



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

**DESARROLLO DE UN SIMULADOR DE EFICIENCIA
ENERGÉTICA PARA LA EVALUACIÓN DE
PROYECTOS DE SUSTITUCIÓN DE MOTORES DE
INDUCCIÓN DE EFICIENCIA ESTÁNDAR POR
MOTORES DE INDUCCIÓN DE ALTA EFICIENCIA**

**TESIS PROFESIONAL QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

PRESENTA:

HUGO HERRERA GUTIÉRREZ

DIRECTOR DE TESIS: DRA. MARÍA DEL CARMEN TORRES SALAZAR

CUERNAVACA, MORELOS

DICIEMBRE, 2020

RESUMEN

En el presente proyecto de tesis, se expone el desarrollo de un simulador de eficiencia energética de motores de inducción, el cual se compone de dos elementos principales: un Hardware y un Software.

El desarrollo del Hardware tiene por finalidad la obtención de parámetros eléctricos del motor de eficiencia estándar a evaluar, con el objeto de enviarlos al Software mediante una comunicación serial. El Software tiene la función de recibir esta información y procesarla para calcular el factor de carga, la eficiencia a la cual opera el motor evaluado y su consumo, así como calcular el consumo de un motor propuesto de alta eficiencia, con el objeto de estimar el ahorro energético que se espera obtener por la sustitución del motor evaluado por el motor propuesto.

Para el desarrollo del Hardware se utilizó como elemento principal una placa de Arduino DUE, mientras que para el desarrollo del Software se utilizó el entorno Visual Studio 2019, basado en el lenguaje de programación C#.

El desempeño del Simulador se probó en un motor de 15 HP de una máquina troqueladora de una empresa ubicada en la ciudad de Cuernavaca, Morelos. Los resultados del simulador mostraron que el motor actual trabaja a un factor de carga del 31% y a una eficiencia del 84%, proponiendo el simulador un motor de alta eficiencia de reemplazo de 7.5 HP, estimándose con esta acción un ahorro energético de 1,743 kWh al año. Con la finalidad de comprobar la incertidumbre de la medición del Hardware del simulador, se utilizó en la prueba mencionada el analizador de redes Dranetz Power Visa, obteniéndose los siguientes errores relativos máximos: 1.7% en la medición de tensión, 12.8% en la medición de corriente y 6.2% en la medición del factor de potencia.



ABSTRACT

In this thesis project, the development of an induction motor energy efficiency simulator is exposed, which is made up of two main elements: Hardware and Software.

The purpose of the hardware development is to obtain the electrical parameters of the standard efficiency motor to evaluate, in order to send them to the Software through serial communication. The Software has the function of receiving this information and processing it to calculate the load factor, the efficiency at which the evaluated motor operates and its consumption, as well as to calculate the consumption of a proposed high-efficiency motor, in order to estimate the energy savings expected to be obtained by replacing the evaluated motor with the proposed motor.

For the development of the Hardware, an Arduino DUE board was used as the main element, while for the development of the Software the Visual Studio 2019 environment was used, based on the C # programming language.

The performance of the Simulator was tested in a 15 HP motor of a die-cutting machine of a company located in the Cuernavaca City. The simulator results showed that the current motor works at a load factor of 31% and an efficiency of 84%, the simulator proposing a 7.5 HP replacement high-efficiency motor, estimating with this action an energy saving of 1,743 kWh per year. In order to check the measurement uncertainty of the simulator Hardware, the Dranetz Power Visa network analyzer was used in the aforementioned test, obtaining the following maximum relative errors: 1.7% in the voltage measurement, 12.8% in the current measurement and 6.2% in the power factor measurement.



AGRADECIMIENTOS

A Dios por permitirme vivir y alcanzar una meta más.

A mis padres Oscar y María Esperanza por darme la vida y su apoyo incondicional en cada etapa de mi vida.

A mi esposa Erika por su constante motivación y apoyo.

A la Dra. María del Carmen Torres Salazar por su dirección y asesoría para la realización del presente trabajo.

A Dr. Luis Cisneros Villalobos, Dr. J. Guadalupe Velázquez Aguilar, Dr. Mario Acosta Flores, Dr. Mario Limón Mendoza, Dra. Viridiana Aydeé León Hernández, Dr. Roy López Sesenes por la transmisión de sus conocimientos a través de la impartición de las materias de maestría.

A CONACyT por la beca otorgada.

A la FCQel por el apoyo en el uso de sus instalaciones y el préstamo del Analizador de Redes Dranetz Power Visa.



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Maestría en Ingeniería Eléctrica y Electrónica



Facultad de Ciencias
Químicas e Ingeniería

DEDICATORIA

A mis hijos Valeria, Fátima y Hugo Emmanuel, por ser mi bella fuente de inspiración.



ÍNDICE

RESUMEN	2
ABSTRACT	3
AGRADECIMIENTOS	4
DEDICATORIA	5
ÍNDICE	6
LISTA DE TABLAS	7
LISTA DE FIGURAS	8
CAPITULO 1. INTRODUCCIÓN.....	11
CAPITULO 2. MARCO TEÓRICO.....	18
CAPITULO 3. DESARROLLO	25
CAPITULO 4. APLICACIÓN Y CASO DE ESTUDIO.....	65
CAPITULO 5. CONCLUSIONES Y RECOMENDACIONES	74
REFERENCIAS	76
APÉNDICE: TABLAS DE EFICIENCIA DE MOTORES ESTÁNDAR Y DE ALTA EFICIENCIA.....	79



LISTA DE TABLAS

Tabla 3.1. Parámetros eléctricos del sensor SCT013-030.

Tabla 3.2. Cálculo del divisor de tensión en Excel.

Tabla 4.1. Comparativo de mediciones y cálculo de errores absolutos y relativos.

LISTA DE FIGURAS

- Figura 1.1. Consumo mundial de energía a través del tiempo.
- Figura 1.2. Suministro mundial de energía por fuente primaria.
- Figura 2.1. Relación entre la potencia eléctrica de entrada, la potencia mecánica de salida y las pérdidas de un motor eléctrico.
- Figura 2.2. Elementos de diseño de un motor de alta eficiencia.
- Figura 2.3. Curva característica de eficiencia (eje vertical) vs factor de carga (eje horizontal).
- Figura 2.4. Efecto de la variación de tensión en los parámetros eléctricos para motores de inducción.
- Figura 2.5. Efecto del porcentaje de desbalance de tensión en la eficiencia.
- Figura 2.6. Efecto del rebobinado del motor en su eficiencia.
- Figura 3.1. Bloques del simulador de eficiencia energética.
- Figura 3.2. Bloques que conforman el Hardware del simulador.
- Figura 3.3. Puertos de la placa Arduino DUE utilizados.
- Figura 3.4. Sensores de corriente SCT013-030.
- Figura 3.5. Diagrama de conexión del sensor SCT013-030.
- Figura 3.6. Curva de respuesta característica del sensor SCT013-030, tomada de su ficha técnica.
- Figura 3.7. Sensores de tensión.
- Figura 3.8. Arreglo propuesto con LM158AN, para el acondicionamiento final de la señal de corriente (Multisim).
- Figura 3.9. Oscilogramas del acondicionamiento de la señal de corriente con LM158AN (Multisim).
- Figura 3.10. Divisor de tensión propuesto (Multisim).
- Figura 3.11. Oscilogramas del divisor de tensión (Multisim).
- Figura 3.12. Circuito acondicionador de señales propuesto para factor de potencia (Multisim).

Figura 3.13. Oscilogramas del circuito acondicionador de señales para factor de potencia (Multisim).

Figura 3.14. Principio de operación para el cálculo del factor de potencia.

Figura 15. Circuito acondicionador de señal propuesto para cálculo de factor de potencia (Multisim).

Figura 16. Oscilogramas del circuito acondicionador de señal para factor de potencia.

Figura 3.17. Hardware del simulador de eficiencia energética de motores de inducción.

Figura 3.18. Máquina de estados finitos como método de programación.

Figura 3.19. Configuración de librerías y pines para los LCD's utilizados.

Figura 3.20. Declaración de variables.

Figura 3.21. Función void setup().

Figura 3.22. Función void loop().

Figura 3.23. Función corrienteA().

Figura 3.24. Función tensionA().

Figura 3.25. Función FP1().

Figura 3.26. Códigos para el envío de información al puerto serial y LCD's.

Figura 3.27. Ventana de bienvenida del Software.

Figura 3.28. Ventana del entorno de Simulación Energética.

Figura 3.29. Sección de datos del motor de eficiencia estándar evaluado.

Figura 3.30. Sección de datos del motor de alta eficiencia propuesto.

Figura 3.31. Sección de botones.

Figura 3.32. Sección de gráficos.

Figura 3.33. Sección de Instrumentos.

Figura 3.34. Sección de parámetros.

Figura 3.35. Código fuente de la ventana principal.

Figura 3.36. Librerías del código fuente de la ventana de simulación energética.

Figura 3.37. Código fuente de la ventana de simulación energética, parte 1.

Figura 3.38. Código fuente de la ventana de simulación energética, parte 2.



- Figura 3.39. Código fuente de la ventana de simulación energética, parte 3.
- Figura 3.40. Código fuente de la ventana de simulación energética, parte 4.
- Figura 3.41. Código fuente de la ventana de simulación energética, parte 5.
- Figura 3.42. Código fuente de la ventana de simulación energética, parte 6.
- Figura 3.43. Código fuente de la ventana de simulación energética, parte 7.
- Figura 3.44. Código fuente de la ventana de simulación energética, parte 8.
- Figura 3.45. Código fuente de la ventana de simulación energética, parte 9.
- Figura 3.46. Código fuente de la ventana de simulación energética, parte 10.
- Figura 3.47. Código fuente de la ventana de simulación energética, parte 11.
- Figura 3.48. Código fuente de la ventana de simulación energética, parte 12.
- Figura 3.49. Código fuente de la ventana de simulación energética, parte 13.
- Figura 4.1. Conexión de sensores en tableros eléctricos energizados.
- Figura 4.2. Parametrización de datos de entrada en el Software.
- Figura 4.3. Gráficos y paneles de instrumentos del Software.
- Figura 4.4. Resultados arrojados por el Software (valores en azul).
- Figura 4.5. Datos almacenados en la Tabla del Software.
- Figura 4.6. Ejecutando la prueba de desempeño del simulador en campo.
- Figura 4.7. Lecturas de tensiones y corrientes del Analizador de Redes.
- Figura 4.8. Lecturas de potencias y factor de potencia del Analizador de Redes.

CAPITULO 1. INTRODUCCIÓN

En el presente trabajo se desarrolla un simulador como una herramienta para la evaluación técnica-económica de proyectos [1] de sustitución de motores de eficiencia estándar por motores de alta eficiencia o eficiencia Premium, utilizando un método no invasivo que permite llevar a cabo la simulación sin la necesidad de interrumpir el proceso productivo del motor o grupo de motores que se deseen evaluar, desde luego, tomando todas las medidas de seguridad eléctrica necesarias [2].

El simulador consiste en un desarrollo tanto de Hardware como de Software [3]; se utiliza una placa de Arduino DUE [4] como Hardware principal para la adquisición de parámetros eléctricos de los motores a evaluar, llevando a cabo dentro de su código un procesamiento de la información inicialmente obtenida por los sensores de tensión y corriente, como el cálculo de valores RMS a partir de los valores instantáneos, así como el cálculo del factor de potencia. Estos datos calculados dentro del código de Arduino son llevados por medio de una comunicación serial en tiempo real a una plataforma desarrollada en Visual Studio (Software), utilizando el lenguaje de programación orientado a objetos y procedimientos C# [5], en donde se realiza la simulación energética a partir de una segunda serie de cálculos como el factor de carga y la eficiencia a la cual se encuentra trabajando el motor evaluado.

El simulador, arroja como resultado una tasa de ahorro energético que correspondería al sustituirse el motor evaluado por un motor de alta eficiencia propuesto por éste.

El Software basa su funcionamiento en el procedimiento mostrado en el curso-taller Promotores de Ahorro y Uso Eficiente de Energía Eléctrica [6] organizado por el Fideicomiso para el Ahorro de Energía Eléctrica (FIDE) en conjunto con la Comisión Nacional de Energía Eléctrica (CNEE) de Guatemala, para la evaluación energética de motores eléctricos.



1.1. Revisión histórica y actual de la eficiencia energética

El enfoque del presente proyecto se relaciona directamente con el término “Eficiencia Energética”, mismo que en la actualidad tiene importancia a nivel mundial, ya que mediante dicho concepto se basan muchas de las estrategias para lograr hacer un uso racional de la energía, con lo que se busca en consecuencia reducir globalmente las emisiones de gases de efecto invernadero hacia la atmósfera, manteniendo el nivel de vida actual.

De acuerdo con la Agencia Internacional de Energía (IEA por sus siglas en inglés), en todo el mundo se espera un incremento en la demanda de energía [7] (ver Figura 1.1), esto como resultado del incremento de la población mundial proyectado. Según estudios del mismo organismo, más del 80% de la producción de energía provienen de fuentes de energía primaria no renovables [7] (ver Figura 1.2), mismas que, en su proceso de conversión a energía útil emiten gases de efecto invernadero.

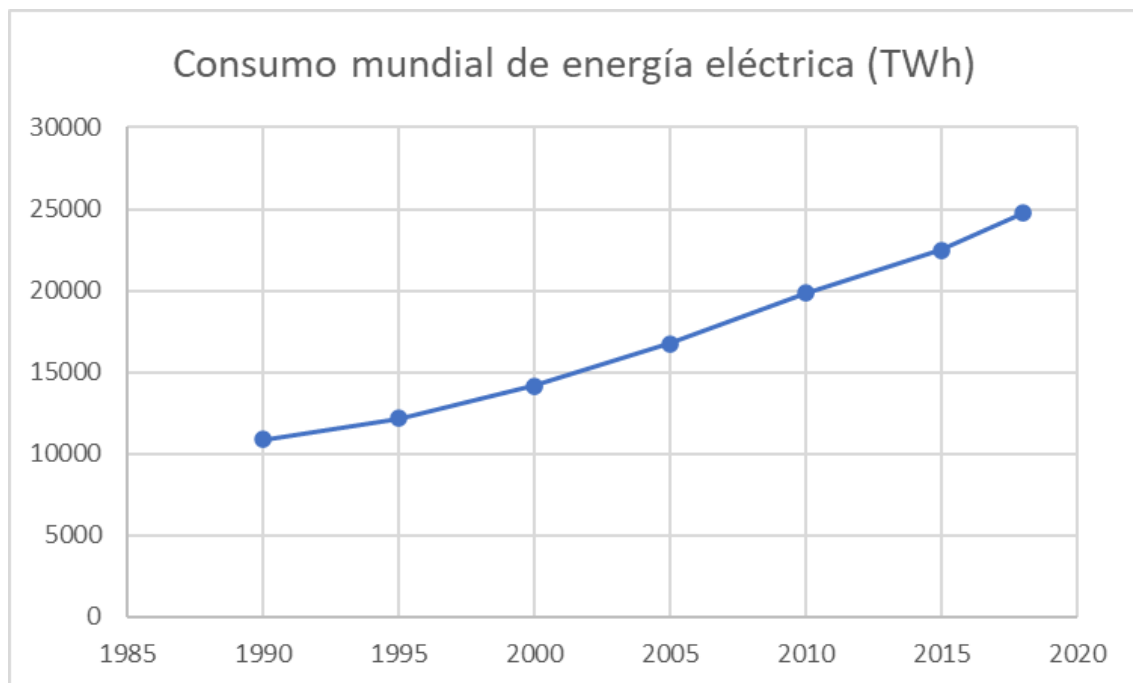


Figura 1.1. Consumo mundial de energía a través del tiempo. Fuente: [8]

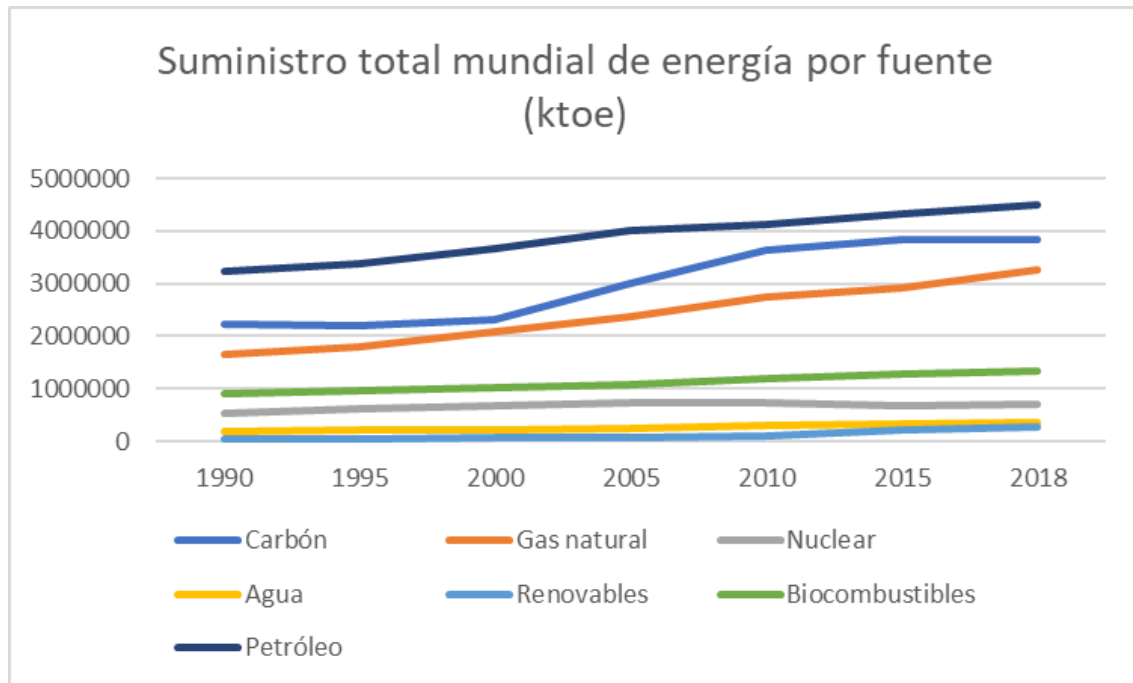


Figura 1.2. Suministro mundial de energía por fuente primaria. Fuente: [9]

En la gráfica de la Figura 1.2, se ve claramente que las fuentes primarias de energía “verdes” tienen una utilización mínima respecto a las provenientes de recursos fósiles, durante todo el periodo de tiempo graficado. En consecuencia, todo proyecto de eficiencia energética tiene impacto no solo en términos de competitividad y normatividad, sino también en cuidado medioambiental y sustentabilidad, ya que un proceso eficiente significa hacer lo mismo con menor energía.

Por lo anteriormente comentado, en la actualidad existe una creciente preocupación en las empresas, por establecer medidas que permitan lograr una mayor eficiencia en los procesos productivos, a fin de mitigar el incremento en el costo de los recursos energéticos y que estas medidas permitan incrementar su competitividad al reducir el gasto por dicho concepto.

1.2. Antecedentes

Uno de los problemas que hoy en día preocupan en demasía, es el incremento de la contaminación, mismo que ha ido de la mano con el desarrollo tecnológico, cuando debiera significar lo contrario. Lo anterior se debe principalmente a que se



ha antepuesto el desarrollo tecnológico a la conservación del medio ambiente; sin embargo, a medida que ha incrementado este problema se ha puesto mayor atención a la búsqueda de soluciones que lo mitiguen o lo reduzcan. Una de estas soluciones es el incremento de la eficiencia energética en los procesos, de tal manera que se obtengan los mismos resultados productivos con la menor cantidad posible de recursos energéticos. Los resultados de esta solución han sido tan significativamente positivos que en la actualidad existen normas y organismos nacionales e internacionales que regulan el uso eficiente de la energía, así como entidades que vigilan el cumplimiento de las mismas. Particularmente en México, es la Comisión Nacional para el Uso Eficiente de la Energía (CONUEE) la encargada de emitir y vigilar las regulaciones en materia del uso de energía, siendo las normas NOM-014-ENER-2004 [10] y NOM-016-ENER-2016 [11] las que regulan los límites de eficiencia energética en motores eléctricos de inducción.

Derivado de lo anterior, se hace necesario el desarrollo de herramientas de apoyo para la evaluación energética de motores eléctricos, con la finalidad de justificar la sustitución cuando no se haga un uso eficiente de la energía de acuerdo con las regulaciones establecidas.

1.3. Descripción del problema

A pesar de la existencia de regulaciones de carácter obligatorio en materia de eficiencia energética como las citadas en el punto anterior, en la mayoría de los casos, las empresas no encuentran la justificación económica para sustituir un motor obsoleto (con baja eficiencia) por un motor de alta eficiencia debido a su mayor costo, sin darse cuenta que en un gran porcentaje de los casos pueden obtenerse periodos de recuperación de la inversión rentables; el problema está en que no cuentan con las técnicas, procedimientos ni herramientas para cuantificar el ahorro energético y económico que se logra obtener con tales proyectos.



1.4. Justificación

El presente proyecto puede ser una herramienta fundamental para la justificación técnica-económica de proyectos de sustitución de motores obsoletos por motores de alta eficiencia, o bien, para la vigilancia del cumplimiento normativo en materia de eficiencia energética de motores de inducción.

Su bajo costo de producción, su método no invasivo para usarlo aún sin interrumpir el proceso productivo y su gran versatilidad al ser programable para aplicaciones específicas, sumado al impacto social y medioambiental por relacionarse con el tema global de eficiencia energética lo convierten en un proyecto de gran interés.

1.5. Objetivos

1.5.1. General

Desarrollar un simulador de eficiencia energética, basado en Hardware y Software, que permita evaluar energéticamente un motor de eficiencia estándar en operación y proyectar un ahorro energético al considerar la sustitución por un motor de alta eficiencia.

1.5.2. Específicos

- Utilizar una placa de Arduino DUE como elemento principal de Hardware, para la medición de los parámetros eléctricos necesarios para la evaluación energética de motores de inducción trifásicos de eficiencia estándar que se encuentren en operación.
- Llevar a cabo el acondicionamiento de las señales eléctricas de tensión y corriente obtenidas por los sensores para que puedan interactuar con la placa de Arduino DUE.
- Crear un código en Arduino para que a partir de las señales instantáneas de tensión y corriente se puedan calcular los valores eficaces (RMS) y el factor de potencia, imprimiendo dichos valores a través del puerto serial cada 10 segundos.



- Desarrollar un Software de Simulación, utilizando la plataforma Visual Studio 2019, versión libre Community, donde se reciban los datos enviados por comunicación serial desde Arduino y se procesen para evaluar el desempeño energético tanto del motor de eficiencia estándar medido como el motor de alta eficiencia proyectado.

1.6. Estado del Arte

En la actualidad es muy común encontrar simuladores para un sinnúmero de aplicaciones y en casi todos los campos, siendo de gran ayuda para hacer proyecciones y tomar mejores decisiones.

En la página oficial de la CONUEE [12] se puede acceder a una caja de herramientas, donde se observan algunos simuladores, como el de desempeño de luminarios en vialidades, o el de eficiencia electromecánica en sistemas de bombeo de agua; sin embargo, no se distingue alguno de eficiencia energética de motores eléctricos.

En lo que respecta a simuladores de eficiencia energética de motores, en la página web del fabricante de motores WEG [13], se puede acceder a un simulador para conocer el consumo y el costo que representaría el uso de un determinado motor, en función de las horas de operación y el costo de energía introducidos en los campos editables, así como el retorno de la inversión de un motor de alta eficiencia para sustituir a un motor de eficiencia estándar, donde se pide como datos de entrada las eficiencias de los motores considerados, así como el costo de los mismos. El inconveniente de este simulador es que la simulación se lleva a cabo solo con los datos de eficiencias introducidos, dejando fuera los factores de carga bajo los cuales operan los motores.

Respecto a los sistemas que miden y registran parámetros eléctricos, en la actualidad hay mucha variedad, desde equipos sencillos que solo miden tensión o corriente, hasta complejos analizadores que registran contenido armónico y eventos transitorios.



Es preciso mencionar que hay sistemas basados en Hardware y Software para el diagnóstico y prueba de motores eléctricos, como los desarrollados por PdMA Corporation [14]; sin embargo, son desarrollos altamente costosos y generalmente para aplicaciones específicas. En contraste, el simulador desarrollado que se expone en el presente trabajo es de bajo costo y es programable, dando la posibilidad de ajustarse a aplicaciones determinadas.

CAPITULO 2. MARCO TEÓRICO

La función de un motor eléctrico es convertir la energía eléctrica en energía mecánica para realizar un trabajo útil. En este proceso de conversión, parte de la energía eléctrica se convierte en pérdidas que no se aprovechan en la flecha del motor como energía mecánica, tal como se puede observar en la Figura 2.1.

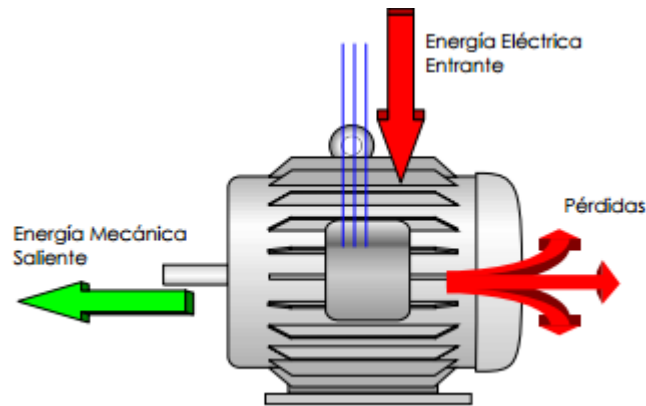


Figura 2.1. Relación entre la potencia eléctrica de entrada, la potencia mecánica de salida y las pérdidas de un motor eléctrico.

De acuerdo con [15], las pérdidas principales en un motor de inducción se generan de acuerdo con su naturaleza como a continuación se describe:

- Pérdidas en los conductores del rotor y el estator. Estas pérdidas son del tipo I^2R y pueden reducirse con el incremento del calibre de los conductores del rotor y el estator, así como la calidad de su fabricación.
- Pérdidas en el núcleo magnético. Estas se presentan en dos componentes, las pérdidas por corriente Eddy y las pérdidas por el fenómeno de histéresis, incluyendo las pérdidas superficiales en la estructura magnética del motor. Dichas pérdidas pueden reducirse usando mejores aceros en la fabricación del motor, así como reduciendo el estrecho del entrehierro.
- Pérdidas mecánicas por fricción y ventilación. Se dan por la fricción de los rodamientos y por la resistencia que ofrece el aire al ventilador y otros

elementos rotativos del motor. Pueden reducirse con rodamientos de menor coeficiente de fricción y mejorando el diseño del ventilador.

De lo anteriormente comentado, se puede deducir que la principal diferencia entre un motor de eficiencia estándar y un motor de alta eficiencia estriba precisamente en que el segundo tiene menos pérdidas que el primero, como resultado del mejoramiento en el diseño de su construcción, tal como se ilustra en la Figura 2.2. Esta situación origina en consecuencia que el motor de alta eficiencia realiza el mismo par pero con menor potencia eléctrica; bajo dicho principio, se basa la estrategia de eficiencia energética de reemplazo de motores de inducción de eficiencia estándar por motores de inducción de alta eficiencia, como una medida de ahorro energético.

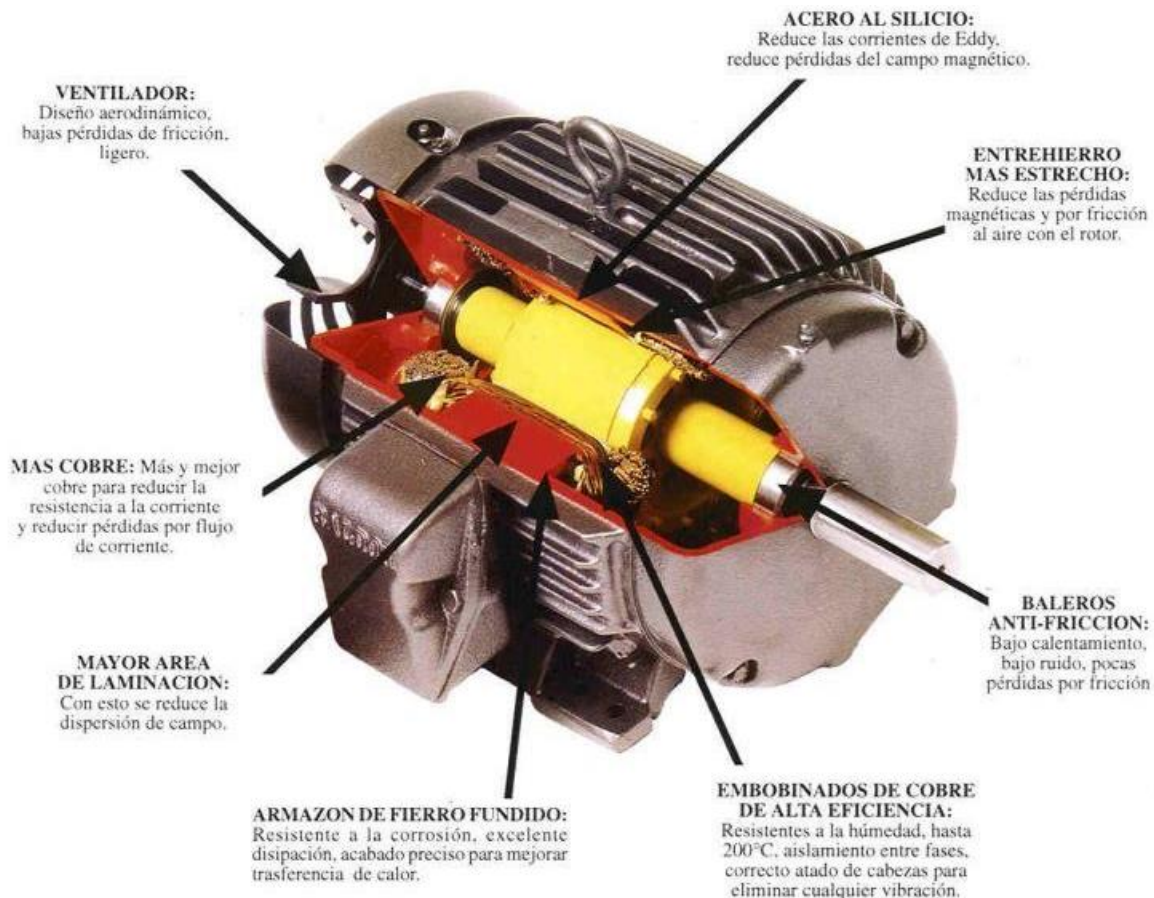


Figura 2.2. Elementos de diseño de un motor de alta eficiencia.



De acuerdo con [16], la eficiencia de un motor eléctrico se define como la razón entre la potencia mecánica de salida y la potencia eléctrica de entrada, pudiéndose determinar también si se conocen las pérdidas y cualquiera de los dos valores, tal como se puede observar en las siguientes ecuaciones:

$$eficiencia = \frac{potencia\ mecánica\ de\ salida}{potencia\ eléctrica\ de\ entrada} \quad \langle 1 \rangle$$

$$eficiencia = \frac{potencia\ eléctrica\ de\ entrada - pérdidas}{potencia\ eléctrica\ de\ entrada} \quad \langle 2 \rangle$$

$$eficiencia = \frac{potencia\ mecánica\ de\ salida}{potencia\ mecánica\ de\ salida + pérdidas} \quad \langle 3 \rangle$$

Existen diversos métodos para la determinación de la eficiencia energética de motores eléctricos de inducción polifásicos, como los citados en [16], sin embargo, no todos ellos tienen una utilidad práctica cuando se trata de evaluar el desempeño energético de un motor en operación. El método que se propone debido a su característica no invasiva es el método de determinación del factor de carga referido a la potencia de placa [6].

MÉTODO DE DETERMINACIÓN DEL FACTOR DE CARGA REFERIDO A LA POTENCIA DE PLACA

Este método toma en cuenta el factor de carga al cual opera el motor y su curva característica de eficiencia obtenida del fabricante. La eficiencia nominal mostrada en la placa de datos de los motores se refiere a la eficiencia a plena carga y ésta varía en función de la carga demandada al motor.

En síntesis, el método consiste en determinar la energía eléctrica que demanda y la eficiencia al factor de carga al cual opera el motor a evaluar y con este dato determinar la potencia mecánica demandada en la flecha. Conociendo la potencia

mecánica que demanda la carga a la flecha del motor evaluado, se determina la energía eléctrica y la eficiencia a la cual operará el motor de alta eficiencia propuesto.

Factor de Carga y eficiencia

El Factor de Carga de un motor es un índice de la potencia que desarrolla en operación respecto a la que puede desarrollar nominalmente, determinándose con la siguiente ecuación:

$$FC = \frac{\text{Potencia demandada en la flecha}}{\text{Potencia de placa}} \quad (4)$$

Donde FC es el Factor de Carga.

Como se indica en [17], los motores eléctricos de eficiencia estándar trabajan en su máxima eficiencia a un factor de carga de 0.75, mientras que los motores de alta eficiencia lo hacen a 0.85%. En la Figura 2.3 se muestra la curva característica de eficiencia vs factor de carga de un motor WEG de 15 HP de alta eficiencia obtenida de los datos proporcionados por el fabricante, donde en este caso la máxima eficiencia se obtiene al 90% de cargabilidad.

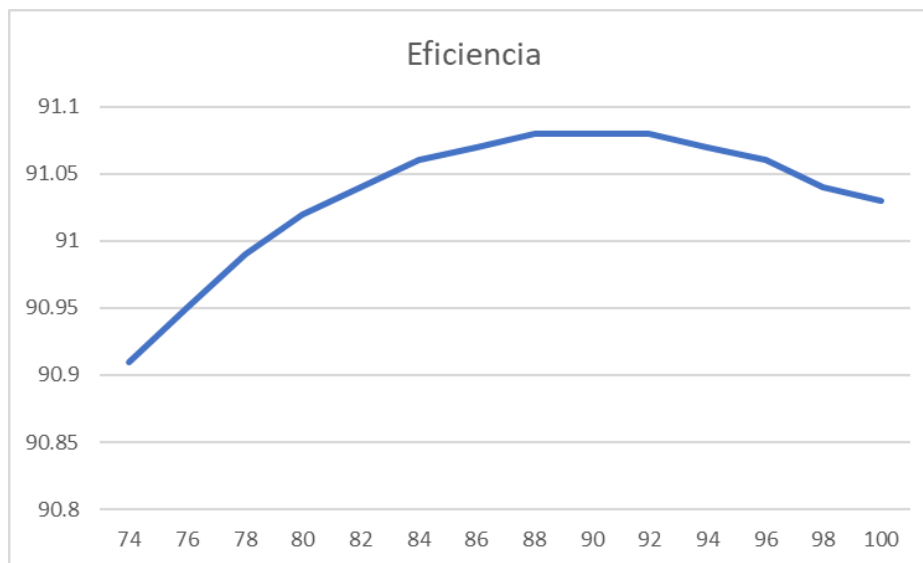


Figura 2.3. Curva característica de eficiencia (eje vertical) vs factor de carga (eje horizontal). Fuente: [18].

Cuando no se tiene la información de la eficiencia a un determinado factor de carga, se pueden hacer uso de las técnicas de interpolación lineal si se conocen los valores inmediatos superiores e inferiores correspondientes, para tener mejor precisión.

Efecto de la variación de voltaje en la eficiencia

De acuerdo con [6], cuando un motor opera a una tensión diferente a la marcada en su placa de datos, se debe hacer un ajuste a su eficiencia de acuerdo a lo mostrado en la Figura 2.4.

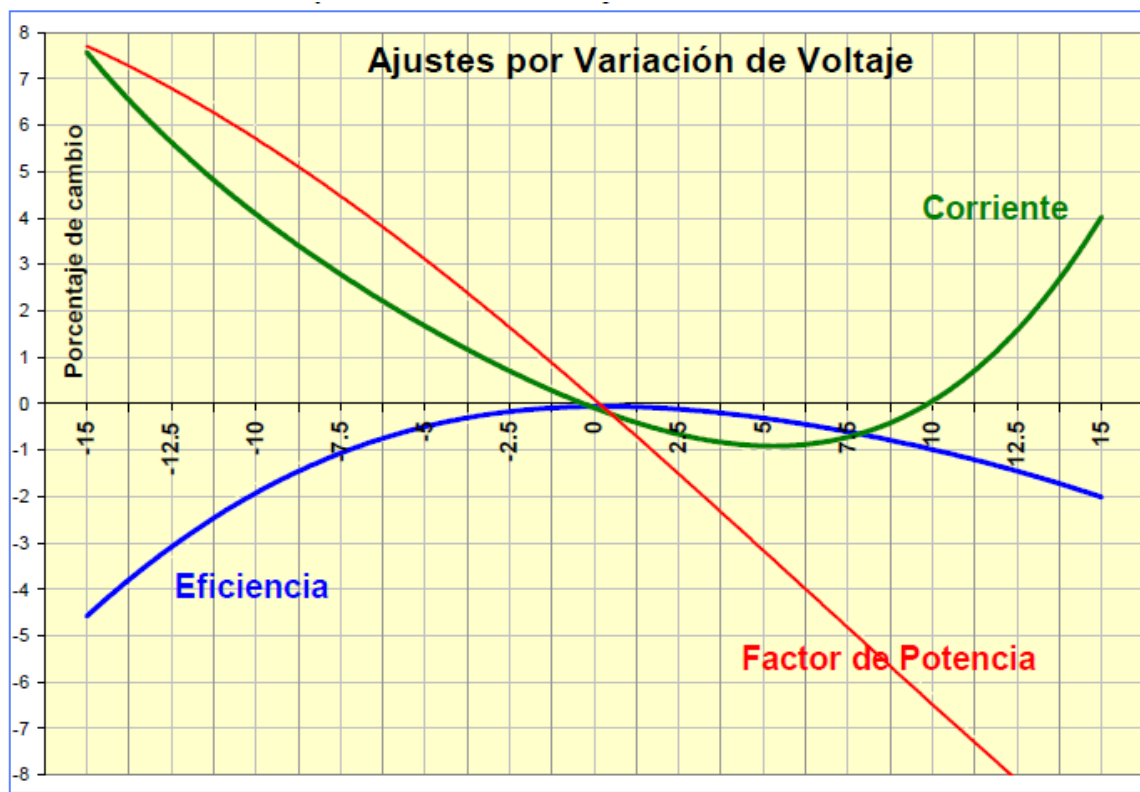


Figura 2.4. Efecto de la variación de tensión en los parámetros eléctricos para motores de inducción. Fuente [6].

La variación porcentual ($\%Vt$) de la tensión de operación respecto a la tensión de placa se obtiene con la siguiente ecuación:

$$\%Vt = \left[\frac{\text{tensión de operación}}{\text{tensión de placa}} \right] \times 100 \quad (5)$$

Efecto del desbalance de tensión en la eficiencia

De acuerdo con [19], en las máquinas asíncronas el desbalance de tensión produce efectos negativos, tales como el aumento del consumo de la potencia activa, el aumento de las pérdidas y, por ende, la disminución de la eficiencia. En la Figura 2.5 se presenta una gráfica que se utiliza en [17], para la determinación del ajuste de la eficiencia de motores de inducción en función del porcentaje de desbalance de tensión.

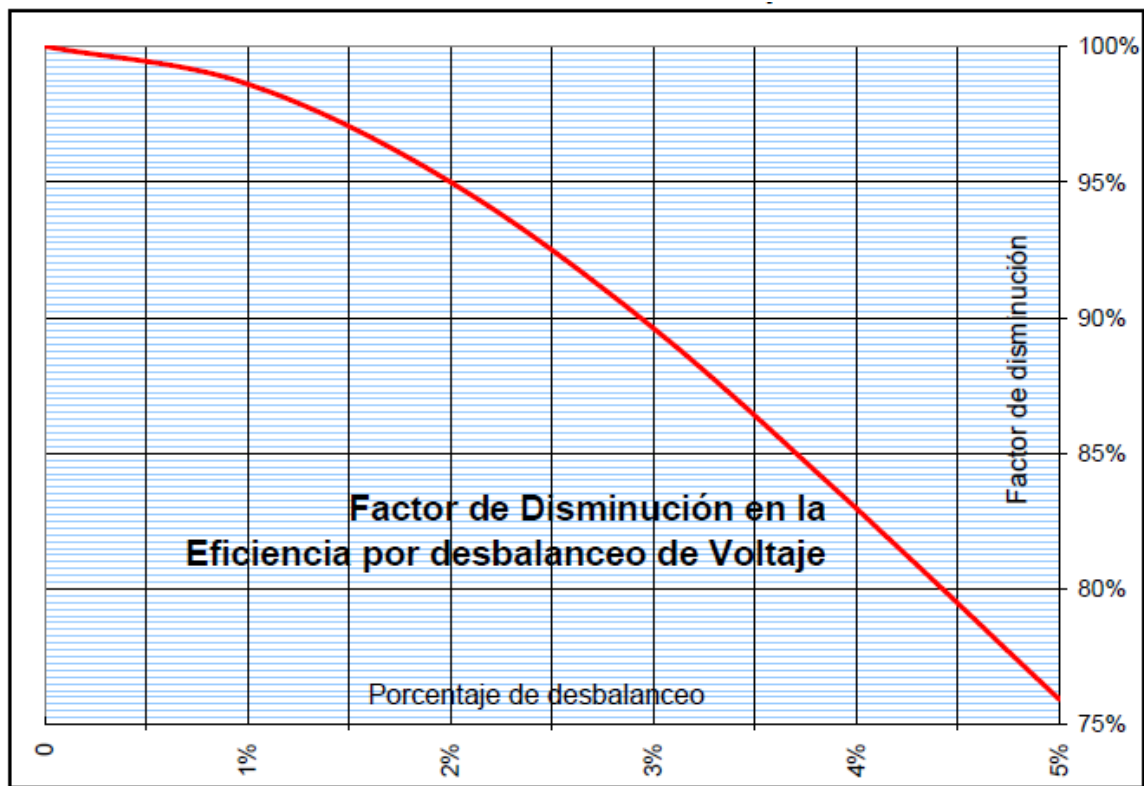


Figura 2.5. Efecto del porcentaje de desbalance de tensión en la eficiencia.

Fuente: [6].

El porcentaje de desbalance de tensión %Dt se determina con la siguiente ecuación:

$$\%Dt = \left[\frac{\text{Máxima diferencia al promedio}}{\text{tensión promedio}} \right] \times 100 \quad (6)$$

Efecto del rebobinado del motor en la eficiencia

De acuerdo con [20] la mayoría de los motores pequeños (hasta 30 HP) reducen su eficiencia de 1 a 5 % al ser rebobinados o reparados, o inclusive mas cuando ésta práctica se ha hecho mas de una vez o se ha hecho sin controles específicos.

En la Figura 2.6 se muestra un gráfico de un estudio realizado [20], mostrando el efecto en la eficiencia que provoca el rebobinado llevado a cabo sin controles específicos (Grupo A) vs el rebobinado llevado a cabo con controles específicos (Grupo B). Claramente se observa que cuando el rebobinado se lleva a cabo sin controles específicos la reducción en su eficiencia se da en mayor porcentaje.

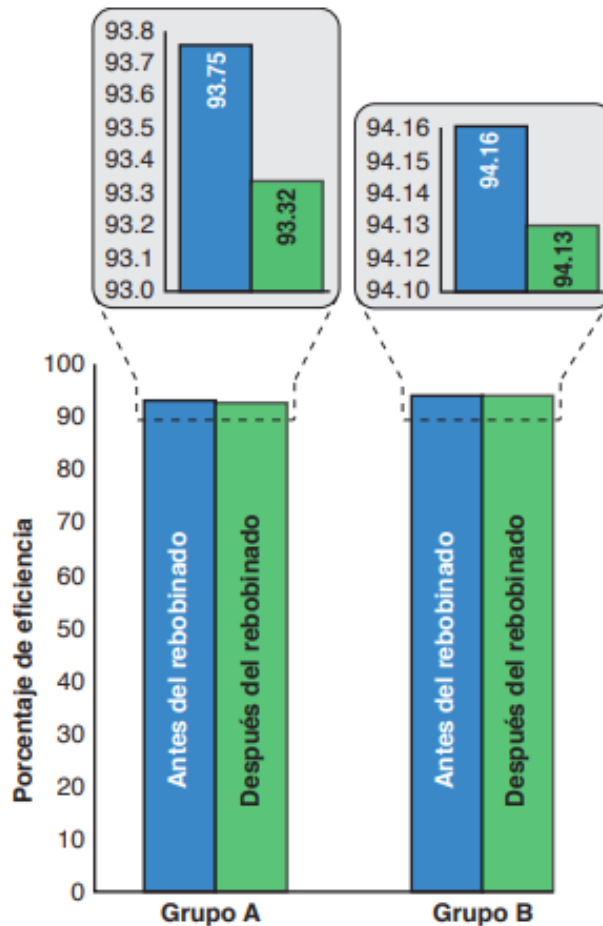


Figura 2.6. Efecto del rebobinado del motor en su eficiencia. Fuente: [20].

CAPITULO 3. DESARROLLO

El presente trabajo consistió en el desarrollo de dos bloques principalmente, uno denominado Hardware y otro denominado Software, siendo ambos bloques complementarios uno del otro y necesarios para llevar a cabo la simulación del ahorro de energía que se estima obtener al sustituir un motor o grupo de motores de eficiencia estándar por motores de eficiencia Premium. El Hardware se encarga de obtener la información necesaria del motor a evaluar, específicamente los parámetros eléctricos del motor en operación, y por medio de una comunicación serial en tiempo real, llevar esta información al Software, donde se procesan los cálculos matemáticos necesarios para obtener como resultado el ahorro de energía estimado al comparar el rendimiento del motor evaluado con el rendimiento estimado de un motor de alta eficiencia (o eficiencia Premium como también se le conoce).

En la Figura 3.1 se muestra el diagrama a bloques del Simulador completo.

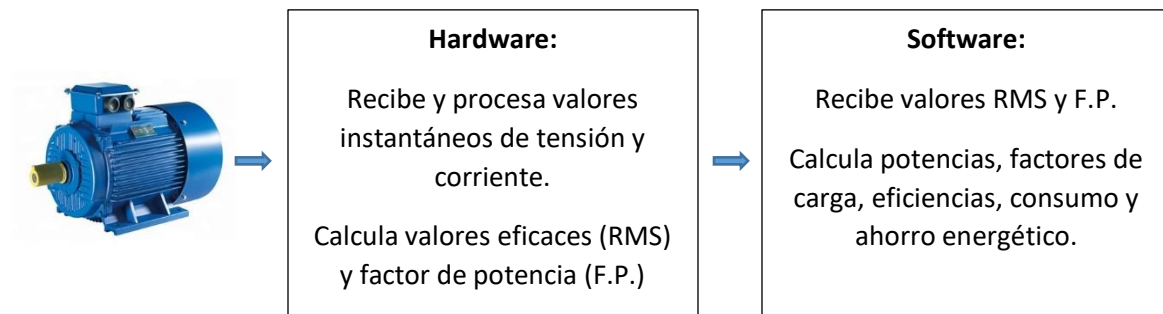


Figura 3.1. Bloques del simulador de eficiencia energética.

3.1. Desarrollo de Hardware

Por definición, Hardware, se refiere a la parte física o tangible de un sistema [3]. Por lo tanto, el desarrollo del Hardware consistió en la especificación, dimensionamiento, selección, adquisición y armado de todos los elementos físicos del simulador. El elemento principal del Hardware es una placa de Arduino DUE, misma que a su vez es la interfaz entre el Hardware y el Software.

La interfaz entre el motor a evaluar y el Hardware son los sensores de tensión y de corriente, los cuales se conectan a las entradas analógicas disponibles en la placa de Arduino, previamente acondicionadas adecuadamente las señales obtenidas. Estos sensores estarán entregando continuamente mientras estén conectados a las entradas analógicas, valores instantáneos de tensión y de corriente.

En la Figura 3.2 se muestra el diagrama a bloques que en su conjunto forman el Hardware del simulador.

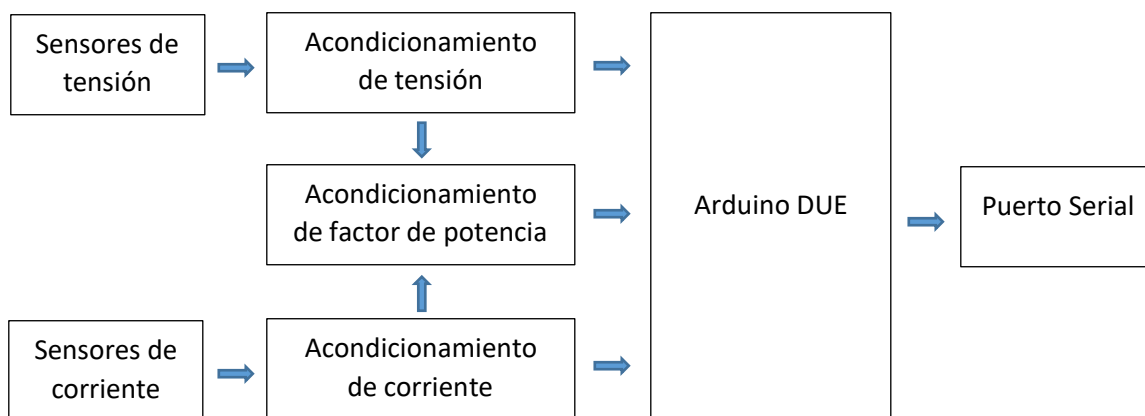


Figura 3.2. Bloques que conforman el Hardware del simulador.

A continuación, se describe el procedimiento empleado para el desarrollo de cada uno de los bloques del Hardware.

3.1.1. Especificación y selección de la placa de Arduino.

Para la realización del Hardware se utilizó una placa de Arduino DUE dado que es un Hardware completamente libre, y gracias a ello resulta posible implementarlo en el desarrollo de cualquier proyecto. Arduino DUE es la primera placa basada en un microcontrolador ARM de 32 bits [4]; dentro de las características de esta placa que favorecen a la realización del presente proyecto es la disponibilidad de 12 entradas analógicas, de las cuales se utilizan 9 como a continuación se describe:

- Entradas analógicas A0, A1 y A2 para las señales acondicionadas de corriente fase A, corriente fase B y corriente fase C, respectivamente.

- Entradas analógicas A3, A4 y A5 para las señales acondicionadas de tensión fase A, tensión fase B y tensión fase C, respectivamente.
- Entradas analógicas A6, A7 y A8 para las señales acondicionadas de factor de potencia fase A, factor de potencia fase B y factor de potencia fase C, respectivamente.

Es importante comentar que Arduino DUE trabaja a una tensión de 3.3 V.C.D. y un valor superior a éste en cualquiera de sus entradas analógicas puede dañar la placa; por esta razón se tuvo especial cuidado en el diseño del acondicionamiento de las señales.

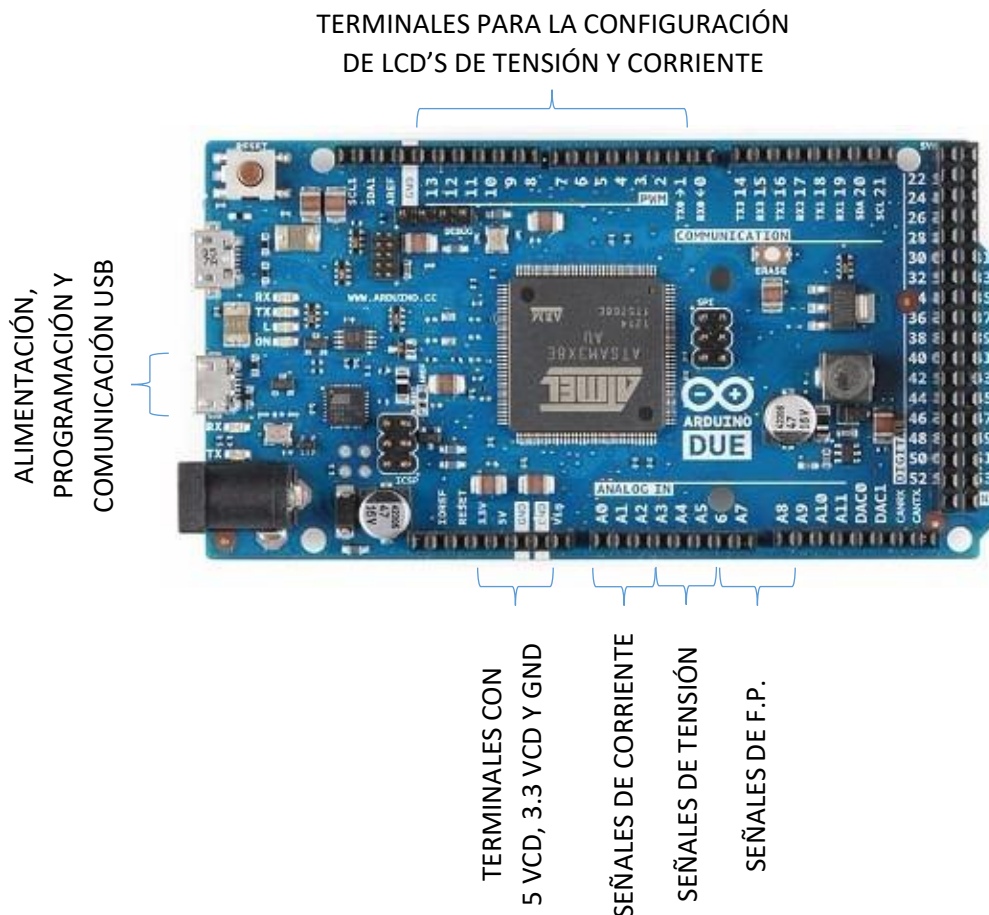


Figura 3.3. Puertos de la placa Arduino DUE utilizados. Fuente: [4].

En la Figura 3.3 se puede apreciar de forma gráfica y resumida lo anteriormente comentado.

3.1.2. Especificación y selección de sensores de tensión y corriente.

3.1.2.1. Sensores de corriente.

Para las mediciones de corriente se utilizaron sensores de tipo núcleo abierto, con la finalidad de encajar en el método no invasivo de medición propuesto. El modelo del sensor es el SCT013-030 de la marca YHDC y se puede apreciar en la Figura 3.4, que se han identificado adecuadamente (“Ia”, “Ib” e “Ic”) para la correcta conexión en las fases a medir, así como su característica no invasiva al poder abrir su núcleo para su enlace con el conductor.



Figura 3.4. Sensores de corriente SCT013-030.

En la Tabla 3.1 se muestran los parámetros eléctricos de interés y en la Figura 3.5 se muestra el diagrama de conexión, información obtenida de su ficha técnica.

Tabla 3.1. Parámetros eléctricos del sensor SCT013-030.

Entrada nominal (RMS)	30 A
Máxima entrada	60 A
Salida nominal (RMS)	1 V
Exactitud	$\pm 1 \%$
Linealidad	$< 0.2 \%$

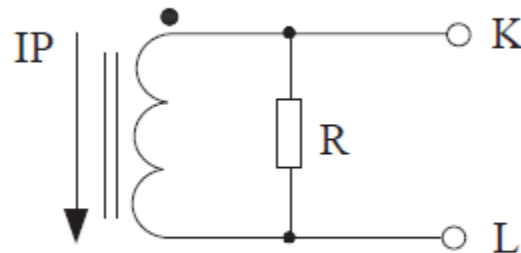


Figura 3.5. Diagrama de conexión del sensor SCT013-030.

La resistencia R que incorpora tiene la función de generar una caída de voltaje KL en función de la corriente I_P que circula entre el núcleo, generando una respuesta como la mostrada en la Figura 3.6.

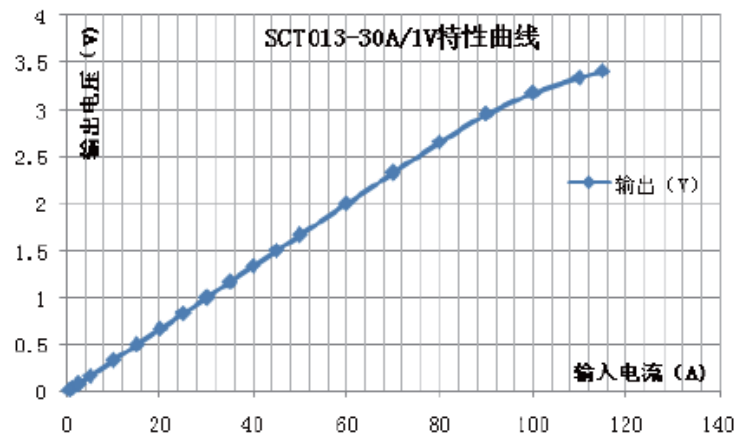


Figura 3.6. Curva de respuesta característica del sensor SCT013-030, tomada de su ficha técnica.

3.1.2.2. Sensores de tensión.

Los sensores de tensión son simplemente cables con un caimán en el extremo que conecta con el sistema a medir. Se seleccionaron cables para que puedan trabajar con tensiones de 600 VCA, ya que el simulador está diseñado para medir motores que trabajan a 460 VCA. En la Figura 3.7 se pueden apreciar estos sensores, que por comodidad en su otro extremo también pueden ser desacoplados de las terminales correspondientes del Hardware.



Figura 3.7. Sensores de tensión.

3.1.3. Acondicionamiento de las señales de corriente.

Una vez estudiada la ficha técnica del sensor de corriente, y conociendo su forma de operar, se procedió al acondicionamiento de la señal de tal manera que pudiera llevarse a las terminales analógicas de Arduino sin que éste sufriera daños. Los datos mostrados en la gráfica de la Figura 3.6 son valores RMS, lo que implica que el sensor bajo estudio trabaja con corriente alterna; en otras palabras, cuando por el sensor circulan $30 A_{(RMS)}$, en su salida se obtendrá $1 V_{(RMS)}$; esto representa un problema para Arduino, ya que no trabaja con tensiones negativas. Para resolver esto, se hace uso de un amplificador operacional como seguidor de tensión [21], pero con la característica de eliminar la parte negativa. El circuito LM158AN cumple



esta función y en la Figura 3.8 se puede apreciar el arreglo propuesto en Multisim y los correspondientes oscilogramas en la Figura 3.9. Para modelar los sensores de corriente se utilizó una fuente de voltaje trifásica en configuración estrella, ya que los sensores tendrán un punto en común. Se configuro un voltaje en la fuente de $1 V_{(RMS)}$ para simular la circulación de una corriente primaria de $30 A_{(RMS)}$.

En el oscilograma de la izquierda, con una configuración de $1 V/Div$, se puede apreciar la onda alterna de las tres señales de corriente en la entrada del circuito LM158AN, mientras que en el oscilograma de la derecha se aprecia las salidas correspondientes, observándose la eliminación de la parte negativa de las señales de corriente. Es muy importante tomar en cuenta que no se estarán procesando los medios ciclos negativos de las ondas de corriente y se tendrá que hacer un ajuste en el código de Arduino para compensar dichos periodos; sin embargo, el propósito del presente proyecto no es realizar un proyecto de análisis de calidad de la energía, en donde esta situación representaría dejar fuera del análisis los transitorios que pudiesen presentarse en los periodos recortados. Para fines de medición de consumo eléctrico, considerar los ciclos positivos medidos como iguales a los negativos recortados no representa una afectación a los resultados esperados.

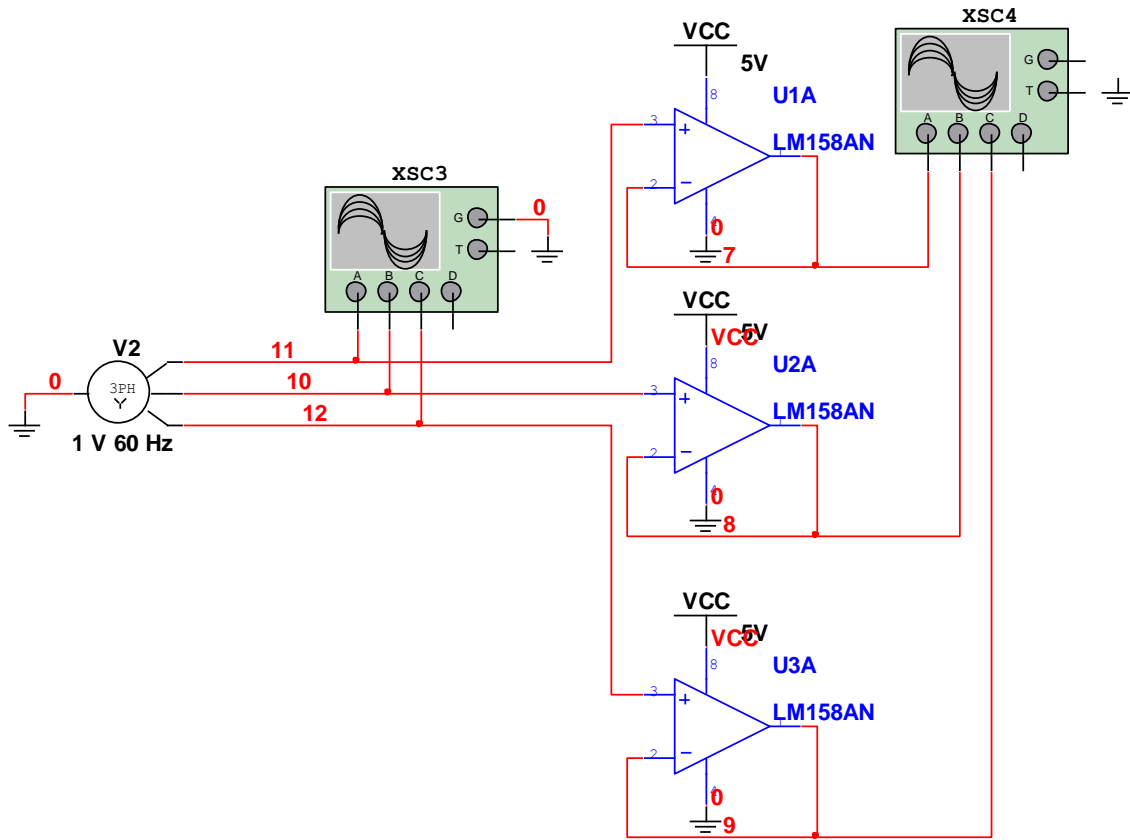


Figura 3.8. Arreglo propuesto con LM158AN, para el acondicionamiento final de la señal de corriente (Multisim).

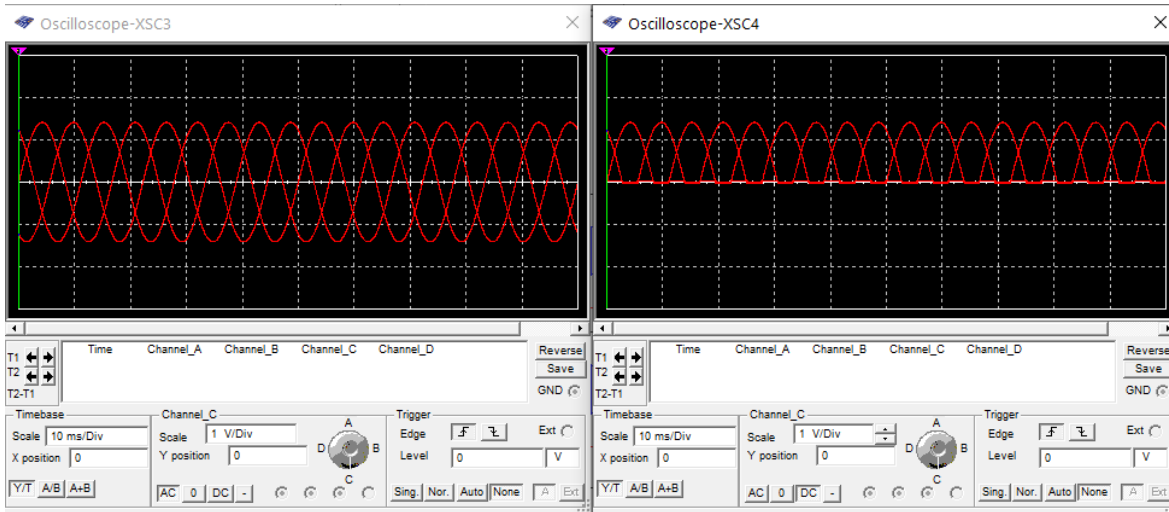


Figura 3.9. Oscilogramas del acondicionamiento de la señal de corriente con LM158AN (Multisim).



3.1.4. Acondicionamiento de las señales de tensión.

Como primer paso, fue reducir el nivel de tensión, sin que se afectara la cualidad de la forma de onda. Esto se logró con un divisor de tensión, el cual se dimensionó con la ayuda de la Ley de Ohm y la Ley de tensión de Kirchhoff [22]. Se propuso una resistencia de entrada de $3.3 \text{ M}\Omega$ con la finalidad de que se tuviera una alta impedancia de entrada, y la segunda resistencia se dimensionó para que se tuviera en ésta, una caída de tensión apropiada para trabajar con Arduino DUE (de 0 a 3.3 VCD); en la Tabla 3.2 se muestran los resultados de los cálculos realizados en Excel.

Tabla 3.2. Cálculo del divisor de tensión en Excel.

Tensión eficaz (RMS)	460
Tensión de pico (Vp)	650.44
R1	3300000
R2	15000
Caída de tensión en R1 (V)	647.50
Caída de tensión en R2 (V)	2.94
I (A)	0.00020
Potencia en R1 (W)	0.12705
Potencia en R2 (W)	0.00058

Como apoyo en el diseño del divisor de tensión, se recurrió al programa Multisim, donde se llevó a cabo la simulación correspondiente, pudiéndose apreciar en la Figura 3.10 el diseño propuesto y en la Figura 3.11 los oscilogramas correspondientes.

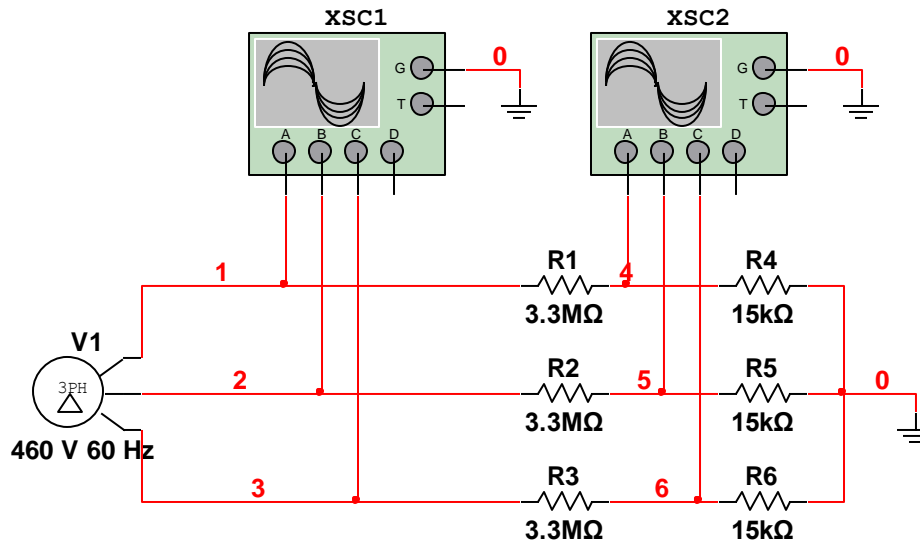


Figura 3.10. Divisor de tensión propuesto (Multisim).

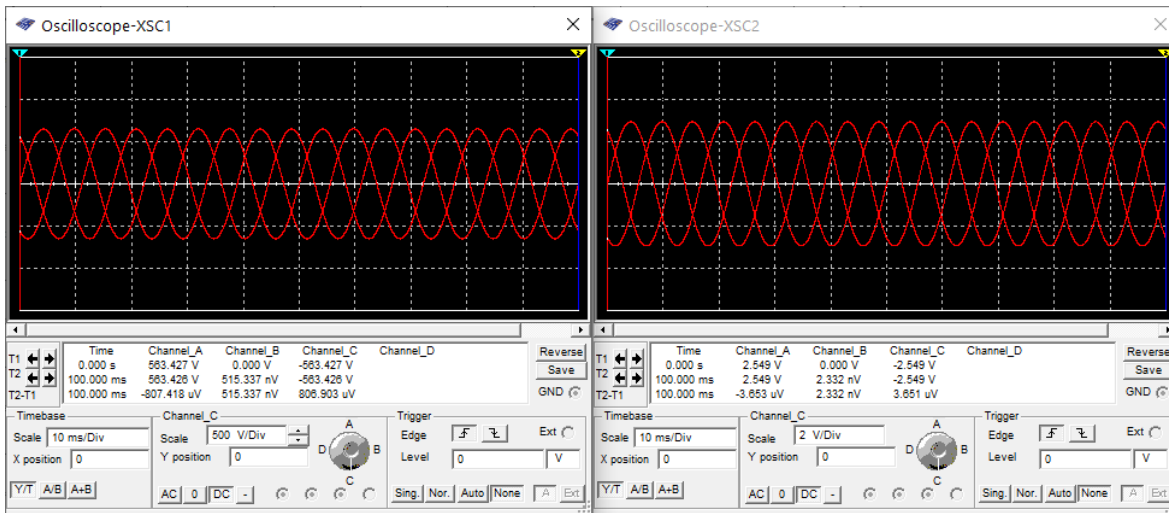


Figura 3.11. Oscilogramas del divisor de tensión (Multisim).

En el oscilograma de la izquierda se muestra la forma de onda de la tensión de alimentación a medir (460 V_(RMS)), que con una escala de 500 V/Div se aprecia un valor de pico idéntico al calculado en Excel (650.44 V_(P)). De la misma forma, en el oscilograma de la derecha se muestra la forma de onda de la tensión resultante en el divisor de tensión, que con una escala de 2 V/Div se aprecia un valor idéntico al obtenido en Excel (2.94 V_(P)). Para finalizar el dimensionamiento del divisor de tensión se calculan las potencias que se estarán disipando en forma de calor en los



resistores, a fin de seleccionar el valor de potencia adecuado de los mismos. En la Tabla 3.2 se muestran dichos cálculos y de ellos se deduce que resistores de $\frac{1}{2}$ W tendrán una capacidad de disipación adecuada.

El acondicionamiento de la señal de tensión hasta ahora resultante no es suficiente para que puedan llevarse a las entradas analógicas de Arduino, ya que en los oscilogramas de la Figura 3.11 se aprecian los ciclos negativos, mismos que terminarían dañando la placa de Arduino. Para solucionar este problema, se recurre nuevamente al circuito LM158AN utilizado en el acondicionamiento de las señales de corriente, obteniéndose resultados similares a los mostrados en el oscilograma de la derecha de la Figura 3.9 para las señales de corriente.

3.1.5. Acondicionamiento de señales para factor de potencia.

Como se describe gráficamente en el diagrama de bloques mostrado en la Figura 3.2, las señales ya acondicionadas de tensión y corriente se llevan tanto a las entradas analógicas de Arduino correspondientes como al bloque de cálculo del factor de potencia. Aquí, se utiliza el amplificador operacional LM139 como comparador [21], con la finalidad de obtener una señal cuadrada a partir de los semiciclos de tensión y corriente llevados a sus entradas. El arreglo propuesto se muestra en la Figura 3.12 y los oscilogramas correspondientes se visualizan en la Figura 3.13.

Finalmente, las ondas cuadradas de tensión y corriente generadas para cada una de las fases se llevan al circuito 74LS08, que básicamente integra compuertas lógicas AND [23], con el objeto de obtener el desfase entre ambas señales a partir de la información del ancho de pulso, tal como se puede interpretar en la Figura 3.14, siendo evidente que para obtener el tiempo de desfase basta con sustraer al periodo del semiciclo el tiempo del ancho del pulso resultante en las salidas de las compuestas AND; el código de Arduino se encargara de llevar a cabo estos cálculos, para posteriormente obtener el factor de potencia con las reglas de conversión correspondiente [24]. En las Figuras 15 y 16 se muestra el circuito simulado en Multisim, así como los correspondientes oscilogramas.

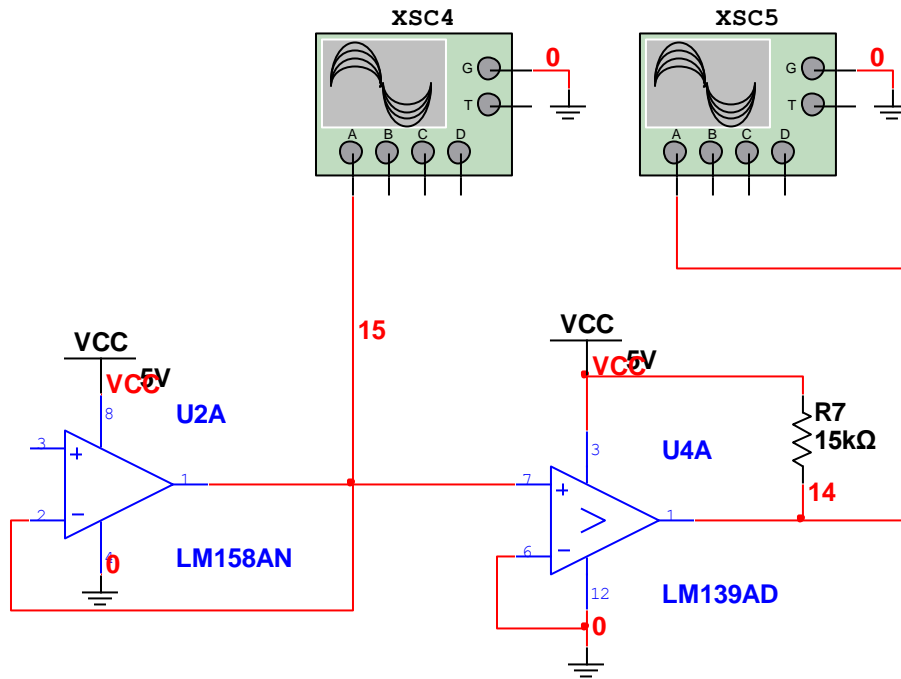


Figura 3.12. Circuito acondicionador de señales propuesto para factor de potencia (Multisim).

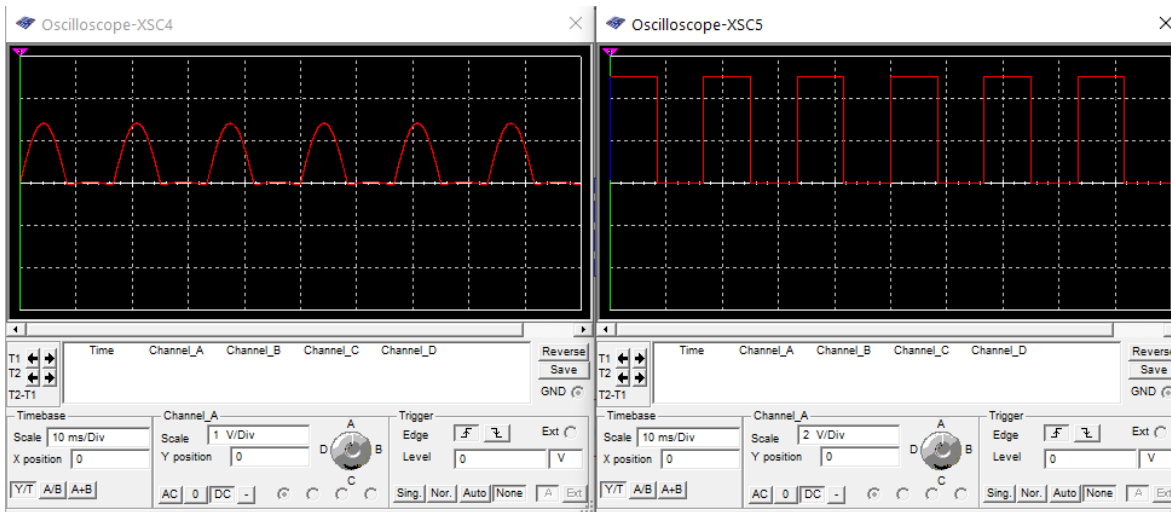


Figura 3.13. Oscilogramas del circuito acondicionador de señales para factor de potencia (Multisim).

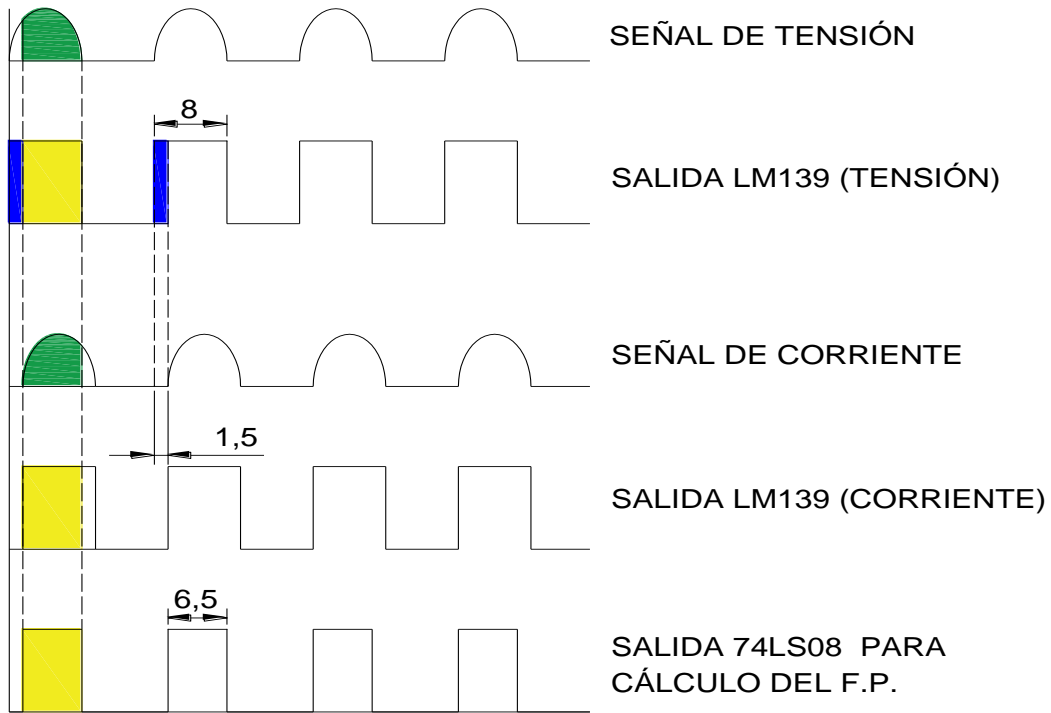


Figura 3.14. Principio de operación para el cálculo del factor de potencia.

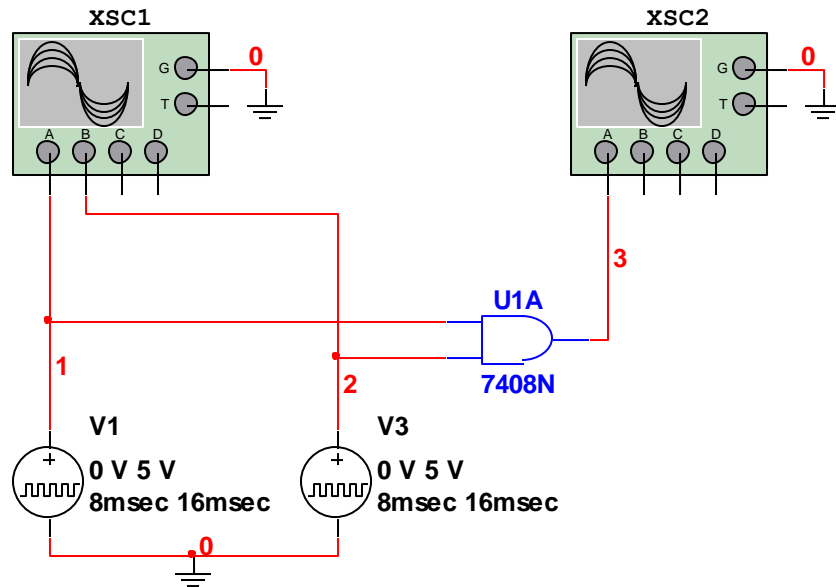


Figura 15. Circuito acondicionador de señal propuesto para cálculo de factor de potencia (Multisim).

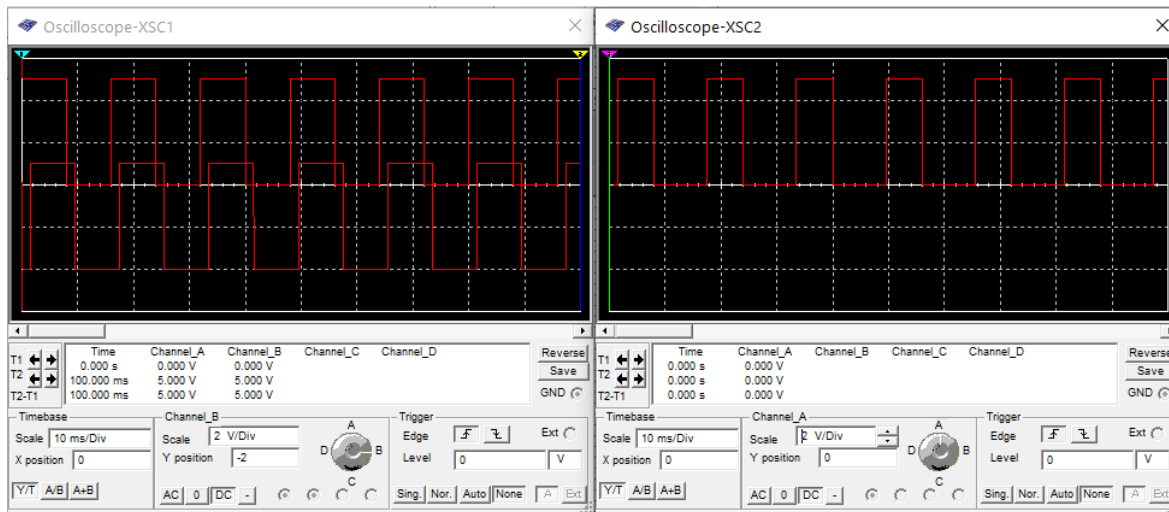


Figura 16. Oscilogramas del circuito acondicionador de señal para factor de potencia.

3.1.6. Protecciones eléctricas del Hardware.

No se consideró circuitería para la regulación de la tensión de alimentación del Hardware, ya que ésta es a través del puerto USB de la computadora que tiene instalado el Software desarrollado.

3.1.6.1. Fusibles.

Debido a que el simulador trabajará con tensiones peligrosas (460 VCA), se consideró la instalación de fusibles de acción rápida en las terminales de entrada de tensión del Hardware; ante una posible condición de corto circuito dentro del Hardware, los fusibles operarán incluso antes de que la protección termomagnética de la máquina en evaluación opere.

3.1.6.2. Diodos Zener.

Con la finalidad de proteger las entradas analógicas de Arduino para las señales de tensión, se instalaron diodos Zener de 3.3 V como reguladores, de tal manera, que, ante una sobretensión superior a este valor, entrarán en su condición de avalancha [21], absorbiéndose la sobretensión generada. En la Figura 3.17 se muestra la terminación del Hardware, mismo que se integró dentro de un maletín para que

pueda trasladarse de una forma práctica. A este desarrollo se le dio el nombre de: “Power Efficiency Analyzer Hardware”.

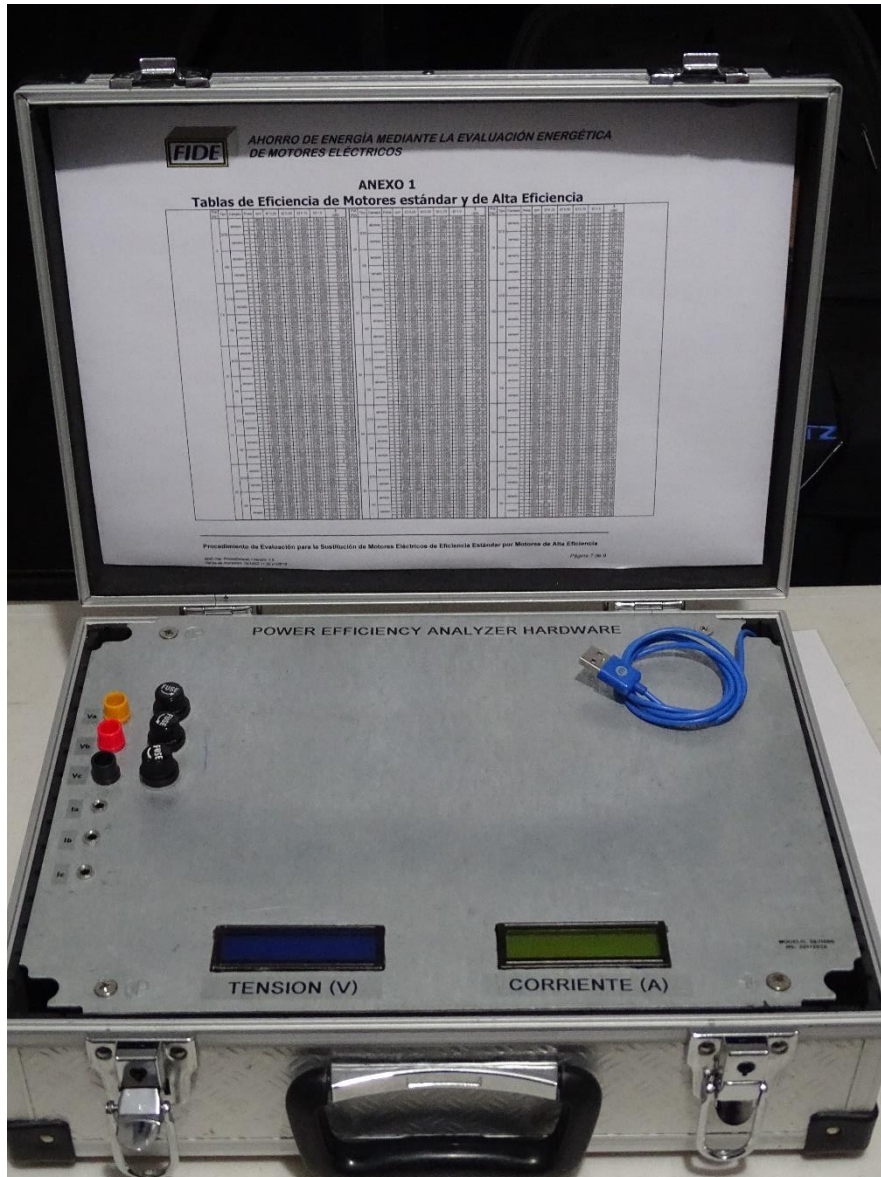


Figura 3.17. Hardware del simulador de eficiencia energética de motores de inducción.

3.1.6. Código de Arduino.

El código de Arduino tiene la función de calcular los valores RMS de tensión y corriente a partir de las señales instantáneas recibidas de los sensores

correspondientes en sus entradas analógicas. Así mismo, calcula el factor de potencia de las tres fases a partir del ancho de pulso que corresponde al tiempo que están en fase las señales de tensión y corriente. Debido a que lo anterior representa un sistema multitarea, se utilizó el concepto de máquina finita [25] para la programación de Arduino, quedando como se muestra en el diagrama de flujo simplificado de la Figura 3.18, llevándose a cabo en su totalidad en un lapso de 10 segundos, por lo que el envío de datos a través del puerto serial es a esa frecuencia. Cabe mencionar que se puede reducir dicho tiempo, incluso a cada segundo; sin embargo, para fines de la evaluación energética no se considera necesario.

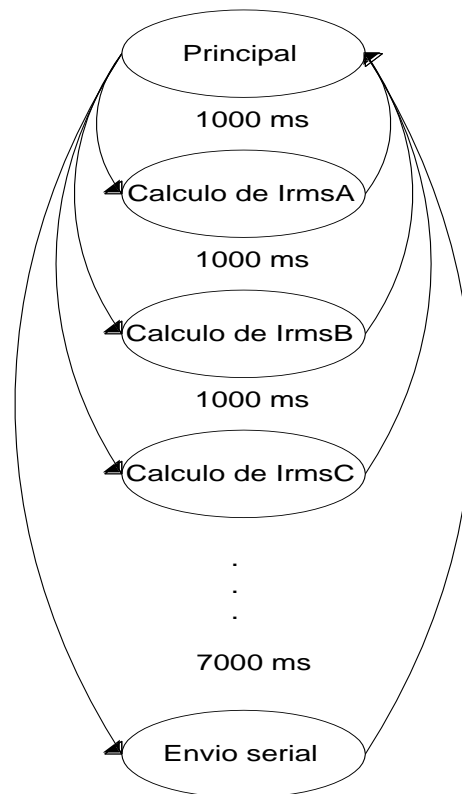


Figura 3.18. Máquina de estados finitos como método de programación.

El código de Arduino inicia con la configuración de librerías y los pines donde se conectarán los dos display de 7 segmentos, uno para visualizar los valores de tensión RMS calculados y el otro para visualizar los valores de corriente RMS. Esta parte del código se muestra en la Figura 3.19.



```
#include <LiquidCrystal.h>           // Cargar librería de lcd.  
LiquidCrystal lcdv(13, 12, 5, 4, 3, 2); // Configuración del lcd de tension  
LiquidCrystal lcdi(13, 12, 11, 10, 9, 8); // Configuración del lcd de corriente
```

Figura 3.19. Configuración de librerías y pines para los LCD's utilizados.

En la Figura 3.20 se muestra la declaración de variables utilizadas dentro del código de Arduino; se muestran los comentarios con la finalidad de indicar el propósito de cada variable.

```
float IrmsA;           // Corriente RMS fase A  
float IrmsB;           // Corriente RMS fase B  
float IrmsC;           // Corriente RMS fase C  
float sensorIa;        // Variable para el sensor de corriente instantánea fase A  
float sensorIb;        // Variable para el sensor de corriente instantánea fase B  
float sensorIc;        // Variable para el sensor de corriente instantánea fase C  
int VrmsA;            // Tensión RMS fase A  
int VrmsB;            // Tensión RMS fase B  
int VrmsC;            // Tensión RMS fase C  
float sensorVa;        // Variable para el sensor de tensión instantánea fase A  
float sensorVb;        // Variable para el sensor de tensión instantánea fase B  
float sensorVc;        // Variable para el sensor de tensión instantánea fase C  
static float OMEGA=(2.0*PI*60.0); // Velocidad angular a 60 Hz  
static float DEGRAD=(2.0*PI/360.0); // Conversión de grados a radianes  
static float RADDEG=(360.0/(2.0*PI)); // Conversión de radianes a grados  
float Drms1;           // Valor RMS del defase entre Va e Ia  
float Drms2;           // Valor RMS del defase entre Vb e Ib  
float Drms3;           // Valor RMS del defase entre Vc e Ic  
float sensorD1;        // Variable para la salida de la compuerta lógica AND entre Va e Ia  
float sensorD2;        // Variable para la salida de la compuerta lógica AND entre Vb e Ib  
float sensorD3;        // Variable para la salida de la compuerta lógica AND entre Vc e Ic  
float cosF11;          // Factor de potencia entre Va e Ia  
float cosF12;          // Factor de potencia entre Vb e Ib  
float cosF13;          // Factor de potencia entre Vc e Ic
```

Figura 3.20. Declaración de variables.

En la Figura 3.21 se muestra la función void setup(), donde se configura el puerto serial y los display de cristal líquido (LCD) utilizados. Se puede observar que se canceló la impresión de etiquetas dentro del LCD; esto debido a que cada uno



desplegará el valor de tres variables y la etiqueta sacrifica espacio dentro del mismo. Esto se resolvió etiquetando físicamente el borde superior de cada LCD para saber su función (ver Figura 3.17).

```
void setup() {  
  Serial.begin(9600);           // Configura velocidad del puerto para comunicación  
  Serial1.begin(9600);         // Configura velocidad del puerto para comunicación con PC  
  Serial.println("Conexion establecida"); // Imprime primer mensaje  
  lcdi.begin(16, 2);           // Define tamaño del lcd de corriente  
  //lcdi.print("Corriente(A)"); // Usar cuando se requiera etiqueta en el LCD  
  lcdv.begin(16, 2);           // Define tamaño del lcd de tension  
  //lcdv.print("Tension(V)");   // Usar cuando se requiera etiqueta en el LCD  
}
```

Figura 3.21. Función void setup().

En la Figura 3.22 se muestra la función void loop() como una máquina de estados finitos, con un tiempo total de 10 segundos cada ciclo de ejecución.

```
void loop() { // Código que estara constantemente ejecutandose  
  corrienteA(); // tarda 1000 ms en ejecutarse  
  corrienteB(); // tarda 1000 ms en ejecutarse  
  corrienteC(); // tarda 1000 ms en ejecutarse  
  tensionA(); // tarda 1000 ms en ejecutarse  
  tensionB(); // tarda 1000 ms en ejecutarse  
  tensionC(); // tarda 1000 ms en ejecutarse  
  FP1(); // tarda 1000 ms en ejecutarse  
  FP2(); // tarda 1000 ms en ejecutarse  
  FP3(); // tarda 1000 ms en ejecutarse  
  envio_serial(); // tarda 300 ms en ejecutarse  
  envio_LCD1(); // tarda 300 ms en ejecutarse  
  envio_LCD2(); // tarda 300 ms en ejecutarse  
  delay (100);  
}
```

Figura 3.22. Función void loop().

En la Figura 3.23 se expone el código del primer estado finito, el cual corresponde al cálculo de la corriente RMS de la fase A. En las tres primeras líneas dentro de la

función se declaran e inicializan las variables que se utilizan, y por medio de la función `while()`, cada milisegundo se estará sumando el cuadrado del valor instantáneo obtenido por el sensor de corriente correspondiente, hasta completar 1 segundo, para posteriormente sacar la raíz cuadrada de la sumatoria de cuadrados obtenida entre el número de muestras, dado que esa es la fórmula para obtener un valor eficaz o RMS [22]. Debe aclararse que previo a la aplicación de la fórmula mencionada, la sumatoria de cuadrados se multiplica por 2; esto por los semiciclos negativos que han sido eliminados durante el acondicionamiento de la señal.

Finalmente, con el objeto de eliminar corrientes parásitas, mediante la función `if()` se hacen cero todos los valores de corriente eficaz que resulten menores a 1 A; esto no afecta la simulación ya que se pretenden medir motores que demanden corrientes mayores a este valor.

```
void corrienteA(){
  float ondala=0;
  long retardo=millis();
  int s=0;
  while(millis()-retardo<1000){
    sensorIa=analogRead(A0)*(1.414/1024.0);
    IrmsA=sensorIa*30;
    ondala=ondala+sq(IrmsA);
    s=s+1;
    delay(1);
  }
  ondala=ondala*2;
  IrmsA=sqrt((ondala)/(s));
  if(IrmsA<1){
    IrmsA=0;
  }
}
```

Figura 3.23. Función `corrienteA()`.

Este código es similar para el resto de los estados finitos, mostrándose en las Figuras 3.24 y 3.25 las funciones `tensionA()` y `FP1()`, respectivamente, para que puedan visualizarse las diferencias.



```
void tensionA(){
float ondaVa=0;
long retardo=millis();
int s=0;
while(millis()-retardo<1000){
  sensorVa=analogRead(A3)*(3.3/1024.0);
  VrmsA=sensorVa*220.0;
  ondaVa=ondaVa+sq(VrmsA);
  s=s+1;
  delay(1);
}
ondaVa=ondaVa*2;
VrmsA=sqrt((ondaVa)/(s));
if(VrmsA<1){
  VrmsA=0;
}
}
```

Figura 3.24. Función tensionA().



```
void FP1(){
  float defase1=0;
  float td1;
  float FI1;
  long retardo=millis();
  int s=0;
  while(millis()-retardo<1000){
    sensorD1=analogRead(A6)*(3.3/1024.0);
    Drms1=sensorD1*1.0;
    defase1=defase1+Drms1;
    s=s+1;
    delay(1);
  }
  defase1=defase1*2;
  Drms1=sqrt((defase1)/(s));
  td1=(16-Drms1);
  FI1=((td1)*(OMEGA)*(RADDEG)/1000.0);
  cosFI1=cos(FI1*DEGRAD);
  if(lrmsA<1){
    cosFI1=0;
  }
}
```

Figura 3.25. Función FP1().

La diferencia más significativa se da en las funciones FPx(), donde se tienen que aplicar las conversiones de radianes a grados y viceversa, para obtener el factor de potencia a partir del tiempo de desfaseamiento entre las señales de tensiones y corrientes [24].

Finalmente, en la Figura 3.26 se muestran los códigos de los estados finitos creados para el envío de información a través del puerto serial y hacia los LCD's correspondientes.



```
void envio_serial(){
  Serial.println(String (VrmsA)+(",")+VrmsB)+(",")+VrmsC)+(",")+IrmsA)+(",")+IrmsB)+(",")+IrmsC)+(",")+co
  delay (300);
}

void envio_LCD1(){
  lcdv.setCursor(0, 0);           // Imprime en primera línea
  lcdv.print("A=");
  lcdv.print(VrmsA);             // Imprime tension RMS Fase A
  lcdv.setCursor(0, 1);         // Imprime en segunda línea
  lcdv.print("B=");
  lcdv.print(VrmsB);             // Imprime tension RMS Fase B
  lcdv.print(" ");
  lcdv.print("C=");
  lcdv.print(VrmsC);             // Imprime tension RMS Fase C
  delay (300);
}

void envio_LCD2(){
  lcdi.setCursor(0, 0);          // Imprime en primera línea
  lcdi.print("A=");
  lcdi.print(IrmsA, 1);          // Imprime corriente RMS Fase A
  lcdi.setCursor(0, 1);         // Imprime en segunda línea
  lcdi.print("B=");
  lcdi.print(IrmsB, 1);          // Imprime corriente RMS Fase B
  lcdi.print(" ");
  lcdi.print("C=");
  lcdi.print(IrmsC, 1);          // Imprime corriente RMS Fase C
  delay (300);
}
```

Figura 3.26. Códigos para el envío de información al puerto serial y LCD's.

El Hardware cumple su objetivo al enviar por el puerto serial una cadena de caracteres con la información de corrientes y tensiones eficaces o RMS y los factores de potencia medidos. El Software se encargará de llevar a cabo el resto de los cálculos para la simulación de eficiencia energética, a partir de la información recibida.

3.2. Desarrollo del Software

El Software se desarrolló en la plataforma Visual Studio 2019, versión Community, misma que es de uso libre. El lenguaje de programación utilizado es C# y es una programación orientada a objetos y procedimientos [3]. Dentro de la plataforma de Visual Studio se utilizó el entorno de programación de Windows Forms, donde se diseñaron las ventanas gráficas bajo las cuales se lleva a cabo la simulación de eficiencia energética. El desarrollo del Software está basado en el procedimiento para la evaluación energética de motores desarrollado por el Fideicomiso para el Ahorro de Energía Eléctrica (FIDE) en conjunto con la Comisión Nacional de Energía Eléctrica (CNEE) de Guatemala [6].

3.2.1. Programación de objetos.

En la Figura 3.27 se muestra la ventana de bienvenida creada para el inicio del Software, que como puede apreciarse, se ha llamado “Power Efficiency Analyzer Software”, idéntico al nombramiento del Hardware, para enfatizar la correlación entre ambos desarrollos. De inicio se comenta que los botones “Configuración” y “Simulación Económica” son por ahora solo proyecciones para una versión futura mejorada del Software. El presente desarrollo es unidireccional, es decir, la información va del Hardware al Software; mediante la Configuración se pretenderá el poder llevar a cabo la parametrización desde el Software, de las mediciones realizadas por el Hardware, tal como se parametriza un analizador de redes comercial para cada tipo de sistema a medir. Así mismo, mediante la Simulación Económica, el Software llevará a cabo los cálculos de las inversiones, así como de los periodos de recuperación por ahorro de energía. El alcance del presente desarrollo se limita al cálculo del ahorro de energía simulado por sustituir el motor de eficiencia estándar evaluado por un motor de alta eficiencia propuesto, por lo que el único botón habilitado en este Menú es el de “Simulación Energética”.



Figura 3.27. Ventana de bienvenida del Software.

En la Figura 3.28 se puede apreciar la ventana que se abre al oprimir el botón de “Simulación Energética”, misma que será el entorno tanto para la evaluación energética del motor medido por el Hardware como para el cálculo del ahorro de energía esperado al llevar a cabo la sustitución por el motor de alta eficiencia propuesto.

A continuación, se describirán cada uno de los componentes de este entorno, así como su propósito.

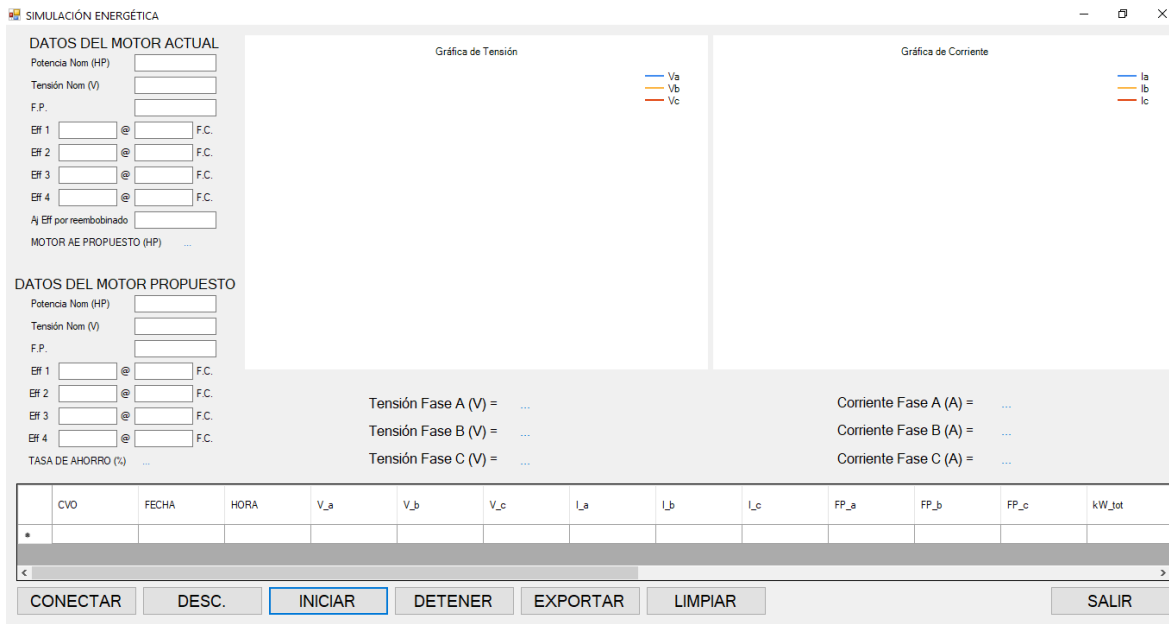



Figura 3.28. Ventana del entorno de Simulación Energética.

En la Figura 3.29 se muestra la sección donde se introducen los datos de placa del motor de eficiencia estándar a evaluar, así como sus eficiencias para distintos factores de carga; esta última información debe investigarse en las curvas de eficiencia indicadas por el fabricante, o en su defecto, recurrir a eficiencias típicas cuando la placa de datos no sea legible o no se cuente con la información suficiente de eficiencia energética. En el APÉNDICE se presenta una tabla de apoyo [6] con las eficiencias típicas de motores estándar y motores de alta eficiencia, misma que como se puede observar en la Figura 3.17, se ha colocado en la parte interna del maletín del Hardware para que pueda usarse ante la ausencia de esta información. Se puede observar, además, que al final de esta sección hay un campo nombrado “MOTOR AE PROPUESTO (HP)”; es aquí donde el simulador establece la capacidad en HP del motor de alta eficiencia propuesto, que como se verá más adelante en el código de programación, selecciona aquel que trabaje a un factor de carga del 80%, ya que este factor se recomienda [6] por presentar la máxima eficiencia del motor.



 SIMULACIÓN ENERGÉTICA

DATOS DEL MOTOR ACTUAL

Potencia Nom (HP)

Tensión Nom (V)

F.P.

Eff 1 @ F.C.

Eff 2 @ F.C.

Eff 3 @ F.C.

Eff 4 @ F.C.

Aj Eff por reembobinado

MOTOR AE PROPUESTO (HP) ...

Figura 3.29. Sección de datos del motor de eficiencia estándar evaluado.

En la Figura 3.30, se muestra la sección donde se introducen los datos del motor de alta eficiencia, los cuales, en la actualidad, los fabricantes de motores se ven obligados a presentar a los consumidores. Aun así, la misma tabla mostrada en el APÉNDICE puede utilizarse para completar estos campos si no se tiene dicha información del fabricante. De igual forma que en la sección anterior, se puede observar que al final de esta sección hay un campo nombrado “TASA DE AHORRO (%)”; aquí el simulador indica el porcentaje de ahorro energético al compararse los rendimientos del motor evaluado y el motor propuesto.

DATOS DEL MOTOR PROPUESTO

Potencia Nom (HP)	<input type="text"/>
Tensión Nom (V)	<input type="text"/>
F.P.	<input type="text"/>
Eff 1	<input type="text"/> @ <input type="text"/> F.C.
Eff 2	<input type="text"/> @ <input type="text"/> F.C.
Eff 3	<input type="text"/> @ <input type="text"/> F.C.
Eff 4	<input type="text"/> @ <input type="text"/> F.C.
TASA DE AHORRO (%)	...

Figura 3.30. Sección de datos del motor de alta eficiencia propuesto.

En la Figura 3.31 se muestra la sección de botones, mismos que se describen a continuación:

CONECTAR. Mediante este botón se establece la conexión serial entre el Hardware y el Software. Es necesario presionarlo antes de presionar el botón “INICIAR”.

DESC. (Desconectar). Desconecta la comunicación serial entre el Hardware y el Software.

INICIAR. Inicia el muestreo del Hardware. Antes de presionar este botón se deben llenar los campos editables de datos de los motores actual y propuesto, y haber establecido previamente la conexión serial con el botón “CONECTAR”

DETENER. Detiene el muestreo del Hardware; para continuar el muestreo se debe presionar nuevamente el botón “INICIAR”.

EXPORTAR. Exporta el muestreo de datos desde que se inician las mediciones hasta que se detienen, en una hoja nueva de Excel.

LIMPIAR. Limpia el muestreo de datos realizado.

SALIR. Sale de la aplicación.



Figura 3.31. Sección de botones.

En la Figura 3.32 se muestra la sección donde se grafican cada uno de los datos de tensión y corriente que envía el Hardware cada 10 segundos.

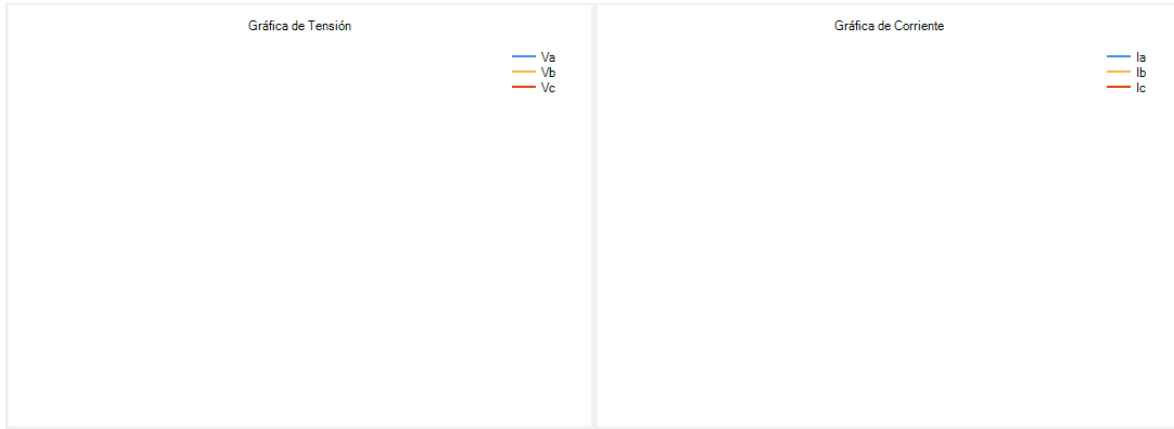


Figura 3.32. Sección de gráficos.

Así mismo, en la Figura 3.33 se muestra el cuadro de instrumentos, donde se indican los últimos datos de tensión y corriente enviados por el Hardware.

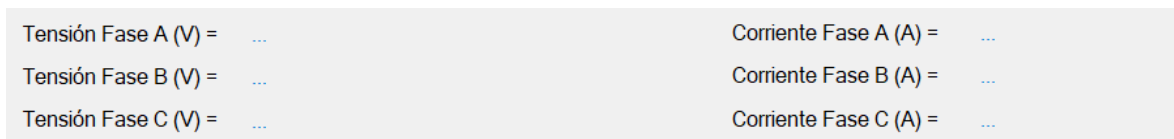


Figura 3.33. Sección de Instrumentos.

Finalmente, en la Figura 3.34 se muestra la sección donde se registran todos y cada uno de los datos de tensión, corriente y factor de potencia enviado por el Hardware, así como los calculados por el mismo Software, como las potencias real, aparente y reactiva, el factor de potencia total, el factor de carga, la eficiencia a la cual opera el motor actual, la potencia mecánica demandada en la flecha, la demanda eléctrica del motor propuesto, el ahorro en demanda y el ahorro en consumo.

CVO	FECHA	HORA	V _a	V _b	V _c	I _a	I _b	I _c	FP _a	FP _b	FP _c	kW _{tot}
*												

Figura 3.34. Sección de parámetros.



3.2.2. Programación de procedimientos.

A continuación, se describe cómo se programaron cada uno de los objetos indicados en la sección anterior. En la Figura 3.35 se muestra el código fuente de la ventana principal (ver Figura 3.27) donde básicamente se muestran las librerías cargadas automáticamente por el entorno de programación, y el código para cada uno de los tres botones creados. La sentencia `formxfrm = new formx(); frmShow();` sirve para abrir la ventana correspondiente al hacer clic en el botón. Como ya se indicó anteriormente, tanto el botón `“CONFIGURACIÓN”` como `“SIMULACIÓN ECONÓMICA”` abren ventanas sin objetos, y pretender ser botones para una versión mejorada del Software donde se pueda configurar las mediciones a realizar por el Hardware, así como llevarse a cabo la simulación económica del remplazo del motor actual por el motor de alta eficiencia propuesto.

El código fuente de la ventana de `“SIMULACIÓN ENERGÉTICA”` se comenta por secciones ya que es muy extenso, donde en la Figura 3.36 se muestran las librerías utilizadas, destacándose la librería `“using offis = Microsoft.Office.Interop.Excel;”` la cual es clave para la exportación de datos a Excel.

En la Figura 3.37 se muestra la declaración de dos variables, `“líneaNo”` como un tipo de dato `int` e inicializado en 1 y `“Dato21”` como un dato tipo `Double`. La primera es para numerar de 1 en 1 y en forma consecutiva el muestreo de datos del Hardware, que como ya se mencionó es cada 10 segundos. La segunda se utiliza para almacenar el valor acumulado del ahorro en consumo.

En la Figura 3.38 se muestran las sentencias para configurar el puerto serial, mismo que debe corresponder al ocupado por el Hardware, en este caso el COM3. La primera sentencia es para establecer la conexión en el puerto serial al hacer clic en el botón `“CONECTAR”` y la segunda sentencia cierra el puerto serial al hacer clic en el botón `“DESC.”`.



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Simulador_de_eficiencia_de_motores
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form2 frm = new Form2();
            frm.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form3 frm = new Form3();
            frm.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Form4 frm = new Form4();
            frm.Show();
        }
    }
}
```

Figura 3.35. Código fuente de la ventana principal.



```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.IO.Ports;  
using System.Windows.Forms.DataVisualization.Charting;  
using offis = Microsoft.Office.Interop.Excel;  
using System.Threading;
```

Figura 3.36. Librerías del código fuente de la ventana de simulación energética.

```
namespace Simulador_de_eficiencia_de_motores  
{  
    public partial class Form3 : Form  
    {  
  
        int lineaNo = 1;  
        Double Dato21;  
  
        public Form3()  
        {  
            InitializeComponent();  
        }  
    }  
}
```

Figura 3.37. Código fuente de la ventana de simulación energética, parte 1.

```
private void button5_Click(object sender, EventArgs e)
{
    serialPort1.PortName = "COM3";
    serialPort1.Open();
}

private void button6_Click(object sender, EventArgs e)
{
    serialPort1.Close();
}
```

Figura 3.38. Código fuente de la ventana de simulación energética, parte 2.

En la Figura 3.39 se muestran las sentencias para inicializar un timer al hacer clic en el botón “INICIAR”, ajustado a 10 segundos como la tasa de muestreo del Hardware, para que en cada ciclo se ejecuten los cálculos correspondientes en el código. Así mismo, con el botón “DETENER” se detiene el timer y con ello la ejecución del código.

```
private void button1_Click(object sender, EventArgs e)
{
    Timer1.Enabled = true;
}

private void button2_Click(object sender, EventArgs e)
{
    Timer1.Enabled = false;
}
```

Figura 3.39. Código fuente de la ventana de simulación energética, parte 3.

En la Figura 3.40 se muestran las sentencias para que al hacer clic en el botón “EXPORTAR” se lleve a cabo la exportación de los datos registrados, desde el inicio

del muestreo hasta su finalización. La sentencia “dataGridView1.Rows.Clear();” limpia de los datos registrados.

```
private void button3_Click(object sender, EventArgs e)
{
    Microsoft.Office.Interop.Excel.Application objExcel = new Microsoft.Office.Interop.Excel.Application();
    objExcel.Visible = true;
    Microsoft.Office.Interop.Excel.Workbook objbook = objExcel.Workbooks.Add(System.Reflection.Missing.Value);
    Microsoft.Office.Interop.Excel.Worksheet objSheet = (Microsoft.Office.Interop.Excel.Worksheet)objbook.Worksheets.get_Item(1);

    for (int s = 0; s < dataGridView1.Columns.Count; s++)
    {
        Microsoft.Office.Interop.Excel.Range myrange = (Microsoft.Office.Interop.Excel.Range)objSheet.Cells[1, s + 1];
        myrange.Value2 = dataGridView1.Columns[s].HeaderText;
    }

    for (int s = 0; s < dataGridView1.Columns.Count; s++)
    {
        for (int j = 0; j < dataGridView1.Rows.Count; j++)
        {
            Microsoft.Office.Interop.Excel.Range myrange = (Microsoft.Office.Interop.Excel.Range)objSheet.Cells[j+2, s + 1];
            myrange.Value2 = dataGridView1[s, j].Value;
        }
    }
}

private void button4_Click(object sender, EventArgs e)
{
    dataGridView1.Rows.Clear();
}
```

Figura 3.40. Código fuente de la ventana de simulación energética, parte 4.

En la Figura 3.41 se muestra la parte del código que estará ejecutándose en cada ciclo del timer. Se comienza con la declaración de variables: “entrante” que es la cadena de datos tipo string enviada por el Hardware a través del puerto serial; “tiempo” del tipo int para la numeración consecutiva del eje horizontal de las gráficas de tensión y corriente; “línea” del tipo int para el incremento de filas en la tabla datos de Excel; y el “delimitador” del tipo char para la separación de los datos enviados por el Hardware en forma de cadena. Con las siguientes sentencias, se lee la cadena de datos que esté presente en el puerto serial y se separan dentro de una matriz por medio del delimitador ‘,’ mismo que se utiliza en el código de Arduino para separar los diferentes datos enviados. Con la sentencia “DateTime =



DateTime.Now;” se establece la fecha y hora en el instante en que se lleva a cabo le ejecución del código.

```
private void Timer1_Tick(object sender, EventArgs e)
{
    string entrante;
    int tiempo = 0;
    int linea = 1;
    char delimitador = ',';
    entrante = serialPort1.ReadLine();
    string cadena = entrante;
    string[] arr = cadena.Split(delimitador);
    string Dato1 = arr[0];
    string Dato2 = arr[1];
    string Dato3 = arr[2];
    string Dato4 = arr[3];
    string Dato5 = arr[4];
    string Dato6 = arr[5];
    string Dato7 = arr[6];
    string Dato8 = arr[7];
    string Dato9 = arr[8];
    DateTime dateTime = DateTime.Now;
```

Figura 3.41. Código fuente de la ventana de simulación energética, parte 5.

En la Figura 3.42 se muestran las sentencias para la programación de las gráficas de tensión y corriente, así como el cuadro de instrumentos, ya descritos en las secciones anteriores. Se puede deducir claramente, que las variables “Dato1”, “Dato2”, “Dato3”, “Dato4”, “Dato5” y “Dato6” corresponden a los datos de tensión fase A, tensión fase B, tensión fase C, corriente fase A, corriente fase B y corriente fase C, respectivamente, obtenidos de la cadena de caracteres recibidos del Hardware. Por lo tanto, las variables “Dato7”, “Dato8” y “Dato9” observadas en la Figura 3.41 corresponden a los datos de factor de potencia fase A, factor de potencia fase B y factor de potencia fase C, respectivamente, obtenidos de la cadena de caracteres recibidos del Hardware.



```
this.chart2.Series["Va"].Points.AddXY(tiempo, Dato1);  
this.chart2.Series["Vb"].Points.AddXY(tiempo, Dato2);  
this.chart2.Series["Vc"].Points.AddXY(tiempo, Dato3);  
this.chart1.Series["Ia"].Points.AddXY(tiempo, Dato4);  
this.chart1.Series["Ib"].Points.AddXY(tiempo, Dato5);  
this.chart1.Series["Ic"].Points.AddXY(tiempo, Dato6);  
tiempo = (tiempo + 1);  
  
label44.Text = Dato1;  
label45.Text = Dato2;  
label46.Text = Dato3;  
label47.Text = Dato4;  
label48.Text = Dato5;  
label49.Text = Dato6;
```

Figura 3.42. Código fuente de la ventana de simulación energética, parte 6.

En la Figura 3.43 se inicia el código para llevar a cabo los cálculos dentro del Software a partir de los datos recibidos del Hardware. La sentencia para el “Dato10” corresponde al cálculo de potencia real trifásica; nótese que se obtiene multiplicando la tensión por la corriente por el factor de potencia de cada fase y sumándose los tres resultados [24], divididos entre 1000 al final para indicar unidades en kW. La sentencia del “Dato11” es similar a la anterior, con la diferencia de que no entra en la operación el factor de potencia, por lo que se obtiene como resultado la potencia aparente, en kVA. Con la sentencia del “Dato12” se obtiene la potencia reactiva en kVAr, al aplicar la fórmula del triángulo de potencias [24] con los ya obtenidos de potencia real y potencia aparente. Finalmente, la sentencia del “Dato13” calcula el factor de potencia total al dividir la potencia real entre la potencia aparente [24].

```
Double Dato10 = Math.Truncate((((Convert.ToDouble(Dato1) * Convert.ToDouble(Dato4) * Convert.ToDouble(Dato7)) + ...  
    (Convert.ToDouble(Dato2) * Convert.ToDouble(Dato5) * Convert.ToDouble(Dato8)) + (Convert.ToDouble(Dato3) * ...  
    Convert.ToDouble(Dato6) * Convert.ToDouble(Dato9)))) / 1000) * 100) / 100;  
Double Dato11 = Math.Truncate((((Convert.ToDouble(Dato1) * Convert.ToDouble(Dato4)) + (Convert.ToDouble(Dato2) * ...  
    Convert.ToDouble(Dato5)) + (Convert.ToDouble(Dato3) * Convert.ToDouble(Dato6)))) / 1000) * 100) / 100;  
Double Dato12 = Math.Truncate(Math.Sqrt((Dato10 * Dato10) + (Dato11 * Dato11)) * 100) / 100;  
Double Dato13 = Math.Truncate(Dato10 / Dato11 * 100) / 100;
```

Figura 3.43. Código fuente de la ventana de simulación energética, parte 7.



En la Figura 3.44 se muestra la sentencia para el “Dato14” que corresponde al cálculo del factor de carga, que por [6], se obtiene multiplicando la potencia eléctrica por la eficiencia nominal y dividiéndose a su vez entre el producto de la potencia en HP por 0.746 que es la equivalencia en kW. Por lo anteriormente comentado se deduce que “textBox10.Text” es el campo editable de Eff1 a factor de carga 1.00 y “textBox7.Text” es el campo editable de Potencia Nom (HP) de la sección de datos del motor actual.

```
Double Dato14 = Math.Truncate(((Dato10 * (Convert.ToDouble(textBox10.Text))) / ((Convert.ToDouble(textBox7.Text)) * 0.746)) ...  
* 100) / 100;
```

Figura 3.44. Código fuente de la ventana de simulación energética, parte 8.

En la Figura 3.45 se muestra el código para llevar a cabo una interpolación de eficiencias, que con el dato del factor de carga al cual se encuentra trabajando el motor se estime la eficiencia de acuerdo a los datos introducidos en los campos editables de eficiencia a distintos factores de carga de la sección de datos del motor actual, por lo que la variable “Dato15” corresponde al ajuste de eficiencia de acuerdo al factor de carga calculado previamente.

En la Figura 3.46 se muestra una serie de sentencias para llevar a cabo el ajuste final de la eficiencia a la cual trabaja el motor, considerándose tres criterios: ajuste por variación de tensión, ajuste por desbalance de tensión y ajuste por rebobinados del motor, tal como se sugiere en [6], siendo las variables “FAvt”, “FAdt” y “FArem” las variables para el Factor de Ajuste por variación de tensión, Factor de Ajuste por desbalance de tensión y Factor de Ajuste por re-embobinados del motor, respectivamente. La variable “Dato16” es entonces el valor de eficiencia ajustada a la cual trabaja el motor evaluado.



```
Double y0 = Convert.ToDouble(textBox10.Text);
Double y1 = Convert.ToDouble(textBox11.Text);
Double y2 = Convert.ToDouble(textBox12.Text);
Double y3 = Convert.ToDouble(textBox13.Text);
Double x0 = Convert.ToDouble(textBox14.Text);
Double x1 = Convert.ToDouble(textBox15.Text);
Double x2 = Convert.ToDouble(textBox16.Text);
Double x3 = Convert.ToDouble(textBox17.Text);
Double[] x = { x3, x2, x1, x0 };
Double[] y = { y3, y2, y1, y0 };
Double[] l = new double[4];
Double Dato15 = 0;
for (int i = 0; i < 4; i = i + 1)
{
    l[i] = 1;
    for (int j = 0; j < 4; j = j + 1)
    {
        if (i != j)
        {
            l[i] = l[i] * ((Dato14 - x[j]) / (x[i] - x[j]));
        }
    }
    Dato15 = Math.Truncate((Dato15 + l[i] * y[i]) * 100) / 100;
}
```

Figura 3.45. Código fuente de la ventana de simulación energética, parte 9.

```
Double TPromStd = (Convert.ToDouble(Dato1) + Convert.ToDouble(Dato2) + Convert.ToDouble(Dato3)) / 3 * Math.Sqrt(3);
Double VTstd = (TPromStd / Convert.ToDouble(textBox8.Text)) - 1;
Double FAvt = VTstd * (0.07 - (1.334 * VTstd)) - 0.0009;
Double DTa = Math.Abs((Convert.ToDouble(Dato1) * Math.Sqrt(3)) - TPromStd);
Double DTb = Math.Abs((Convert.ToDouble(Dato2) * Math.Sqrt(3)) - TPromStd);
Double DTc = Math.Abs((Convert.ToDouble(Dato3) * Math.Sqrt(3)) - TPromStd);
Double[] DT = { DTa, DTb, DTc };
Double maxDT = DT.Max();
Double DTstd = maxDT / TPromStd;
Double FAdt = 1 - DTstd * (0.0113 + 0.0073 * DTstd);
Double FArem = Convert.ToDouble(textBox18.Text);
Double Dato16 = Math.Truncate((FAdt * (Dato15 - FAvt - FArem)) * 100) / 100;
```

Figura 3.46. Código fuente de la ventana de simulación energética, parte 10.



En la Figura 3.47 se muestra el código para determinar la potencia en la flecha (variable “Dato17”), misma que es la demanda mecánica del motor actual y la misma que se demandará al motor de alta eficiencia propuesto. A partir de este punto, se realizan los mismos cálculos que los anteriores, pero en sentido inverso a partir de la potencia mecánica que demanda la maquinaria impulsada por el motor actual, a fin de determinar la nueva demanda eléctrica para el motor de alta eficiencia propuesto (variable “Dato18”).

```
Double Dato17 = Math.Truncate(((Dato16 * Dato10 / 0.746) * 100) / 100;
Double MotPropAE = Math.Truncate((Dato17 / 0.75) * 100) / 100;
label16.Text = MotPropAE.ToString();
Double FCae = Dato17 / Convert.ToDouble(textBox30.Text);
Double Y0 = Convert.ToDouble(textBox20.Text);
Double Y1 = Convert.ToDouble(textBox21.Text);
Double Y2 = Convert.ToDouble(textBox22.Text);
Double Y3 = Convert.ToDouble(textBox23.Text);
Double X0 = Convert.ToDouble(textBox24.Text);
Double X1 = Convert.ToDouble(textBox25.Text);
Double X2 = Convert.ToDouble(textBox26.Text);
Double X3 = Convert.ToDouble(textBox27.Text);
Double[] X = { X3, X2, X1, X0 };
Double[] Y = { Y3, Y2, Y1, Y0 };
Double[] L = new double[4];
Double EffFC = 0;
for (int a = 0; a < 4; a = a + 1)
{
    L[a] = 1;
    for (int b = 0; b < 4; b = b + 1)
    {
        if (a != b)
        {
            L[a] = L[a] * ((FCae - X[b]) / (X[a] - X[b]));
        }
    }
    EffFC = Math.Truncate((EffFC + L[a] * Y[a]) * 100) / 100;
}
Double VTae = (TPromStd / Convert.ToDouble(textBox29.Text)) - 1;
Double FAvtAE = VTae * (0.07 - (1.334 * VTae)) - 0.0009;
Double EffFCaj = Math.Truncate((FAdt * (EffFC - FAvtAE)) * 100) / 100;
Double Dato18 = Math.Truncate((Convert.ToDouble(textBox30.Text) * FCae * 0.746 / EffFCaj) * 100) / 100;
```

Figura 3.47. Código fuente de la ventana de simulación energética, parte 11.



La serie de cálculos llevados a cabo en cada ciclo del timer finaliza con las sentencias mostradas en la Figura 3.48, donde se determina el ahorro en demanda (variable "Dato19") restándose la potencia eléctrica demandada del motor actual menos la potencia eléctrica demandada por el motor de alta eficiencia propuesto, así como el ahorro en consumo eléctrico (variable "Dato20") considerando que el muestreo es cada 10 segundos y el consumo acumulado (variable "Dato21"). Aquí se aprecia también la sentencia para determinar la tasa de ahorro, donde simplemente se divide la potencia eléctrica del motor propuesto entre la potencia eléctrica del motor actual, multiplicado por 100 para expresarse en por ciento.

```
Double Dato19 = Math.Truncate((Dato10 - Dato18) * 100) / 100;  
Double Dato20 = Math.Truncate((Dato19 * 10 / 3600) * 10000000) / 10000000;  
Double tasa = Math.Truncate((Dato19 / Dato10 * 100) * 100) / 100;  
label35.Text = tasa.ToString();  
Dato21 = Dato21 + Math.Truncate((Dato20) * 10000000) / 10000000;
```

Figura 3.48. Código fuente de la ventana de simulación energética, parte 12.

La programación orientada a procedimientos termina con las sentencias mostradas en la Figura 3.49, donde simplemente se llevan los datos provenientes del Hardware y los datos calculados en el Software a una hoja de Excel, donde se irán incrementando las filas de datos en cada ciclo del timer.



```
linea = dataGridView1.Rows.Add();
dataGridView1.Rows[linea].Cells[0].Value = lineaNo;
dataGridView1.Rows[linea].Cells[1].Value = dateTime.ToShortDateString();
dataGridView1.Rows[linea].Cells[2].Value = dateTime.ToLongTimeString();
dataGridView1.Rows[linea].Cells[3].Value = Dato1;
dataGridView1.Rows[linea].Cells[4].Value = Dato2;
dataGridView1.Rows[linea].Cells[5].Value = Dato3;
dataGridView1.Rows[linea].Cells[6].Value = Dato4;
dataGridView1.Rows[linea].Cells[7].Value = Dato5;
dataGridView1.Rows[linea].Cells[8].Value = Dato6;
dataGridView1.Rows[linea].Cells[9].Value = Dato7;
dataGridView1.Rows[linea].Cells[10].Value = Dato8;
dataGridView1.Rows[linea].Cells[11].Value = Dato9;
dataGridView1.Rows[linea].Cells[12].Value = Dato10;
dataGridView1.Rows[linea].Cells[13].Value = Dato11;
dataGridView1.Rows[linea].Cells[14].Value = Dato12;
dataGridView1.Rows[linea].Cells[15].Value = Dato13;
dataGridView1.Rows[linea].Cells[16].Value = Dato14;
dataGridView1.Rows[linea].Cells[17].Value = Dato15;
dataGridView1.Rows[linea].Cells[18].Value = Dato16;
dataGridView1.Rows[linea].Cells[19].Value = Dato17;
dataGridView1.Rows[linea].Cells[20].Value = Dato18;
dataGridView1.Rows[linea].Cells[21].Value = Dato19;
dataGridView1.Rows[linea].Cells[22].Value = Dato20;
dataGridView1.Rows[linea].Cells[23].Value = Dato21;
linea++;
lineaNo++;
dataGridView1.FirstDisplayedScrollingRowIndex = dataGridView1.RowCount - 1;
}
```

Figura 3.49. Código fuente de la ventana de simulación energética, parte 13.



CAPITULO 4. APLICACIÓN Y CASO DE ESTUDIO

En este Capítulo se muestran los resultados obtenidos al aplicar el Simulador de Eficiencia Energética desarrollado en un entorno real, con la finalidad de verificar su funcionalidad. Su desempeño se probó en una empresa ubicada en la Ciudad de Cuernavaca, en una máquina industrial impulsada por un motor de eficiencia estándar de 15 HP. A continuación, se describe el procedimiento llevado a cabo.

Procedimiento

Paso No.1

Se tramitó el permiso de alto riesgo, correspondiente a trabajos eléctricos, con la finalidad de demostrar que el simulador trabaja bajo un método no invasivo que permite llevar a cabo la simulación sin la necesidad de sacar de operación la máquina. Posteriormente, se acordonó el área de trabajo para evitar el ingreso de personal no calificado.

Paso No. 2

Se conectaron los sensores de tensión y corriente a las entradas correspondientes, tanto del Hardware del simulador como del analizador de redes utilizado de apoyo, así como el cable de comunicación serial a la computadora del Software, tal como se aprecia en la Figura 4.1.

Paso No. 3

Con el Equipo de Protección Personal (EPP) apropiado, se abrieron los tableros eléctricos correspondientes y se llevó a cabo la conexión de los sensores de tensión y corriente en las terminales del arrancador del motor, tanto del Hardware del simulador como del analizador (ver Figura 4.1.). Gracias al método no invasivo del Hardware al utilizar sensores de corriente de núcleo abierto, no fue necesario sacar de operación la máquina para llevar a cabo las conexiones.




Figura 4.1. Conexión de sensores en tableros eléctricos energizados.

Paso No. 4

Se inicializó el Software y se estableció la comunicación serial entre el Hardware y el Software.

Paso No. 5

Con la ayuda de la tabla mostrada en el APÉNDICE, se parametrizaron los datos de entrada como se observa en la Figura 4.2, introduciendo los datos de placa del motor a evaluar y los del motor de alta eficiencia propuesto.

 SIMULACIÓN ENERGÉTICA

DATOS DEL MOTOR ACTUAL

Potencia Nom (HP)	<input type="text" value="15"/>
Tensión Nom (V)	<input type="text" value="460"/>
F.P.	<input type="text" value="0.60"/>
Eff 1	<input type="text" value="0.8815"/> @ <input type="text" value="1.00"/> F.C.
Eff 2	<input type="text" value="0.8943"/> @ <input type="text" value="0.75"/> F.C.
Eff 3	<input type="text" value="0.8902"/> @ <input type="text" value="0.50"/> F.C.
Eff 4	<input type="text" value="0.8477"/> @ <input type="text" value="0.25"/> F.C.
Aj Eff por reembobinado	<input type="text" value="0.00"/>
MOTOR AE PROPUESTO (HP)	...

DATOS DEL MOTOR PROPUESTO

Potencia Nom (HP)	<input type="text" value="7.5"/>
Tensión Nom (V)	<input type="text" value="460"/>
F.P.	<input type="text" value="0.90"/>
Eff 1	<input type="text" value="0.9060"/> @ <input type="text" value="1.00"/> F.C.
Eff 2	<input type="text" value="0.9088"/> @ <input type="text" value="0.75"/> F.C.
Eff 3	<input type="text" value="0.9019"/> @ <input type="text" value="0.50"/> F.C.
Eff 4	<input type="text" value="0.8431"/> @ <input type="text" value="0.25"/> F.C.
TASA DE AHORRO (%)	...

Figura 4.2. Parametrización de datos de entrada en el Software.

Cabe mencionar que de inicio se introdujeron los datos del motor de alta eficiencia propuesto considerando una capacidad nominal igual a la del motor evaluado; esto con la finalidad de determinar el factor de carga actual y verificar la capacidad del motor que propone el Software del simulador.

Paso 6.

Se inicializó la simulación y el Software comenzó a recibir, procesar y almacenar los datos recibidos del Hardware por medio de la comunicación serial establecida, con una velocidad de 1 muestreo cada 10 segundos. En la Figura 4.3 se muestra el área de gráficos y los paneles de instrumentos de tensión y corriente, donde en el gráfico se visualizan los datos históricos y en los paneles de instrumentos se visualizan los últimos datos muestreados.

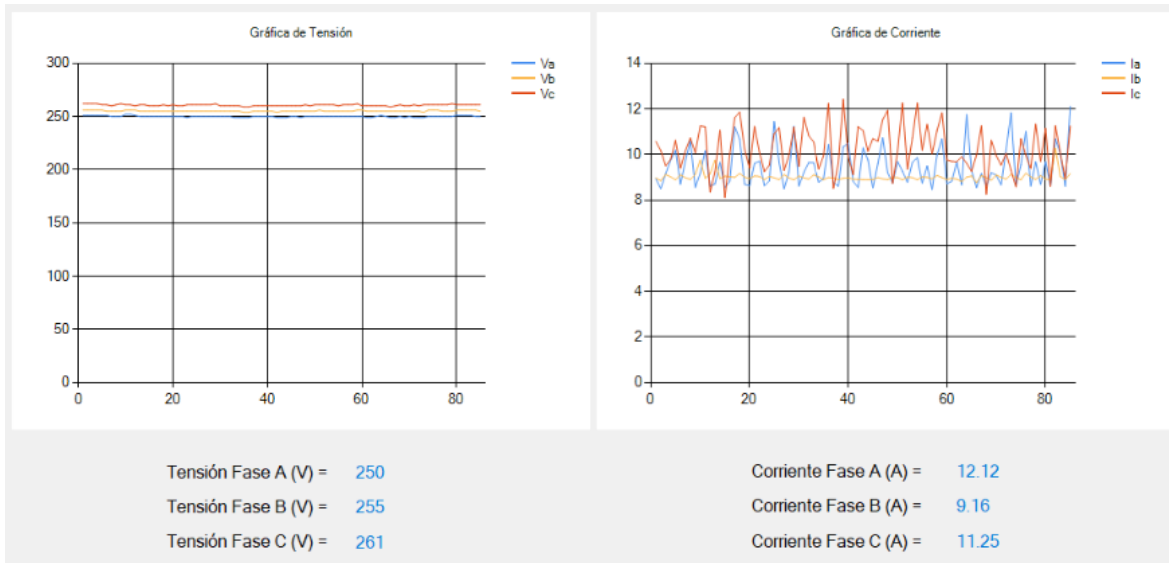


Figura 4.3. Gráficos y paneles de instrumentos del Software.

Paso 7.

Se verificó la potencia del motor de alta eficiencia que propone el simulador en función del factor de carga calculado por el Software, encontrándose que el motor actual está sobredimensionado, requiriéndose en la flecha una potencia de 7.04 HP, seleccionándose por tanto el motor de alta eficiencia con el valor de potencia estándar inmediato superior, en este caso, 7.5 HP (ver Figura 4.4).

DATOS DEL MOTOR ACTUAL

Potencia Nom (HP)

Tensión Nom (V)

F.P.

Eff 1 @ F.C.

Eff 2 @ F.C.

Eff 3 @ F.C.

Eff 4 @ F.C.

Aj Eff por rebobinado

MOTOR AE PROPUESTO (HP) 7.04

DATOS DEL MOTOR PROPUESTO

Potencia Nom (HP)

Tensión Nom (V)

F.P.

Eff 1 @ F.C.

Eff 2 @ F.C.

Eff 3 @ F.C.

Eff 4 @ F.C.

TASA DE AHORRO (%) 4.52

Figura 4.4. Resultados arrojados por el Software (valores en azul).

Paso 8.

Finalmente, se llevó a cabo la exportación de datos a Excel, mismos que se van almacenando en la tabla del Software como se muestra en la Figura 4.5.

FC	Eff a FC	Eff Aj a FC	Pot_Mec_Std	Dem Mot Prop	Ahorro en Demanda (kW)	Ahorro en Consumo (kWh)	Ahorro Acum (kWh)
0.31	0.84	0.84	4.53	3.75	0.28	0.0007777	0.0616065

Figura 4.5. Datos almacenados en la Tabla del Software.



Análisis de resultados

El Software del simulador determinó que el motor actual trabaja a un factor de carga de 31% y a una eficiencia ajustada de 84% (ver Figura 4.5), proponiendo por tanto un motor de alta eficiencia de 7.04 HP (ver Figura 4.4) para que trabaje a un régimen de carga cercano al 80% como se recomienda.

Así mismo arroja, como resultado de la acción de reemplazar el motor de eficiencia estándar de 15 HP actualmente instalado en la máquina por un motor de alta eficiencia de 7.5 HP, una tasa de ahorro de 4.52% (ver Figura 4.4), lo que representa un ahorro en demanda de 0.25 kW.

Con los datos proporcionados por el personal de producción, donde dicha máquina opera alrededor de 7,000 horas al año, se calcula que esta acción de reemplazo del motor representa un ahorro anual de 1,743 kWh.

Con dichos datos puede iniciarse el análisis económico correspondiente, donde se debe tomar en cuenta la tarifa eléctrica correspondiente y el costo de la inversión, con la finalidad de encontrar el ahorro económico y el periodo de recuperación de la inversión. Como se señaló en el Capítulo 3, esto puede incluirse en el Software del simulador en una versión futura.

La Figura 4.6 se muestra como evidencia de la prueba de desempeño del simulador llevada a cabo.



Figura 4.6. Ejecutando la prueba de desempeño del simulador en campo.

Comparación de las mediciones del Hardware del simulador con las mediciones de un Analizador de redes

Con la finalidad de comprobar la incertidumbre de las mediciones hechas por el Hardware del simulador, se utilizó al mismo tiempo que se llevó a cabo la prueba, el Analizador de Redes Dranetz Power Visa, registrando las mediciones mostradas en la Figura 4.7 para tensiones y corriente y en la Figura 4.8 para potencias y factor de potencia.

En la Tabla 4.1 se realiza el comparativo de las mediciones llevadas a cabo tanto por el Hardware del simulador como por el Analizador de redes, y se calculan los correspondientes errores absolutos y relativos.

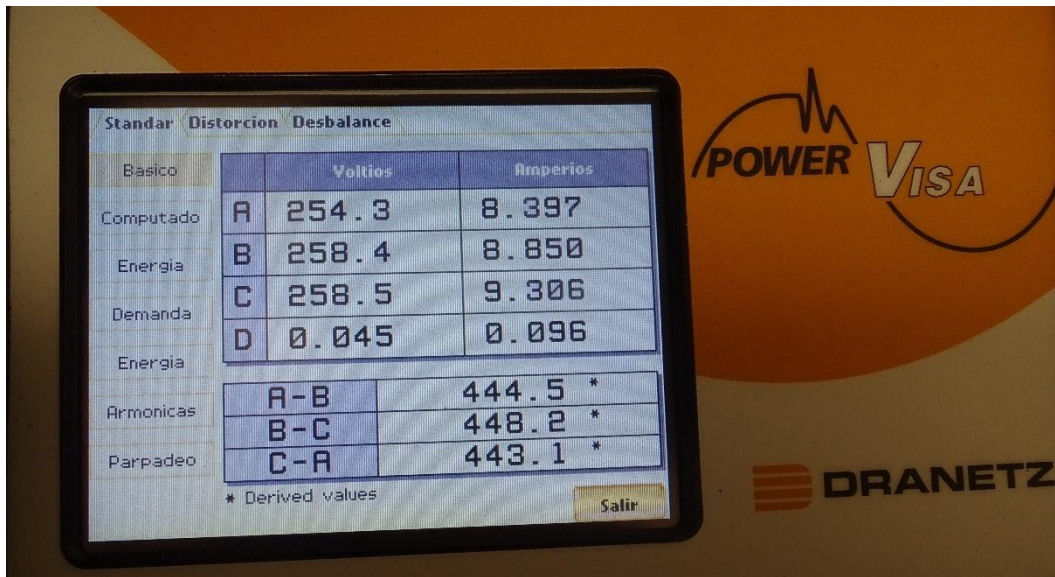


Figura 4.7. Lecturas de tensiones y corrientes del Analizador de Redes.

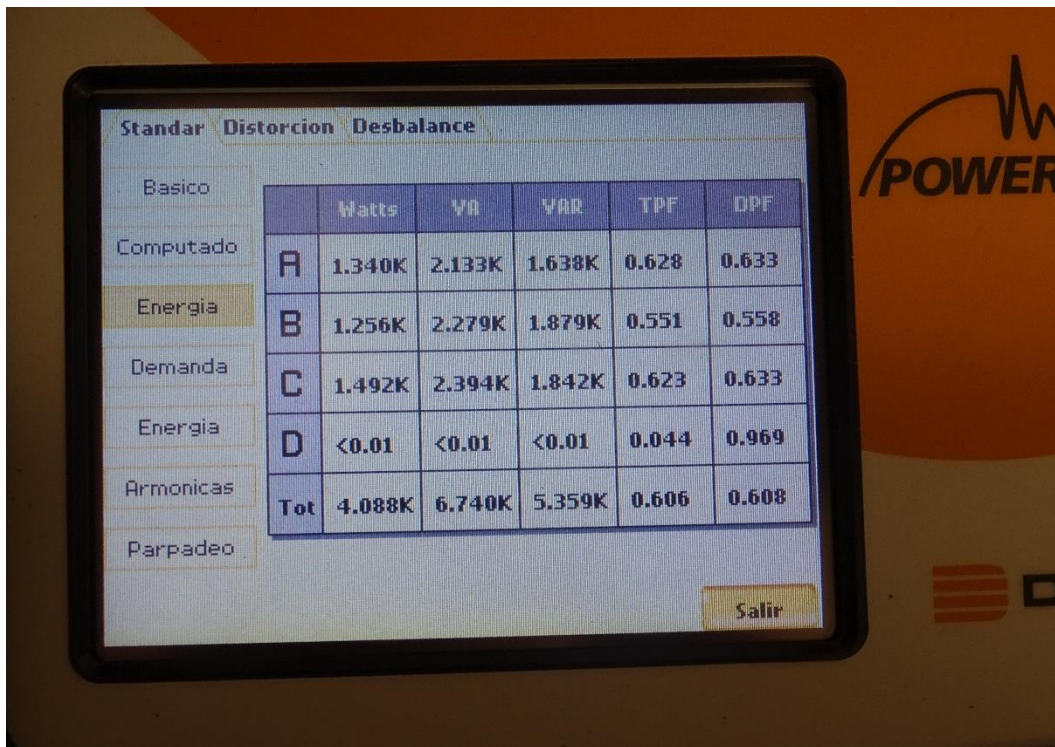


Figura 4.8. Lecturas de potencias y factor de potencia del Analizador de Redes.



Tabla 4.1. Comparativo de mediciones y cálculo de errores absolutos y relativos.

HORA	V_a	V_b	V_c	I_a	I_b	I_c	FP_a	FP_b	FP_c
Mediciones del Hardware	250.0	255.2	260.7	9.5	9.0	10.1	0.6	0.6	0.6
Mediciones del Analizador de redes	254.3	258.4	258.5	8.397	8.85	9.306	0.62	0.55	0.62
error absoluto	4.3	3.2	2.2	1.1	0.2	0.8	0.0	0.0	0.0
error relativo	1.7	1.2	0.8	12.8	1.7	8.8	6.3	5.7	6.3



CAPITULO 5. CONCLUSIONES Y RECOMENDACIONES

Se desarrolló un simulador de eficiencia energética de motores de inducción, compuesto por un Hardware, para el cual se utilizó como elemento principal una placa de Arduino DUE y un Software, que se desarrolló en el entorno de programación Visual Studio 2019.

Con el desarrollo del Hardware se logró la obtención de parámetros eléctricos del motor de eficiencia estándar a evaluar.

Con el desarrollo del Software se logró recibir la información del Hardware mediante una comunicación serial y procesarla para calcular el factor de carga, la eficiencia a la cual opera el motor evaluado y su consumo, así como calcular el consumo de un motor propuesto de alta eficiencia, obteniéndose la estimación del ahorro energético por la sustitución del motor evaluado por el motor propuesto.

Los resultados del simulador muestran que éste puede ser una herramienta de apoyo para la evaluación técnica-económica de proyectos de sustitución de motores, probando su desempeño en un motor de 15 HP de una máquina troqueladora de una empresa ubicada en la ciudad de Cuernavaca, Morelos. Los resultados del simulador evidenciaron que el motor evaluado trabaja a un factor de carga del 31% y a una eficiencia del 84%, proponiendo el simulador un motor de alta eficiencia de reemplazo de 7.5 HP, estimándose con esta acción un ahorro energético de 1,743 kWh al año. Se comprobó la incertidumbre de la medición del Hardware del simulador, utilizando en la prueba de desempeño mencionada el analizador de redes Dranetz Power Visa, obteniéndose los siguientes errores relativos máximos: 1.7% en la medición de tensión, 12.8% en la medición de corriente y 6.2% en la medición del factor de potencia.

El simulador puede mejorar en sus dos componentes: respecto al Hardware, se puede utilizar un controlador mas robusto como la tarjeta de desarrollo Fpga Altera De2-115, con la finalidad de determinar si con su entorno de programación se pueden reducir los errores indicados en la Tabla 4.1. Por otro lado, en la actualidad



es muy importante que los desarrollos entren dentro de las regulaciones establecidas, como el estándar IEC 61000-4-30 para equipos de medición, siendo ampliamente recomendable que se tomara en cuenta este estándar para lograr con ello un Hardware de medición Clase A. Respecto al Software, se debe complementar con el desarrollo de la parte económica, a fin de facilitar aun más la evaluación de proyectos de eficiencia energética por sustitución de motores eléctricos.

Finalmente, el desempeño real del presente desarrollo se puede verificar llevando a la práctica la sustitución del motor estándar evaluado por el motor de alta eficiencia propuesto, donde la simulación energética realizada debe corresponder con las mediciones de consumo correspondientes.



REFERENCIAS

- [A. López Miranda y D. Lankenau Caballero, Administración de proyectos, la clave para la
1 coordinación efectiva de actividades y recursos, Ciudad de México: Pearson Educación de
] México, S.A. de C.V., 2017.
- [S. d. T. y. P. Social, «Diario Oficial de la Federación,» [En línea]. Available:
2 http://dof.gob.mx/nota_detalle.php?codigo=5227363&fecha=29/12/2011. [Último acceso: 19
] 12 2020].
- [G. J. Bronson, C++ para ingeniería y ciencias, Ciudad de México: Thomson, 2007.
3
]
- [Arduino, «Arduino,» [En línea]. Available: <https://store.arduino.cc/usa/duemil>. [Último acceso: 19
4 12 2020].
]
- [Microsoft, «Microsoft,» [En línea]. Available: [https://docs.microsoft.com/es-
5 es/dotnet/csharp/programming-guide/](https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/). [Último acceso: 19 12 2020].
]
- [M. Fideicomiso para el Ahorro de Energía Eléctrica y G. Comisión Nacional de Energía Eléctrica,
6 «Promotores de ahorro y eficiencia de energía eléctrica,» Guatemala, 2010.
]
- [IEA, «IEA,» [En línea]. Available: <https://www.iea.org>. [Último acceso: 19 12 2020].
7
]
- [IEA, «IEA,» [En línea]. Available: [https://www.iea.org/data-and-
8 statistics?country=WORLD&fuel=Energy%20consumption&indicator=TotElecCons](https://www.iea.org/data-and-statistics?country=WORLD&fuel=Energy%20consumption&indicator=TotElecCons). [Último
] acceso: 19 12 2020].
- [IEA, «IES,» [En línea]. Available: [https://www.iea.org/data-and-
9 statistics?country=WORLD&fuel=Energy%20supply&indicator=TPESbySource](https://www.iea.org/data-and-statistics?country=WORLD&fuel=Energy%20supply&indicator=TPESbySource). [Último acceso:
] 19 12 2020].
- [C. N. p. e. A. d. Energía, «Diario Oficial de la Federación,» [En línea]. Available:
1 [http://dof.gob.mx/nota_detalle.php?codigo=4919668&fecha=19/04/2005#:~:text=Esta%20No
0 rma%20Oficial%20Mexicana%20establece,inducci%C3%B3n%2C%20tipo%20jaula%20de%20ar
\] dilla%2C](http://dof.gob.mx/nota_detalle.php?codigo=4919668&fecha=19/04/2005#:~:text=Esta%20Norma%20Oficial%20Mexicana%20establece,inducci%C3%B3n%2C%20tipo%20jaula%20de%20ardilla%2C). [Último acceso: 19 12 2020].



- [C. N. p. e. U. E. d. I. Energía, «Diario Oficial de la Federación,» [En línea]. Available:
1 http://www.dof.gob.mx/nota_detalle.php?codigo=5460981&fecha=15/11/2016. [Último
1 acceso: 19 12 2020].
]
- [C. N. p. e. U. E. d. I. Energía, «CONUEE,» [En línea]. Available:
1 <https://www.gob.mx/conuee/acciones-y-programas/herramientas-y-aplicaciones-conuee>.
2 [Último acceso: 19 12 2020].
]
- [WEG, «WEG,» [En línea]. Available:
1 http://ecatalog.weg.net/tec_cat/retornoinvestmotor_web.asp?cd_mercado=000U. [Último
3 acceso: 19 12 2020].
]
- [PDMA, «PDMA,» [En línea]. Available: <https://pdma.com/about-PdMA.php>. [Último acceso: 19
1 12 2020].
4
]
- [E. C. Quispe y L. F. Mantilla, «ResearchGate,» 12 2004. [En línea]. Available:
1 https://www.researchgate.net/publication/277105627_Motores_electricos_de_alta_eficiencia_5_caracteristicas_electromecanicas_ventajas_y_aplicabilidad. [Último acceso: 15 12 2019].
5
]
- [IEEE, *IEEE Standard Test Procedure for Polyphase Induction Motors and Generators*, New York,
1 USA, 2004.
6
]
- [F. p. e. A. d. E. Eléctrica, «Procedimiento para la evaluación Energética de Motores Eléctricos,»
1 2002.
7
]
- [WEG, «WEG,» [En línea]. Available: <https://www.weg.net/catalog/weg/US/es/Motores-El%C3%A9ctricos/Motor-de-Inducc%C3%ADon---NEMA/Usos-General-ODP-TEFC/ODP-Hierro-Gris/W40-/General-High-Efficiency-15-HP-4P-254T-3F-230-460-V-60-Hz-IC01---ODP---Con-pies/p/10360239>. [Último acceso: 15 12 2020].
8
]
- [J. Policarpo, G. de Abreu y A. Eigeles, «Induction Motor Thermal Aging Caused by Voltage
1 Distortion and Imbalance: Loss of Useful Life and Its Estimated Cost,» *IEEE Transactions on*
9 *Industry Applications*, vol. 38, nº 1, 2002.
]



[Electrical Apparatus Service Association, Inc. y Association of Electrical and Mechanical Trades,
2 «El Efecto de la Reparación/Rebobinado en la Eficiencia del Motor,» 2003.

0

]

[R. Boylestad y L. Nashelsky, *Electrónica: Teoría de circuitos y dispositivos electrónicos*,
2 Naucalpan de Juárez: Pearson Educación de México, S.A. de C.V., 2009.

1

]

[W. Hayt Jr., J. Kemmerly y S. Durbin, *Análisis de circuitos en ingeniería*, Ciudad de México:
2 McGraw-Hill, 2007.

2

]

[R. Tocci y N. Widmer, *Sistemas digitales, principios y aplicaciones*, Ciudad de México: Pearson
2 Educación, 2003.

3

]

[J. Grainger y W. Stevenson Jr., *Análisis de sistemas eléctricos de potencia*, Naucalpan de
2 Juárez: McGraw-Hill, 1996.

4

]

[N. T. JA, «Youtube,» [En línea]. Available: <https://www.youtube.com/watch?v=iPepH8Co4YI>.
2 [Último acceso: 19 12 2020].

5

]

[edx, «iea,» [En línea]. Available: <https://www.iea.org/countries/mexico>. [Último acceso: 15 02
2 2020].

6

]

[D. Valdivia, «Motores de alta eficiencia,» *Electro Industria*, 2005.

2

7

]



APÉNDICE: TABLAS DE EFICIENCIA DE MOTORES ESTÁNDAR Y DE ALTA EFICIENCIA. FUENTE: [17].

Pot (hp)	Tipo	Carcaza	Polos	rpm	Ef 0.25	Ef 0.50	Ef 0.75	Ef 1.0	\$ USD
5	STD	abierto	2	3600	0.7615	0.8247	0.8374	0.8270	259.74
			4	1800	0.7435	0.8152	0.8422	0.8320	255.81
			6	1200	0.7735	0.8118	0.8285	0.8383	389.54
			8	900	0.7020	0.8003	0.8267	0.8218	704.44
		cerrado	2	3600	0.7633	0.8424	0.8608	0.8575	323.35
			4	1800	0.7840	0.8528	0.8633	0.8549	292.05
			6	1200	0.7674	0.8204	0.8443	0.8470	520.09
			8	900	0.7230	0.8090	0.8365	0.8350	1,027.65
	AE	abierto	2	3600	0.8080	0.8637	0.8797	0.8753	421.20
			4	1800	0.8358	0.8927	0.8963	0.8898	390.98
			6	1200	0.8266	0.8793	0.8900	0.8891	500.69
			8	900	0.8170	0.8845	0.8965	0.8900	867.53
		cerrado	2	3600	0.8267	0.8811	0.8938	0.8894	466.61
			4	1800	0.8429	0.8963	0.9055	0.8959	427.51
			6	1200	0.8375	0.8918	0.9014	0.8931	676.06
			8	900	0.8130	0.8828	0.8945	0.8900	917.31
7.5	STD	abierto	2	3600	0.8090	0.8563	0.8616	0.8483	323.58
			4	1800	0.8044	0.8523	0.8588	0.8492	328.05
			6	1200	0.7675	0.8460	0.8570	0.8488	601.33
			8	900	0.7430	0.8358	0.8506	0.8498	906.86
		cerrado	2	3600	0.7933	0.8619	0.8758	0.8586	433.72
			4	1800	0.7985	0.8679	0.8783	0.8722	438.13
			6	1200	0.7941	0.8449	0.8595	0.8567	729.14
			8	900	0.7420	0.8332	0.8743	0.8329	1,155.98
	AE	abierto	2	3600	0.8630	0.9066	0.9057	0.8897	531.52
			4	1800	0.8431	0.9019	0.9088	0.9060	459.55
			6	1200	0.8362	0.8933	0.9034	0.9029	702.41
			8	900	0.8360	0.8945	0.9010	0.8940	1,047.37
		cerrado	2	3600	0.8495	0.8993	0.9080	0.9024	566.00
			4	1800	0.8561	0.9077	0.9150	0.9102	577.99
			6	1200	0.8482	0.9017	0.9111	0.9078	924.33
			8	900	0.8294	0.8867	0.8960	0.8857	1,388.40
10	STD	abierto	2	3600	0.8123	0.8649	0.8783	0.8684	336.32
			4	1800	0.8181	0.8653	0.8696	0.8539	394.32
			6	1200	0.8513	0.8646	0.8808	0.8747	663.30
			8	900	0.7950	0.8562	0.8676	0.8570	1,021.28
		cerrado	2	3600	0.8328	0.8866	0.8873	0.8680	533.50
			4	1800	0.8274	0.8780	0.8843	0.8754	516.29
			6	1200	0.8035	0.8613	0.8741	0.8689	888.81
			8	900	0.7980	0.8630	0.8735	0.8650	1,513.53
	AE	abierto	2	3600	0.8345	0.8980	0.9054	0.8995	523.75
			4	1800	0.8697	0.9137	0.9162	0.9083	533.10
			6	1200	0.8633	0.9130	0.9187	0.9140	920.04
			8	900	0.8585	0.9053	0.9113	0.9050	1,223.95
		cerrado	2	3600	0.8723	0.9119	0.9177	0.9110	687.38
			4	1800	0.8758	0.9142	0.9196	0.9092	658.21
			6	1200	0.8634	0.9066	0.9140	0.9094	1,102.71
			8	900	0.8333	0.8989	0.9069	0.9008	1,717.37
15	STD	abierto	2	3600	0.8605	0.8886	0.8919	0.8726	532.03
			4	1800	0.8477	0.8902	0.8943	0.8815	534.83
			6	1200	0.7973	0.8643	0.8731	0.8675	810.49
			8	900	0.8135	0.8708	0.8764	0.8600	1,314.82
		cerrado	2	3600	0.8219	0.8728	0.8878	0.8770	662.58
			4	1800	0.8028	0.8713	0.8853	0.8790	746.10
			6	1200	0.7954	0.8578	0.8761	0.8763	1,173.98
			8	900	0.7683	0.8785	0.8815	0.8660	1,851.85
	AE	abierto	2	3600	0.8754	0.9136	0.9149	0.9049	604.68
			4	1800	0.8728	0.9261	0.9298	0.9253	763.59
			6	1200	0.8634	0.9134	0.9199	0.9186	1,079.71
			8	900	0.8665	0.9100	0.9130	0.9050	1,560.87
		cerrado	2	3600	0.8691	0.9119	0.9206	0.9159	903.84
			4	1800	0.8964	0.9236	0.9276	0.9243	865.01
			6	1200	0.8738	0.9155	0.9220	0.9258	1,451.13
			8	900	0.8488	0.9066	0.9102	0.9008	2,130.05
20	STD	abierto	2	3600	0.8736	0.9002	0.9037	0.8886	560.07
			4	1800	0.8339	0.8857	0.8888	0.8789	649.26
			6	1200	0.8606	0.8902	0.8945	0.8844	1,004.07
			8	900	0.8045	0.8783	0.8883	0.8800	1,542.78
		cerrado	2	3600	0.8140	0.8798	0.8934	0.8981	811.25
			4	1800	0.8294	0.8876	0.8970	0.8880	919.57
			6	1200	0.8181	0.8849	0.8912	0.8841	1,386.78
			8	900	0.8493	0.8830	0.8908	0.8818	2,082.49
	AE	abierto	2	3600	0.8754	0.9100	0.9137	0.9004	699.80
			4	1800	0.8926	0.9295	0.9314	0.9216	825.67
			6	1200	0.8827	0.9239	0.9272	0.9202	1,330.42
			8	900	0.8870	0.9210	0.9230	0.9144	1,858.19
		cerrado	2	3600	0.8898	0.9208	0.9277	0.9195	1,134.27
			4	1800	0.8964	0.9301	0.9341	0.9280	1,045.60
			6	1200	0.8857	0.9242	0.9269	0.9178	1,733.34
			8	900	0.8800	0.9164	0.9201	0.9088	2,652.33



Pot (hp)	Tipo	Carcaza	Polos	rpm	Ef 0.25	Ef 0.50	Ef 0.75	Ef 1.0	\$ USD
25	STD	abierto	2	3600	0.8510	0.8950	0.9017	0.8886	720.77
			4	1800	0.8714	0.9056	0.9085	0.8932	756.75
			6	1200	0.8544	0.8932	0.8990	0.8913	1,205.28
			8	900	0.8430	0.8818	0.8913	0.8825	1,800.50
		cerrado	2	3600	0.8406	0.8907	0.9021	0.8883	955.50
			4	1800	0.8691	0.9121	0.9171	0.9055	1,073.55
			6	1200	0.8411	0.8901	0.8994	0.8915	1,630.81
			8	900	0.7890	0.8720	0.8850	0.8817	2,516.80
	AE	abierto	2	3600	0.8804	0.9271	0.9305	0.9264	966.96
			4	1800	0.8919	0.9357	0.9387	0.9331	1,011.44
			6	1200	0.8910	0.9350	0.9372	0.9281	1,548.08
			8	900	0.8905	0.9237	0.9260	0.9147	2,182.27
		cerrado	2	3600	0.8835	0.9220	0.9292	0.9234	1,334.00
			4	1800	0.9109	0.9372	0.9401	0.9352	1,254.78
			6	1200	0.9050	0.9348	0.9366	0.9274	2,183.35
			8	900	0.8833	0.9195	0.9216	0.9084	2,946.94
30	STD	abierto	2	3600	0.8651	0.8998	0.9043	0.8916	847.99
			4	1800	0.8821	0.9168	0.9157	0.8986	897.26
			6	1200	0.9073	0.9068	0.9093	0.9019	1,370.92
			8	900	0.8355	0.8865	0.8960	0.8868	2,030.60
		cerrado	2	3600	0.8543	0.8956	0.9049	0.8930	1,071.69
			4	1800	0.8871	0.9168	0.9210	0.9090	1,221.27
			6	1200	0.8387	0.9009	0.9078	0.8994	1,902.79
			8	900	0.8330	0.8894	0.8952	0.8830	2,907.32
	AE	abierto	2	3600	0.8726	0.9218	0.9275	0.9290	1,125.62
			4	1800	0.9001	0.9380	0.9411	0.9334	1,166.14
			6	1200	0.9003	0.9377	0.9383	0.9338	1,748.50
			8	900	0.8960	0.9260	0.9298	0.9251	2,518.83
		cerrado	2	3600	0.8827	0.9237	0.9290	0.9244	1,565.49
			4	1800	0.9138	0.9406	0.9424	0.9358	1,495.02
			6	1200	0.9092	0.9352	0.9370	0.9296	2,505.96
			8	900	0.8830	0.9246	0.9296	0.9234	3,604.64
40	STD	abierto	2	3600	0.8883	0.9090	0.9111	0.8970	1,093.72
			4	1800	0.8939	0.9028	0.9109	0.9046	1,137.34
			6	1200	0.8718	0.9063	0.9122	0.9058	1,973.21
			8	900	0.8500	0.8975	0.9035	0.8935	2,410.85
		cerrado	2	3600	0.8450	0.8833	0.8945	0.8919	1,623.70
			4	1800	0.8770	0.9011	0.9119	0.9131	1,641.83
			6	1200	0.8523	0.8933	0.9021	0.8995	2,771.64
			8	900	0.8650	0.9036	0.9080	0.8986	3,556.28
	AE	abierto	2	3600	0.8879	0.9316	0.9363	0.9331	1,473.55
			4	1800	0.9110	0.9484	0.9500	0.9407	1,481.48
			6	1200	0.9083	0.9376	0.9434	0.9401	2,368.84
			8	900	0.8990	0.9270	0.9294	0.9221	2,957.70
		cerrado	2	3600	0.8854	0.9249	0.9338	0.9321	2,023.95
			4	1800	0.9147	0.9428	0.9462	0.9411	1,911.28
			6	1200	0.9050	0.9380	0.9430	0.9384	3,381.76
			8	900	0.8905	0.9273	0.9318	0.9228	4,347.79
50	STD	abierto	2	3600	0.8704	0.8997	0.9122	0.9055	1,441.00
			4	1800	0.8892	0.9116	0.9167	0.9091	1,278.77
			6	1200	0.8833	0.9200	0.9129	0.9133	2,367.43
			8	900	0.8790	0.9113	0.9113	0.8988	3,055.49
		cerrado	2	3600	0.8571	0.8892	0.9047	0.9018	2,049.00
			4	1800	0.8853	0.9198	0.9259	0.9208	2,073.40
			6	1200	0.8680	0.9121	0.9178	0.9130	3,240.71
			8	900	0.8716	0.9069	0.9121	0.9036	4,414.52
	AE	abierto	2	3600	0.8982	0.9291	0.9348	0.9326	1,685.74
			4	1800	0.9133	0.9487	0.9484	0.9409	1,671.03
			6	1200	0.9123	0.9399	0.9445	0.9408	2,704.00
			8	900	0.9080	0.9363	0.9373	0.9274	3,648.04
		cerrado	2	3600	0.9049	0.9349	0.9397	0.9350	2,613.50
			4	1800	0.9275	0.9494	0.9489	0.9430	2,276.15
			6	1200	0.9113	0.9342	0.9428	0.9377	3,823.19
			8	900	0.9010	0.9387	0.9410	0.9299	5,430.02
60	STD	abierto	2	3600	0.8856	0.9157	0.9185	0.9084	1,661.08
			4	1800	0.8507	0.9079	0.9156	0.9141	1,790.62
			6	1200	0.8820	0.9166	0.9184	0.9108	2,627.38
			8	900	0.9043	0.9275	0.9277	0.9179	3,689.31
		cerrado	2	3600	0.8697	0.8835	0.8999	0.9035	2,992.55
			4	1800	0.8824	0.9175	0.9252	0.9237	2,728.27
			6	1200	0.8618	0.9101	0.9174	0.9135	4,008.46
			8	900	0.8695	0.9148	0.9190	0.9100	4,971.85
	AE	abierto	2	3600	0.9113	0.9346	0.9394	0.9376	1,990.71
			4	1800	0.9125	0.9469	0.9499	0.9485	2,012.35
			6	1200	0.9110	0.9459	0.9495	0.9466	3,365.62
			8	900	0.9140	0.9428	0.9440	0.9362	4,338.62
		cerrado	2	3600	0.8977	0.9311	0.9412	0.9395	3,424.97
			4	1800	0.9224	0.9470	0.9511	0.9463	3,295.43
			6	1200	0.9044	0.9410	0.9467	0.9434	4,521.67
			8	900	0.9005	0.9365	0.9375	0.9264	5,947.62



Pot (hp)	Tipo	Carcaza	Polos	rpm	Ef 0.25	Ef 0.50	Ef 0.75	Ef 1.0	\$ USD
75	STD	abierto	2	3600	0.8675	0.9066	0.9163	0.9105	2,289.58
			4	1800	0.8788	0.9224	0.9255	0.9192	2,154.75
			6	1200	0.8890	0.9234	0.9224	0.9132	3,038.24
			8	900	0.8580	0.9135	0.9188	0.9112	4,240.21
		cerrado	2	3600	0.8698	0.8867	0.9047	0.9113	3,577.94
			4	1800	0.8966	0.9173	0.9293	0.9272	3,553.95
			6	1200	0.8905	0.9121	0.9189	0.9184	4,510.27
			8	900	0.8340	0.9060	0.9127	0.9047	6,612.23
	AE	abierto	2	3600	0.9042	0.9369	0.9430	0.9438	2,641.17
			4	1800	0.9248	0.9512	0.9542	0.9509	2,402.93
			6	1200	0.9150	0.9487	0.9489	0.9483	3,816.87
			8	900	0.9070	0.9422	0.9464	0.9435	5,073.49
		cerrado	2	3600	0.9061	0.9400	0.9454	0.9427	4,382.14
			4	1800	0.9283	0.9482	0.9523	0.9500	4,101.98
			6	1200	0.9098	0.9430	0.9487	0.9463	5,510.17
			8	900	0.8991	0.9388	0.9433	0.9391	7,658.52
100	STD	abierto	2	3600	0.8903	0.9212	0.9248	0.9158	2,770.17
			4	1800	0.8858	0.9221	0.9268	0.9184	2,469.84
			6	1200	0.8735	0.9216	0.9276	0.9233	3,721.41
			8	900	0.8685	0.9157	0.9200	0.9183	5,534.26
		cerrado	2	3600	0.8843	0.8987	0.9110	0.9147	4,703.18
			4	1800	0.8910	0.9267	0.9340	0.9337	4,890.60
			6	1200	0.8680	0.9211	0.9297	0.9260	5,859.82
			8	900	0.8790	0.9215	0.9265	0.9155	8,448.70
	AE	abierto	2	3600	0.9090	0.9364	0.9403	0.9386	3,212.21
			4	1800	0.9293	0.9516	0.9546	0.9508	3,060.77
			6	1200	0.9050	0.9427	0.9480	0.9462	4,694.63
			8	900	0.9170	0.9460	0.9490	0.9437	6,430.45
		cerrado	2	3600	0.8927	0.9347	0.9437	0.9442	5,699.27
			4	1800	0.9270	0.9502	0.9551	0.9518	5,172.45
			6	1200	0.9190	0.9446	0.9505	0.9476	7,186.40
			8	900	0.9110	0.9420	0.9452	0.9373	9,552.08
125	STD	abierto	2	3600	0.8840	0.9195	0.9255	0.9199	3,658.28
			4	1800	0.9220	0.9373	0.9373	0.9318	3,290.95
			6	1200	0.8855	0.9272	0.9275	0.9224	4,535.98
			8	900	0.9182	0.9417	0.9422	0.9356	10,267.76
		cerrado	2	3600	0.8808	0.9069	0.9234	0.9278	6,759.57
			4	1800	0.8830	0.9222	0.9314	0.9318	5,632.68
			6	1200	0.8900	0.9300	0.9363	0.9323	6,445.89
			8	900	s/d	0.9310	0.9320	0.9240	12,537.20
	AE	abierto	2	3600	0.8957	0.9450	0.9485	0.9475	4,574.54
			4	1800	0.9308	0.9549	0.9550	0.9523	3,553.83
			6	1200	0.9145	0.9480	0.9503	0.9465	5,455.02
			8	900	0.9220	0.9490	0.9240	0.9463	7,262.45
		cerrado	2	3600	0.9060	0.9430	0.9496	0.9494	7,464.76
			4	1800	0.9271	0.9494	0.9559	0.9545	7,019.23
			6	1200	0.9238	0.9484	0.9538	0.9497	8,424.06
			8	900	0.9240	0.9478	0.9492	0.9414	11,217.16
150	STD	abierto	2	3600	0.8943	0.9215	0.9263	0.9213	4,673.50
			4	1800	0.9110	0.9254	0.9343	0.9323	3,604.85
			6	1200	0.9000	0.9295	0.9325	0.9300	5,117.84
			8	900	0.9270	0.9450	0.9420	0.9620	10,900.50
		cerrado	2	3600	0.8770	0.9103	0.9256	0.9280	8,060.94
			4	1800	0.8995	0.9295	0.9375	0.9376	6,741.48
			6	1200	0.9080	0.9398	0.9420	0.9343	8,304.62
			8	900	0.9020	0.9450	0.9420	0.9240	15,356.90
	AE	abierto	2	3600	0.9193	0.9442	0.9474	0.9450	4,843.67
			4	1800	0.9238	0.9532	0.9578	0.9572	5,335.59
			6	1200	0.9200	0.9504	0.9534	0.9507	6,087.16
			8	900	0.9185	0.9483	0.9507	0.9480	8,396.31
		cerrado	2	3600	0.9173	0.9470	0.9523	0.9500	8,706.23
			4	1800	0.9302	0.9500	0.9563	0.9569	8,222.63
			6	1200	0.9310	0.9525	0.9569	0.9551	10,096.97
			8	900	0.9193	0.9488	0.9502	0.9452	13,502.34
200	STD	abierto	2	3600	0.8843	0.9244	0.9338	0.9325	6,545.93
			4	1800	0.9106	0.9389	0.9421	0.9351	5,107.57
			6	1200	0.9190	0.9280	0.9413	0.9318	7,430.48
			8	900	s/d	0.9363	0.9383	0.9253	8,193.25
		cerrado	2	3600	0.9113	0.9157	0.9287	0.9308	10,239.75
			4	1800	0.8980	0.9408	0.9473	0.9427	8,708.70
			6	1200	0.9251	0.9429	0.9451	0.9357	12,997.18
			8	900	s/d	0.9423	0.9490	0.9420	13,734.72
	AE	abierto	2	3600	0.9380	0.9480	0.9533	0.9498	8,135.08
			4	1800	0.9313	0.9591	0.9630	0.9588	6,695.58
			6	1200	0.9330	0.9548	0.9604	0.9510	7,700.55
			8	900	s/d	0.9578	0.9573	0.9503	9,718.71
		cerrado	2	3600	0.9178	0.9438	0.9518	0.9527	10,994.88
			4	1800	0.9407	0.9553	0.9591	0.9586	10,410.64
			6	1200	0.9350	0.9548	0.9579	0.9545	13,542.37
			8	900	0.9250	0.9514	0.9541	0.9483	14,367.86



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Facultad de Ciencias
Químicas e Ingeniería

FACULTAD DE CIENCIAS QUÍMICAS e INGENIERÍA
Programas educativos de calidad reconocidos por CIEES, CACEI y CONACYT
SGC certificado en la norma ISO 9001:2015

Secretaría de Investigación y Posgrado

FORMATO T-2
NOMBRAMIENTO DE COMISIÓN REVISORA
Y DICTAMEN DE REVISIÓN

Cuernavaca, Mor., a 20 de enero de 2021.

Dr. Luis Cisneros Villalobos
Dr. Mario Limón Mendoza
Dr. J. Guadalupe Velásquez Aguilar
Dr. José Gerardo Vera Dimas
Dra. María del Carmen Torres Salazar
PROFESORES DE LA FCQeI
P R E S E N T E

Por este conducto, me permito informarle que ha sido asignado como integrante de la Comisión Revisora de la tesis que presenta el Ing. Hugo Herrera Gutiérrez titulada: DESARROLLO DE UN SIMULADOR DE EFICIENCIA ENERGÉTICA PARA LA EVALUACIÓN DE PROYECTOS DE SUSTITUCIÓN DE MOTORES DE INDUCCIÓN DE EFICIENCIA ESTÁNDAR POR MOTORES DE INDUCCIÓN DE ALTA EFICIENCIA realizada bajo la dirección de la Dra. María del Carmen Torres Salazar en el Programa Educativo de Maestría en Ingeniería Eléctrica y Electrónica. Agradezco su valiosa participación en esta Comisión y quedo a sus órdenes para cualquier aclaración o duda al respecto.

A T E N T A M E N T E
Por una humanidad culta

DRA. VIRIDIANA AYDEÉ LEÓN HERNÁNDEZ
ENCARGADA DE DESPACHO DE LA DIRECCIÓN DE LA FCQEI

D I C T A M E N

DRA. VIRIDIANA AYDEÉ LEÓN HERNÁNDEZ
ENCARGADA DE DESPACHO DE LA DIRECCIÓN DE LA FCQEI
P R E S E N T E

En respuesta a su amable solicitud para formar parte de la Comisión Revisora de la tesis mencionada y una vez realizada la revisión correspondiente, me permito informarle que mi VOTO es:.

DICTAMEN

NOMBRE	VOTO	FIRMA
Dr. Luis Cisneros Villalobos		
Dr. Mario Limón Mendoza		
Dr. J. Guadalupe Velásquez Aguilar		
Dr. José Gerardo Vera Dimas		
Dra. María del Carmen Torres Salazar		

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

VIRIDIANA AYDEE LEON HERNANDEZ | Fecha:2021-02-03 15:46:21 | Firmante

FICRghd7zgspr6r3iEPIWLMhM1v4qllMnKzFvuxMQTwoVVhwCWN4/HskJRRrPfcJNYbiR1lc5jvs+Vy7kjNOUSW/zvAo6YsUBX7L4BT/Y4oM4VKUoQCGXanYTXfcG6fzSYIqEEDFKV9TrtYmfu2nEN7lZaCuGwmG6t/RsX+9s0DaBjxYm/mR2rfxl04gjpX0YCTt7HouLEFL0T1Kl/mpoBSe2ZqmfREyvQHZ+7vgSytipjY97vrSC11w0g8B8gm3c30qTacsLL8nnm+KilM6vt5SejKfT3JbDQAARcLdHM+R98RCcarMVKC2CPf1UKKrX4jh6c22twpenQx527kBnA==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:



[g32De5](#)

<https://efirma.uaem.mx/noRepudio/LMROCJVw7Q80IRxPS5a9HGRh2HHPProA2>



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

LUIS CISNEROS VILLALOBOS | Fecha:2021-02-08 14:39:04 | Firmante

Y9PJSrMD+JP0T8Mjn5CB5XzVy8Qmfd+Cju/5GEPDQiRR9m/A2KfQRb9O+AJNdPp9SOJQRSyK9QFDRJC77bqzK7K+EQXH/CdfIDMzyGRXxf94x9iJU2yqLh9TRqmnCXe9jFWKuscAdAzwGvx3yhNQmWnwalUKCD6uv+oQsReJewDpcKg5Dh/mUnvrC1aGABi/tPDI2TcV+4rJfNmXFXEewQmr7DEPwVTrSnSuH8SZ1wTsTDqTolca7chXdJxDSaPANdQZdH8VPoeFb3YYYjFA99AqlcSHm5B40MmAzwwU+FgzJhcGt7WnvG17kNQRilVfDZdj2WvLtGqJBNU93Q==

MARIA DEL CARMEN TORRES SALAZAR | Fecha:2021-02-08 14:39:07 | Firmante

zrMNH/yTNJASSnfPBwu6zvvtkAbZXEXGG74bmTsjCkcxXbXacTbj2AvJ/Vnxz3JMUN99j7MfiRCooTvIRDPhKUMkSaPBvYVdzkWquUhrKHxeYKzm0wGj652QVspugZHMvWc9BYRsZTCXWESVZZjdGhL2pvqRx5OdAtZVbHf9Knw1RIsDfSbV2GOUjzoNjrMGdckqEb+oLUSgHydQRh7ML9Ds6P3RCzUgwm8NNZ3RJBtraJyB/PIBeQc58h/6B8YPojnccx2dlSV00tEneDDjfwYxyCkQ1cUgG4hjlYSF6LrP41hM+dkl9X3oPlwU7cM93z9d0z57ribb3nXO1bMw==

JOSE GERARDO VERA DIMAS | Fecha:2021-02-08 15:16:41 | Firmante

g4CkbRCewWwhGDichSonSST+GOANSFsPXUia7mmTycTOC/5K+yF56BDZdtXdcNdqzwP8QNbeHaTcDHLmJNWJu5KsYRXV8rILa651BnMewmsiu29B40zPr8GOXHlzpShETgUE/9TrCv8uFb/HIODR9JXrwDyOlxh7+LX/op7Jl2eMtsR8qJGmC+CimJEuC6qL9NCgYpNoWoYRXE9QxezLNUApsEoTd4QFpauXSmYHx4RUFjNM9RniNDuKAgx9QaUolBwJpxlX4tTC0ShDlp10hnz1ff6vk5QxUzYo0bdSDuxqF/whdVUXKHADEIPq85PWAcVhS9rf2/M4kgGgAMDEw==

MARIO LIMON MENDOZA | Fecha:2021-02-08 19:29:36 | Firmante

IQBQnZmjtbb+BRT4PLhd/Fr3srUhZ5KkmFqMNFBCBtYqxjZr7FaTan+dvqO5psGrB56unEqotpm/483LjRoknyRuhv6oitYJK6OT7n+bmrz9ltZWskZayll2kkGsQra7gcx2mhzG/7/U5sGi24fQ6PemTO5sqvZKuDgF+EL9nA/V4VP5ce475aWppq7SoTiRRtUezLS5Lsw6pANXAcc4ZaRf+FB3qfalhd8cojVyr3T9JuN4aM0+5podL5ZaUIBMhMbdbpG2UwQJxqRkMBo1dRpbOPw9992GrQP052K9HmZENwoaNiqPCIAA1qmbYRzxDvfyaiR/YFpTB4vWfoNfUQ==

J GUADALUPE VELASQUEZ AGUILAR | Fecha:2021-02-09 10:35:20 | Firmante

efw3pFn6Qi3+nZVv06568NPrQnQhZik4YMac1o8QhtOHsaNUQiDEXG/lgc5bQUuq2EfN/LZNtZnkr+qJCAC2j7xujZAo5rYIZQUgOmysj8DPF6VuG3tMwVTBAGg/38Xs1faAEo1CRpgxsWblglD8WicNgHoJqJ85ou0ftqtGQD9Xlr3JbCl4InK4fBRx2NFewgIRBRAWRzFCj1pVLGqG07RilnsAw8aPx0lfU8m97Cx8nMq1ZBvR/a2VzFJ0CFHuCEJCoENSgfGXzCQlVxwqvjCOWqQeEeQ0zD5AV9nrGOU/bq7jr9oqvQadCRnysvNZ82yOP0gGoFFbax1cFTvA==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:



IXZ1MQ

<https://efirma.uaem.mx/noRepudio/ESIIVSE1FSJ06wzGNUNN7NPvFdXDrz5>

