el propósito de determinar la duración de un pulso de consumo, su amplitud, desplazar el gráfico en el tiempo, ampliarlo, etc. En este caso se observó el comportamiento en el intervalo de tiempo comprendido entre el minuto 5, segundo 56 hasta el minuto 10 segundo 8.

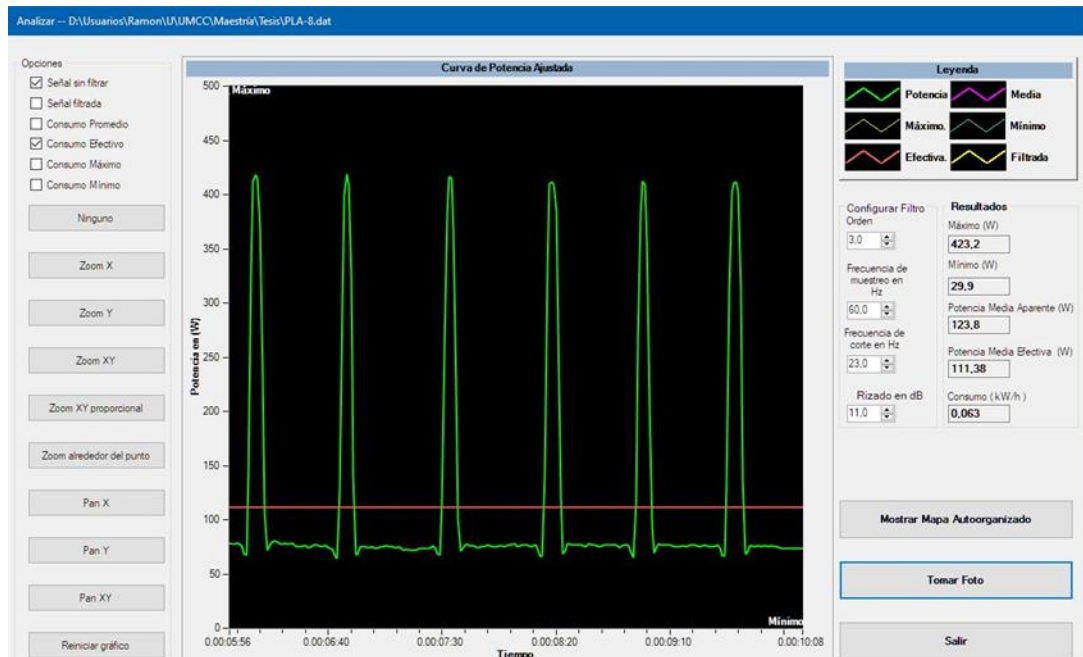


Figura 3.4. Ventana de análisis. Elaboración propia.

En la Figura 3.5 se puede observar la medición del ancho de un pulso de consumo alto comprendido entre el minuto 6, segundo 4 hasta el minuto 6, segundo 13, lo que permite calcular el ancho del mismo en 9 segundos de duración. También se puede apreciar los resultados del máximo consumo en esta impresión que fue de 423,2 W, el mínimo 29,9 W.

La potencia media aparente 123,8 W, la potencia media efectiva 111,38 W (mostrada en el gráfico en color rojo) con un consumo total durante el proceso de 0,063 kW/h por cada pieza impresa. Además, en algunos casos resulta conveniente filtrar la señal original para su análisis. En ese caso se aplica un filtro Chebyshev pasa bajos, el que puede ajustarse en sus parámetros dinámicamente para adecuarlo a la necesidad de los usuarios según Figura 3.6.

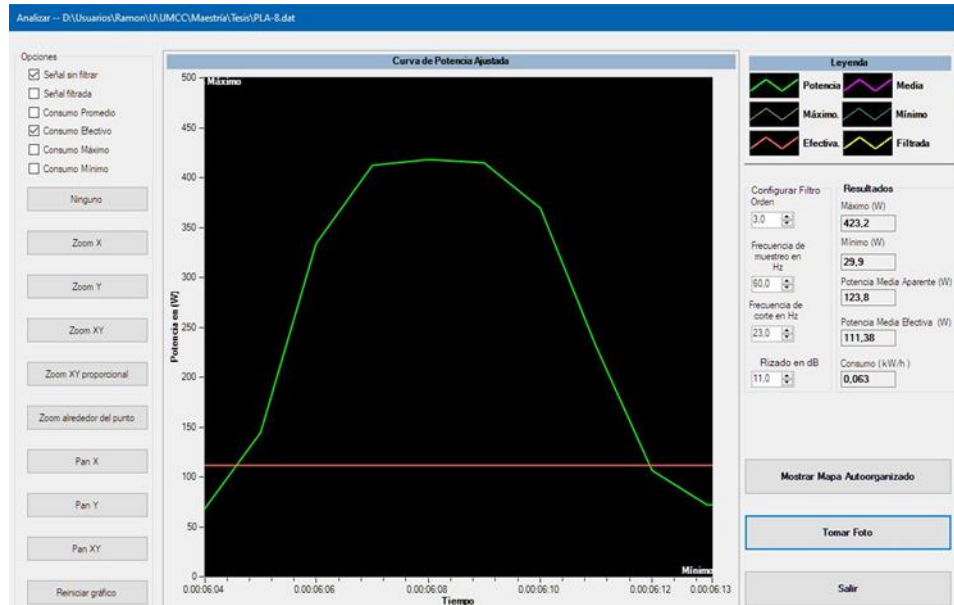


Figura 3.5. Selección de un pulso de consumo. Elaboración propia.

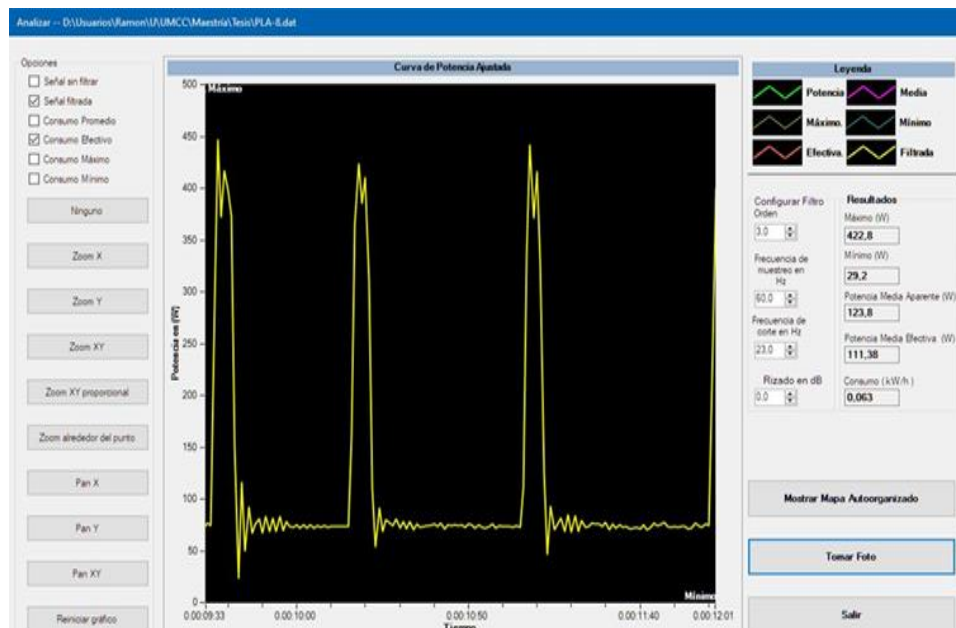


Figura 3.6. Señal filtrada con Chebyshev pasa bajos. Elaboración propia.

3.3 Caso de estudio 2: Monitoreo y medición del consumo de un aire acondicionado tipo Split de 1 tonelada de capacidad

Una de las causas más comunes de derroche de energía eléctrica es el uso inapropiado de equipos de climatización, particularmente equipos domésticos. Sus propietarios incurren en grandes gastos y por ende en un incremento en el consumo de combustible fósil para la generación de esa energía.

En el siguiente caso de estudio se muestra el comportamiento del consumo eléctrico de un aire acondicionado tipo Split, de 3 430 W (11 703,66 BTU/h), durante 9 horas, 17 minutos y 26 segundos (Figura 3.7). Las condiciones de operación del mismo se fijaron a 25 °C para el enfriamiento, ventilador en mínimo y temperatura ambiente inicial de 30 °C, en el horario comprendido entre las 11:00 pm y 8:17 am del día siguiente (Figura 3.8).



| CHIGO Room Air-conditioner | |
|---|-----------------------|
| Model | KFTX-002 |
| Model of indoor unit | KFTX-002 |
| Model of outdoor unit | KFTX-002 |
| Electric shock protection class | I |
| Rated volt | 220V~ |
| Rated frequency | 60Hz |
| Cooling capacity | 3430W |
| Cooling power input | 1070W |
| Cooling current input | 4.9A |
| Indoor discharge air-flow | ≥500m ³ /h |
| Rated power input | 1500W |
| Rated current input | 6.8A |
| Indoor/Outdoor noise | 29~40/54dB(A) |
| Refrigerant | R410A/1200g |
| HP. PS | 4.2MPa |
| LP. PS | 2MPa |
| Indoor max operating pressure of heat exchanger | 4.2MPa |
| Indoor unit weight | 10kg |
| Date of manufacture | |
| Number of manufacture | |

Figura 3.7. Ficha técnica del aire acondicionado. Elaboración propia.

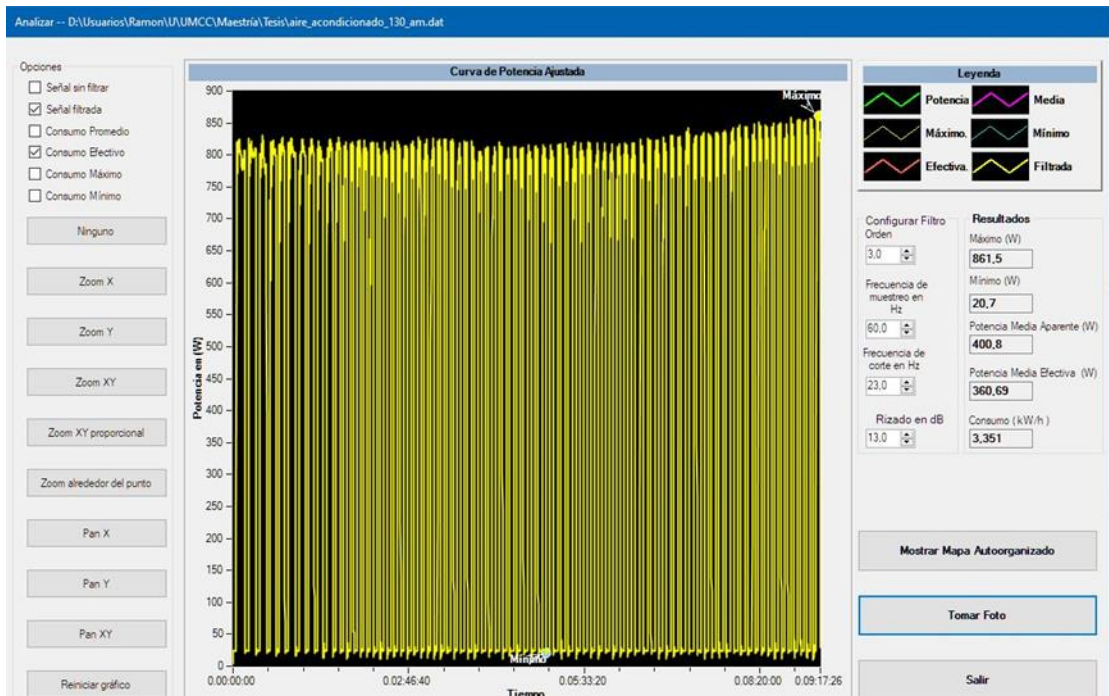


Figura 3.8. Consumo de un aire acondicionado en 9 horas y 17 minutos.

Se puede apreciar (Figura 3.8) como los ciclos de mayor consumo son más anchos en la primera hora y a partir de ahí comienza el estrechamiento de estos cada vez más, lo que significa una disminución en el consumo durante ese horario. En la Figura 3.9 se observa el detalle del mayor consumo antes de la transición de la primera hora a la siguiente.

A diferencia de la impresora 3D, el aire acondicionado presenta una carga inductiva que ocasiona picos que pueden distorsionar el análisis por lo que se empleó el filtro Chebyshev pasa bajos para disminuir elementos de la señal que no son de interés.

Con la información ofrecida por el sistema de monitoreo el especialista puede tomar decisiones sobre la configuración más económica dadas las condiciones de operación, así como, tener una imagen real del comportamiento del equipo durante su operación.

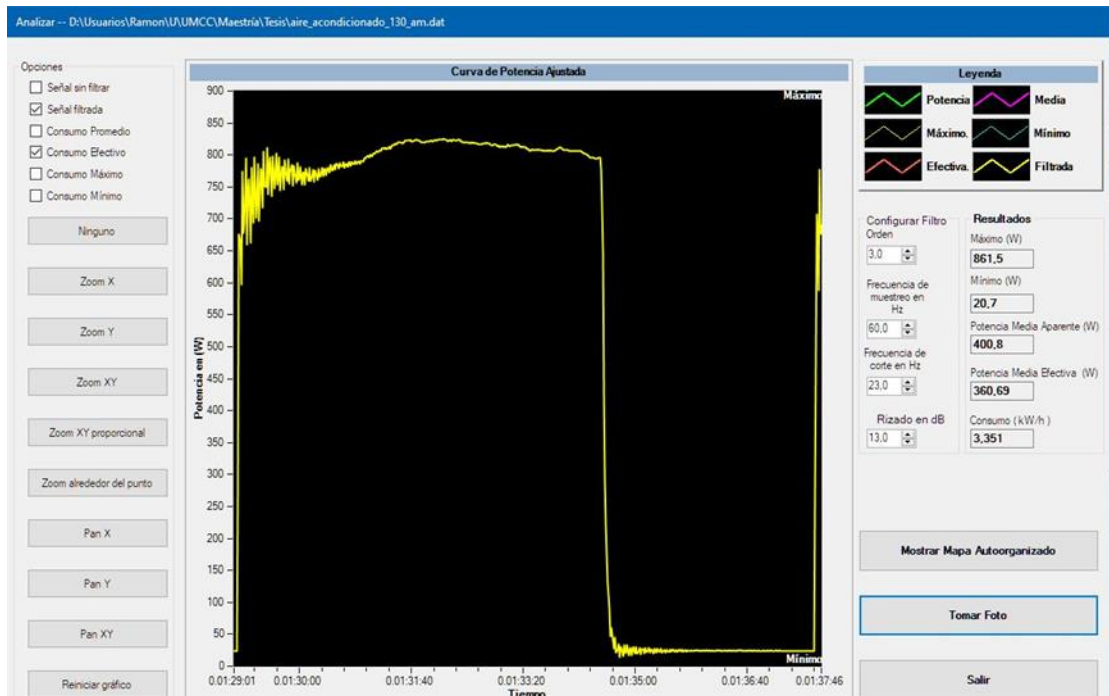


Figura 3.9. Comparación del pulso de consumo con el standby. Elaboración propia.

3.4 Conclusiones parciales del capítulo

Una vez finalizado el presente capítulo, se arribó a las siguientes conclusiones parciales:

1. La comparación de las mediciones hechas con las realizadas mediante el instrumento de precisión Fluke 376 resultaron en una alta coincidencia, por lo que se demostró la validez de las mediciones realizadas con el sistema de monitoreo desarrollado.
2. Se demostró la utilidad del sistema cuando se aplica en un sistema de fabricación, lo que permitió establecer con exactitud el consumo de energía eléctrica en este proceso, así como, facilitar la determinación del diseño más apropiado y sostenible.
3. Permitted observar detalladamente el comportamiento del consumo de energía eléctrica de un aire acondicionado durante más de 9 horas lo que facilitó observar la duración de

los intervalos de mayor consumo y su distribución en el tiempo con precisión de segundos, lo que hace posible el análisis preciso para la adopción de medidas de ahorro.

4. Se demostró la utilidad de las herramientas incorporadas a la aplicación de escritorio ya que facilitó el análisis detallado de los datos obtenidos.

CONCLUSIONES

Como resultado final del trabajo desarrollado se arribó a las siguientes conclusiones:

1. El dispositivo acondicionador que se diseñó y fabricó, acondicionó apropiadamente las señales provenientes del sensor.
2. El programa que se desarrolló para el monitoreo y control de la interfaz respecto al sensor que se instaló, permitió medir, registrar, visualizar, analizar el comportamiento y determinar el consumo eléctrico total.
3. La calibración del sistema obtenido mediante el empleo de un instrumento de precisión (FLUKE 376) demostró la exactitud de las mediciones obtenidas.
4. La red neuronal multicapa desarrollada para la creación del mapa autoorganizado de patrones, funcionó según lo esperado y quedó a la espera de la determinación de los patrones de consumo.
5. En la validación de los casos estudiados, facilitó hacer un análisis detallado del consumo de energía eléctrica de cada uno y permitió la adopción de medidas apropiadas para contribuir a su sostenibilidad.

RECOMENDACIONES

Basadas en las conclusiones obtenidas y para la extensión futura del presente trabajo, se realizan las siguientes recomendaciones:

1. Concluir la determinación de los patrones de consumo para utilizar toda la funcionalidad de los mapas autoorganizados.
2. Incorporar a la solución el Arduino Ethernet Shield para eliminar la dependencia de la conexión USB para la transmisión de los datos.
3. Adicionar el *hardware* que permita leer el voltaje y el factor de potencia, es decir, sensores con estos propósitos para eliminar la necesidad de estimar el factor de potencia y leer y anotar el voltaje de referencia.
4. Realizar la tramitación pertinente para el registro del *software* en el registro de propiedad intelectual.
5. Difundir la existencia del sistema de monitoreo para ser aplicado a la producción donde sea posible.
6. Establecer asociaciones que permitan la fabricación y distribución del sistema de monitoreo, así como, la incorporación de los sensores necesarios para la medición del factor de potencia y el voltaje de línea.

REFERENCIAS BIBLIOGRÁFICAS

- Abur, A., Alvarado, F., Bel, C., Cañizares, C., Pidre, J., Navarro, A., . . . Expósito, A. (2002). *Análisis y operación de sistemas de energía eléctrica*: McGraw-Hill.
- Acosta, T. R., Méndez, G. L. C., y Lom, M. F. (2017). Sistema de monitoreo de consumo eléctrico inalámbrico y analizador de la calidad de la energía. *Universidad Autónoma de Ciudad Juárez*.
- Albán Guerrero, J. E., y Paguay Llamuca, N. J. (2017). *Diseño e implementación de un sistema SCADA con comunicación profibus para el control y monitoreo de procesos industriales en el laboratorio de automatización de la Facultad de Mecánica*. Escuela Superior Politécnica de Chimborazo, Riobamba-Ecuador.
- Andana Quijada, J. H., y Duran Figueroa, R. A. (2015). *Software para la Implementación de Mapas Auto-Organizados de Kohonen (SOM) para la clasificación de las acciones de la bolsa de Chile*. Universidad del Bio-Bio, Chillán.
- Ashlock, D., y Warren, A. (2015). Guía de acondicionamiento de señales para ingenieros. Retrieved from <http://ni.com/compactdaq>
- Astudillo, V., y Cadena, J. (2016). *Desarrollo de un prototipo para el seguimiento y cuidado de jardines urbanos basado en la recopilación de datos a través de la plataforma arduino*. Póntica Universidad Católica del Ecuador.,
- Barragán, H. (2019). *¿Que es Wiring?* (<http://wiring.org.co>). Universidad de los Andes, Bogotá. (ISSN 2011-3188)
- Barreto, G., A., J., M., C., Mota, L., M, G., R, S. y., y Frota, A. (2004). *Nonstationary time series prediction using local models based on competitive neural networks*. . Paper presented at the Proceedings of the 17th international conference on Innovations in applied artificial intelligence.
- Butow, E., y Ryan, T. (2002). *C# Your visual blueprint for building .NET applications*. Inc. 909 Third Avenue New York, NY 10022: Hungry Minds.
- Candia, L. D., Rodríguez, A. S., Castro, N., Bazán, P., Ambrosi, V. M., y Díaz, F. J. (2018). *Mejoras en maquinaria industrial con IoT: hacia la industria 4.0*. Paper presented at the XXIV Congreso Argentino de Ciencias de la Computación (La Plata, 2018).
- Cao, L. (2003). Support vector machines experts for time series forecasting. *Neurocomputing*, 321 – 339.
- Carretero Peña, A., y García Sánchez, J. M. (2012). *Gestión de la eficiencia energética: cálculo del consumo, indicadores y mejora*. (© AENOR (Asociación Española de Normalización y Certificación) Ed.).

- Cervantes Suarez, C. A. (2017). *Identificación de patrones de trayectorias vehiculares utilizando el algoritmo mapas Auto-Organizados*. Universidad de Guayaquil, Guayaquil-Ecuador.
- Charte Ojeda, F. (2002). *Microsoft Visual C#.Net*. Madrid, España: EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2002
- Civerchia, F., Bocchino, S., Salvadori, C., Rossi, E., Maggiani, L., Petracca, M., y DOI. (2017). Industrial Internet of Things Monitoring Solution for Advanced Predictive Maintenance Applications. *Journal of Industrial Information Integration*, Page 550. doi:10.18502/keg.v3i1.1458
- Cruz Vega, M., Oliete Vivas, P., Morales Rios, C., González, L. C., Cendón Martín, B., y Hernández Seco, A. (2015). *Las tecnologías IOT dentro de la industria conectada: Internet of things*: EOI Esc. Organiz. Industrial.
- Curay Aldana, R. (2017). *Estudio de las redes de sensores aplicados a la medicina para monitorización de pacientes*. Universidad Nacional de Piura, Piura.
- Díaz Ortiz, J. J., y Romero Suárez, M. A. (2017). *Desarrollo e implementación de un aplicativo web, utilizando la metodología SCRUM, para mejorar el proceso de atención al cliente en la Empresa Z Aditivos S.A*. Universidad Autónoma de Perú, Lima-Perú.
- Díaz Pérez, A. J., y Serra López, L. A. (2015). Competencias específicas en los estudiantes de programación del Instituto Universitario Politécnico Santiago Mariño. *Revista Ethos Venezolana*, 7(1).
- Energy, O. (2019). CT sensors - Interfacing with an Arduino. Retrieved from <https://openenergymonitor.org/forum-archive/node/156.html>
- Fluke Corp. (2010). Manual de uso del multímetro Fluke 374, 375 y 376. Retrieved from <http://us.fluke.com/usen/support/manuals>
- García, J. L., y Toledo, R. V. (2004). Estimadores cuantitativos de la emisión de contaminantes atmosféricos por combustión del petróleo crudo cubano y su efecto económico. *Ingeniería Energética*, 25(2), 51-57.
- González Hernández, W., Estrada Sentí, V., y Martínez Llantada, M. (2004). Contribución al desarrollo de la creatividad en los estudiantes de la Educación Superior a través de la enseñanza de la programación.
- Hernández Cevallos, M. I., y Ledesma Marcalia, D. A. (2010). DESARROLLO DE UN SISTEMA SCADA PARA LA MEDICIÓN DE VOLTAJES CON SISTEMAS EMBEBIDOS PARA EL LABORATORIO DE MECATRÓNICA DE LA FACULTAD DE MECÁNICA.

- Herrera, J., Barrios, M., y Pérez, S. (2014). Diseño e implementación de un sistema scada inalámbrico mediante la tecnología zigbee y arduino. *Prospect*, Vol. 12, págs. 65-72.
- Herrero Herranz, J. C., y Sánchez Allende, J. (2015). Una mirada al mundo de Arduino. *Revista de Ciencia, Tecnología y Medio Ambiente.*, VOLUMEN XIII.
- Kovacs, J. (2007). C# .Net History Lesson.
- Lage Dávila, A. (2007). Conectando la ciencia a la economía: las palancas del socialismo. *Revista Cuba Socialista*(45), 2-26.
- Lara Cortez, P. (2012). Acondicionamiento de señales. Introducción a la Mecatronica. In.
- Lara Sobrino, J. d. J. (2019). *Aplicación web para la gestión de solicitudes de almacén en la Dirección Provincial de Bufetes Colectivo Matanzas*. (Tesis de Grado.). Universidad de Matanzas, Matanzas.
- Liao, C.-K., Chi, P.-L., Chang, C.-Y. J. I. T. o. M. T., y Techniques. (2007). Microstrip realization of generalized Chebyshev filters with box-like coupling schemes. 55(1), 147-153.
- López Suárez, R. d. J. (2019). *Diseño de una interfaz para el acondicionamiento de señales eléctricas: Importancia científica, tecnológica y social*. CD de Monografías 2019. Monografía. Universidad de Matanzas. .
- Manqueros Avilés, V., González Vargas, L., Cordero Escamilla, A., Astorga, G., y Torres Pérez, J. (2019). Implementación de un Sistema Remoto de Monitoreo y Registro de la Demanda y Consumo Eléctrico en Tiempo Real para un Sistema de Baja Tensión.
- Martín Cantalejo, S. (2015). *Plataforma de control domótico escalable y gestionable mediante dispositivos móviles*.
- Martin del Rio, B., y Sanz Molina, A. (2001). *Redes Neuronales y Sistemas Difusos*.
- Martin, S. F. (2018). *Adquitectura de software SIEM para el monitoreo en línea del consumo de energía eléctrica en un entorno residencial utilizando un algoritmo inteligente*. Universidad Autónoma Querétaro,
- Martínez Contreras, A., Varela, D. T., y Peñaloza, F. G. (2019). Diseño de una red neuronal distribuida entre dispositivos Raspberry Pi conectados a Internet. *Research in Computing Science*(ISSN 1870-4069), 148(147).
- Microsoft Corp. (2020). Microsoft Visual Studio Licensing.
- MIKROELECTRON. (2020). HWCT004 Current Transformer 50A/50MA. Retrieved from <https://www.aliexpress.com/item/1900336188.html>

- Montoro, J. M. (2016). *Instrumento virtual basado en arduino, para la medida y caracterización de circuitos lineales*. (Trabajo Fin de Grado). Escuela Politécnica Superior de Linares, Retrieved from <http://tauja.ujaen.es/handle/10953.1/3486>
- Morán Álvarez, A. (2012). *Análisis y predicción de perfiles de consumo energético en edificios públicos mediante técnicas de minería de datos*. Universidad de Oviedo, Gijón.
- Narváez Serrano, Á. F. (2007). *Clasificación de imágenes multiespectrales de la región sur del Ecuador mediante Redes Neuronales*. Universidad Técnica Particular de Loja, Loja-Ecuador.
- Neuta Chiguasuque, E. A. (2013). *Tarjeta de adquisición de datos por ethernet basada en microcontrolador*. (PROGRAMA TECNOLOGÍA EN ELECTRÓNICA). Corporación Universitaria Minuto de Dios., Bogotá D.C.
- NI Corporation. (2020). Measurement Studio. Retrieved from <https://www.ni.com/en-us/shop/software/products/measurement-studio.html>
- Novillo Vicuña, J., Hernandez Rojas, D. L., Mazón Olivo, B., y Correa Elizaldes, K. D. (2018). Monitoreo inalámbrico de señales eléctricas de voltaje 110/220V a través de Arduino. *Alternativas*, 19(1), 55-62.
- Nunes Silva, I., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L. H. B., y Reis Alves, S. F. d. (2017). *Artificial Neural Network Architectures and Training Processes*. Switzerland.
- Núñez Barreto, Á. A. (2020). El costo de un kilowatt hora. Retrieved from <https://www.unionelectrica.cu/el-coste-de-un-kilowatt-hora/>
- Núñez Jover, J., y Montalvo Arriete, L. F. (2015). La política de ciencia, tecnología e innovación en Cuba y el papel de las universidades. *Revista Cubana de Educación Superior*, 34(1), 29-43.
- Peña Daniel, L. A. (2019). *Sistema para la gestión del plan de capacitación de la empresa perforación y reparación capital de pozos de petróleo y gas*. Universidad de Matanzas, Facultad de Ciencias Técnicas.
- Pérez Barreto, J. (2019). Diseño de una red basada en el bus CAN para la adquisición de datos.
- Pérez, E. J. (2017). Implementación de Filtros Pasa Baja: Bessel y Chebyshev.
- Pessanha Santos, N. (2008). Introdução ao Arduino. *Revista Programar*, 17, 39-44.
- Pressman, R. S. (2010). *Ingeniería del software. UN ENFOQUE PRÁCTICO* (7ma ed.): McGraw-Hill.

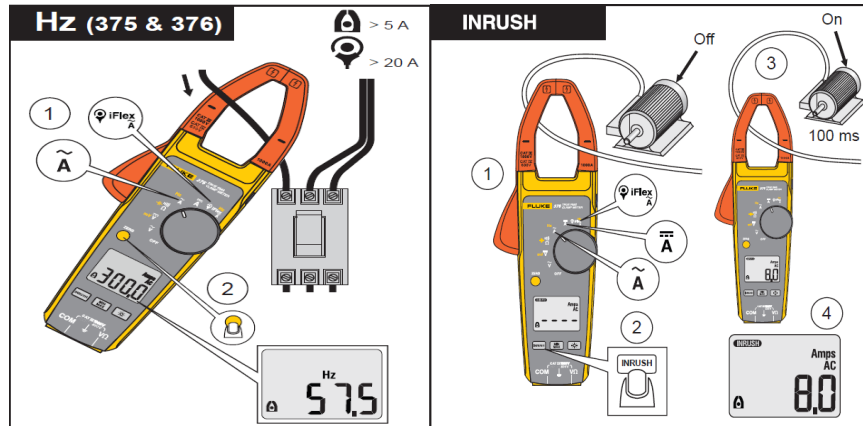
- Rastogi, N., Mehra, R., y Engineering, C. (2013). Analysis of butterworth and chebyshev filters for ecg denoising using wavelets. *IOSR Journal of Electronics*, 6(6), 37-44.
- Rey Guevara, C. F. (2017). *Estudio de la efectividad de la aplicación de la metodología ágil de desarrollo Scrum*. Universidad Tecnológica Israel.,
- Rivas Cárdenas, F. E. (2017). diseño y construcción de una aplicación móvil para la medición y monitoreo de señales de corriente eléctrica utilizando arduino.
- Rojas, C., y Humberto, J. (2017). La Cuarta Revolución Industrial o Industria 4.0 y su Impacto en la Educación Superior en Ingeniería en Latinoamérica y el Caribe. *Universidad Antonio Nariño, Colombia*.
- Ruiz Ariza, J. D. (2019). *Sistema de Control y Monitoreo de Consumo Energético para Equipos de Climatización Orientado a Internet de las Cosas (IoT)*. UNIVERSIDAD DE LA COSTA, Colombia.
- Ruiz Gutiérrez, J. M. (2007). Manual de Programación Arduino.
- Salazar Serna, C. A., y Correa Ortiz, L. C. (2011). Buses de campo y protocolos en redes industriales. *Ventana Informática*, pp 83-109.
- Sancho Caparrini, F. (2014). Mapas Auto-Organizados. Retrieved from <http://www.cs.us.es/~fsancho/?e=76>
- Scull Echeverría, L. (2018). *Sistema Informático para la gestión de los procesos de Órdenes de Trabajos en la Asociación Económica Internacional Aguas Varadero*. Universidad de Matanzas., Matanzas.
- SPA, E. T. B. (2015). IMPACTO DE LA AGENDA DE ENERGÍA EN LA MITIGACIÓN DEL CAMBIO CLIMÁTICO. *INFORME FINAL-14/05/2015 ENERGY TO BUSINESS SPA RIVER CONSULTORES*.
- Suárez Rodríguez, J. A., y Beatón Soler, P. A. (2007). Estado y perspectivas de las energías renovables en cuba. *Tecnología Química*, 27(3), 75-82.
- Suárez Rodríguez, J. A., Beaton Soler, P. A., y Faxas Escalona, R. (2011). Estado y perspectivas de la energía fósil en Cuba. *Tecnología Química*, 31(2), 88-94.
- Tapia, V. (2017). Industria 4.0 – Internet de las Cosas. *UTCiencia" Ciencia y Tecnología al servicio del pueblo"*, 1(1), 51-60.
- Trujillano, J., March, J., y Sorribas, A. (2004). Aproximación metodológica al uso de redes neuronales artificiales para la predicción de resultados en medicina. *Técnicas Emergentes*, 59-67.

- Val Román, J. L. d. (2016). *Industria 4.0: la transformación digital de la industria*. Paper presented at the Valencia: Conferencia de Directores y Decanos de Ingeniería Informática, Informes CODDII.
- Valido, L. A. (2019). *Visualizador de la Interfaz Hombre-Máquina del Sistema ROFLEXIN/LC para Dispositivos Móviles*. (MSc.). Universidad de Matanzas,
- Vásquez, J. W. (2008). Evolución y nuevas tendencias de los sistemas de control – PAC (Controladores analógicos programables). *Inventum*, 5, 80-86.
- Vega, A. M., Santamaría, F., y Rivas, E. (2014). Internet de los objetos empleando arduino para la gestión eléctrica domiciliaria. *Revista Escuela de Administración de Negocios*(77), 24-41.
- Villalva Castañeda, L. A. (2017). *Aplicación de Scrum en el desarrollo de software en TeamSoft S.A.C*. Universidad César Vallejo., Lima.
- Wikipedia. (2019). Filtro de Chebyshev. In *Wikipedia*.

ANEXO I

Multímetros Fluke. Especificaciones eléctricas

Corriente CA



Rango

- Modelos 374 y 375600,0 A
- Modelo 376.....999,9 A

Resolución0,1 A

Precisión

- 2 % ± 5 dígitos (10 – 100 Hz)
- 2,5 % ± 5 dígitos (100-500 Hz)

Factor de cresta (59/60 Hz)

- 3 a 500 A (únicamente modelos 375 y 376)
- 2,5 @ 600 A
- 1,42 @ 1.000 A
- Agregar 2 % para F. C. > 2