



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**PROPUESTA DE DISEÑO, DE UN SISTEMA DE MONITOREO DE UN
MOTOR TRIFÁSICO DE INDUCCIÓN, CON ROTOR DE JAULA DE ARDILLA,
UTILIZANDO ARDUINO COMO INTERFAZ DE ADQUISICIÓN DE SEÑALES
MEDIANTE LABVIEW**

Carlos René Gómez Patzán

Asesorado por el Ing. Gustavo Benigno Orozco Godínez

Guatemala, julio de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**PROPUESTA DE DISEÑO, DE UN SISTEMA DE MONITOREO DE UN
MOTOR TRIFÁSICO DE INDUCCIÓN, CON ROTOR DE JAULA DE ARDILLA,
UTILIZANDO ARDUINO COMO INTERFAZ DE ADQUISICIÓN DE SEÑALES
MEDIANTE LABVIEW**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

CARLOS RENÉ GÓMEZ PATZÁN

ASESORADO POR EL ING. GUSTAVO BENIGNO OROZCO GODÍNEZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRICISTA

GUATEMALA, JULIO DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Guillermo Antonio Puente Romero
EXAMINADOR	Ing. Armando Gálvez Castillo
EXAMINADOR	Ing. Fernando Alfonso Moscoso Lira
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PROPUESTA DE DISEÑO DE UN SISTEMA DE MONITOREO DE UN MOTOR TRIFÁSICO DE INDUCCIÓN, CON ROTOR DE JAULA DE ARDILLA, UTILIZANDO ARDUINO COMO INTERFAZ DE ADQUISICIÓN DE SEÑALES MEDIANTE LABVIEW

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 8 mayo 2018

Carlos René Gómez Patzán

Guatemala, 29 de febrero de 2020.

Ingeniero José Aníbal Silva
Coordinador Área de Electrotecnia
Escuela Mecánica Eléctrica
Facultad de Ingeniería
USAC.

Estimado Ingeniero:

De acuerdo con la designación efectuada por la Dirección de Escuela, me permito informarle que he procedido a asesorar el Trabajo de Graduación titulado: **PROPUESTA DE DISEÑO, DE UN SISTEMA DE MONITOREO DE UN MOTOR TRIFÁSICO DE INDUCCIÓN, CON ROTOR DE JAULA DE ARDILLA, UTILIZANDO ARDUINO COMO INTERFAZ DE ADQUISICIÓN DE SEÑALES MEDIANTE LABVIEW**, desarrollado por el estudiante CARLOS RENÉ GÓMEZ PATZÁN, carne 2010-20760 y, encontrándolo satisfactorio en su contenido y resultados, me permito dar aprobación al mismo, remitiéndolo a esa Coordinación para el trámite pertinente, en el entendido que el Autor y este Asesor somos responsables del contenido y conclusiones del Trabajo.

Agradeciendo la atención a la presente, me es grato suscribirme, deseándole éxitos en sus labores.

Atentamente,



Ing. Gustavo Benigno Orozco Godínez.
Colegiado 1,879
ASESOR

ING. GUSTAVO B. OROZCO
COLEGIADO 1879



ESCUELA DE
INGENIERÍA MECÁNICA ELÉCTRICA
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

REF. EIME 23072020a
23 Julio 2020

Señor Director
Ing. Armando Alonso Rivera Carrillo
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Señor Director:

Me permito dar aprobación al trabajo de graduación titulado en modalidad de tesis: **PROPUESTA DE DISEÑO, DE UN SISTEMA DE MONITOREO DE UN MOTOR TRIFÁSICO DE INDUCCIÓN, CON ROTOR DE JAULA DE ARDILLA, UTILIZANDO ARDUINO COMO INTERFAZ DE ADQUISICIÓN DE SEÑALES MEDIANTE LABVIEW**, del estudiante; Carlos René Gómez Patzán con numero de carnet 1845846470301, que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑAD A TODOS

JOSE ANIBAL SILVA DE LOS ANGELES
ING ELECTRONICO
COLEGIADO No 5067

Ing. José Anibal Silva de los Angeles
CC. Básicas Y Electrotecnia



REF. EIME 60. 2021.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; CARLOS RENÉ GÓMEZ PATZÁN titulado; PROPUESTA DE DISEÑO, DE UN SISTEMA DE MONITOREO DE UN MOTOR TRIFÁSICO DE INDUCCIÓN, CON ROTOR DE JAULA DE ARDILLA, UTILIZANDO ARDUINO COMO INTERFAZ DE ADQUISICIÓN DE SEÑALES MEDIANTE LABVIEW, procede a la autorización del mismo.

Ing. Armando Alonso Rivera Carrillo

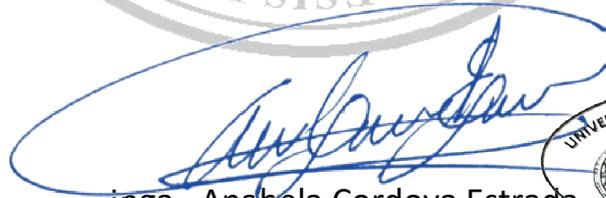


GUATEMALA, 8 DE ABRIL 2021.

DTG. 284-2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **PROPUESTA DE DISEÑO, DE UN SISTEMA DE MONITOREO DE UN MOTOR TRIFÁSICO DE INDUCCIÓN, CON ROTOR DE JAULA DE ARDILLA, UTILIZANDO ARDUINO COMO INTERFAZ DE ADQUISICIÓN DE SEÑALES MEDIANTE LABVIEW**, presentado por el estudiante universitario: **Carlos René Gómez Patzán**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



inga. Anabela Cordova Estrada
Decana



Guatemala, julio de 2021

AACE/asga

ACTO QUE DEDICO A:

- Dios** Por darme sabiduría, fuerzas, salud, permitirme crecer como persona y ser humano día tras día.
- Mis padres** Ofelia Patzán Pérez y René Gómez Tubac por sus enseñanzas, su apoyo incondicional en todo momento y darme lo necesario para poder estar aquí.
- Mi esposa** Yoselin Paola López Us de Gómez, por ser importante influencia en mi carrera, su apoyo, cariño, comprensión y la paciencia que me tuvo en los momentos más difíciles.
- Mi hija** Sofía Gabriela Gómez López, por recibirme siempre con un abrazo y un beso, por ser la fuente de inspiración para lograr este triunfo.
- Mi tío** Wenceslao Gómez Car, por sus bendiciones, enseñanzas, valores inculcados, su cariño y ayuda emocional.
- Mis amigos** Por su ayuda, apoyo durante este trayecto del saber.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por permitirme desarrollar mi carrera como profesional en tan prestigiosa casa de estudio.
Facultad de Ingeniería	En especial a la Escuela de Ingeniería Mecánica Eléctrica, por los consejos de los catedráticos durante mi proceso profesional y darme los valores necesarios para el ejercicio profesional.
Mi asesor	Ing. Gustavo Benigno Orozco Godínez, por brindarme sus conocimientos y darme parte de su tiempo para la realización de este trabajo de graduación.
Mi familia	Por brindarme su amor, cariño, comprensión y apoyo para ser un buen profesional.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. MARCO TEÓRICO.....	1
1.1. Motor eléctrico	1
1.2. Aspectos generales del motor asíncrono.....	3
1.2.1. Estator	3
1.2.2. Rotor.....	3
1.2.2.1. Rotor de jaula de ardilla.....	4
1.2.2.2. Rotor bobinado	4
1.2.3. Consideraciones generales de un motor eléctrico de rotor de jaula de ardilla	5
1.3. Funcionamiento del motor trifásico de inducción.....	7
1.3.1. Ley de Faraday.....	8
1.3.2. Ley de Lorentz.....	9
1.4. Corriente alterna.....	11
1.4.1. Valor eficaz	13
1.5. Corriente alterna trifásica.....	14
1.6. Campos magnéticos giratorios	15
1.7. Circuito equivalente del motor asíncrono trifásico	18
1.7.1. Arranque del motor de inducción	21

1.8.	Placa característica del motor	22
1.9.	Aplicaciones de las máquinas de inducción	23
1.10.	Sistema de monitoreo	25
1.11.	Parámetros implicados.....	26
1.11.1.	Tensión eléctrica	26
1.11.2.	Corriente eléctrica	26
1.11.3.	Potencia eléctrica	27
1.11.3.1.	Potencia activa	27
1.11.3.2.	Potencia reactiva.....	27
1.11.3.3.	Potencia aparente	27
1.11.3.4.	Factor de potencia.....	28
1.11.4.	Energía eléctrica	28
1.11.5.	Frecuencia en corriente alterna.....	28
1.11.6.	Velocidad del rotor	28
1.11.7.	Temperatura.....	28
1.12.	Fallas en motores eléctricos de inducción.....	30
2.	ARDUINO	31
2.1.	Tipos de Arduino	31
2.1.1.	Arduino UNO	31
2.1.2.	Arduino DUE	32
2.1.3.	Arduino TRE.....	32
2.1.4.	Arduino Leonardo.....	32
2.1.5.	Arduino MEGA 2560	33
2.1.6.	Arduino Mega ADK.....	33
2.1.7.	Arduino BT	33
2.1.8.	Arduino Micro	33
2.1.9.	Arduino Nano	34
2.1.10.	Arduino YUN	34

2.1.11.	Arduino Ethernet.....	34
2.1.12.	Arduino FIO	35
2.1.13.	Arduino ZERO	35
2.2.	Características Arduino UNO	36
2.3.	Datos técnicos del Arduino UNO	37
2.3.1.	Pines digitales.....	38
2.3.2.	Pines analógicos.....	39
2.3.3.	Pines de alimentación.....	39
2.3.4.	Otros pines	40
2.4.	Microcontrolador	41
2.5.	Componentes en la placa base	42
2.6.	Entorno de programación Arduino IDE	45
2.6.1.	Configuración de IDE.....	47
2.7.	Estructura Sketch	48
2.8.	Compilación y carga	51
3.	SOFTWARE LABVIEW	57
3.1.	Entorno de desarrollo LabVIEW	58
3.1.1.	Virtual Instruments en LabVIEW.....	58
3.1.2.	Panel frontal.....	58
3.1.3.	Paleta de controles	59
3.2.	Diagrama de bloques	60
3.2.1.	Nodos de diagrama de bloques	61
3.3.	Terminales.....	62
3.4.	Funciones	62
3.5.	SubVIs	64
3.6.	Cables	65
3.7.	Programación gráfica	66
3.7.1.	Estructuras de ejecución	69

3.7.1.1.	Ciclos While.....	69
3.7.1.2.	Ciclos For	70
3.7.1.3.	Estructura case	72
3.7.1.4.	Variables	73
3.8.	Conexión entre LAbVIEW y Arduino	75
4.	SISTEMA DE MEDICIÓN UTILIZANDO ARDUINO Y LABVIEW	77
4.1.	Medición de parámetros eléctricos.....	77
4.2.	Medición de voltaje.....	77
4.2.1.	Trasformador de potencial.....	78
4.3.	Programa para medición de voltaje.....	84
4.4.	Medición de corriente	86
4.5.	Medición de temperatura.....	93
4.5.1.	Sensor de temperatura.....	94
4.5.2.	Programa para medir la temperatura	95
4.6.	Diagrama de conexión desde motor hasta ordenador	99
4.6.1.	Protección eléctrica del cable.....	99
4.6.2.	Protección mecánica	100
4.6.3.	Instalación del transformador de voltaje	101
4.6.4.	Instalación del transformador de corriente	102
4.6.5.	Instalación del sensor de temperatura	103
4.7.	Distancia de instalación.....	104
	CONCLUSIONES.....	107
	RECOMENDACIONES	109
	BIBLIOGRAFÍA.....	111

ÍNDICE DE ILUSTRACIONES

1.	Estator y rotor de un motor de inducción.....	5
2.	Composición completa de un motor eléctrico de jaula	6
3.	Imán moviéndose sobre conductores cortocircuitados.....	10
4.	Grafica de corriente alterna.....	12
5.	Grafica de corriente alterna desfasada 120° eléctricos.....	15
6.	Campo magnético giratorio producido en un motor de inducción trifásico de dos polos.....	17
7.	Circuito equivalente del motor de inducción por fase	19
8.	Circuito equivalente por fase más aproximado	20
9.	Curvas del par de torsión en función de la velocidad	23
10.	Arduino UNO R3.....	37
11.	Arduino IDE.....	46
12.	Directorio de guardado de un sketch.....	48
13.	Estructura básica de un sketch	49
14.	Programa semáforo escrito en Arduino IDE.....	54
15.	Programa de aplicación de grafica de Arduino IDE.....	55
16.	Grafica de función trigonométrica Seno	56
17.	Panel frontal de un VI.....	59
18.	Paleta de controles del panel frontal	60
19.	Diagrama de bloques de un VI.....	61
20.	Paleta de funciones.....	64
21.	Creacion de subVI en el diagrama de bloques.....	65
22.	Paleta de Herramientas.....	67
23.	Representación gráfica del ciclo While.....	70
24.	Ciclo For aplicado a un programa con Wait	71

25.	Programa con Structure case	72
26.	Variables locales y globales.....	74
27.	Diagrama de bloque de un programa elemental.....	75
28.	Ruta de librerías de compatibilidad.....	76
29.	Trasformado de potencial	79
30.	Circuito rectificador de voltaje.....	82
31.	Divisor de voltaje.....	84
32.	Diagrama de bloques del programa medio de voltaje.....	85
33.	Panel del programa de medición de voltaje	86
34.	Sensor de corriente no invasivo.....	87
35.	Transformado de corriente.....	88
36.	Circuito acondicionador de corriente.....	90
37.	Diagrama de bloques del programa medición de corriente.....	92
38.	Panel frontal del programa de medición de corriente.....	93
39.	Modulo sensor de temperatura	94
40.	Diagrama de bloques del programa de medición de temperatura	97
41.	Interfaz entre diagrama de bloques y panel de control de usuario.....	98
42.	Cables para instrumentación PLTC	100
43.	Diagrama de instalación	102
44.	Instalación del sensor de temperatura KY-028	104
45.	Diagrama de conexión del sistema de monitoreo	105

TABLAS

I.	Lista de componentes de la figura anterior	7
II.	Especificaciones técnicas del microcontrolador	44
III.	Tiempo requerido para ejecutar algunos comandos en Atmega328	52
IV.	Tipos de datos más comunes	66
V.	Funciones principales de la paleta de herramientas	68
VI.	Distancia recomendada para cable con par blindado.....	104

LISTA DE SÍMBOLOS

Símbolo	Significado
A	Amperio
E	Campo eléctrico
B	Campo magnético
I²C	Circuito integrado
PC	Computadora personal
API	Conjunto de funciones, rutinas y subrutinas
CA/CD	Convertor de corriente CA a CD
CA	Corriente alterna
CD	Corriente directa
C/s	Colombios por segundo
ICSP	Entrada para programación serial
∅	Flujo magnético
ε	Fuerza electromotriz
GHz	gigahercio
Hz	hercio
kΩ	Kilo ohmios
kB	Kilobyte
kWh	Kilowatt-hora
MB	Megabyte
MHz	Megahercio
RAM	Memoria de acceso aleatorio
ROM	Memoria de lectura
μs	Microsegundo

mA	Miliamperio
ms	Milisegundo
PWM	Modulación por ancho de pulsos
HDMI	Norma de video de alta definición
PBC	Placa de circuito impreso
SPI	Protocolo de datos serial
USB	Puerto de conexión
UARTs	Puerto serie de microcontroladores
IDE	Software de programación para Arduino
WIFI	Tecnología de interconexión inalámbrica
SMD	Tecnología de montaje superficial
VA	Voltamperio
VAr	Voltamperio reactivo
V	Voltio

GLOSARIO

Arandela	Pieza delgada generalmente circular y con un orificio en el centro.
Baudio	Unidad de medida utilizada en telecomunicaciones, representa el número de símbolos por segundo en un medio de transmisión digital.
Bits	Unidad de medida utilizada en telecomunicaciones, representa el número de símbolos por segundo en un medio de transmisión digital.
Bobina	Espiras de conductor eléctrico aislado, arrollado repetidamente.
Bootloader	Gestor de arranque, es un programa diseñado exclusivamente para preparar todo lo necesario para iniciar el sistema operativo.
Brida	Elemento circular que une dos componentes de un sistema de tuberías. Permite desmontaje sin operaciones destructivas.
Calibre	Es una referencia de clasificación del espesor de los conductores eléctricos, mide la sección circular de los mismos.

Chaveta	Pieza de sección rectangular o cuadrada que se inserta entre dos elementos que deben ser solidarios entre sí, para transmitir potencia y evitar que se produzcan deslizamientos entre ellas.
Condensador	Dispositivo electrónico que se basa en almacenar energía en forma de campo eléctrico, también se conoce con el nombre de capacitor.
Cursor	Señal, normalmente parpadeante, indica en una pantalla de computadora la posición en la que se puede realizar una acción.
Electromagnetismo	Magnetismo producido por una corriente eléctrica.
Empaque	También conocido como; junta mecánica, junta de estanqueidad o empaquetadura, son componentes de material adaptable que se utiliza para sellar bien uniones entre dos piezas de metal.
Equidistante	Se refiere a algo que se encuentra a igual distancia entre dos puntos.
Escobillas	Son bloques de grafito, proporcionan conexión eléctrica entre una parte rotatoria y una parte fija.
Ferromagnetismo	Es la propiedad de materiales principalmente hierro, cobalto, níquel y otros de, adquirir una fuerte imantación.

Impedancia	Se refiere a la resistencia de los dispositivos alimentados por corriente alterna, posee magnitud y fase.
Iteraciones	Significa repetir varias veces un proceso para alcanzar una meta.
Prensaestopas	Forro para ejes giratorios generalmente en forma de anillos o retenes.
Termistor	Componente electrónico, su valor varía en función de la temperatura a la cual se somete.
Toolkit	Conjunto de herramientas de software para utilizarse para un propósito particular.

RESUMEN

El presente trabajo de graduación consiste en investigar, diseñar y combinar de manera adecuada, hardware con software, para que estos sean capaces de formar un sistema de medición en tiempo real, capaz de mostrar al usuario los datos obtenidos de un motor trifásico de corriente alterna, con rotor tipo jaula de ardilla. La interfaz del usuario busca ser lo más gráfica posible, siendo capaz de imitar medidores eléctricos industriales.

El primer capítulo aborda la teoría básica de motores eléctricos, haciendo énfasis en su funcionamiento, también considera la corriente alterna trifásica, sus características, también sus parámetros asociados al funcionamiento de motores eléctricos. En el capítulo dos se presenta la plataforma Arduino abarcando sus principales variantes comerciales, su composición física, describiendo sus principales características; contiene ejemplos de sus comandos más elementales, al igual que su funcionamiento y aplicación.

El tercer capítulo trata el software de desarrollo LabVIEW. Describe sus conceptos y características de programación, muestras sus interfaces gráficas de interacción, herramientas hasta finalizar con la conexión entre LabVIEW con Arduino.

Por último, en el cuarto capítulo se diseña el software en LabVIEW capaz de obtener los datos procesados por Arduino provenientes del motor trifásico, mediante sensores adecuados para medir corriente, voltaje y temperatura del motor.

OBJETIVOS

General

Proponer un diseño de sistema de monitoreo para un motor trifásico de inducción, utilizando Arduino como interfaz de adquisición de señales para LabVIEW.

Específicos

1. Exponer los fundamentos del funcionamiento de un motor trifásico de inducción tipo jaula de corriente alterna, sus parámetros asociados y la importancia de un sistema de monitoreo.
2. Presentar la plataforma Arduino como hardware, especificaciones y principales modelos disponibles, también, su software libre para la programación del mismo.
3. Enseñar el software LabVIEW y su funcionamiento básico como herramienta de programación gráfica de ingeniería hasta prepararlo para su aplicación en conjunto de Arduino.
4. Diseñar un programa de medición de parámetros del motor eléctrico tales como tensión, corriente y temperatura para mostrarlos en un ordenador en tiempo real.

INTRODUCCIÓN

Considerando la cantidad de aplicaciones de los motores trifásicos, principalmente de índole industrial, es necesario prevenir mantenimientos o paradas imprevistas puesto que, tratándose de motores críticos, representan paradas de producción, por lo tanto, representa pérdida económica. Surge la necesidad de aplicar nuevas alternativas que permitan identificar, establecer y actuar sobre problemas potenciales para facilitar o mejorar las técnicas actuales de predecir o detectar fallas.

Las herramientas computacionales con funcionalidades parecidas a LabVIEW permiten desarrollar aplicaciones para solventar este tipo de necesidades, puesto que, gracias a su lenguaje de programación gráfico, permite programar de manera considerablemente intuitiva, facilitando la adquisición de señales de instrumentación y monitoreo. Otra característica de este software es la capacidad de integración con hardware dedicado, o para uso general como lo es Arduino, este mismo, se utiliza como medio entre la máquina eléctrica con un ordenador ejecutando LabVIEW.

Mediante investigación, pruebas con las funciones, aprovechando la capacidad gráfica que ofrece LabVIEW, se pretende desarrollar un sistema capaz de mostrar en tiempo real, en un ordenador, parámetros específicamente voltaje, corriente y temperatura de un motor trifásico, datos que pueden ser utilizables para establecer mejoras en los mantenimientos preventivos de este tipo de máquinas.

1. MARCO TEÓRICO

1.1. Motor eléctrico

Es una maquina eléctrica encargada de transformar energía eléctrica en energía mecánica, logrando este efecto basándose en inducción electromagnética. El funcionamiento se desarrolla introduciendo una corriente en la maquina por medio de una fuente externa. son máquinas eléctricas fabricadas con una parte estática y una parte giratoria. Existen algunos con la capacidad de realizar el proceso inverso, convirtiendo energía mecánica en eléctrica.

Es posible clasificar los motores electricos, basándose en el tipo de corriente que utilizan, existen tres grupos, cada tipo tiene variación en sus componentes, su estructura determina la interacción de los flujos electromagnéticos, estas diferencias brindan características diferentes en especial la fuerza o par de torsión del motor.

- Motores universales: su nombre se deriva de la capacidad de funcionar con corriente directa o con corriente alterna, gracias a su característica de poder operar con ambas corrientes, son utilizados en gran cantidad de aplicaciones, a nivel profesional, industrial o doméstico.
- Motores de corriente directa: generalmente tienen un diseño variado, su característica principal es que permite modificar la velocidad del rotor con simplemente variar su tensión. Dependiendo de su función son de mayor coste por consiguiente sus aplicaciones a nivel industrial son limitadas.

- Motores de corriente alterna: generalmente diseñados para funcionar a velocidades constantes, se consideran sencillos, de bajo coste y ampliamente usados en aplicaciones industriales, de la misma forma de utilización doméstica.

El motor de corriente alterna, se encuentra en el grupo de los más utilizados a nivel industrial dado a que, las redes de distribución de la energía eléctrica se realizan en corriente alterna: generación, transformación, transporte y distribución. Son considerablemente más económicos, por tanto, se utilizan en la mayoría de procesos con un mantenimiento menos complejo, aunque el avance constante en electrónica de potencia ha incrementado las aplicaciones de estos.

Los de motores de corriente alterna, son fabricados en variedades, sin embargo, dentro de las clasificaciones más generales se pueden encontrar; motores sincrónicos y motores de inducción.

- Motor síncrono: se llama así por su velocidad constante, independientemente de la variación de carga. El rotor gira a velocidad constante, en conjunto con la velocidad de su campo magnético, proveniente del estator. Tienen la capacidad de funcionar como generadores, convirtiendo energía mecánica en eléctrica, este régimen de generador es mayormente utilizado para aplicaciones en estas máquinas.
- Motor de Inducción: también conocido como motor asíncrono, se caracteriza por su velocidad variable, dependiendo de la variación de carga. Son máquinas de corriente alterna, una característica de estos es que, su rotor no tiene capacidad de girar a la misma rapidez del campo magnético del estator. Mientras más grande sea el torque que entrega el rotor, más

grande es la diferencia entre las frecuencias del rotor y la frecuencia del estator.

1.2. Aspectos generales del motor asíncrono

A nivel industrial, la mayoría de máquinas mecánicas que utilizan movimiento rotatorio, están basados en motores de inducción; también llamado motor asíncrono, su nombre se deriva del método que se utiliza para transformar la potencia eléctrica en potencia mecánica, son alimentados por corriente alterna trifásica. Todas las máquinas rotatorias, se basan de dos partes principales:

1.2.1. Estator

Es base principal de la máquina, siendo esta fija permite la libertad al girar al rotor, el estator no tiene movimiento mecánico, pero sí movimiento magnético gracias a corrientes trifásicas. fabricado por una carcasa en la que está fijadas un conjunto de chapas de material llamado acero magnético, acero al silicio, contiene ranuras encargadas de almacenar los devanados inductores.

1.2.2. Rotor

Mecánicamente es la parte giratoria de la máquina. Colocado sobre un eje que, rota libremente en el interior del estator, al igual que este se fabrica de un núcleo de chapas de acero al silicio, este material se usa para disminuir las pérdidas asociadas a los campos magnéticos rotatorios, como las corrientes parásitas y las producidas por histéresis. Se encuentran apiladas formando un cilindro, en el interior del cual se dispone una parte de cobre, que se puede dividir en dos tipos más utilizados:

1.2.2.1. Rotor de jaula de ardilla

Conocido también como rotor cortocircuitado, motor de jaula, es un cilindro montado en un eje. Compuesto por barras conductoras de aluminio o de cobre entre espacios determinados formados por las chapas de acero eléctrico, ambos extremos de las barras en cortocircuito con anillos, formando así una jaula cilíndrica.

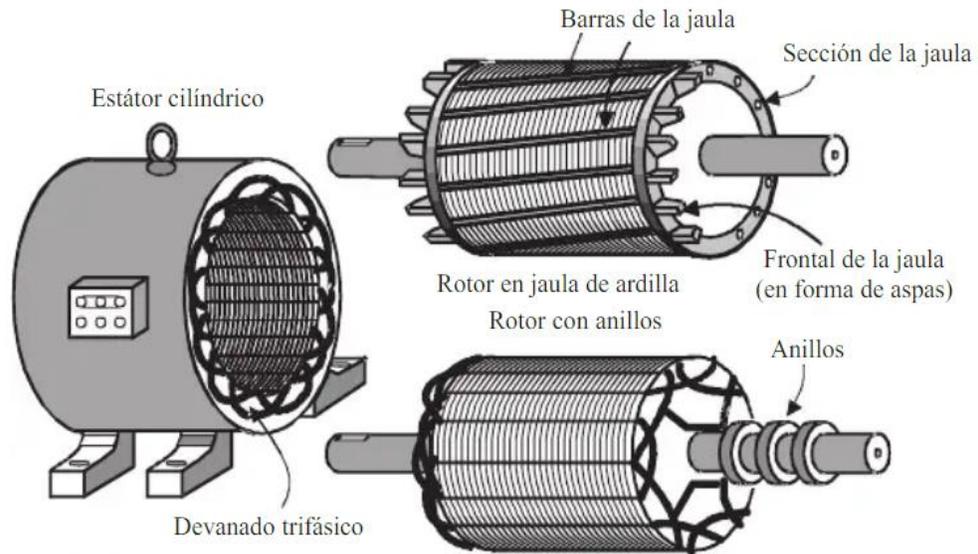
Un inconveniente considerable en motores con rotor de jaula, es que, durante el arranque requiere una corriente intensa, en comparación con la nominal; entre 4 y 7 veces. Depende principalmente de la capacidad nominal del motor y de la efectividad de la resistencia del rotor en condiciones de arranque, también es necesario considerar un bajo factor de potencia, a pesar de ello, el par de arranque suele ser catalogado bajo.

1.2.2.2. Rotor bobinado

Al igual que el rotor de jaula, este tiene la variante de poseer bobinas o devanados de cobre; lleva 3 grupos de estas, se conectan a sendos anillos deslizantes colocados en el eje, mediante escobillas de carbón que conectan el rotor a unas resistencias que se pueden variar hasta poner el rotor en corto circuito. Son motores mayormente nombrados como motores trifásicos de rotor bobinado.

Este tipo de rotores está actualmente extinguiéndose, estudios consideran que, el 95 % de los motores de inducción en aplicaciones industriales son de jaula de ardilla. En la figura 1 se muestra un dibujo de un estator de motor trifásico, simultáneamente contiene los rotores de inducción bobinado y tipo jaula de ardilla.

Figura 1. **Estator y rotor de un motor de inducción**



Fuente: FRAILE MORA, Jesús. *Maquinas Eléctricas*. p. 169.

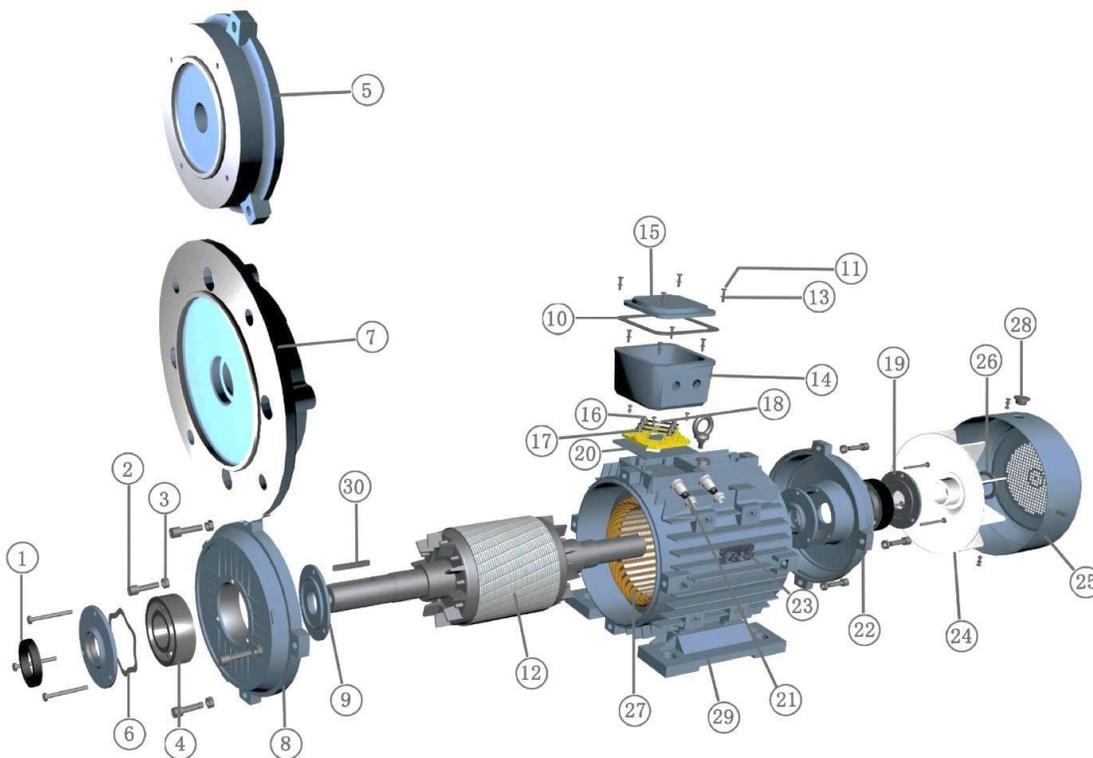
1.2.3. **Consideraciones generales de un motor eléctrico de rotor de jaula de ardilla**

Dada la gran aplicación de los motores eléctricos trifásicos de inducción, existen muchas variaciones del mismo, principalmente variedad de tipos específicos de carcasas de motores eléctricos en el mercado, los cuales se pueden clasificar de manera muy general en dos tipos básicos: los totalmente cerrados y los abiertos.

En la figura 2 se muestran todos los componentes mecánicos y eléctricos de un motor trifásico con rotor de jaula de ardilla con una carcasa cerrada, son de múltiples posiciones, indica que, las patas son desmontables, esto permite fácilmente el cambio de forma constructiva; sujeción, al girar la caja de bornes en

intervalos de 90 grados, facilitando su instalación, permitiendo libre acceso a su caja borneras.

Figura 2. **Composición completa de un motor eléctrico de jaula**



Fuente: Alren.es. *Despiece de motor eléctrico de aluminio.*

www.alren.es/documentacion/disenio-mecanico/despiece. Consulta: junio 2019.

Tabla I. **Lista de componentes de la figura anterior**

1. Junta de aceite	16. Bornes
2. Tornillos	17. Bloque de conexión
3. Arandela	18. Bloque de bornera
4. Cojinete o rodamiento	19. Tapa de rodamiento
5. Brida	20. Base de bornes
6. Reten	21. Prensaestopas
7. Escudo delantero	22. Escudo trasero
8. Tapa de rodamiento	23. Placa de características
9. Junta	24. Ventilador
10. Empaque	25. Tapa del ventilador
11. Tornillo	26. Empaque
12. Rotor de jaula	27. Estator
13. Arandela de fijación	28. Tapón
14. Base caja de bornera	29. Patas de sujeción
15. Tapa de caja de bornera	30. chaveta

Fuente: Alren.es. *Despiece de motor eléctrico de aluminio.*

www.alren.es/documentacion/disenio-mecanico/despiece. Consulta: junio 2019.

1.3. **Funcionamiento del motor trifásico de inducción**

El funcionamiento del motor de inducción trifásico está basado en la aplicación de la ley de Faraday, al mismo tiempo se considera la fuerza generada en un conductor descubierta por Lorenz. Otro efecto implicado es la generación del campo magnético giratorio o campo magnético rotatorio.

1.3.1. Ley de Faraday

Establece que el voltaje en un circuito cerrado es proporcionalmente directo a la rapidez en que varía sobre el tiempo la cantidad de flujo magnético atravesando una superficie bordeada, esta se describe con la siguiente ecuación:

$$\oint_C \vec{E} \cdot d\vec{l} = -\frac{d}{dt} \int_S \vec{B} \cdot d\vec{S}$$

Donde:

- E es el campo eléctrico
- dl es un elemento diferencial de longitud del circuito representado por c
- B es el campo magnético
- S es una superficie y su contorno es C

Simplificando la ecuación anterior se obtiene la forma sencilla de la ecuación matemática conocida como la ley de Faraday, conocida también como fuerza electromotriz o voltaje inducido. Esta descripción matemática es entendible si se considera el siguiente experimento: al ejercer movimiento acercándose o alejándose a un imán permanente hacia un conductor eléctrico que no se encuentre excitado por ninguna fuente de energía eléctrica, al realizar una medición con un amperímetro en el conductor, será notable que aparecerá una corriente, esta misma desaparece si el imán no varía su posición en el tiempo.

$$\varepsilon = -\frac{d\Phi}{dt}$$

Donde:

- ε es una fuerza electromotriz
- Φ es el flujo magnético

1.3.2. Ley de Lorentz

Fundamentada en que si una partícula cargada, circulando a una rapidez, en un punto donde existe presencia de un campo magnético de cierta intensidad, la partícula experimenta una fuerza llamada fuerza de Lorentz.

$$\vec{F} = q \cdot \vec{v} \times \vec{B}$$

Donde:

- F es la fuerza de Lorentz
- q es la carga de la partícula
- v rapidez de la partícula
- B el campo magnético donde se somete la partícula

Considerando una partícula bajo efecto de un campo eléctrico, fuerza eléctrica en conjunto con un campo magnético; fuerza magnética, la fuerza electromagnética total o fuerza de Lorentz sobre dicha partícula viene dada por:

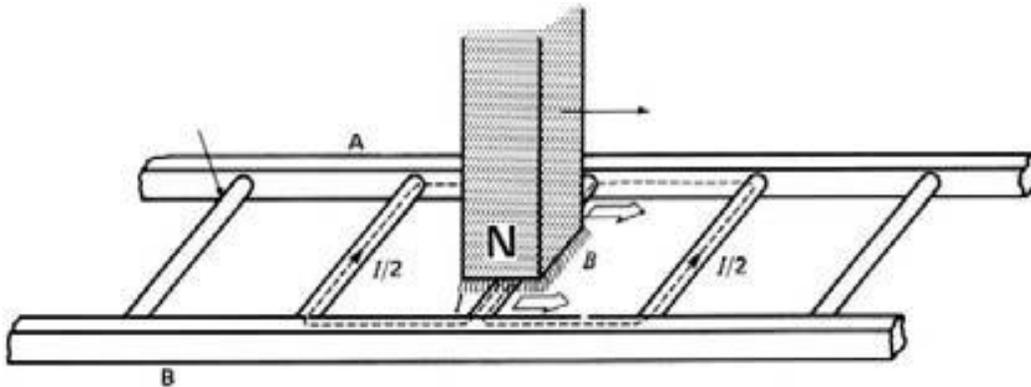
$$\vec{F} = q\vec{E} + q \cdot \vec{v} \times \vec{B}$$

La ley de Lorentz describe una fuerza inducida por el campo electromagnético que recibe una partícula cargada, o una corriente eléctrica, ésta

en conjunto con la ley de Faraday son las bases del funcionamiento de los motores eléctricos.

Para comprender la aplicación de estos teoremas a los motores eléctricos, considérese el siguiente ejemplo, una composición de conductores de longitud l , sus extremos se cortocircuitan mediante dos barras conductoras A y B, considérese la figura 3; acto seguido, se desplaza un imán permanente colocado sobre la escalera conductora, que se mueve con una rapidez v hacia la derecha, interactuando así su campo magnético B con los conductores.

Figura 3. **Imán moviéndose sobre conductores cortocircuitados**



Fuente: THEODORE, Wildi. *Maquinas eléctricas y sistemas de potencia*. p. 268.

Considerando el movimiento del campo magnético del imán sobre los conductores suceden los siguientes efectos, se induce un voltaje descrito por la ecuación $E = Blv$ en cada conductor mientras el flujo magnético este variando sobre ellos. El voltaje inducido genera al instante una corriente eléctrica I , acto seguido, esta fluye por el conductor debajo de la cara del polo N , fluyendo a través de las barras extremas y regresa a través de los demás conductores.

Si se considera que el conductor transporta corriente, ésta a su vez provoca campo magnético en el conductor; ley de Ampere. Queda bajo el efecto del campo magnético variable del imán en movimiento, experimenta una fuerza mecánica; descrita por la ley de Lorentz. Ésta fuerza actúa en dirección para arrastrar el conductor tratando de seguir el campo magnético variable.

Si la escalera conductora está libre para moverse, se acelerará hacia la derecha producto de la interacción entre campos magnéticos. Sin embargo, mientras ésta adquiera mayor velocidad, la diferencia de velocidades entre ambos; el imán y conductores será menos pronunciada, en consecuencia, el voltaje inducido E en conjunto con la corriente I disminuirán, afectando la fuerza que actúa en los conductores, por lo tanto, esta tiende a reducirse. Si existiera la condición que la escalera pueda moverse a la misma rapidez que el campo magnético, el voltaje inducido y por ende la corriente, al igual que la fuerza que arrastra la escalera serían cero dado que no existe variación de campo magnético.

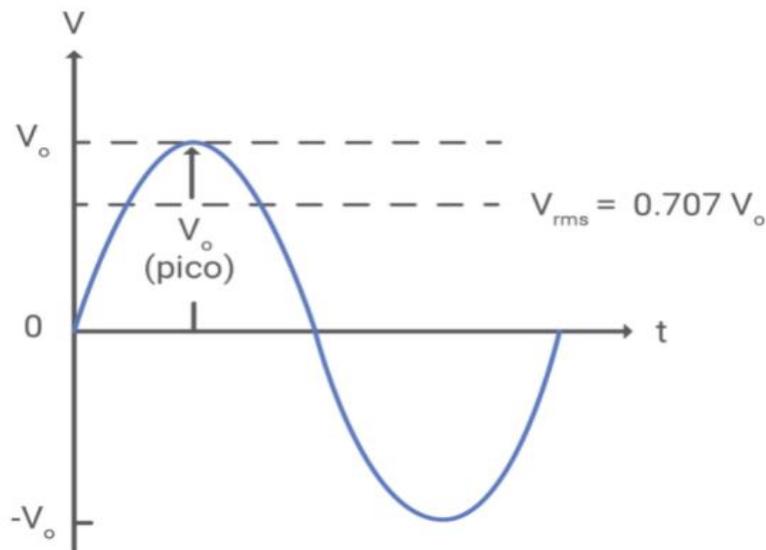
1.4. Corriente alterna

Entender el funcionamiento de los motores de inducción implica comprender el comportamiento de la corriente alterna, considerase como una variación de flujo eléctrico, donde su magnitud, como su dirección cambian periódicamente. Gráficamente, su forma es generalmente asociada a una onda senoidal, véase la figura 4, este tipo de corriente posee características particulares que las hace distinguir unas de otras. Sus características principales son:

- Forma de onda: es la curva representativa en cada instante del desarrollo de la tensión (o la intensidad).

- Amplitud: se define como la distancia entre el punto más alejado de una onda y el punto de equilibrio o medio, también se lo conoce como valor pico o valor máximo de pico.
- Frecuencia: se refiere al número de veces que se repite la onda en la unidad de tiempo
- Fase: es el ángulo descrito por la onda en un instante determinado

Figura 4. **Gráfica de corriente alterna**



Fuente: Tecdigital. *Valores eficaces de corriente y tensión.*

www.tecdigital.tec.ac.cr/repo/rea/electronica/el-2114/un_2/23_valores_eficaces_de_corriente_y_tensin.html. Consulta: 22 de junio de 2021.

1.4.1. Valor eficaz

Se refiere al valor cuadrático medio de corriente alterna, equivalente al valor de corriente directa capaz de producir la misma cantidad de trabajo en una carga resistiva determinada. Conceptualmente el valor eficaz es aplicable para estudiar formas de onda periódicas, también conocido como valor RMS; éste definido con la función:

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

Considerando la función $f(t)$ para un periodo $T = 2\pi$

$$f(t) = V_{max} \sin(t)$$

$$V_{ef} = \sqrt{\frac{1}{2\pi} \int_0^{2\pi} [V_{max} \sin(t)]^2 dt}$$

$$V_{ef} = \sqrt{\frac{V_{max}^2}{2\pi} \int_0^{2\pi} \sin^2(t) dt}$$

$$V_{ef} = V_{max} \sqrt{\frac{1}{2\pi} \int_0^{2\pi} \left[\frac{1 - \cos(2t)}{2} \right] dt}$$

$$V_{ef} = V_{max} \sqrt{\frac{1}{2\pi} \left[\frac{t}{2} - \frac{\sin(2t)}{4} \right]_0^{2\pi}}$$

$$V_{ef} = V_{max} \sqrt{\frac{1}{4\pi} (2\pi)}$$

$$V_{ef} = V_{max} \sqrt{\frac{1}{2}}$$

$$V_{ef} = \frac{V_{max}}{\sqrt{2}} \approx 0.707 V_{max}$$

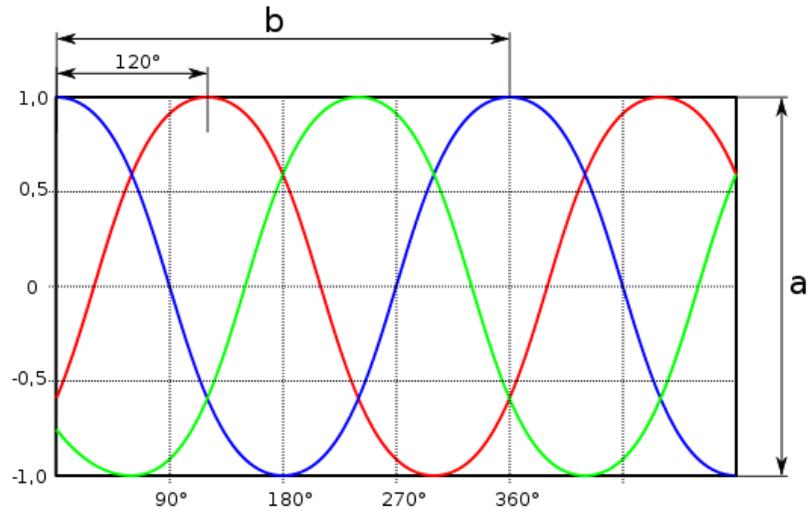
1.5. Corriente alterna trifásica

Las redes de distribución eléctrica operan con corriente alterna trifásica, y para ahorrar costos, a nivel industrial se utiliza de esta manera; la mayor parte de máquinas eléctricas funcionan con corriente alterna trifásica, está formada por tres corrientes alternas monofásicas, denominadas fases, tienen la característica de tener igual frecuencia como amplitud; por lo tanto, igual valor eficaz en las cuales existe una diferencia de 120° eléctricos.

Para generar corriente trifásica se utilizan máquinas eléctricas compuestas de tres devanados o grupos de devanados, estos se enrollan sobre piezas equidistantes entre sí. Estos son sometidos a un campo magnético que gira en el interior de la máquina, por lo tanto, considerando la ley de Faraday, se genera en los devanados tres tensiones iguales, pero desfasadas 120° eléctricos que se describen con la expresión para los tres voltajes transportados:

$$V_1 = V_{max} \sin(\omega t), \quad V_2 = V_{max} \sin\left(\omega t + 2\frac{\pi}{3}\right), \quad V_3 = V_{max} \sin\left(\omega t - 2\frac{\pi}{3}\right)$$

Figura 5. **Gráfica de corriente alterna desfasada 120° eléctricos**



Fuente: ECURED. *Corriente trifásica*. www.ecured.cu. Consulta: 10 de julio de 2019.

Dentro de las principales ventajas que tiene la energía eléctrica trifásica, se tiene el transporte de la misma. Las líneas de transmisión son beneficiadas, al conducir corriente alterna en alto voltaje reduce el calibre de los conductores que se utilizan (por el transporte de intensidad baja), esto ayuda a transportar a largas distancias utilizando bajas intensidades de corriente, evitando así pérdidas por el efecto Joule. Otra ventaja de utilizar energía eléctrica en CA es la utilización de transformadores pues estos se encuentran diseñados para operar con la misma.

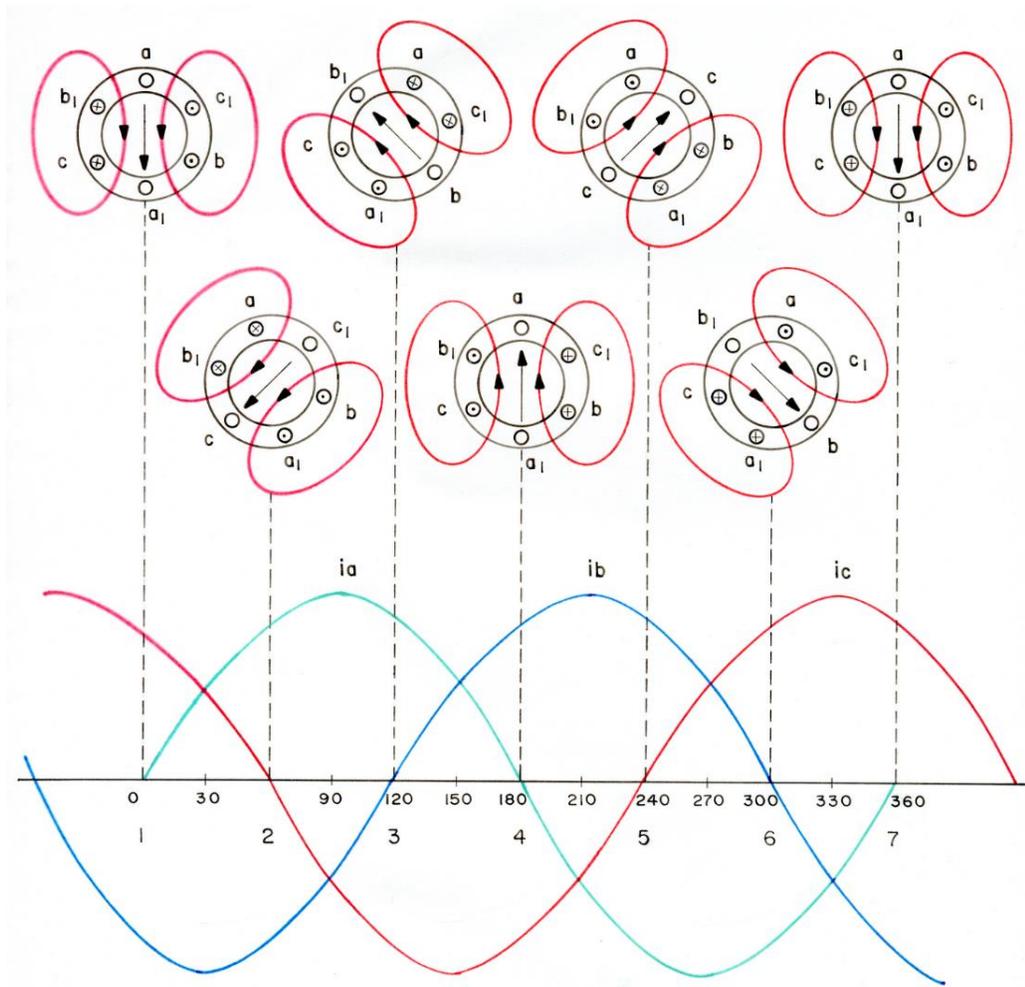
1.6. Campos magnéticos giratorios

El estator de las máquinas rotatorias construidas con el fin de operar con corriente trifásica, está formado de 3 bobinas o grupos de las mismas, separadas 120° geométricamente, al hacer circular en estas las 3 corrientes provenientes de la red trifásica equilibrada, produce una onda rotativa de f.m.m, ésta a su vez

genera un campo magnético resultante que varía senoidalmente en la periferia del estator.

La figura 6 muestra cómo se forma el campo resultante de la aplicación de corrientes equilibradas, también desfasadas 120° en el estator de un motor, este campo magnético gira a la misma velocidad angular con la que está variando la onda sinodal, esta interacción con el rotor, inducirá corrientes en la barras o devanados de cobre cortocircuitados, convirtiendo el rotor en un imán que por magnetismo acción y repulsión interactúa con el campo magnético rotatorio del estator. Generando así un movimiento en el rotor del motor el cual gira hasta alcanzar su velocidad angular nominal.

Figura 6. **Campo magnético giratorio producido en un motor de inducción trifásico de dos polos**



Fuente: JFB Ingeniería. *Campo magnético giratorio*. www.jfbingenieria.blogspot.com/2010/03/campo-magnetico-giratorio.html. Consulta: 15 de julio de 2019.

La velocidad del campo magnético giratorio se denomina velocidad de sincronismo, se expresa con la ecuación:

$$n = \frac{60f}{P} = \frac{120f}{p}$$

Donde:

f: frecuencia de la red a la que está conectada la máquina, en Hz

P: número de pares de polos de la máquina

p: número de polos de la máquina

n: velocidad de sincronismo de la máquina, en rpm

La rapidez angular del rotor es inferior a la rapidez de giro del campo magnético en el estator, velocidad de sincronismo, producto de la oposición al movimiento del rotor por los cojinetes, fricción con el aire y la carga acoplada al eje del rotor. A la diferencia de velocidades se le conoce como deslizamiento, se expresa en tanto por ciento, en referencia a la velocidad de sincronismo, viene dado por la expresión:

$$S = \left(\frac{n_s - n}{n_s} \right) \cdot 100$$

Donde:

n_s : velocidad de sincronismo

n: velocidad del rotor

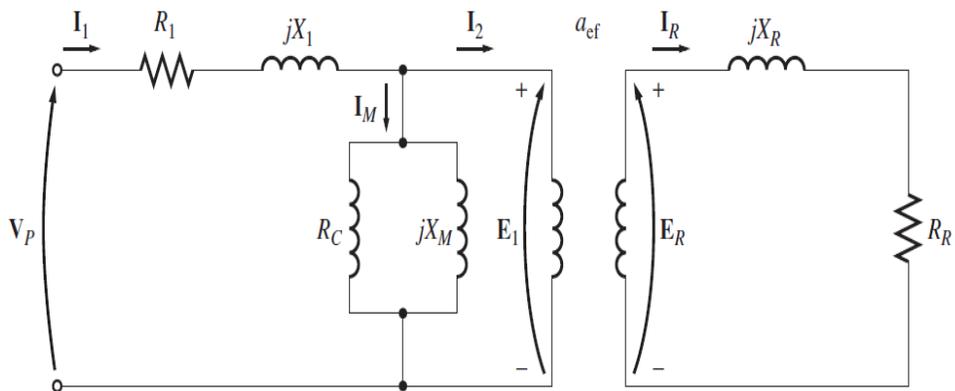
1.7. Circuito equivalente del motor asíncrono trifásico

El circuito equivalente de cualquier máquina eléctrica muestra la variedad de parámetros de la máquina, para la máquina de inducción se usa el modelo de transformador, este modelo representa el fenómeno de inducción magnética, es decir, la inducción de voltaje que genera sobre el rotor. El circuito tiene como objetivo representar una red electromagnética que explique el comportamiento

de la máquina, en la figura 7 se muestra un circuito aproximado compuesto por resistencias e inductancias, por fase, de un motor asíncrono.

Dentro de esta red se tiene, la resistencia del cobre del estator R_1 , la reactancia de dispersión generada por el flujo magnético que no sigue el camino del material ferromagnético, sino que se escapa a en el aire X_1 , la tensión del estator E_1 acoplado al secundario E_R mediante un transformador ideal con relación de vueltas a_{eff} . El voltaje E_R inducido en el rotor produce una corriente en el conductor eléctrico del rotor R_R . la componente R_C representa pérdidas en el núcleo ferromagnético del estator por corrientes de Foucault, jX_M se refiere a la reactancia de magnetización, jX_R es la reactancia de dispersión del rotor. Las corrientes I_1 corresponde a la corriente de línea, I_M es la corriente de magnetización. I_R es la corriente total del rotor, I_2 es la corriente del estator.

Figura 7. **Circuito equivalente del motor de inducción por fase**



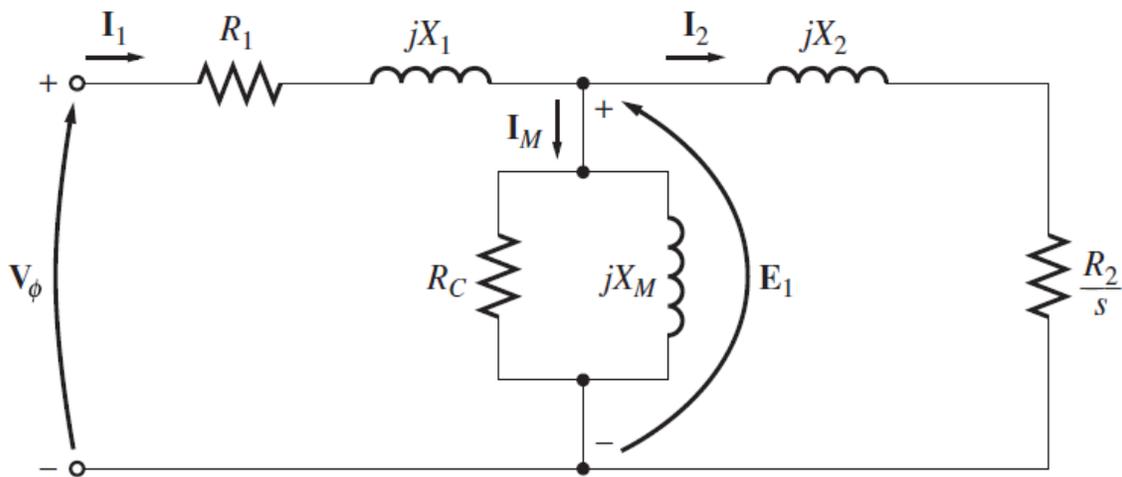
Fuente: CHAPMAN, Stephen J. *Maquinas Eléctricas*. p .237.

El circuito equivalente de la maquina es considerablemente sencillo, se obtiene de los ensayos de vacío; motor sin carga mecánica, ensayo a corto circuito; con el rotor bloqueado, si las medidas o cálculos son realizados

precisamente, es posible calcular de igual manera las corrientes por fase del motor, factor de potencia, par motor, pérdidas y rendimiento del motor.

Simplificando el circuito, donde es notable que esta máquina se puede describir como un transformador que rota donde su entrada es un sistema de corriente trifásica. La diferencia con el transformador, se encuentra en la salida del circuito, esta representa potencia eléctrica desde los devanados secundarios, en el motor, las salidas en los devanados secundarios se encuentran en corto circuito, de tal forma que no existe salida de potencia eléctrica, es potencia mecánica, por tal motivo es posible establecer el siguiente circuito equivalente más aproximado.

Figura 8. **Circuito equivalente por fase más aproximado**



Fuente: CHAPMAN, Stephen J. *Maquinas Eléctricas*. p. 240.

1.7.1. Arranque del motor de inducción

Un motor trifásico, cuando inicia su arranque, es necesario excitarlo con un flujo de corriente mucho mayor a la corriente nominal. La corriente requerida es considerablemente más grande, a tal punto de ser capaz de provocar una caída de tensión que afecte al funcionamiento de los receptores del entorno. Para minimizar estos problemas, se crearon reglas o normas que limitan el uso de motores que arranquen de forma directa, si superan cierta potencia.

Considerando que, en motores de rotor tipo jaula, únicamente se puede acceder a sus terminales del devanado del estator mediante la caja de bornes. La variedad de arranques, básicamente consisten en hacer variar la tensión de alimentación del estator, para ello, es necesario considerar la intensidad de arranque de un motor de inducción, esta, al ser más alta que la intensidad nominal, extender el tiempo de arranque, produce considerable elevación de temperatura, afectando de manera perjudicial los devanados de cobre y aislante del motor. Por lo tanto, puede afectar perjudicialmente no solo a la maquina sino también, la instalación eléctrica en su entorno.

Para reducir el impacto producto de corrientes en el instante de arranque de un motor, se emplean técnicas de arranque. En función de su rotor, entiéndase la construcción de su rotor de jaula de ardilla o con anillos deslizantes, dentro de sus principales métodos de arranque se encuentran los siguientes:

- Arranque directo
- Arranque estrella delta
- Arranque mediante resistencias estatificas
- Arranque por autotransformador
- Arranque por rotor bobinado

- Arranque electrónico

1.8. Placa característica del motor

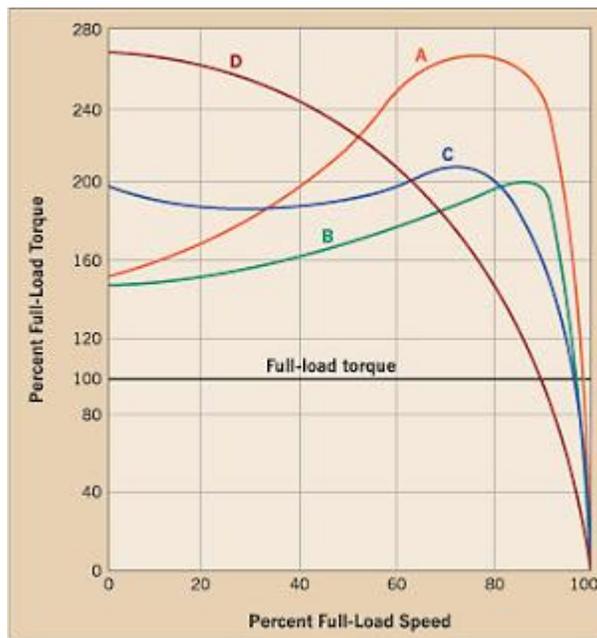
De acuerdo a los estándares industriales DIN y NEMA, el motor debe contener una placa con datos llamada placa característica, esta tiene que estar sujeta al motor con remaches del mismo material que las placas, se tiene que colocar de modo que pueda leerse incluso cuando la máquina se encuentre en servicio. Los datos e información que mínimamente debe contener la placa de datos, igual que placas auxiliares, considerando cualquier motor de corriente alterna monofásico o trifásico, en forma indeleble, en lugar visible, son los siguientes: (el orden no tiene relevancia, en generalmente es a criterio del fabricante).

- Nombre del fabricante
- Tipo de maquina
- Clase de aislamiento
- Potencia nominal
- Intensidad de corriente nominal
- Tensión nominal
- Factor de potencia
- Corriente de arranque
- Velocidad nominal
- Frecuencia nominal
- Eficiencia
- Carcaza o *frame*
- Tipo de servicio

1.9. Aplicaciones de las máquinas de inducción

La norma NEMA clasifica los motores en diferentes categorías, las cuales se diferencian principalmente por el par de arranque, esto se logra modificando a nivel constructivo la disposición, en conjunto con la geometría del rotor tipo jaula, obteniendo más capacidad al aplicar mayor o menor corriente. En la figura 9 se muestran los distintos tipos de comportamiento de los motores, siendo el eje horizontal el porcentaje de velocidad de trabajo, y en el eje vertical, el torque de la carga aplicado.

Figura 9. **Curvas del par de torsión en función de la velocidad**



Fuente: Apoyo Electrico. *Motores y Controles Eléctricos*. www.apoyoelectrico.blogspot.com/2016/03/placa-caracteristica.html. Consulta: 13 de junio de 2018.

La aplicación de motores trifásicos de inducción, es ampliamente variada, a nivel doméstico o aplicaciones industriales. Algunos de las aplicaciones de motores trifásicos asíncronos, tratándose de corriente trifásica de tipo inducción de rotor jaula de ardilla son los siguientes:

Para motores NEMA clases A y B, esta normado que estos aplican relativamente par de arranque pequeño, en condiciones de estado estacionario, sus aplicaciones van desde maquinas o herramientas, bombas centrífugas de impulsión, ventiladores, y sopladores.

Para un motor nema con clasificación C, en aplicaciones para accionar cargas tipo mecánicas que requieren mayor par mecánico, a su vez menor corriente durante el arranque, sus aplicaciones se limitan a situaciones en las que es considerado difícil el arranque como en bombas hidráulicas con carga, molinos, compresores de pistón, elevadores, y bandas transportadoras.

Para motores NEMA clasificados D, con enorme par de arranque, a la vez baja corriente al momento del mismo, sus principales aplicaciones se enfocan para servicios pesados en arranque, por ejemplo, cargas para troquelar, estas necesitan el alto par con aplicación a carga instantánea. En otras palabras, se utilizan en dispositivos de uso intermitente, entre aplicaciones de estos dispositivos se tienen grúas, prensas, taladros y cortadoras.

Otras aplicaciones de índole industrial, son dispositivos de triturado, embotelladoras e hiladoras. Otras aplicaciones laborales más específicas donde se utilizan este tipo de máquinas eléctricas son, industria textil, papelera, petrolera e industria cementera entre otras.

Las aplicaciones de un motor trifásico de inducción no se limitan a usos en aserraderos, molinos de grano y fábricas de pólvora, una característica favorable que tienen estos en la operación de cada uno de estos trabajos es la baja tasa de incendios producto del material que trabaja, considerados altamente inflamable.

1.10. Sistema de monitoreo

La mayoría de fallos en máquinas rotatorias ocurren según el funcionamiento anormal de los componentes internos del motor (según Mario Castelli cerca del 90 %). Considerado que el mantenimiento correctivo involucra paradas no programadas o maniobras costosas de operación del motor o elementos que deriven en el funcionamiento continuo de este.

Los requerimientos actuales de calidad consideran cada vez más necesaria la utilización de sistemas con capacidad de monitorizar o detectar fallas, ayudando a reducir significativamente la interrupción de la producción. Un sistema de monitoreo es la combinación de componentes que actúan conjuntamente, cumplen el objetivo de observar el curso de uno o más parámetros para detectar problemas, a la vez que, es posible prevenir eventuales anomalías.

Dado que los motores eléctricos asíncronos se encargan del correcto funcionamiento a nivel producción en la industria. Es necesario determinar o diagnosticar fallas. Con el fin de prevenirlas, se utilizan sensores o dispositivos adecuados, estos pueden ser empleados para recolectar datos de parámetros durante el funcionamiento en motores eléctricos.

1.11. Parámetros implicados

Considerado como imprescindible y orientativo un parámetro es un dato primordial para tasar o apreciar una cierta circunstancia. desde, una cierta situación puede comprenderse o ubicarse en contexto.

Aplicados a ingeniería eléctrica principalmente considerando motores eléctricos, entre los principales parámetros que se pueden obtener ya sea directa o indirectamente, mediante sensores resaltan principalmente por facilidad de acceso los siguientes:

1.11.1. Tensión eléctrica

También denominada voltaje y en otras ocasiones diferencia de potencial, es una magnitud física que impulsa a los electrones a lo largo de un conductor en un circuito eléctrico cerrado, provocando el flujo de una corriente eléctrica. En el Sistema Internacional de Unidades se mide en Voltios.

1.11.2. Corriente eléctrica

Conocido también por intensidad eléctrica, hace referencia a la carga eléctrica atravesando una sección del conductor en la unidad de tiempo. En el Sistema Internacional de Unidades se expresa en C/s, siendo mejor conocido como amperio.

1.11.3. Potencia eléctrica

Es la relación de paso de energía de un flujo por unidad de tiempo, indica, la cantidad de energía entregada o absorbida por un elemento en un tiempo determinado, cuando se trabaja en corriente alterna esta se clasifica en 3 tipos de potencias:

1.11.3.1. Potencia activa

También llamada potencia media, real o verdadera dado a que es la que se aprovecha como potencia útil, es debida a los dispositivos resistivos en la red eléctrica y su unidad de medida es el vatio.

1.11.3.2. Potencia reactiva

Es la potencia que demandan las cargas inductivas y capacitivas en el circuito, cumple la función de generar campo magnético o campo eléctrico respectivamente, esta no se transforma en trabajo efectivo, su unidad de medida es el voltamperio reactivo.

1.11.3.3. Potencia aparente

Se refiere a la potencia total consumida por la carga, se puede obtener como la suma vectorial de las potencias activa y reactiva, también se puede obtener realizando el producto de los valores eficaces de la intensidad por la tensión. Su unidad de medida es el voltamperio.

1.11.3.4. Factor de potencia

Se obtiene como la relación entre la potencia activa, y la potencia aparente, da una medida de la capacidad de una carga de absorber potencia activa. Esta es una cantidad adimensional.

1.11.4. Energía eléctrica

Energía es la capacidad de realizar un trabajo en un periodo de tiempo, en la rama eléctrica, se define como el trabajo realizado por una maquina eléctrica en una hora, su medición se estandariza como kilovatio-hora.

1.11.5. Frecuencia en corriente alterna

Constituye un fenómeno físico que se repite cíclicamente un número determinado de veces durante un segundo de tiempo y puede abarcar desde uno hasta millones de ciclos por segundo o hercios.

1.11.6. Velocidad del rotor

Número de giros que el rotor realiza en un minuto de tiempo, se expresa en revoluciones por minuto. Para los motores de CA, cualquier cambio en la frecuencia provoca un cambio proporcional en la velocidad del rotor.

1.11.7. Temperatura

El termino temperatura en la mayoría de textos se define como una magnitud física que indica la cantidad de calor de un cuerpo, objeto o del medio ambiente. El aumento de temperatura por efecto de pérdidas eléctricas y

mecánicas están relacionadas a las condiciones de trabajo del motor. En condiciones normales de operación, la máquina puede recalentarse principalmente si opera ininterrumpidamente. La temperatura se mide en el Sistema Internacional de Unidades en grados kelvin sin embargo la escala más utilizada es la de grados Celsius también llamada centígrada.

Para obtener mayor información del funcionamiento se necesita monitorizar más parámetros de la máquina. No obstante, es contraproducente porque la cantidad y variedad de sensores a utilizarse en motores eléctricos. Pueden aumentar el costo y dificultar de la sensorización requerida, también es considerable que algún elemento en los dispositivos del sistema de monitoreo falle, pudiendo esta ocasionar una detención no programada.

La medición de temperatura y el análisis de la vibración ha sido una técnica frecuentemente utilizada para el monitoreo de motores de inducción. Pero, en la actualidad es considerable el monitoreo de al menos la parte eléctrica del motor, principalmente para analizar el comportamiento de la corriente del bobinado del estator. La mayoría de análisis sugieren evaluar el espectro de frecuencia de la corriente del estator, esto contribuye a predecir fallas en el motor.

La mayoría de las técnicas de monitoreo incluyen sensores vulnerables o de complicada instalación en la máquina, es considerable también un costo elevado. No obstante, monitorear la corriente del estator puede proveer información muy útil, con la ventaja de no invadir internamente el motor, reduciendo así costos de instalación.

1.12. Fallas en motores eléctricos de inducción

Los motores eléctricos pueden fallar por diferentes causas, por sobre tensión, sobrecarga mecánica, por falta de mantenimiento entre otros, estas se pueden clasificar en 2 grupos: fallas eléctricas que suelen afectar el aislamiento, este tipo se caracteriza por dañar parcial o totalmente el bobinado del estator, se conocen como fallas de cortocircuito en el devanado del estator. Las fallas mecánicas, son otro tipo de problemas en los motores, suelen afectar el rotor o componentes asociados al mismo.

Las fallas de índole mecánico en motores eléctricos provocan daños en rodamientos, quiebre de barras o anillos del rotor, entrehierros irregulares principalmente excentricidades estáticas y dinámicas, lo que conlleva a desbalances del mismo.

Los problemas por sobrecalentamiento en motores se producen principalmente por obstrucción del ventilador, otros influyentes considerable son sobrecargas en la maquina o falta de ventilación externa, estos motivos causan mayor parte de los daños de la parte eléctrica en motores de inducción.

2. ARDUINO

Es una placa electrónica compuesta por dispositivos integrados, así también una parte de software de código abierto, diseñado para aplicaciones en electrónica o proyectos aplicados a varias disciplinas. Tiene la capacidad de detectar el medio ambiente, recibiendo señales en sus entradas de un conjunto de sensores. En función de su programación pueden influir sus alrededores controlando directa o indirectamente luces, señales, motores y otros actuadores.

2.1. Tipos de Arduino

A partir del 2005, desde que salieran a la venta en Italia los modelos de Arduino, se categorizan en placas de desarrollo, placas de expansión o *shields*, kits, accesorios e impresoras en tres dimensiones. Arduino cuenta con diferentes prototipos y modelos de placas a elegir, en función de la complejidad de la aplicación asociada a esta. Cada modelo cuenta con características particulares con pequeñas variantes que las diferencian de las demás.

La popularidad de la placa es tal que, se tienen *shields* oficiales, así como *shields* modificadas por la comunidad aprovechando que se trata de hardware y software libre; dentro de las placas oficiales más utilizadas se encuentran las siguientes:

2.1.1. Arduino UNO

Arduino de gama de entrada, todas las *shields* se diseñan para utilizarse sobre esta placa, puesto que se considera el modelo estándar, cuenta con

14 pines entrada/salida digital, de las cuales 6 se pueden usar como PWM, además cuenta con 6 entradas analógicas, también cuenta con I2C, más adelante se especifican datos acerca de este tipo de placa.

2.1.2. Arduino DUE

Considerado como la primera placa basada en un microcontrolador AT91SAM3X8E de 32 *Bits* ARM, tiene 54 entradas/salidas digitales, tiene 12 entradas analógicas, utilizables como PWM, 2 buses TWI, SPI y 4 UARTs. Funcionan todos los módulos basados en 3,3 V, no soporta 5V este puede dañar la placa. Tiene la capacidad de suministrar 130mA en sus pines, es capaz de funcionar a 84 MHz, posee adicionalmente interno, dos puertos USB para poder controlar periféricos.

2.1.3. Arduino TRE

Considerada la primera placa fabricada en Estados Unidos, se encuentra basado en un procesador Sitara AM335x capaz de alcanza 1 GHz, 512 Mb de RAM DDR3L, puerto Ethernet 10/100, 1 puerto USB 2.0, 4 puertos USB Host, salida de vídeo HDMI y audio, también contiene un zócalo para tarjetas Micro-SD.

2.1.4. Arduino Leonardo

Placa que incluye 12 entradas analógicas, 20 entrada/salidas digitales. A diferencia del resto de placas, con el microcontrolador ATmega32u4 no posee un procesador secundario para controlar el puerto USB. Por lo tanto, no necesita de ninguna conversión serie-USB además tiene más pines de interrupciones externas, cuenta con comunicación TWI, SPI y dos UART.

2.1.5. Arduino MEGA 2560

Arduino basado en un microcontrolador ATmega2560 de ahí su nombre. Dotado con 54 entradas/salidas digitales, 16 con función PWM, incluye 16 entradas analógicas, 4 UART, agrega dos modos PWI y otro SPI. Cada pin puede suministrar hasta 40 mA, tiene 6 interrupciones externas, es considerable su compatibilidad con todos los *shields* de Arduino.

2.1.6. Arduino Mega ADK

Placa con características iguales que el Arduino Mega 2560, con la variación de incluir conexión USB *Host* interface, para conectarlo con dispositivos basados en Android basados en el chip MAX3421e. capacitado para habilitar su aplicación con la Android Open *Accessory Development Kit*.

2.1.7. Arduino BT

Originalmente basado en el ATmega168, con un oscilador a 16 MHz agrega el módulo Bluegiga WT11 compatible con comunicación en serie inalámbrica Bluetooth, integra 14 pines entradas/salidas digitales; 6 con capacidad de PWM, 6 entradas analógicas y un encabezado ICSP, capaz de funcionar a 5 V, con capacidad de suministrar en sus pines de salida un máximo de 40 mA.

2.1.8. Arduino Micro

Acerca de sus características es similar al Arduino Leonardo, la diferencia es el tamaño con el que está construido. Es compatible con las *Shields* de Arduino, sin embargo, físicamente se debe instalar de forma externa, en

concreto, requiere cableado por el usuario, funciona a 5 V y puede suministrar hasta 40 mA en sus pines.

2.1.9. Arduino Nano

Potenciado con un ATmega328. Es similar en cuanto a características al Arduino UNO, las diferencias son el tamaño, tiene solo un conector de alimentación de CC capaz de funcionar con un cable USB Mini-B, es compatible con la mayoría de *shields*, aunque de la misma forma que el Arduino Micro. Consume para funcionar 19 mA y funciona a una tensión de 5 V.

2.1.10. Arduino YUN

Su principal característica consiste en modular su trabajo por separado, a la misma vez de forma complementaria, cuenta con la versatilidad de un Arduino, con un microprocesador ATmega 32 u 48 a una frecuencia de 16 Mhz, por otro lado, un dispositivo con microprocesador Atheros AR9331, capacitado para trabajar con Lilino; Linux basado en OperWrt a 400 Mhz. Las características de la placa son comparables a la placa Leonardo. incluyendo Ethernet, *slot* SD y WiFi incluidos, manipulados mediante Lilino, es compatible con todas las *Shields* y es capaz de trabajar por separado. Por estas características es aconsejable para usar al diseñar dispositivos conectados en red, más en general, proyectos de Internet de las cosas.

2.1.11. Arduino Ethernet

Incluye el microcontrolador ATmega328, básicamente igual que el modelo Arduino UNO con el agregado de conector de red RJ45 reduciendo así el espacio de instalación. Dotado con 14 pines entradas/salidas, 6 entradas con capacidad

analógica, un cristal a 16 MHz, conexión de alimentación, zócalo ICSP, agregando un botón de *RESET*. Es necesario considerar que los pines 10, 11, 12 y 13 están reservados para la interfaz Ethernet, es considerable que estos no se deben utilizar, esto reduce la cantidad de pines disponibles a 9 incluyendo 4 con soporte PWM.

2.1.12. Arduino FIO

Placa Arduino basado en un microcontrolador ATmega328p. Trabaja a 8Mhz, un voltaje de 3,3 V, tiene 14 pines de entradas/salidas digitales; 6 se pueden usar como salidas PWM, puede suministrar 40mA en sus pines, tiene 8 pines de entrada analógicas, incluye un conector para batería y su propio módulo de carga, como un espacio dedicado a instalar un módulo de comunicaciones xBee. Tiene UART TTL e interrupciones permitiendo así ponerlo en modo *Sleep*.

Además, posee tanto TWI (I2C) como SPI. La principal ventaja de este modelo es el bajo consumo durante el modo *Sleep* y la capacidad de programarlo mediante XBEE, sin necesidad de conexión físicamente al computador. Está diseñado para aplicaciones inalámbricas.

2.1.13. Arduino ZERO

Placa de 32 bits basada en un microcontrolador ATMEL Cortex M0 SAMD21 incorpora tecnología EDBG, funciona a 48 MHz, contiene 20 entradas/salidas digitales, también contiene 10 pines capaces de aplicar PWM, el voltaje de operación de estas es de 3,3 V se considera más orientado a proyectos de tipo wearable o Internet de las cosas.

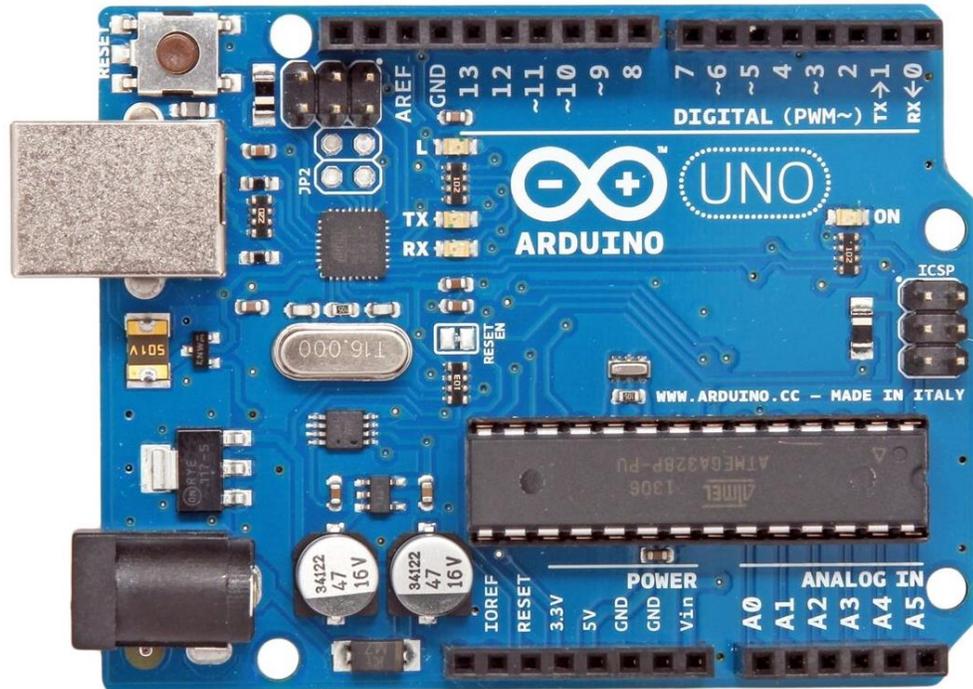
2.2. Características Arduino UNO

Arduino se ha vuelto muy popular entre personas que empiezan con la electrónica, por varias razones, pero destacando que, a diferencia de la mayoría de circuitos programables, Arduino no necesita una parte separada de hardware; un programador para cargar nuevo código al microcontrolador, solo es necesario un cable USB.

Pensado para incluir el hardware necesario y conectarse en conjunto a componentes externos, Arduino UNO es de los tipos más comercializados a nivel mundial, gracias a su simpleza, bajo precio a la vez que amplia disponibilidad. Se utiliza esta variante de Arduino para realizar este sistema de adquisición de señales dado a las ventajas de disposición comercial, capacidad de entradas/salidas, velocidad de transmisión, mediante conexión USB.

Arduino Uno es una PCB con componentes electrónicos, monta sus bases con un microcontrolador ATmega328. Dispone de 14 entradas/salidas digitales, entre estas 6 con posibilidad de PWM (Modulación por ancho de pulsos) de salida, es considerable que los 6 pines restantes soportan señales análogas. Agrega un resonador cerámico de 16 MHz, conexión USB, entrada de alimentación, también posibilidad de cabecera ICSP, con un botón para *reset* físico. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo: led, capacitores, un regulador LM78xx, resistencias, transistores y diodos; solo requiere conectarse a un ordenador con un cable USB o a la corriente eléctrica a través de un adaptador de corriente alterna CA/CD

Figura 10. **Arduino UNO R3**



Fuente: CINJORDIZ, Cesar. *Arduino Uno R3*. www.infootec.net. Consulta: 5 de julio de 2018.

2.3. Datos técnicos del Arduino UNO

La placa Arduino UNO R3, dispone de 14 pines que pueden configurarse como entrada o salida, tales con posibilidad de conexión hacia cualquier dispositivo con la funcionalidad de transmitir o recibir señales digitales de 0 o 5 V, las aplicaciones de estos pines pueden proporcionar amplia variedad de funciones y se dividen en la siguiente categoría:

2.3.1. Pines digitales

Estos pines de la placa Arduino UNO pueden ser utilizados como entradas y salidas de propósito general, para este propósito se utilizan los comandos, por ejemplo: `pinMode ()`, `digitalWrite ()` o `digitalRead ()`. Estos indican la función a disponer de estos, técnicamente, las terminales tienen una resistencia interna conocida como *pull-up* de entre 20 K Ω o 50 K Ω , la misma se encuentra desconectada, a menos que se indique lo contrario, mediante el comando `digitalWrite ()`. Cada pin puede soportar y suministrar hasta 40 mA.

Utilizados para recepción RX o transmisión TX de señales TTL, aparte de ser pines digitales; 0 o 1 respectivamente considerados normales o corrientes, pueden configurarse como pines capaces de producir ondas de pulsos con modulación PWM, con las mismas capacidades de entrada como salida de cualquier otro pin digital, pueden configurarse para un uso diferente. Están preparados para enviar y recibir comunicaciones a otros dispositivos, que por su parte son capaces de realizar el mismo tipo de comunicaciones.

Los pines 2 y 3 están preparados para generar interrupciones en el microcontrolador. Aplicables principalmente para percibir cambios de estado en su entrada, pueden excitarse cuando se encuentra un valor bajo en estas entradas, como flancos de subida o bajada en el modo entrada.

Respecto a los pines 3,5,6,9,10 y 11, se refiere a salidas PWM de 8 bits, en consecuencia, tienen la capacidad de enviar señales como modulación por ancho de pulsos.

2.3.2. Pines analógicos

Arduino UNO solo entiende señales digitales, significa que para comprender señales analógicas es necesario convertirlas en digitales, la placa contiene un conversor ADC de 6 canales. El conversor tiene una resolución de 10 bits; también se puede utilizar en un modo de 8 bits, devolviendo enteros entre 0 y 1023. Estos pines analógicos de Arduino implementan las funcionalidades de los pines digitales. Por lo tanto, si se requiere más pines digitales se pueden usar los pines analógicos. La etiqueta para los pines analógicos es A0, A1, A2, A3, A4 y A5.

2.3.3. Pines de alimentación

Las placas de Arduino tienen dos diferentes valores de voltajes de entrada a través de sus pines, estos se encuentran en el apartado marcado como POWER donde es posible encontrar los pines:

- VIN: con la capacidad de operar entre 6 y 12 Voltios, siendo el voltaje de entrada a la placa Arduino, necesario cuando está siendo alimentado por una fuente externa; sino se está alimentando mediante la conexión USB. Es considerable, al alimentar la placa con menos de 7 voltios, la salida del regulador de tensión a 5 voltios podría verse mermada.
- 5V: es un voltaje de salida, este puede venir de la terminal VIN por medio del regulador integrado en la placa o puede ser suministrado mediante la conexión USB, el voltaje está regulado a 5 Voltios DC en consecuencia no varía su valor en funcionamiento normal.

- 3,3 V: es una fuente de corriente directa con un valor de 3,3 Voltios, generado por el chip regulador de voltaje, la corriente máxima suministrada por el pin puede ser hasta 50 mA.
- GND: son pines que es posible utilizar como tierra o masa.

2.3.4. Otros pines

La placa Arduino tiene otras funciones para lo cual existen algunos pines los cuales tiene la capacidad de realizar otras funciones que incluyen:

- AREF: es un voltaje de referencia utilizado en las entradas análogas. Se puede utilizar mediante la función `analogReference`.
- Reset: Se utilizar para agregar un botón de reinicio adicional al que tiene incluida la placa.
- Pines I2C: (SCL) o (SDA) se encuentran en A4 y A5 respectivamente en la revisión R3. Estos soportan comunicación I2C se utiliza principalmente en aplicaciones internas para la comunicación entre diferentes partes de un circuito, por ejemplo, entre un controlador junto a circuitos periféricos integrados.
- SPI: se comportan como bus serie con arquitectura muy parecida a maestro-esclavo, 10 (SS) selector esclavo, 11 (MOSI) salida maestra/entrada esclava, 12 (MISO) entrada maestra / salida esclava, 13 (SCK) reloj serie.

2.4. Microcontrolador

Es un circuito integrado con capacidad de aceptar programación para ejecutar órdenes almacenadas en su memoria. Trabaja principalmente con tres bloques funcionales que ejecutan una tarea específica, este funcionamiento se asemeja a un ordenador, pero a escala muy básica; microcomputadora, siendo estas; la unidad central de procesamiento, memorias RAM y ROM, incluyendo periféricos de entrada/salida.

Los microcontroladores más usados en las plataformas Arduino son el Atmega168, Atmega328, Atmega1280, ATmega8, actualmente se está ampliando a microcontroladores Atmel con arquitectura ARM como el Atmel SAMD21 o los ST STM32, y también Intel. Arduino UNO tiene 2 variantes, Arduino UNO convencional; microcontrolador Atmega328P en formato DIP o Arduino UNO SMD con microcontrolador Atmega328P.

Un aspecto importante a considerar en estas placas es la memoria, Arduino tiene tres tipos de memoria:

- SRAM: es un tipo de memoria RAM sus siglas provienen de *Static Random Access Memory*, basada en operación con semiconductores, es memoria estática de acceso aleatorio, tiene la capacidad de mantener los datos, con la condición de ser alimentada por la fuente, es donde se indica al Arduino crear y manipular las variables durante su ejecución. Es necesario considerar, que es un recurso limitado que debe gestionarse, especialmente si se utilizan *strings* o cadenas de caracteres de forma intensiva.
- EEPROM: son siglas de *Electrically Erasable Programmable Read-Only Memory*, ROM programable y borrable eléctricamente. Se trata de memoria

no volátil para mantener datos después de un reinicio. Esta se graba desde el programa del microcontrolador, este tipo de memoria tiene un límite de lecturas/escrituras.

- Flash: Tipo de memoria utilizada para almacenar los programas cargados al microcontrolador, en esta memoria también se almacena el *bootloader*. Es posible ejecutar un programa desde la memoria flash, pero no se pueden modificar los mismos, es necesario copiarlos en la SRAM para realizar modificaciones.
- La memoria Flash y la EEPROM son no volátiles, en definitiva, la información continua tras el apagado del Arduino.

2.5. Componentes en la placa base

A nivel de hardware los pines como microprocesador, no son capaces de operar por si solos, estas placas también contienen una variedad de dispositivos para complementar el funcionamiento o para señalización. Dentro de los principales componentes, según la placa Arduino UNO R3; existen algunas revisiones y es posible que cambien el modelo o capacidad de algún componente, se tienen los siguientes componentes básicos:

- Led indicador de recepción (Rx) y transmisión (Tx), encienden respectivamente cuando la tarjeta se envía o recibe datos con el PC.
- Led ON, indica cuando el Arduino está encendido.

- Regulador de voltaje de 5 V generalmente NCP1117, se encarga de convertir el voltaje ingresado por el *plug* de alimentación a un voltaje estabilizado con una corriente de hasta 1A.
- Regulador de voltaje LP2985 de 5 V a 3,3 V capaz de suministrar una corriente de alimentación máxima de 150 mA.
- Microchip de 8-bit ATmega16U2 su función es convertir la comunicación del puerto USB a comunicación serial.
- Condensadores de 47 μ F de capacidad y 25 V, Arduino no utiliza ese límite de voltaje.
- Fusible re-armable de intensidad máxima 500 mA. Incorporado para proteger tanto la placa Arduino como el bus USB de sobrecargas y cortocircuitos.
- Resonador cerámico de 16 MHz para el microcontrolador ATmega328. Los resonadores cerámicos son menos precisos que los cristales osciladores, se trata del pequeño, el cristal grande es de uso del microchip ATmega16U2 y su reloj interno.
- Encapsulados de resistencias.
- Diodo M7 en la entrada de alimentación de la placa. Establece el sentido de circulación de la intensidad. Si en algún caso se produce una contracorriente, cumple la función de bloquearla impidiendo que afecte a la fuente de alimentación.

Tabla II. **Especificaciones técnicas del microcontrolador**

Microcontrolador	Atmega328p	Atmega328
Voltaje de operación	5V	5V
Voltaje de entrada	7-12V	7-12V
recomendado		
Voltaje de entrada limite	6-20V	6-20V
Pines de entrada y salida digital	14 (6 proporcionan PWM)	14 (6 proporcionan PWM)
Pines de entrada analógica	6	6
Intensidad de corriente pin IO	20mA	40mA
Intensidad de corriente pin 3.3V	50mA	50mA
Memoria Flash	32 KB (0.5 KB reservados para el bootloader)	32 KB (2 KB reservados para el bootloader)
SRAM	2KB	2KB
EEPROM	1KB	1KB
Frecuencia de reloj	16 MHz	16 MHz

Fuente: elaboración propia, utilizando Microsoft Excel 2016.

2.6. Entorno de programación Arduino IDE

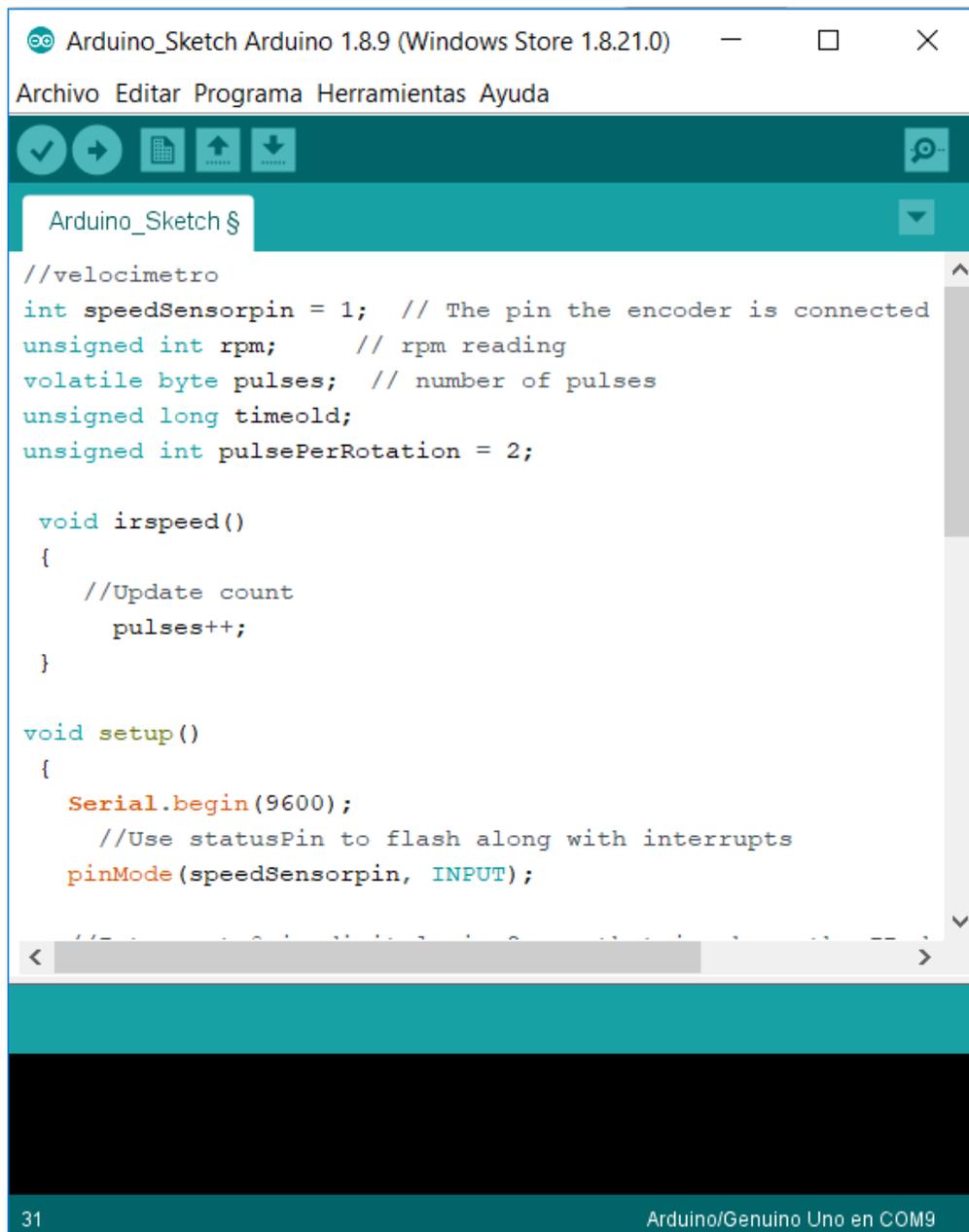
Cualquier placa Arduino desde el punto de vista de hardware, no puede realizar funciones, ni trae alguna por defecto, no tiene capacidad de tomar decisiones o mostrar datos de manera independiente. Todas estas funciones se realizan por medio del programa que se ejecuta en el microcontrolador. El lenguaje de programación utilizado en las placas se llama *Processing* y está basado en el lenguaje C.

IDE de la sigla en inglés *Integrated Development Environment*, es un entorno de desarrollo integrado o software compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para variedad de los mismos.

Diseñado como un *software* de aplicación, el Arduino IDE, se compone de un editor de código, un compilador, depurador y constructor de interfaz gráfica GUI. Soporta funcionalidades aplicadas para las placas Arduino utilizando herramientas para introducir el programa compilado en la memoria flash del hardware mediante el puerto serie.

Actualmente es posible utilizar este software de dos maneras, desde algún navegador web por ejemplo Chrome o Firefox, o de manera local, esto es sin la necesidad de conexión permanente a internet, solo es necesario descargar e instalar el software de IDE, en los sistemas operativos Windows, Mac Os X y distribuciones Linux. La figura 11 muestra una captura de pantalla con Windows, en un IDE de Arduino instalado localmente.

Figura 11. **Arduino IDE**



The image shows a screenshot of the Arduino IDE interface. The window title is "Arduino_Sketch Arduino 1.8.9 (Windows Store 1.8.21.0)". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for checkmark, back, grid, upload, download, and search. The editor shows a sketch named "Arduino_Sketch §" with the following code:

```
//velocimetro
int speedSensorpin = 1; // The pin the encoder is connected
unsigned int rpm;      // rpm reading
volatile byte pulses; // number of pulses
unsigned long timeold;
unsigned int pulsePerRotation = 2;

void irspeed()
{
  //Update count
  pulses++;
}

void setup()
{
  Serial.begin(9600);
  //Use statusPin to flash along with interrupts
  pinMode(speedSensorpin, INPUT);
}
```

The status bar at the bottom left shows the page number "31" and the bottom right shows "Arduino/Genuino Uno en COM9".

Fuente: elaboración propia, utilizando Arduino IDE 1.8.9.

Una vez instalado el IDE, se tiene que seleccionar el tipo de placa y el puerto serie que utilizara el ordenador para la conexión con el hardware, ambas opciones se encuentran en la pestaña herramientas.

Acercas de la figura 11, el software Arduino IDE está formado de 3 partes principales:

- Barra de menú con botones de acceso rápido: estos se encuentran en la parte superior de la interfaz, la barra de menú contiene las pestañas Archivo, Editar, Programa, Herramientas y Ayuda, mientras que los botones de acceso rápido tienen botones especiales, estas funciones son verificar, subir, nuevo, abrir o salvar.
- Editor de Texto o Editor de programación: es la parte más amplia del Arduino IDE, y esta designada para escribir las líneas del programa, este lenguaje es conocido como *processing*.
- Notificaciones o consola: ubicada en la parte inferior y con color negro es la parte de depuración, donde se notifica al programador sobre errores por ejemplo de sintaxis, comunicación entre otros.

2.6.1. Configuración de IDE

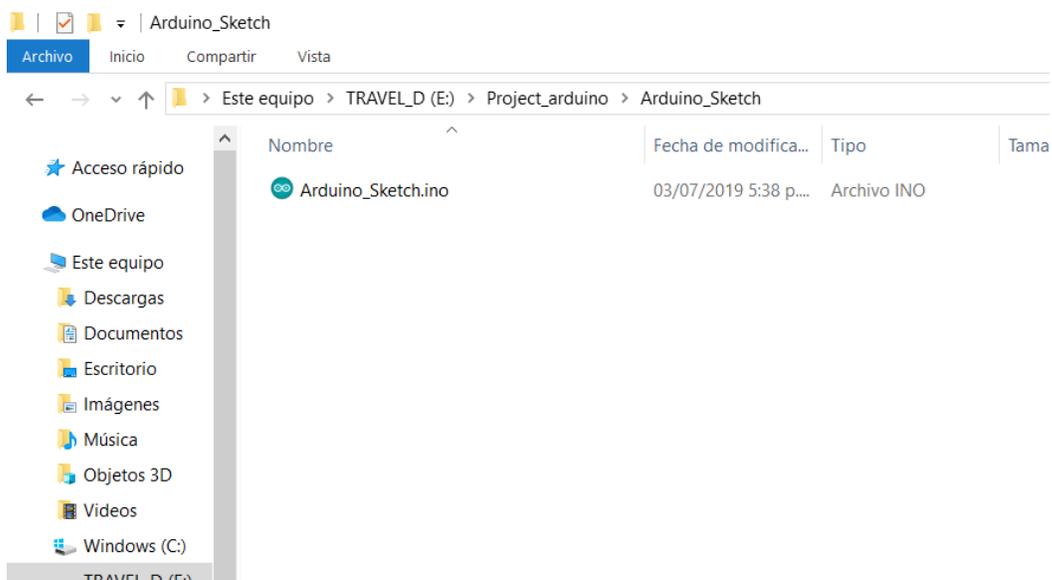
Configurar de manera eficiente el software ayuda a facilitar la edición de programas, mostrar información de compilación, carga de los mismos a la placa y advertencias por problemas de conexión o problemas de guardado, entre otros. Para esto solo se requiere entrar al menú Archivo, luego en preferencias, en este apartado se encuentran las opciones:

- Mostrar número de línea
- Habilitar plegado código
- Uso de editor externo
- Núcleo agresivamente compilado de cache
- Actualización de software IDE
- Cambio de extensión de ficheros

2.7. Estructura Sketch

Se refiere a un proyecto o un programa de Arduino, tiene la extensión .ino. Algo muy Importante, para que funcione, el nombre del fichero debe estar en un directorio con el mismo nombre que el *sketch*, no es indispensable que un *sketch* esté en un único fichero, pero si es necesario que todos los ficheros estén dentro del mismo directorio que el fichero principal.

Figura 12. Directorio de guardado de un sketch



Fuente: elaboración propia, utilizando Windows 10.

La estructura básica de un *sketch* para la placa Arduino es simple, se compone de dos partes principales considérese figura 13, se muestran *Void setup* y *Void loop*, estas son obligatorias, porque encierran bloques que contienen declaraciones o instrucciones de programación indicadas por el usuario.

Figura 13. Estructura básica de un sketch

```
sketch_jul03a
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Fuente: elaboración propia, utilizando Arduino IDE 1.8.9.

setup () es la parte de programación encargada de recolectar la configuración y *loop ()* es el complemento del programa que se ejecuta en iteradas ocasiones; de ahí el término *loop* o bucle. Ambas funciones necesarias para que el programa se ejecute.

La función *setup ()*, debe contener el inicio de los elementos a utilizarse en la programación, es la primera función a ejecutar en el programa y se utiliza para configurar, declarar variables, comenzar a usar librerías, entre otros. Esta función sólo se ejecuta una vez justo después de hacer el *reset*, no se vuelve a ejecutar hasta que no se realice otro *reset* en la placa.

La función bucle *loop* se ejecuta justo después de la función *setup* que contiene el código que se ejecutará continuamente, por ejemplo, lectura de entradas, activación de salidas, entre otros. Esta función puede considerarse la base de los programas de Arduino e incide principalmente en el control activo de la placa.

Dentro de las funciones básicas del Arduino es considerable mencionar las siguientes:

- `pinMode(pin,modo)`: permite indicar el modo de funcionamiento para un pin específico ya sea entrada o salida digital, aplicando (INPUT) para entrada o como salida con (OUTPUT).
- `digitalWrite(pin,valor)`: se utiliza para indicar en modo de salida digital al pin especificado en el paréntesis, asigna el valor HIGH (5V) o LOW (0V).
- `digitalRead(pin)`: indica lectura para un pin digital especificado, HIGH o LOW. Solo es necesario especificar el pin deseado en el paréntesis y considerar las limitaciones electrónicas de la placa.
- `analogRead(pin)`: lee el valor de voltaje en el pin analógico especificado. La placa utiliza 6 canales conectados a un conversor (ADC) de 10 bits.
- `analogWrite(pin,valor)`: establece un valor analógico PWM en un pin especificado. Puede ser usado para controlar la luminosidad de un led o la velocidad de un motor CD de bajo consumo de corriente.
- `delay(número)`: indica al *Sketch* detener su ejecución durante el tiempo especificado en el paréntesis expresado en milisegundos.

- `millis()`: devuelve el número de milisegundos que han transcurrido desde la última vez que Arduino se puso en funcionamiento y el momento en el que se ejecuta este comando. Este comando no acepta parámetros dentro del paréntesis, si se intenta darle alguno el compilador indicara un error. Sin embargo, forma parte de su sintaxis.

El tiempo de ejecución de algunos comandos en Arduino depende del microprocesador a utilizarse. Sin embargo, este se encuentra en medidas de microsegundos, la tabla III muestra el tiempo de ejecución de comandos básicos en un Arduino que trabaja con el ATmega328 corriendo a 16 Mhz.

2.8. Compilación y carga

El termino compilar en programación, se refiere a traducir un lenguaje de programación a un código que la maquina sea capaz de ejecutar, significa traducir el código escrito en el IDE para luego convertirlo a un lenguaje compresible para el microprocesador del Arduino. Una vez escrito el código del programa en el IDE, es posible verificar que se encuentra debidamente escrito a la vez que ordenado para su trasferencia hacia el microprocesador de la placa, ambas opciones las encontramos en el área de botones de acceso rápido, compilar con el botón de verificar, que es un botón con una marca de cotejo; y para cargar, con el botón subir, que es un botón con una flecha apuntando a la derecha.

Tabla III. **Tiempo requerido para ejecutar algunos comandos en Atmega328**

Comandos	Duración	
	μ s	ciclos de CPU
Analod read	110,9	1775
Digital read (pin sin PWM)	3,65	58
Digital read (pin con PWM)	4,53	72
Digital write (pin sin PWM)	3,93	63
Digital write (pin con PWM)	4,84	77
Analod write	8,05	129
milis()	1,32	21
micros()	2,96	47
randon()	140,3	2245

Fuente: Arduino. *Calculation speed tables.*

www.forum.Arduino.cc/index.php?topic=200585.msg1478940#msg1478940. Consulta septiembre 2019.

Acerca de la figura 14, contiene la programación de un *sketch* para Arduino con su software nativo IDE. Consiste en un semáforo compuesto por tres leds, declarados por las constantes rojo, amarillo, verde y pausa. Estas se declaran al inicio del programa, son constantes globales, por lo tanto, es posible acceder a ellas en cualquier parte del programa. Las led se definen con los pines 2, 4 y 7 respectivamente, se agrega una constante llamada pausa sien el valor de 7 segundos; se utiliza como tiempo de retardo, posteriormente se declaran los pines como salidas digitales, siendo su valor de; 5 Voltios CD.

Se inicia el bucle principal, llamado *Void loop*, contiene la programación de encendido, apagado, también contiene la función *for* para simular el parpadeo de cambio de luz del semáforo. Este bucle es infinito, por lo tanto, cuando el

programa ejecute todas las líneas de comandos, vuelve a ejecutar línea tras línea cada comando desde el principio del bucle. Véase figura 14 contiene el programa del semáforo escrito en Arduino IDE.

Después de realizar la compilación del programa, el IDE muestra un mensaje en la zona inferior indicando la realización del mismo. Si en algún caso se encontrase con algún error, muestra mensajes en color rojo en la parte de consola. En la imagen se muestra un mensaje con el tamaño correspondiente del *sketch*, junto el dato de espacio disponible para la elaboración del mismo.

Figura 14. Programa semáforo escrito en Arduino IDE

```
semaforo
//declaracion de constantes globales

int rojo=2;           //asigna pin 2 para LED rojo
int amarillo=4;      //asigna pin 4 para LED amarillo
int verde=7;        //asigna pin 7 para LED verde
int pausa=7000;     //7 segundos tiempo de espera de los LEDs encendidos

// programación principal

void setup() {       //declara los pines 2, 4 y 7 como salida.
  pinMode(rojo, OUTPUT); //LED rojo
  pinMode(amarillo, OUTPUT); //LED amarillo
  pinMode(verde, OUTPUT); //LED verde
}

void loop(){ // Inicia el loop
  digitalWrite(rojo, HIGH); //enciende el LED rojo
  delay(pausa); //espera 7 segundos
  digitalWrite(rojo, LOW); //apaga el LED rojo
  digitalWrite(verde, HIGH); //enciende el LED verde
  delay(pausa); //espera 7 segundos
  for(int i=0; i<4; i++) { //parpadea el LED verde durante 4 veces.
    digitalWrite(verde, LOW);
    delay(500); //espera 0.5 segundos
    digitalWrite(verde, HIGH);
    delay(500);
  }
  digitalWrite(verde, LOW); //cuando terminan las 4 iteraciones, apaga el LED verde
  digitalWrite(amarillo, HIGH); //enciende el LED amarillo
  delay(4000); //espera 4 segundos
  for (int a=0; a<4; a++) { //parpadea el LED amarillo durante 4 veces
    digitalWrite(amarillo, LOW);
    delay(500);
    digitalWrite(amarillo, HIGH);
    delay(500);
  }
  digitalWrite(amarillo, LOW); //cuando termina las 4 iteraciones apaga el LED amarillo
} //Terminado el loop
<
Compilación terminada
Tamaño binario del Sketch: 1,298 bytes (de un máximo de 32,256 bytes)
40
```

Fuente elaboración propia, empleando Arduino IDE 1.8.9.

A partir de la versión 1.6.6 del IDE de Arduino, se dispone la opción para realizar gráficas con las lecturas recibidas por el puerto serie, opción útil en momentos donde es necesario revisión de datos numéricos en forma gráfica, esta opción se encuentra en la pestaña herramientas, luego en el apartado *Serial Plotter*.

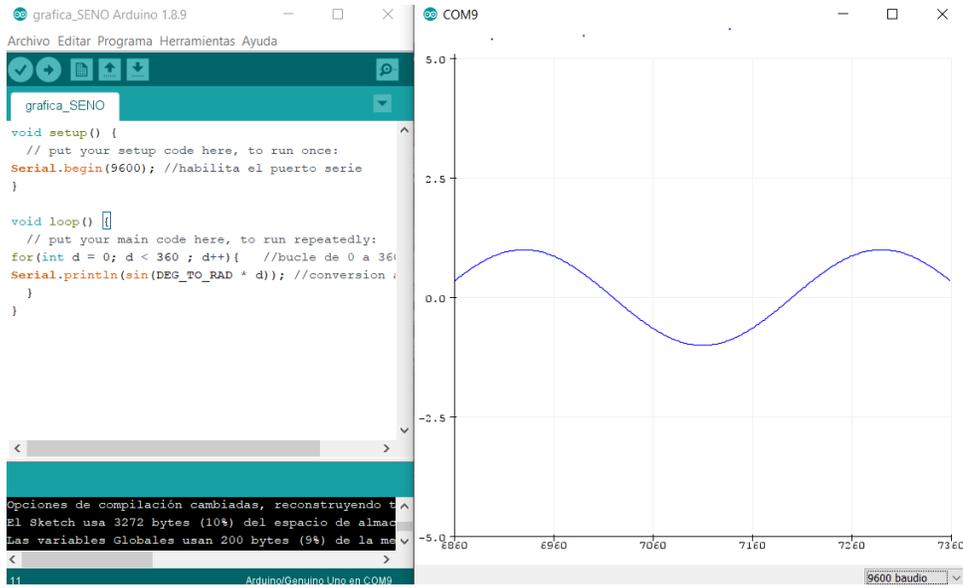
Esta herramienta es capaz de realizar múltiples gráficas en un mismo plano, también posee la característica de funcionar sin necesidad de conexión a internet, realizar cambios automáticamente en el tamaño de la gráfica acomodando el eje Y en función del valor de la variable a medir. El eje X es un eje constante definido por el usuario con una velocidad de comunicación serial de 9600 bits por segundo o baudios, esta velocidad se puede ajustar según la necesidad de la aplicación y se dispone de variantes como 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 o 115200.

Figura 15. Programa de aplicación de gráfica de Arduino IDE

```
void setup() {  
  
  Serial.begin(9600); //habilita el puerto serie  
}  
  
void loop() {  
  
  for(int d = 0; d < 360 ; d++){ //bucle de 0 a 360  
    Serial.println(sin(DEG_TO_RAD * d)); // imprime puerto serie  
                                     // conversion a radianes  
  }  
}
```

Fuente: elaboración propia, empleando Arduino IDE 1.8.9.

Figura 16. Gráfica de función trigonométrica Seno



Fuente elaboración propia, utilizando Arduino IDE 1.8.9.

3. SOFTWARE LABVIEW

LabVIEW es un software para desarrollar aplicaciones de ingeniería o ciencia, su nombre es acrónimo de *Laboratory Virtual Instrument Engineering Workbench*, es una plataforma con entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico, utilizando bloques. Dado a esto se considera sencillo de entender, a la vez que intuitivo, esto facilita visualizar, codificar y crear sistemas que permitan aumentar la eficiencia en procesos que ya se encuentran establecidos o rediseñalos de tal manera que se aumente su desempeño.

Una de las ventajas de LabVIEW es la capacidad de combinarse con otro software, hardware o integrar cualquier dispositivo de adquisición de datos, LabVIEW se aplica en industrias de distribución de energía eléctrica, sector alimenticio, azucareras, medicina u otras que utilicen componentes eléctricos o electrónicos aplicados a sistemas de control y automatización. Gracias a su lenguaje enfocado a programación gráfica ofrece herramientas de alto nivel al alcance de los usuarios sin amplios conocimientos de programación en lenguajes tradicionales, reduciendo así el tiempo de desarrollo de los proyectos.

En principio LabVIEW fue desarrollado para ejecutarse en sistemas operativos MAC, No obstante, dado a su amplio potencial, su demanda creció y ahora está disponible para sistemas como Windows, UNIX o GNU/Linux.

Se utiliza una licencia para estudiantes para el aprendizaje y desarrollo del software, específicamente LabVIEW 15,0 instalado en Microsoft Windows 10 Home de 64 bits.

3.1. Entorno de desarrollo LabVIEW

LabVIEW utiliza programación en formato gráfico o lenguaje G; programación en bloques, conocidos como Instrumentos Virtuales. Estos programas pueden formar categorías de instrumentos, formado así subprocesos, facilitando la construcción de programas más avanzados, a la vez que se reduce el tiempo de desarrollo del mismo, igualando la capacidad de lenguajes de programación convencionales. Debido a esto, la creación de herramientas aplicadas en sistemas de control y medición se ha simplificado.

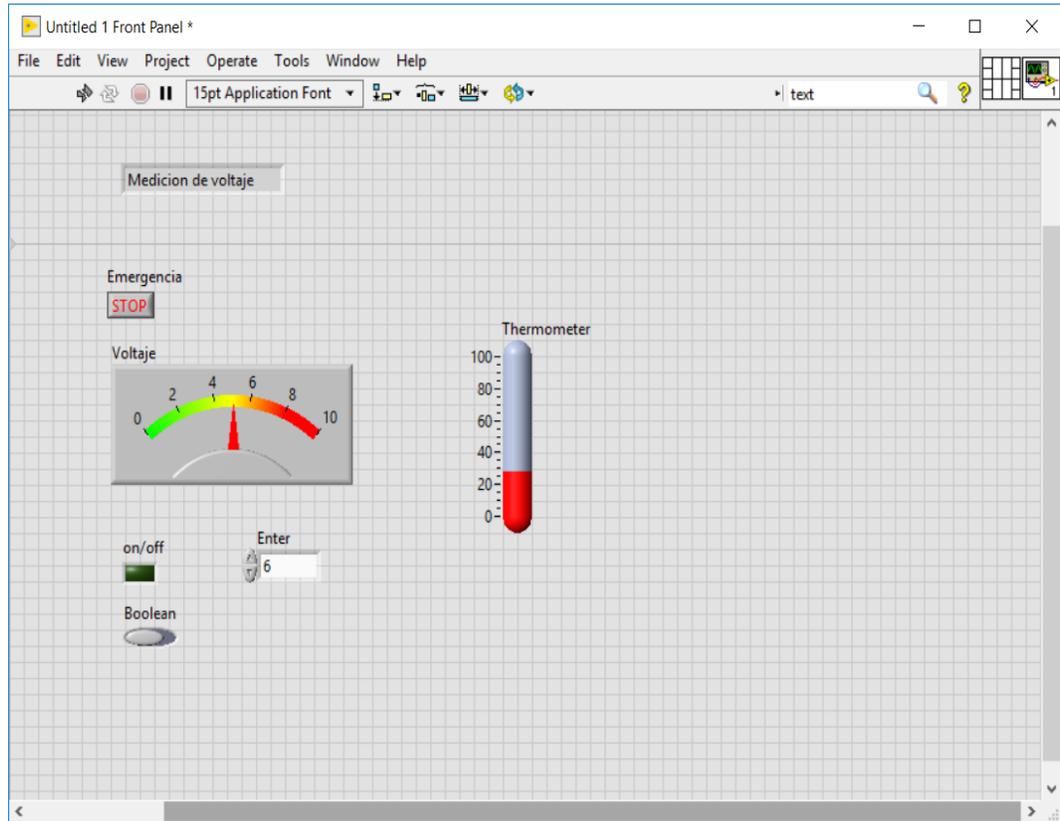
3.1.1. Virtual Instruments en LabVIEW

Los nombres que adquieren los programas o proyectos desarrollados en LabVIEW se llaman VIs, estos buscan imitar el comportamiento y apariencia de dispositivos reales por ejemplo un osciloscopio, multímetro, botones, interruptores, vúmetros, entre otros. Estos pueden formar grados o jerarquías de instrumentos, generando así los subVIs, al utilizarlos se desarrolla un programa más complejo. Los VIs están integrados de una parte interactiva con el usuario en conjunto con otra parte de código fuente, sin embargo, esto no limita la capacidad de aceptar parámetros procedentes de otros VIs.

3.1.2. Panel frontal

Cuando se abre un instrumento virtual nuevo o existente, se muestra una ventana del panel frontal, es la interfaz gráfica del VI con el usuario, se encarga del ingreso de datos junto a una representación de salidas proporcionadas por el programa, incluye indicadores, controles, gráficos y elementos de texto. Cada proyecto debe tener un VI principal, puede contener uno o más subVIs.

Figura 17. Panel frontal de un VI

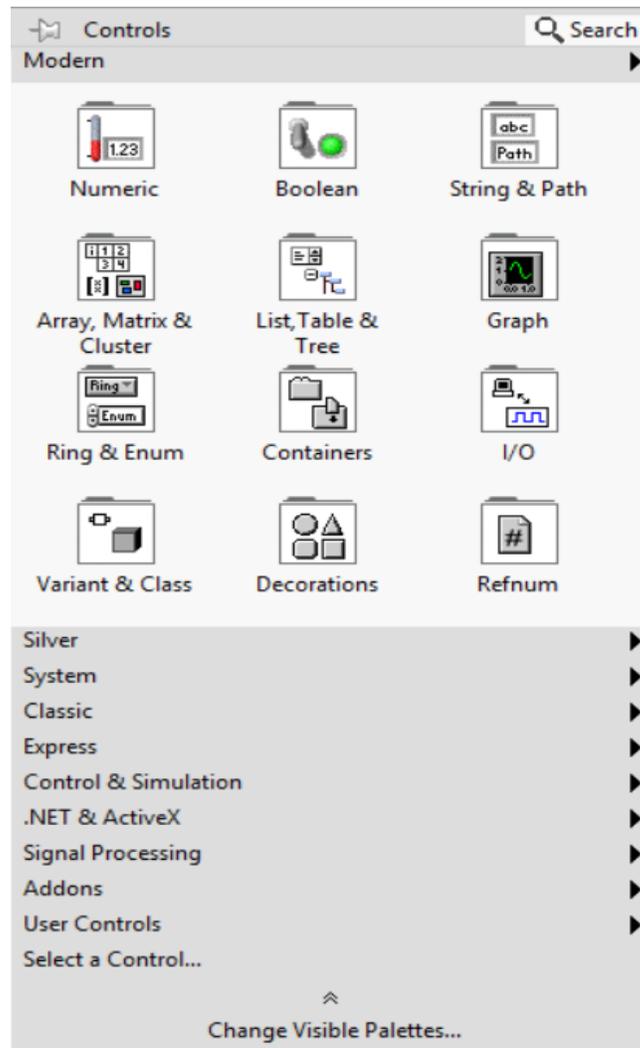


Fuente: elaboración propia, utilizando LabVIEW 2015.

3.1.3. Paleta de controles

Contiene herramientas como indicadores, funciones, controles o apartados de índole estético, son necesarios para diseñar el panel frontal. es posible acceder a esta desde el panel frontal, en la pestaña *View* después *Controls Palette* o dando clic derecho, en algún espacio vacío en la ventana del panel frontal. contiene varias categorías, es posible editarla para mostrar una o varias categorías en función de la necesidad del programador. La figura 18 muestra un panel frontal con funciones aplicables a los VIs.

Figura 18. **Paleta de controles del panel frontal**



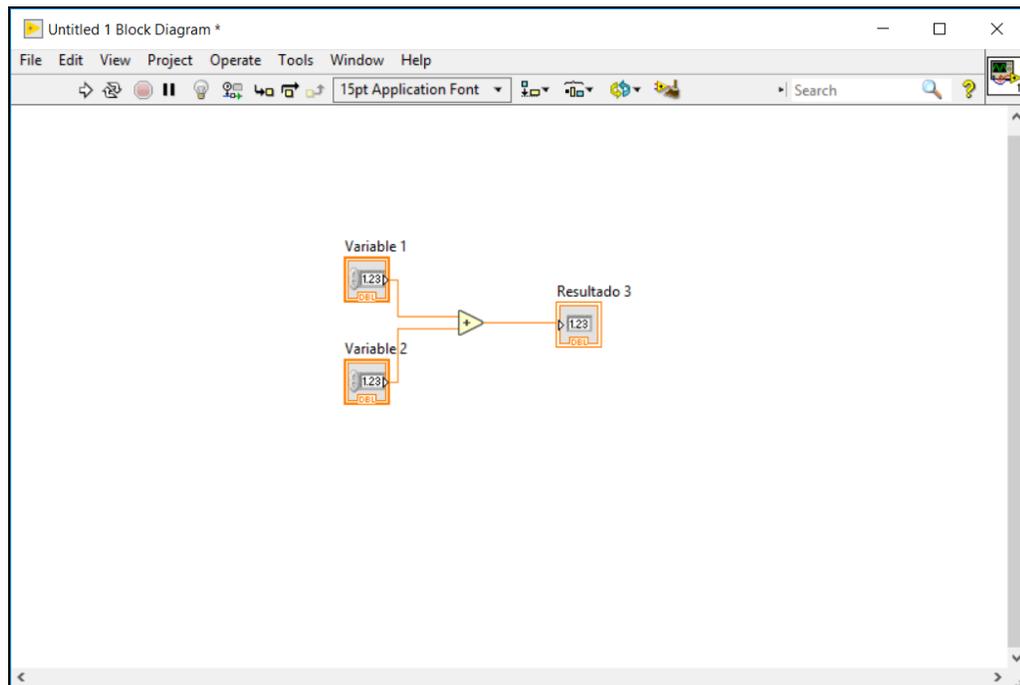
Fuente: elaboración propia, utilizando LabVIEW 15.

3.2. Diagrama de bloques

Es el apartado donde se realiza el código del programa, incluyen terminales, constantes, estructuras, cables, subVIs y funciones que transportan datos en conjunto con los objetos del diagrama de bloques. En este apartado se diseña el

diagrama de flujo del programa, es la parte que no se encuentra accesible al usuario final, analógicamente a programas basados en texto, es la parte de código del programa que no se puede cambiar.

Figura 19. **Diagrama de bloques de un VI**



Fuente: elaboración propia, utilizando LabVIEW 15.

3.2.1. **Nodos de diagrama de bloques**

Cumplen la función de realizar operaciones, instrucciones o subrutinas, valiéndose de entradas y salidas. Los nodos son capaces de realizar operaciones en cuanto un VI se ejecuta, pueden ser funciones, subVIs, Express VIs o estructuras. Las estructuras son parte del control en procesos, por ejemplo, Ciclos *While* o Ciclos *For*.

3.3. Terminales

Su objetivo principal es otorgar una entrada o salida, basándose en el intercambio de datos entre el panel frontal y el diagrama de bloques, tienen un comportamiento comparable a variables o constantes en lenguajes de programación basados en texto. La mayor parte de elementos del panel frontal tienen su terminal correspondiente en el diagrama de bloques, con excepción de etiquetas incluyendo elementos decorativos.

3.4. Funciones

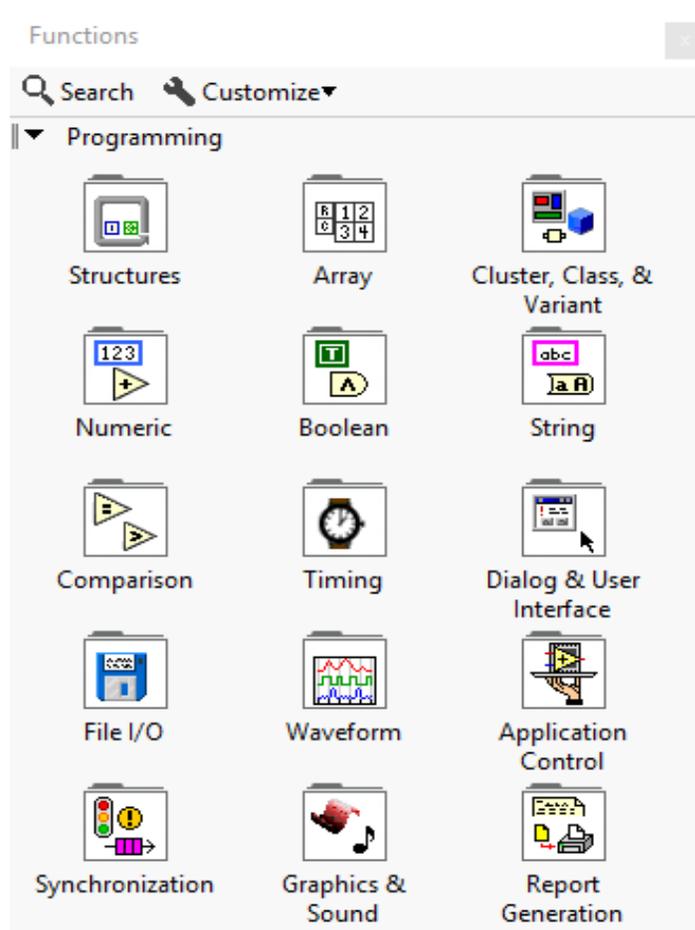
Las funciones son elementos de operación básicas de LabVIEW, estas incluyen funciones booleanas, numéricas, texto, arreglos, o archivos. Estas no cuentan con ventanas gráficas en el diagrama de bloques ni en el panel frontal, sin embargo, si cuenta con paneles conectores. Sus argumentos se especifican en las entradas a la izquierda del icono, y sus salidas a la derecha con las salidas. Los principales tipos de funciones son:

- Funciones numéricas: se utilizan para realizar operaciones aritméticas, trigonométricas, algebraicas, entre otras.
- Funciones booleanas: mediante estas se puede efectuar operaciones con algebra booleana.
- Funciones *string*: permiten manipular cadenas de caracteres y formar dichas cadenas, también contiene la conversión de cadenas de caracteres.
- Funciones *cluster*: se utiliza para realizar operaciones con *clusters*. Básicamente agrupan elementos de datos de varios tipos.

- Funciones de arreglos: se refiere a creación, edición y operaciones con arreglos.
- Funciones de tiempo: se utilizan para realizar operaciones con tiempos determinados, estas influyen especialmente en la ejecución del programa.
- Funciones de comparación: realiza comparaciones entre datos o grupos de datos, incluyendo entradas y salidas, pueden ser variedad de datos.
- Funciones de dialogo e interfaz de usuario: permiten desplegar e interactuar con cuadros de dialogo durante la ejecución del VI.
- Funciones de entrada y salida de archivos: se utilizan para interactuar, leer o editar archivos almacenados en el disco local.

La paleta de funciones se encarga de contener VIs, funciones y constantes, utilizables en el desarrollo del diagrama de bloques. Para acceder a ella solo es necesario entrar a la pestaña en el lado del diagrama de bloques, *View* después en *Functions Palette*. La paleta de funciones se divide en varias categorías; incluyendo las opciones mostrar o su contraparte, ocultar categorías de acuerdo a las necesidades del programador.

Figura 20. **Paleta de funciones**

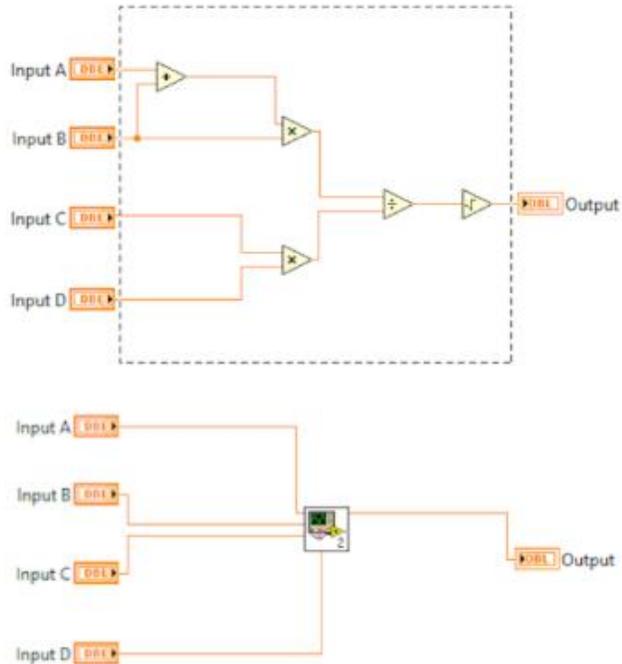


Fuente: elaboración propia, utilizando LabVIEW 15.0.

3.5. **SubVIs**

Se trata de un VI llamado desde el diagrama de bloques a partir de otro VI, no se limita el reutilizar un subVIs en otros VIs. Para realizar esto se crea un panel conector, el panel es un conjunto de terminales en el icono que corresponde a los controles e indicadores de ese VI.

Figura 21. Creacion de subVI en el diagrama de bloques



Fuente: DIGILENT. *Creating a SubVI in LabVIEW*. www.blog.digilentinc.com. Consulta: 21 de julio de 2019.

3.6. Cables

Gráficamente se representan como conexiones entre objetos o funciones, transfieren datos o señales del diagrama de bloques, estos se pueden conectar a varios VIs o funciones que tengan la capacidad de entender los datos, se representan en diferentes colores, estilos y grosores esto en función del tipo de dato en transferencia.

Tabla IV. **Tipos de datos más comunes**

Tipo de Cable	Escolar	Arreglo de 1D	Arreglo en 2D	Color
Numérico				Naranja (punto flotante), Azul (entero)
Booleano				Verde
Cadena de caracteres				Rosa

Fuente: National Instruments Corp. *Programación Gráfica*. www.ni.com. Consulta: 5 de septiembre de 2019.

LabVIEW tiene la capacidad de realizar conexiones automáticas temporales en el diagrama de bloques, esto se logra acercando un objeto seleccionado con otro, el programa sugiere conexiones validas o compatibles. El software conecta las terminales que corresponden mejor, no así terminales incompatibles, dicha función se encuentra en la pestaña de herramientas, después en opciones y por último se elige diagrama de bloques en la lista de categorías.

3.7. Programación gráfica

Tratándose de un lenguaje G la herramienta principal de trabajo es el mouse del ordenador, en LabVIEW es el medio principal de interacciones con el entorno de programación, tiene la capacidad de completar tareas especialmente selección, cableado, resaltar texto entre otras. Sus funciones son variadas y por lo tanto tiene su paleta de herramientas.

Las herramientas del mouse brindan la capacidad de crear, modificar o depurar VIs, son modos de operación del cursor del mouse. El modo de operación

del cursor depende al icono de la herramienta seleccionado. En función de la posición del cursor sobre los objetos o funciones en el diagrama de bloques, LabVIEW selecciona la función apropiada automáticamente. Para mostrar la paleta de herramientas basta con seleccionarla en la pestaña *Tools* después *View* y por ultimo *Tools Palette*.

Figura 22. **Paleta de herramientas**



Fuente elaboración propia, utilizando LabVIEW 15.

La tabla V muestra una breve descripción de las herramientas más comunes que se encuentran en la paleta de herramientas del software.

LabVIEW ejecuta sus programas siguiendo el modelo de flujo de datos. Por ejemplo, los nodos se activan con la condición de tener los datos disponibles en sus correspondientes terminales de entrada, cuando termina su ejecución, procede a enviar los datos a su terminal de salida, por lo tanto, traslada los datos al siguiente nodo en la trayectoria del flujo. El traslado de datos a través de nodos incide el orden de realización de los VIs, esto a su vez afecta las funciones en el diagrama de bloques.

Tabla V. **Funciones principales de la paleta de herramientas**

Icono	Nombre	Descripción
	Selector Automático	Automáticamente se escoge una herramienta de acuerdo a la ubicación del cursor.
	Operación	Se utiliza para cambiar los valores de un control, presionar algún interruptor o interactuar con una función variable
	Posicionamiento	Brinda la capacidad para seleccionar o cambiar el tamaño de los objetos, después se puede mover, copiar o eliminar el objeto
	Etiquetado	Se utiliza para proporcionar el texto en un control, para editar texto y para crear etiquetas
	Cableado	Se selecciona cuando se requiere cablear objetos entre si en el diagrama de bloques
	Menú de acceso directo	Para tener acceso inmediato al menú de objetos con el botón izquierdo del mouse.
	Desplazamiento	Otorga la capacidad de desplazamiento a través de las ventanas sin usar barras de desplazamiento.
	Breakpoint	Establece puntos de pausa en VIs, funciones, nodos y estructuras para detener la ejecución en esa ubicación específica.
	Probe	Establece puntos de prueba en el diagrama de bloques
	Copiar color	Copia colores y después se pueden pegar con la herramienta de Coloreo

Fuente: elaboración propia, utilizando Microsoft Excel 2016.

3.7.1. Estructuras de ejecución

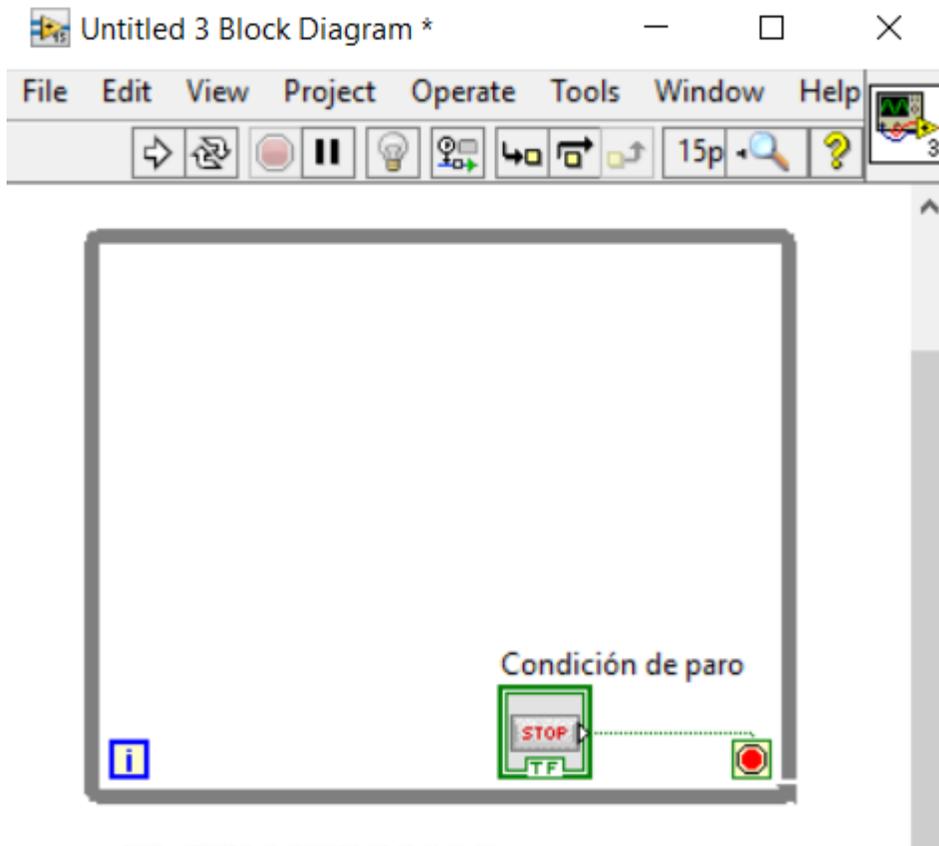
Contiene secciones del programa gráfico, controlan como y donde el código en su interior se ejecuta, son aplicables generalmente cuando es necesaria la ejecución de un mismo conjunto de sentencias cierto número de veces, o reiterarse mientras se cumpla alguna condición. Dentro de las más comunes se encuentran los siguientes:

3.7.1.1. Ciclos While

Es una estructura que repite el código en su interior indefinidamente hasta que se cumpla una condición determinada. Se representa con un rectángulo gris ubicado en la paleta *Structures*. Para aplicarlo basta con seleccionarlo y arrastrarlo hasta el diagrama de bloques que se desea repetir, si el programa no tiene fin determinado por el usuario o el programador, este se ejecutara infinitamente hasta que el usuario lo detenga.

Este ciclo se ejecuta con la condición de conectar un botón de *stop* u otra condición estrictamente necesaria por sintaxis de programación, sirve de paro al terminal *stop*; botón rojo en la parte inferior derecha por defecto en el software, véase figura 23. En conjunto con estas estructuras es posible aplicar los *shift registers* para disponer de los datos obtenidos en iteraciones anteriores.

Figura 23. **Representación gráfica del ciclo While**



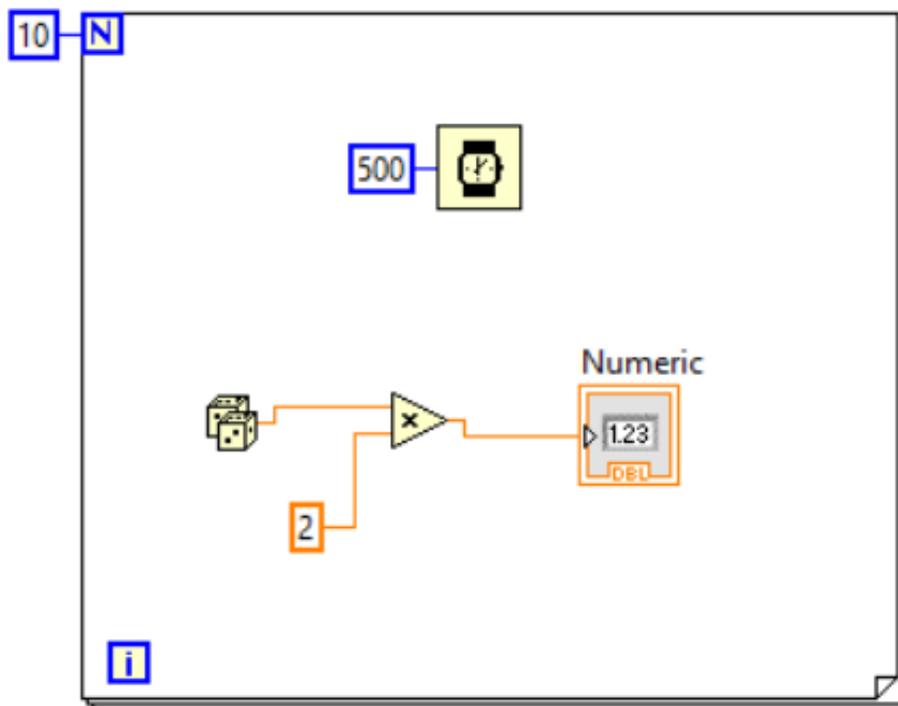
Fuente: elaboración propia, utilizando LabVIEW 15.0.

3.7.1.2. Ciclos For

Es una estructura muy parecida al ciclo *While*, la diferencia consiste en el número de iteraciones de ejecución. Tiene que establecerse, se utiliza principalmente en vectores o arreglos con dimensiones determinadas, permitiendo agregar, modificar o eliminar datos según el índice establecido. Este también se encuentra en la paleta *Structures* y se tiene que considerar la cantidad de iteraciones para el ciclo *For*. Siempre comienza desde cero.

Cuando un ciclo termina de ejecutar una iteración, inmediatamente comienza la siguiente iteración, en ocasiones es necesario controlar la frecuencia, igualmente la temporalización de esta misma, es altamente considerable proporcionar al procesador y a la interfaz del usuario, tiempo para completar las tareas. Por lo tanto, cuando se utiliza un ciclo, se tiene que agregar un *Timing (Wait)*, esto para garantiza que LabVIEW consuma la menor cantidad de memoria del PC. El valor mínimo del *Wait* puede establecerse es 5 ms.

Figura 24. **Ciclo For aplicado a un programa con Wait**

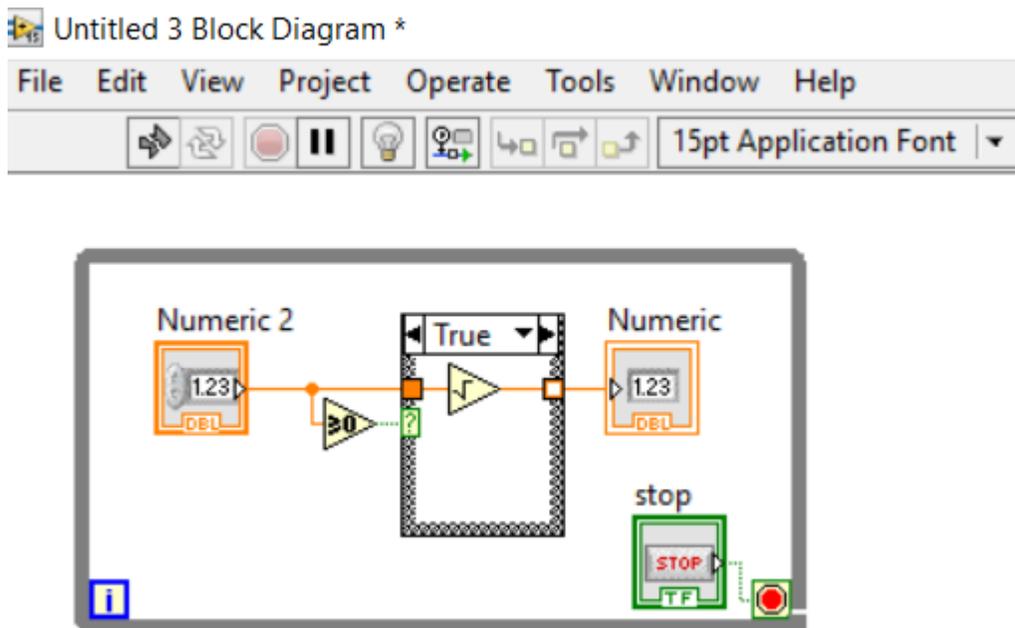


Fuente: elaboración propia, utilizando LabVIEW 15.0.

3.7.1.3. Estructura case

Se utilizan para ejecutar un código diferente en función de su entrada, se compone de al menos dos subdiagramas o casos y solo es posible visualizar uno a la vez, por lo tanto, solamente se puede ejecutar un caso a la vez. Un valor de entrada determina cual subdiagrama se ejecuta, este valor puede ser tipo booleano, numérico entero, secuencia o valor de tipo enumerado al terminal selector de casos.

Figura 25. Programa con Structure case



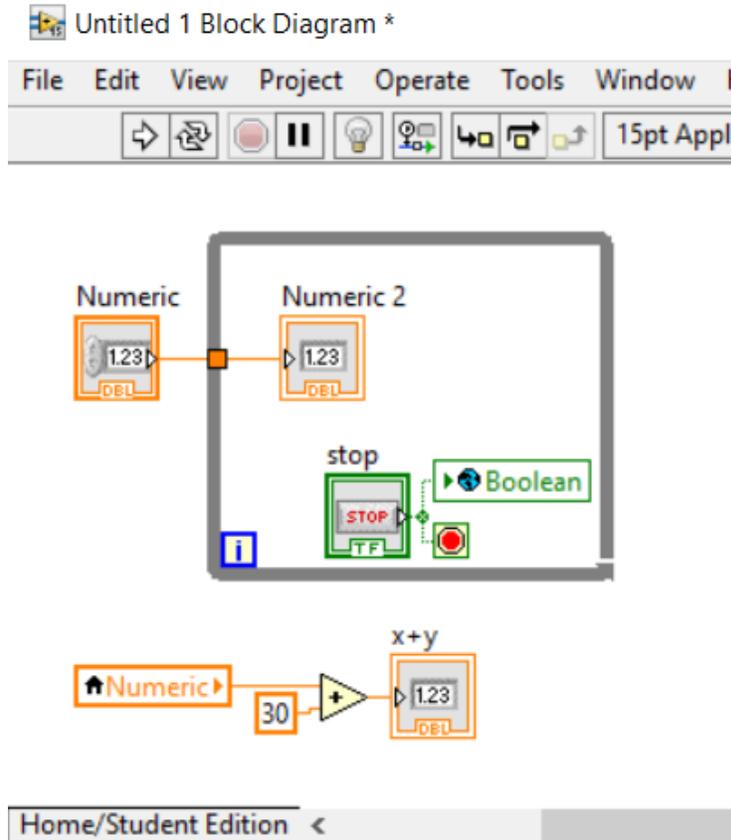
Fuente: elaboración propia, utilizando LabVIEW 15.

3.7.1.4. Variables

Se refieren a elementos de programación que pueden tomar valores o datos manipulables en el código del programa, existen dos tipos de variables; locales y globales. Las variables locales pueden leerse o escribirse en controles o indicadores en el panel frontal de un VI. En el diagrama de bloques se puede asociar a un elemento del programa, para utilizarlo posteriormente en otra parte o en otro bloque del mismo, en definitiva, las variables locales son accesibles a elementos compatibles dentro de un VI, simplificando enormemente el cableado en el diagrama de bloques.

Las variables globales tienen característica de poderse utilizar entre varios VIs para acceder o transportar datos, estas agregan la característica de tener un panel frontal pero no un diagrama de bloques. Los iconos de casa; variable local, mundo; variable global, se utilizan para identificar qué tipo de variable se está trabajado, y estas se pueden configurar en modo de lectura o escritura según la necesidad del programa o función asociada.

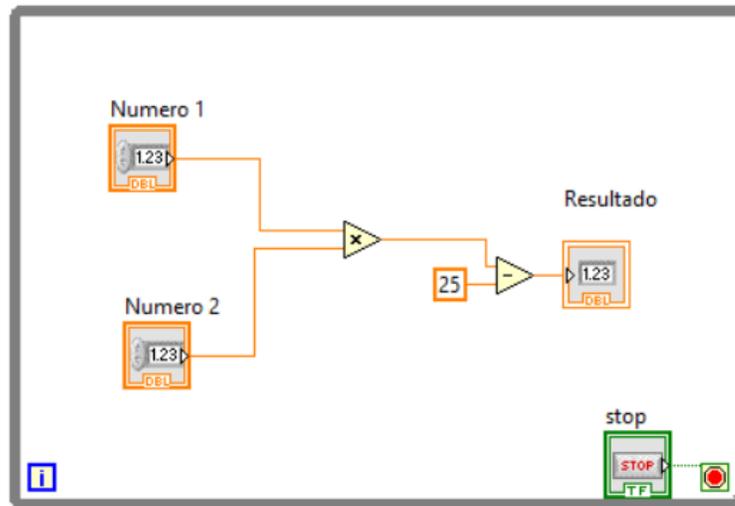
Figura 26. Variables locales y globales



Fuente: elaboración propia, utilizando LabVIEW 15.0.

La figura 27 muestra la programación de flujo de datos aplicado a un ejemplo sencillo, donde se multiplica dos números ingresados por el usuario; variables, para luego, restar 25; una constante, al resultado de la multiplicación, después el resultado es enviado a un indicador con el mismo nombre. La ejecución del diagrama de bloques es de izquierda a derecha, no es dependiente del orden, sino una función, realizándose cuando disponga de los datos correspondientes en sus entradas para luego enviarlos a su salida.

Figura 27. **Diagrama de bloque de un programa elemental**



Fuente: elaboración propia, utilizando LabVIEW 15.0.

3.8. **Conexión entre LAbVIEW y Arduino**

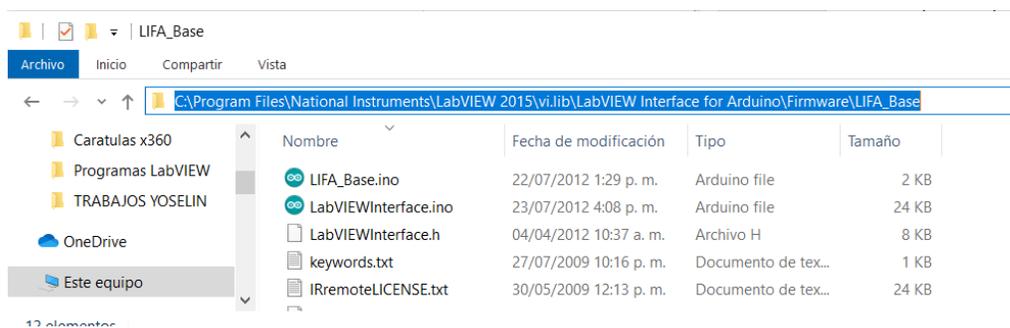
Es necesario habilitar los puertos e interfaces de comunicación desde LabVIEW con la placa de Arduino, para ello es necesario instalar un complemento llamado NI VISA, considerando que este no viene por defecto en el software. La conexión física entre el ordenador y Arduino se realiza mediante un puerto USB tipo A macho en el ordenador, con una entrada USB tipo B macho en el Arduino. VISA es un API, su función principal es la comunicación con buses de instrumentación. Se ofrece gratuitamente el complemento NI VISA desde la página oficial de National Instruments.

El siguiente complemento necesario es el VI *Package Manager*, es un administrador de contenidos, asimismo manejador de descargas de librerías incluyendo diversos *toolkits* para LabVIEW. Una vez instalado, se procede con la descarga de paquetes de compatibilidad con la tarjeta Arduino, este paquete se encuentra directamente con el nombre “LabVIEW Interface for Arduino” es

necesario para establecer la conexión entre el software y el hardware; una vez localizado el paquete se instala en el ordenador con la opción “Install Package”.

Para establecer la conexión, es necesario cargar el programa que permite la manipulación del hardware con el software, este se encuentran con el nombre “LIFA_Base.ino”, ubicado en la dirección “C:\Program_Files\National Instruments\LabVIEW_2015\vi.lib\LabVIEW_Interfacefor_Arduino\Firmware\LIFA_Base”, considerar esta depende de la ruta de instalación del programa LabVIEW y de la versión a utilizarse, antes de cargar el archivo con Arduino IDE se tiene que seleccionar el tipo de tarjeta Arduino con el puerto COM de la computadora.

Figura 28. Ruta de librerías de compatibilidad



Fuente: elaboración propia, utilizando Windows 10.

4. SISTEMA DE MEDICIÓN UTILIZANDO ARDUINO Y LABVIEW

En este capítulo se muestra el diseño, al igual que componentes de los módulos que conforman el sistema de medición, compuesto por sensores electrónicos, un medio de lectura de datos, simultáneamente procesado de los mismos y una interfaz para el usuario. Este módulo se encarga de traducir las señales eléctricas para luego mostrarlas de manera gráfica mediante el panel de control en LabVIEW.

4.1. Medición de parámetros eléctricos

Se realiza utilizando transductores eléctricos, estos son dispositivos que al ser sometidos a una magnitud física; por ejemplo, calor tiene como objetivo representar esa magnitud como una señal eléctrica. En otras palabras, transforman una magnitud física en una señal eléctrica, generalmente una señal normalizada o establecida ya sea de voltaje o corriente; por ejemplo 4 o 20 mA estas son convenientemente registradas y medidas.

4.2. Medición de voltaje

Los sistemas eléctricos de potencia se suministran por generadores trifásicos. Idealmente los generadores les suministran a cargas trifásicas balanceadas, por lo tanto, en los motores trifásicos las cargas tienen impedancias idénticas en las tres fases o considerablemente muy parecidas. Por esta razón, solo es necesario medir el voltaje de una fase, porque teóricamente no existe variación del mismo en un motor trifásico. Para asegurar que el sistema sea

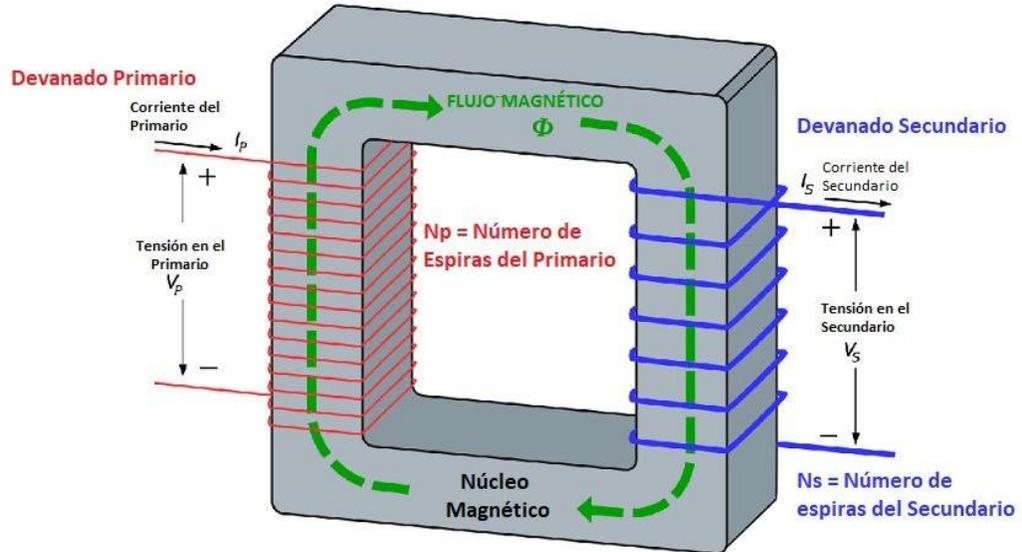
capaz de medir el voltaje trifásico y enviarlo al Arduino, es necesario convertir el voltaje trifásico a un nivel aceptable para dispositivo de lectura.

4.2.1. Transformador de potencial

También se le conoce como transformador de voltaje, tiene el objetivo principal de transferir energía eléctrica de un circuito hacia otro circuito, mediante conductores acoplados inductivamente. En términos básicos se encarga de convertir valores de voltaje, de un valor determinado y transformarlo a otro valor, se basan en dos principios esenciales: el electromagnetismo junto a la inducción electromagnética.

La base del funcionamiento de un transformador de potencial, requiere introducir una corriente alterna en el devanado primario, esto genera un flujo magnético variable en el núcleo del transformador, a su vez el flujo magnético variable influye al devanado secundario, dado que ambos se encuentran acoplados al mismo núcleo. La variación de flujo magnético genera una fuerza electromotriz variable o tensión en el devanado secundario. Los valores de voltaje están afectados en función de la cantidad de vueltas de los devanados.

Figura 29. **Trasformado de potencial**



Fuente: Tecnología Industrial. *Trasformador eléctrico* www.tecnologia-industrial.es/Transformador.htm. Consulta: 3 de octubre de 2019.

de acuerdo a la ley de Faraday el voltaje inducido en la bobina primaria en función del flujo magnético es:

$$V_p(t) = N_p \frac{d\phi_p(t)}{dt}$$

Donde N_p es el número de vueltas de la bobina primaria y $\Phi(t)$ es el flujo magnético, por lo tanto, el voltaje en la bobina secundaria obedece también la ley de Faraday.

$$V_s(t) = N_s \frac{d\phi_s(t)}{dt}$$

Si se considera la inexistencia de pérdidas en el transformador se obtiene que entre las bobinas primaria y secundaria el acople magnético es perfecto, por lo tanto, el flujo magnético en cada bobina es igual, $\Phi_p(t) = \Phi_s(t)$

$$\phi_p(t) = \phi_s(t)$$

Al derivar respecto al tiempo se obtiene:

$$\frac{d\phi_p(t)}{dt} = \frac{d\phi_s(t)}{dt}$$

Al sustituir con ley de Faraday:

$$\frac{V_p(t)}{N_p} = \frac{V_s(t)}{N_s}$$

A la ecuación resultante se le conoce como relación de transformadores se le asigna generalmente como **a**, si la relación de transformación es mayor que 1 el transformador es elevador, si a es menor que 1 el transformador es reductor.

$$\frac{V_p}{V_s} = \frac{N_p}{N_s} = a$$

Lo que sucede con la corriente eléctrica, considerando las ecuaciones anteriores, y evidentemente la condición de no existir pérdidas, implica que la potencia en el primario es igual a la potencia en el secundario:

$$P_p(t) = P_s(t)$$

$$V_p(t) \cdot I_p(t) = V_s(t) \cdot I_s(t)$$

$$\frac{I_p(t)}{I_s(t)} = \frac{V_s(t)}{V_p(t)}$$

Al sustituir la relación de voltajes previamente encontrada

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

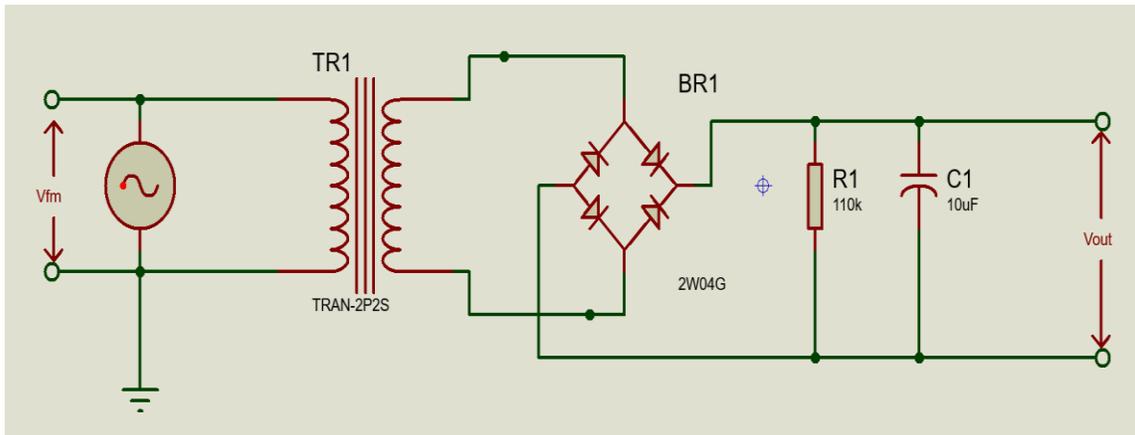
Finalmente

$$\frac{I_p(t)}{I_s(t)} = \frac{N_s}{N_p}$$

La figura 30 corresponde a un circuito rectificador de onda completa, este transforma una señal de corriente alterna de entrada V_{im} ; en este caso voltaje de fase del motor, en voltaje de salida V_{out} ; voltaje de salida, el TR1 es un pequeño transformador de potencial encargado de reducir la onda en un orden de 20:1 indicativo que por cada 20 voltios de corriente alterna en el primario, se obtiene 1 voltio de corriente alterna en el secundario.

Luego del transformador la corriente pasa a un puente rectificador de diodos BR1, su trabajo es convertir la onda de corriente alterna a una forma de voltaje de corriente directa, el capacitor C1 y la resistencia R1 mantienen el valor del voltaje V_{out} evitando rizados o variaciones no deseados en el voltaje, al igual que limitar el valor de corriente en el circuito.

Figura 30. **Circuito rectificador de voltaje**



Fuente: elaboración propia, utilizando Proteus 8.

Básicamente el circuito se encarga de convertir el voltaje alterno de una fase del motor, a un voltaje de corriente directa reducido 20 veces, por ejemplo, si el V_{fm} se encuentra en 240 voltios, dado a la razón de transformación en la salida del transformador; donde se encuentra el puente de diodos se obtiene un voltaje de 12 voltios. Es necesario considerar que se está trabajando con corriente alterna por lo tanto se trabaja con valores pico y valores RMS.

$$V_{pico} = \sqrt{2} \cdot V_{rms}$$

donde:

V_{pico} : Voltaje máximo de onda

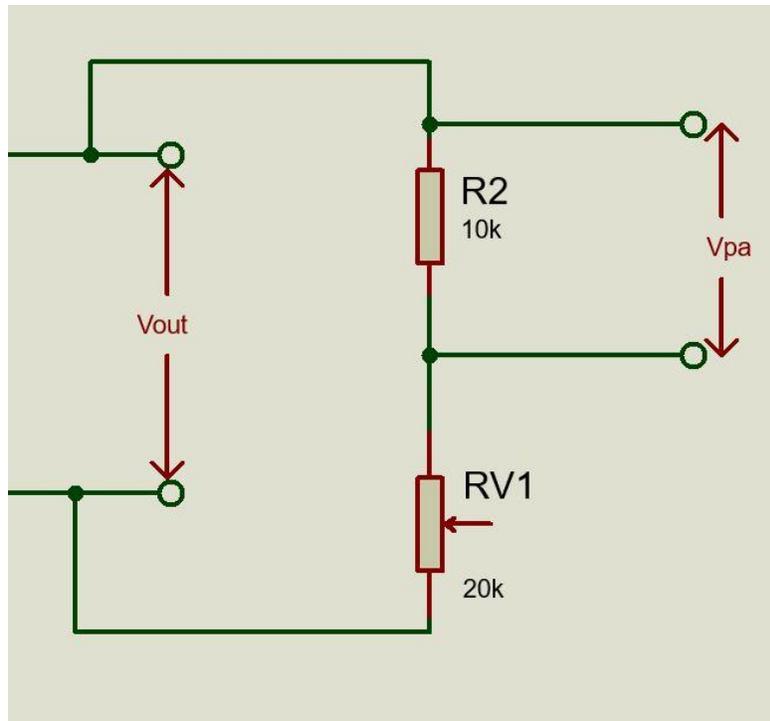
V_{rms} : Valor eficaz de una onda sinusoidal

Este voltaje de 12 voltios es el valor eficaz, esto indica que el valor rectificado puede alcanzar picos de voltaje $= \sqrt{2} \cdot 12 = 16,97 V$. Posteriormente se debe realizar un divisor de voltaje para reducirlo a menos de 5 Voltios límite que acepta Arduino para así evitar que se dañe, el arreglo queda como el que muestra la figura 30, en donde RV1 es una resistencia variable con el fin de variar el voltaje V_{pa} el cual va conectado a los pines analógicos del Arduino.

Este voltaje V_{pa} se tiene que calibrar con el fin de variar, sin sobrepasar el voltaje permitido por el Arduino, por ejemplo, variar entre 3 y 5 voltios, siendo 5 el límite que soporta el motor sin que reciba ningún daño su bobinado interno. Es considerable que, según las normas NEMA, los motores, se diseñan de tal manera que puedan soportar sus cargas nominales de potencia, en voltajes de 10 % arriba o abajo de su valor nominal de la placa. Aun así, las variaciones de voltaje dentro de este rango de 10 % pueden tener algún efecto en la operación del motor.

En este caso es posible simular una variación de voltaje mayor a ese 10 % del nominal, esto para tener un rango mayor, haciendo más evidentes los cambios de voltaje en el motor para fines de simulación. Pero en el campo laboral no tiene que existir mucha diferencia de voltaje si se trata del suministro comercial, esto siempre y cuando no exista ningún fallo en la instalación eléctrica o en la calidad de suministro por el distribuidor de energía eléctrica.

Figura 31. **Divisor de voltaje**



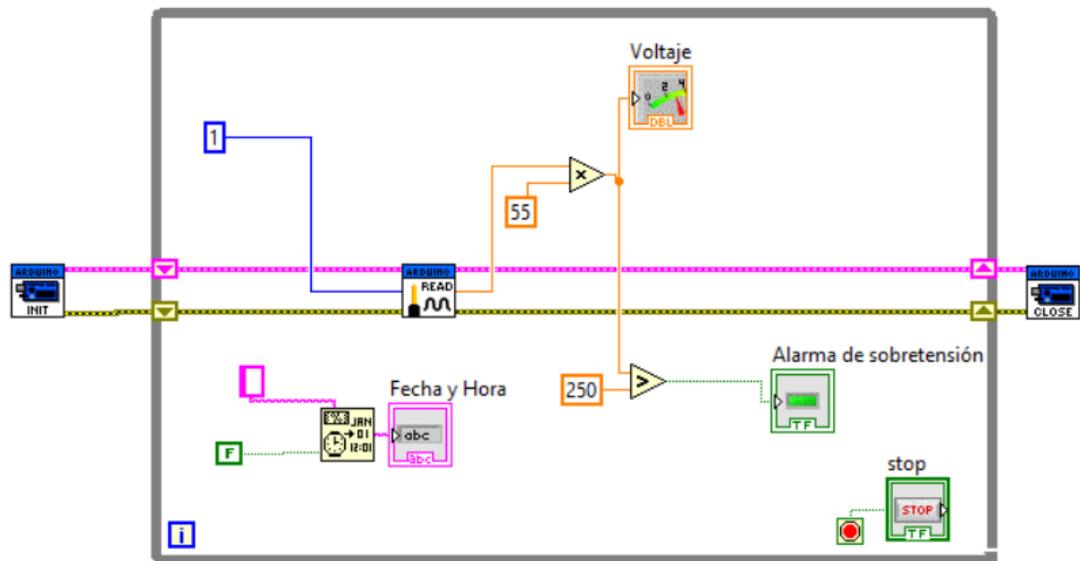
Fuente: elaboración propia, utilizando Proteus 8.

4.3. Programa para medición de voltaje

Primero se declara la tarjeta Arduino, esto con las funciones representadas por los iconos en azul INT y CLOSE, estos tienen que conectarse entre sí indicando el flujo del programa, entre estos se declara la función READ, esto indica al programa, utilizar el pin analógico 1 como receptor de una señal variable entre 0 y 5 voltios, luego a la salida de este icono se coloca un multiplicador para elevar el voltaje de entrada para así representar virtualmente los 240 V AC correspondientes al voltaje del motor trifásico. Este multiplicador puede usarse también como calibrador lógico en el programa, logrando así una medición lo más certera posible.

La salida del multiplicador se conecta con un indicador de voltaje, este en paralelo a un comparador, este se encarga de encender un led virtual indicando sobretensión si el valor de lectura sobrepasa los 250 voltios. El rectángulo de color gris que rodea el diagrama de bloques es una función WHILE LOOP, este solo se detiene cuando se presiona el botón STOP. Al VI se agrega la función Format Date/Time String representada por el icono reloj, para mostrar al usuario fecha y hora actual. considérese la figura 32 que muestra el programa grafico encargado del módulo de medición de voltaje.

Figura 32. Diagrama de bloques del programa medio de voltaje

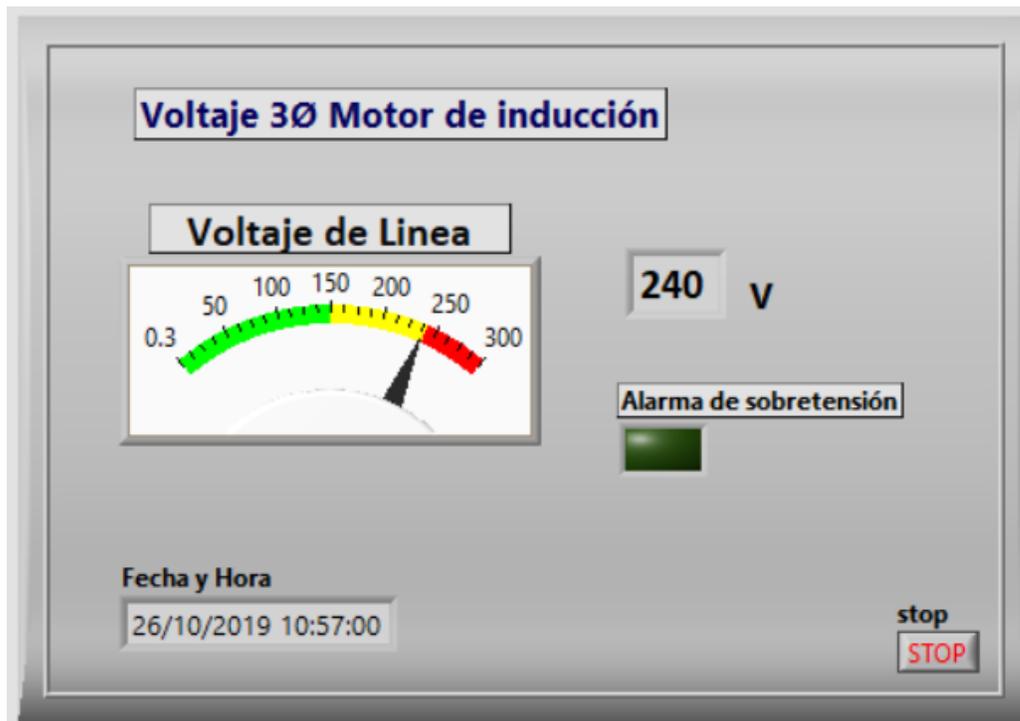


Fuete: elaboración propia, utilizando LabVIEW 15.0.

En conjunto con el diagrama de bloques se programa el panel frontal, este cumple la función de interface gráfica. Contiene un vúmetro analógico virtual indicando en tiempo real el voltaje que se encuentra en las terminales del motor, este también contiene un indicador del valor del voltaje con numeración arábica, un botón de paro (*stop*) este es el único manipulable para el usuario, únicamente cumple la función de detener la ejecución del programa, no así la ejecución del

motor. Por último, se agrega un indicador de fecha y hora, como un led virtual, para funcionar como alarma, activándose cuando sobrepase el voltaje de 250 V, establecido en el diagrama de bloques de la programación.

Figura 33. **Panel del programa de medición de voltaje**



Fuente: elaboración propia, utilizando LabVIEW 15.0.

4.4. **Medición de corriente**

De la misma manera que se obtiene indirectamente el voltaje, se mide la corriente trifásica del motor. Esta se puede lograr utilizando sensores de corriente de inducción, se trata de transformadores de corriente que permiten medir la intensidad de la misma que atraviesa un conductor, siendo su principal ventaja no requerir modificar el conductor por lo que reciben el nombre de no invasivos.

Para este fin se puede utilizar cualquier transformador de corriente, siempre y cuando tenga la capacidad de soportar la corriente máxima posible demandada por la máquina, los sensores SCT-013 basan su funcionamiento en transformadores de corriente, son dispositivos de instrumentación que proporcionan una medición proporcional a la intensidad que atraviesa un circuito. La medición se realiza por inducción electromagnética.

La precisión del sensor según el fabricante se encuentra entre 1 % y 2 %, es necesario considerar que el núcleo ferromagnético se cierre en su totalidad para restar errores de lectura como también problemas de dispersión.

Figura 34. **Sensor de corriente no invasivo**



Fuente: Electrón. *Sensor de corriente no invasivo SCT013-30.*

www.electronperdido.com/shop/sensores/tension-corriente/sensor-corriente-30a-no-invasivo-sct-013-030/. Consulta: 16 de julio de 2019.

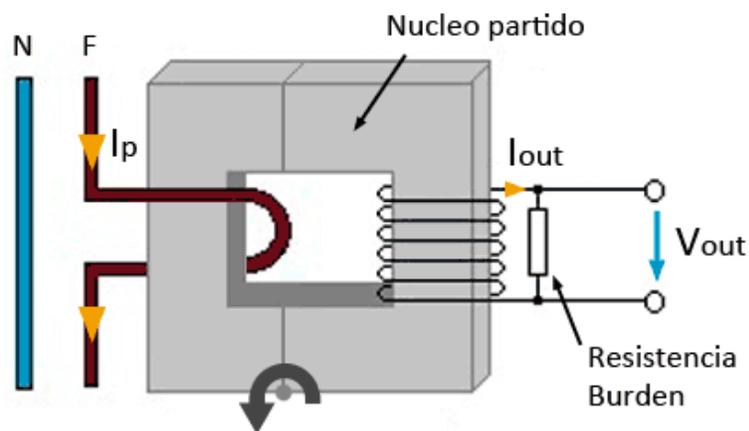
Un transformador de corriente es similar a un transformador de tensión, se basa en los mismos principios de funcionamiento, la diferencia se encuentra en que este, está dedicado a medir la corriente o intensidad, en consecuencia, están diseñados y contruidos de forma distinta. El transformador genera una intensidad

en el secundario, su valor es proporcional a la intensidad que atraviesa el primario.

Cuando la corriente alterna circula por el conductor se genera un flujo magnético en el núcleo, el conductor donde se transporta la corriente que se necesite medir cumple la función de espira primaria, generando un flujo magnético en el núcleo, a su vez esto genera una corriente eléctrica en el devanado secundario.

Al medir corriente alterna en el devanado primario; fase del motor, la intensidad inducida en el secundario es igualmente alterna. Al transportarse por la resistencia burden; interna del sensor, la salida de tensión también es alterna, considerando que las entradas analógicas de la mayoría de microcontroladores, incluidos Arduino, sólo son capaces de medir tensiones positivas sin componentes externos.

Figura 35. **Transformado de corriente**

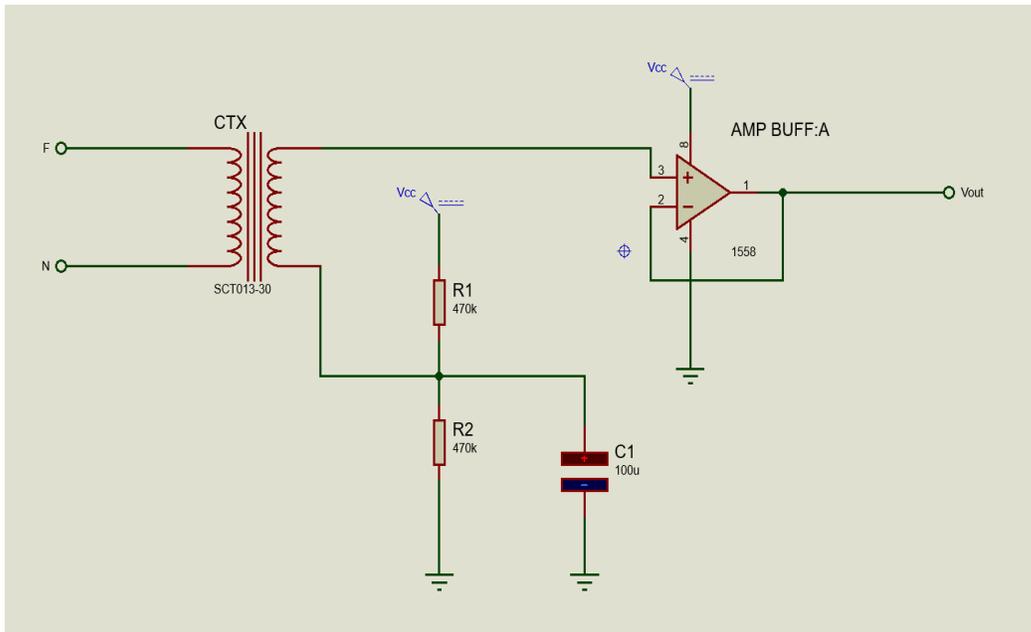


Fuente: LLAMAS, Luis. *Sensor de corriente eléctrica no invasivo*. www.luisllamas.es. Consulta: 20 de julio de 2019.

Para solucionar el problema de las tensiones negativas se aplica un circuito *offset*, se utiliza para obtener un desplazamiento en la amplitud de la señal sinodal para así evitar la señal negativa. Esto se logra mediante dos resistencias junto a un condensador, para proporcionar el punto medio entre GND y V_{CC} , véase la figura 36. Adicional al circuito también se utiliza un amplificador operacional configurado como *buffer* o seguidor de voltaje, este con el fin de aislar la señal de salida respecto a la señal de entrada.

De izquierda a derecha en la figura 36, se muestra el transformador de corriente CTX que convierte el campo magnético generado por el conductor de una de las fases del motor, a una señal de voltaje. Cada amperio en CA genera una variación de voltaje de aproximadamente $\pm 33,33$ mV, mientras más grande sea la corriente transportada por el conductor, más grande será el voltaje generado a las salidas del sensor. El límite de corriente sin saturar el núcleo del SCT013-30 es de 30 amperios AC, cuando este valor se introduzca, se tendrá una señal de voltaje de salida de ± 1 Voltio.

Figura 36. Circuito acondicionador de corriente



Fuente: elaboración propia, utilizando Proteus 8.

Para realizar la calibración de los rangos que reconoce la placa Arduino, es necesario considerar que se trabaja con corriente alterna, esta mayormente para denotar sus valores se emplean en RMS. Por lo tanto, cuando el sensor SCT 013 entregue un valor de ± 1 Voltio el Arduino en realidad estará leyendo el valor pico, este valor $V_{pico} = 1 \times 1,414 = \pm 1,414$ Voltios. En el circuito de la figura 36 contiene el punto *offset*, compuesto por V_{cc} las 2 resistencias y el condensador, si el valor entre las resistencias es de 2,5 V DC, al considerar este valor de tensión junto con el valor que está entregando el sensor, se obtiene un rango de $\pm 1,41 + 2,5 = 1,08$ a 3,92 V, de esta manera solo se manipulan señales positivas, estas se encuentran en el rango del Arduino.

El amplificador operacional del circuito se configura de tal manera que trabaje como un seguidor de voltaje, también se conoce como amplificador *buffer*, esta combinación de ganancia unitaria permite aislamiento de ruido electrónico,

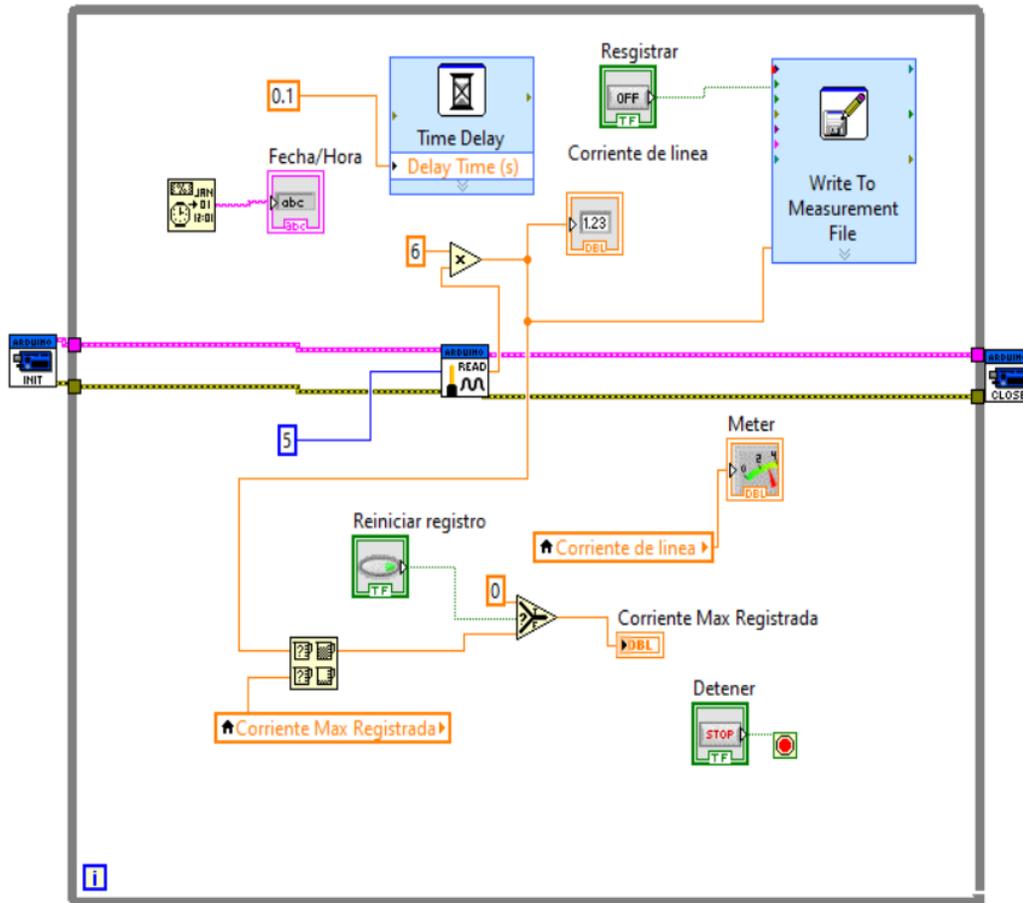
es un circuito con amplificador operacional que tiene una ganancia de un voltio. Entonces, no proporciona ganancia alguna a la señal y se encuentra para aislar el circuito del pin del Arduino evitando ruido o señales que interfieran con la medición de la corriente.

La figura 37 contiene el diagrama de bloques correspondiente al módulo de lecturas de corriente. Primero es necesario declarar la tarjeta Arduino con INT, acto seguido para cerrar el programa con CLOSE, luego se declara el pin de lectura analógico READ y se tiene que asignar al pin a utilizarse, en este caso se utiliza el 5. Para comodidad del usuario se agrega un indicador que muestra datos de fecha junto a la hora del ordenador, también se agrega un temporizador o *Time Delay* para limitar el uso de recursos del ordenador y del Arduino con un tiempo de 0,1 s.

El cable naranja del lado derecho, están conectados a un multiplicador por 6, este se utiliza como calibrador analógico del programa, se conecta a un indicador digital, que a su vez se conecta a una función de registro en conjunto con un botón booleano para activar o desactivar la función. También se conecta a un vúmetro virtual tipo aguja a manera de dar estética a la interfaz del usuario. En la parte inferior se tiene una función encargada de guardar la corriente pico o corriente máxima registrada, para tener el dato de la corriente de arranque del motor y se agrega un seleccionador que resetea el dato a cero en cuanto se presiona el botón de reinicio.

Por último, se agrega un botón para detener la ejecución del programa y para salir. El botón forma parte del *loop* que encierra todo el diagrama de bloques, es necesario declararlo para que LabVIEW no muestre ningún error de programación.

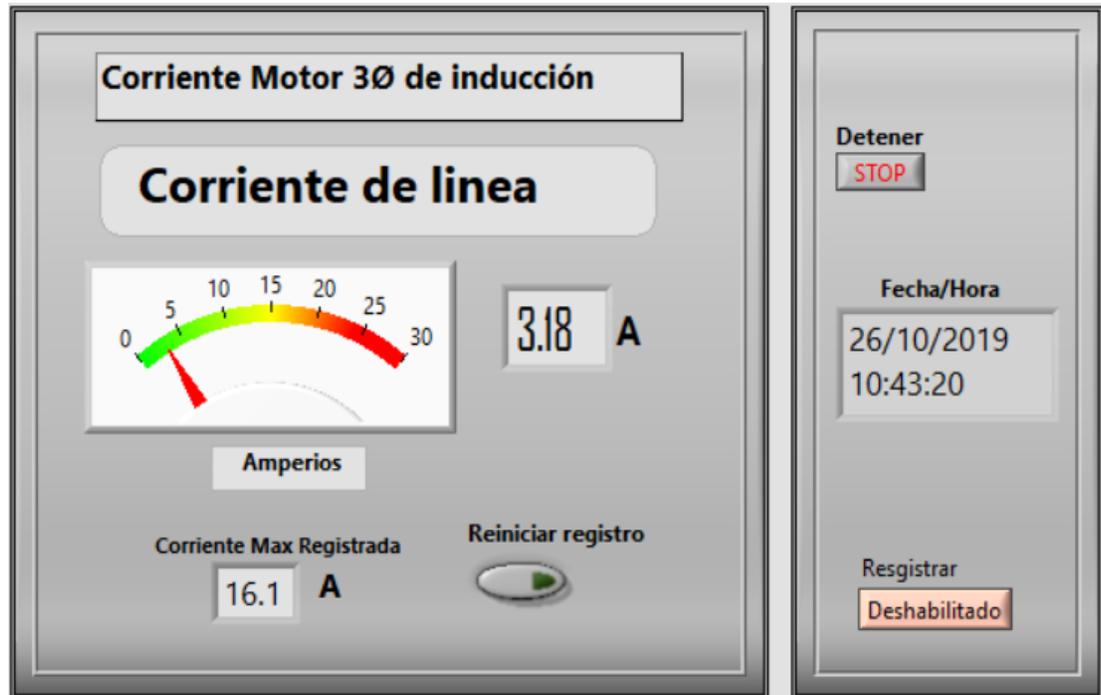
Figura 37. Diagrama de bloques del programa medición de corriente



Fuente: elaboración propia, utilizando LabVIEW 15.0.

El panel frontal accesible para el usuario se muestra en la figura 38, las funciones disponibles para interactuar con el programa son; registro el cual se encarga de guardar el valor de la corriente vs tiempo en una tabla de Microsoft Excel. Es necesario considerar que esta función consume más capacidad de procesamiento del microcontrolador de la placa Arduino. El botón “Reiniciar registro” está en la parte derecha, del indicador de “Corriente Max Registrada”, este indicador toma el valor más alto registrado en el sensor y lo guarda en la memoria, brindando la posibilidad de guardar el valor de la corriente de arranque.

Figura 38. Panel frontal del programa de medición de corriente



Fuente: elaboración propia, utilizando LabVIEW 15.0.

4.5. Medición de temperatura

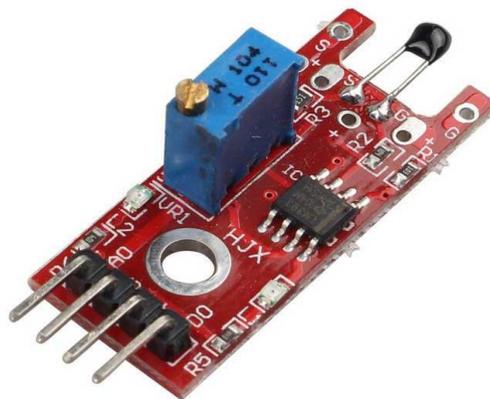
La temperatura es un parámetro muy influyente en el trabajo de un motor, ésta brinda información sobre la calidad o estado del mismo. Si un motor se sobrecalienta, trasfiere el calor a su núcleo ferromagnético, después a su carcasa y componente más importante sus devanados, por ende, estos se deterioran con mayor velocidad disminuyendo drásticamente su vida útil. Se considera que cada incremento de 10 °C en las bobinas de los motores por encima la temperatura de trabajo nominal, reduce la vida útil hasta un 50 % del motor, incluyendo si el sobrecalentamiento es temporal.

4.5.1. Sensor de temperatura

Para el transductor se utiliza el sensor KY-028, este se encuentra compuesto por 3 componentes principales, una parte que contiene el sensor en la parte frontal del módulo termistor NTC, éste se encarga de medir físicamente la temperatura como una señal analógica, para luego enviarla a la segunda parte de su propia placa, donde se realiza una amplificación de señal. En esta parte es posible variar la cantidad de ganancia según se necesite, esto se logra con un potenciómetro integrado en la propia placa del sensor, después se traslada la señal a una salida analógica del módulo.

La última etapa de esta pequeña placa es un comparador, este se encarga de controlar un led en función de la señal, en consecuencia, si la señal se encuentra bajo un valor específico enciende o apaga el led. Su rango de medición según el fabricante se encuentra entre los $-55\text{ }^{\circ}\text{C}$ hasta los $+125\text{ }^{\circ}\text{C}$, compuesto por un termistor NTC el cual disminuye su resistencia cuando la temperatura se eleva.

Figura 39. **Modulo sensor de temperatura**



Fuente: Arduino modules. *Módulo de Sensor de Temperatura Digital para Arduino*, www.Arduinomodules.info/ky-028-digital-temperature-sensor-module/. consulta: julio 2019.

Este módulo cuenta con 4 pines de conexión los cuales es posible conectar de manera directa con Arduino, permite aumentar o disminuir su sensibilidad con el potenciómetro que se encuentra sobre el encapsulado azul. Sus pines vienen etiquetados como:

- Pin A0 : sirve para leer la señal analógica del sensor, se conecta en los pines analógicos del Arduino.
- Pin GND: se conecta a la tierra de toda la circuitería incluyendo el Arduino.
- Pin +: es el pin de alimentación del módulo, soporta voltajes de alimentación de 3,3 V o 5 V DC.
- Pin D0: para conexión del pin digital con Arduino.

4.5.2. Programa para medir la temperatura

Acercas de la figura 40, se indica al software, trabajar con una tarjeta externa (Arduino), se realiza con las funciones INT y CLOSE externas del cuadro gris que cumple la función de realizar un bucle WHILE LOOP hasta que realice la pulsación del botón STOP, luego se declara la constante READ para indicar la utilización del pin analógico 0 encargado de recibir la señal proporcionada por el sensor de temperatura. Esta señal es inversamente proporcional al cambio de la temperatura (se reduce su valor si la temperatura aumenta y viceversa), por lo tanto, se aplica una función inversa $1/X$ para que los indicadores muestren el cambio de valor correspondiente.

Los valores de corriente son pequeños, por lo tanto, se aplica la función de multiplicador X ; este es el calibrador analógico del sensor de temperatura, ésta señal se transfiere a las funciones de Indicador de temperatura; o termómetro Indicador digital, también a la función gráfica y a una función que compara el valor con una constante que se define en función de la temperatura de trabajo del

motor. Cumple la función de cambiar de color un indicador virtual, cuando se supera el valor de la temperatura, dando así una función de alarma de alta temperatura.

Se agregan las funciones de registro, esto para tener la opción de guardar datos de la gráfica en una tabla de Microsoft Excel, en esta se tienen los datos tiempo y valores de temperatura.

Figura 40. Diagrama de bloques del programa de medición de temperatura

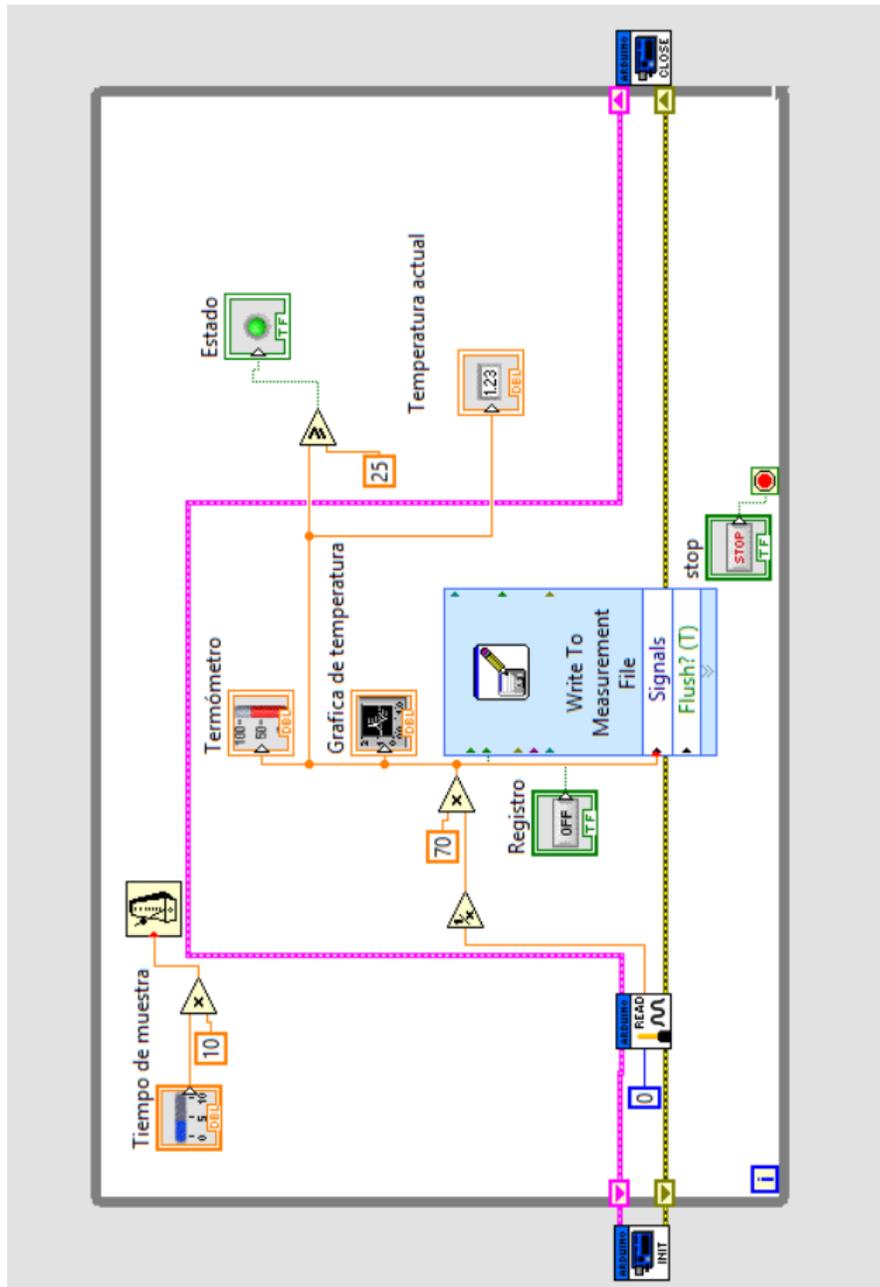
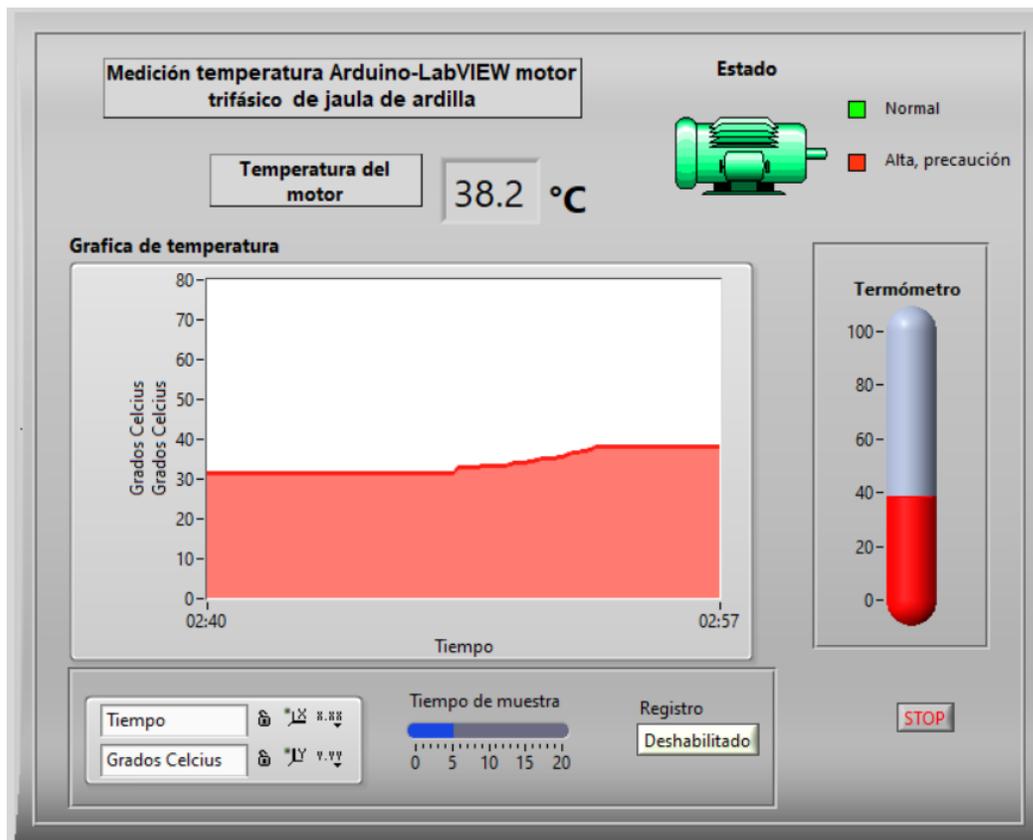


Diagrama de bloques correspondiente al programa de medición de temperatura.

Fuente: elaboración propia, utilizando LabVIEW 15.0.

La siguiente figura 41, corresponde al panel frontal del programa de medición de temperatura, contiene los indicadores termómetro gráfico y digital. Se tiene un led virtual que indica si el estado de la temperatura del motor se encuentra en nivel aceptable o normal, cambiando de color al momento de sobrepasar el límite establecido, contiene un boto STOP que detiene la ejecución del programa, un botón para registrar los datos de temperatura vs tiempo, por último un multiplicador de tiempo para aumentar o disminuir las muestras de temperatura que se gráfica en tiempo real.

Figura 41. **Interfaz entre diagrama de bloques y panel de control de usuario**



Interfaz del diagrama de bloques del programa de medición de temperatura

Fuente: elaboración propia, utilizando LabVIEW 15.

4.6. Diagrama de conexión desde motor hasta ordenador

Para realizar la conexión entre los sensores a la vez que garantizar la integridad, funcionalidad y protección de los mismos es necesario considerar una adecuada instalación de todos los elementos implicados en el sistema de monitoreo, en general es posible prevenir fallas del sistema de medición, considerando una instalación permanente de cada uno de sus componentes.

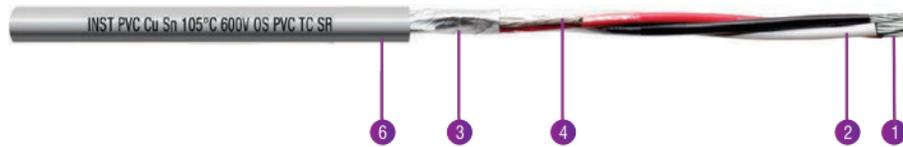
Las distancias entre cada elemento quedan en función de la instalación, no existe una especificación acerca de la máxima distancia permitida para los sensores, pero es aconsejable instalar el ordenador y dispositivos, lo más cerca posible al motor, siempre que no se afecte la integridad de cada uno de ellos.

4.6.1. Protección eléctrica del cable

El conductor recomendable para las señales analógicas, es decir el cable que se encarga de transportar la señal desde los transformadores incluyendo el módulo de temperatura hasta la placa Arduino es conocido como cable para instrumentación y control PLTC, este brinda la cobertura necesaria para transportar las señales analógicas evitando distorsiones o perturbaciones electromagnéticas externas que puedan sufrir las mismas durante el trayecto hasta las placas de circuitos.

Este tipo de conductores están cubiertos con una malla o capa conductora común, actuando como una jaula de Faraday, sin embargo, se recomienda alejar los cables de instrumentación y control a más de 1,5 m de los cables de alta tensión o alta potencia si se encuentran aledaños, esto disminuye considerablemente la probabilidad de ruido inducido por estos conductores.

Figura 42. **Cables para instrumentación PLTC**



Construcción

- 1 Conductor de cobre estañado (Cu Sn) clase B, calibre 16 y 18 AWG.
- 2 Aislamiento en PVC retardante a la llama, resistente a la abrasión, al calor y la humedad.
- 3 Cinta Poliéster aluminizada con 100% de cubrimiento.
- 4 Conductor de drenaje en cobre estañado 20 AWG.
- 5 Hilo de rasgado para facilidad en la instalación.
- 6 Chaqueta en PVC retardante a la llama, resistente a la abrasión, al calor y la humedad, para uso en bandejas tipo TC (Tray Cable).

Núcleo: Conductores individuales cableados.

Fuente: CENTELSA. *Cables para instrumentation y control.*

www.centelsa.com/pdf/CablesparaInstrumentacionyControl.pdf. Consulta: junio 2020.

4.6.2. Protección mecánica

Es necesario evaluar el ambiente físico al que se va someter el sistema de monitoreo, por lo que la protección mecánica del mismo es necesaria, para esto es necesario analizar las variables que puedan interferir en la integridad física del cable, se recomienda considerar los siguientes aspectos:

- Fuego o material inflamable: implica considerar el comportamiento frente al fuego, se debe garantizar resistencia al fuego, en conjunto a evitar propagaciones del mismo.
- Intemperie: principalmente las radiaciones solares, es aconsejable evitar que el cable trabaje a la intemperie expuesto a los rayos UV, o inclemencias provocadas por lluvia, esto ayuda a extender la vida útil del conductor.

- El ambiente de trabajo: los ambientes corrosivos o abrasivos necesitan consideración especial, pero en general se recomienda aplicar canalización del cableado, se utiliza principalmente para brindar protección contra golpes ya sea de maniobra o accidentales, es una excelente medida de protección principalmente si es canalización permanente en pared con metal sólido.

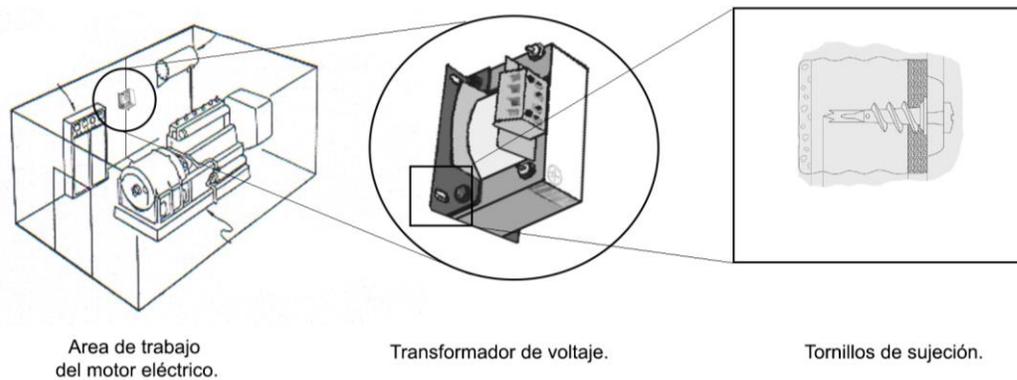
Para garantizar la efectividad de la protección de los cables internos contra las interferencias, el blindaje debe guardar los conductores a la vez evitar discontinuarse entre extremos, también es necesario realizar conexión con la puesta a tierra del sistema eléctrico para evitar la circulación indeseada de ruido o señales que puedan viajar en el blindaje metálico.

4.6.3. Instalación del transformador de voltaje

Las conexiones del transformador de voltaje deben conectarse lo más cercano posible a las borneras del motor, empalmando dos cables de alimentación, en el lado de alto voltaje; devanado primario del lado del motor, en el lado de bajo voltaje; devanado secundario, se conecta la placa del circuito rectificador, éste a su vez, debe instalarse cerca del Arduino y el ordenador que se encargara de ejecutar el programa en LabVIEW, alejándolos del ambiente industrial que pueda afectar el óptimo funcionamiento electrónico.

Para instalar permanente el transformador es necesario sujetarlo, mediante abrazaderas de metal, se aconseja utilizar tornillos con su respectivo espacio perforado en la superficie donde se realizará la instalación del transformador ya sea una mesa, banco de madera, o inclusive la pared si se encuentra cercana; por ejemplo, a 1mt de distancia del motor a monitorear. la instalación debe evitar las vibraciones generadas por el funcionamiento del motor.

Figura 43. **Diagrama de instalación**



Fuente: elaboración propia, utilizando Adobe Photoshop CS6.

4.6.4. Instalación del transformador de corriente

Tratándose de transformadores de corriente no invasivos, la conexión del sensor, solo requiere abrazar de manera adecuada uno de los cables de alimentación de corriente trifásica del motor, considerando que, este no sea interrumpido por cualquier otro campo electromagnético intenso, evitando así perder sensibilidad o lecturas erróneas en el circuito. Otro aspecto importante es alejarlo de vibraciones o movimientos que desacoplen la unión del transformador de corriente. Objetivo que se puede lograr sujetando el transformador de manera permanente con abrazaderas atornilladas a una superficie sólida.

Este transformador es capaz de operar a la intemperie, siempre que el nivel de humedad no sea alto. Su salida se conecta al PCB del circuito acondicionador de corriente; figura 36, este debe instalarse cerca del Arduino y por ende cerca del ordenador, siempre alejado del ambiente industrial que pueda generar daños a la electrónica de los mismos.

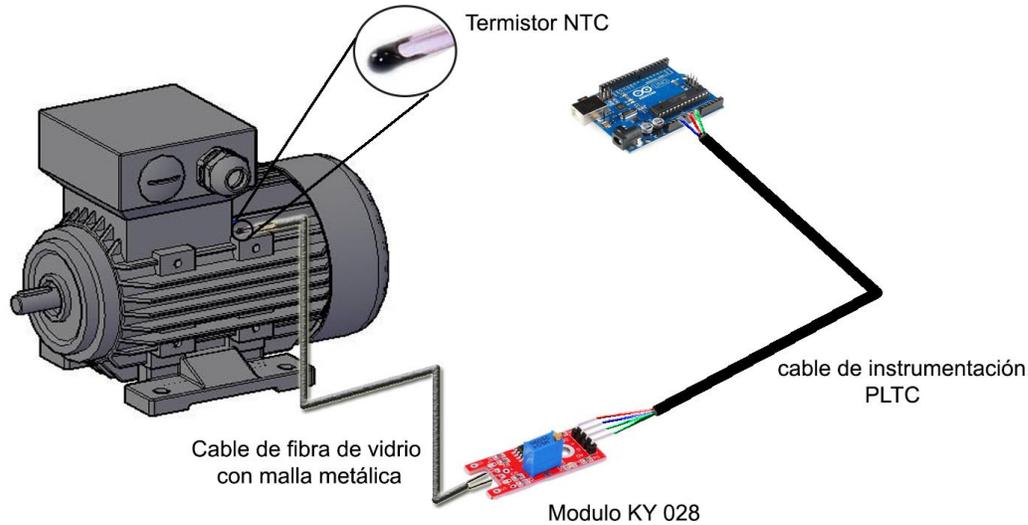
4.6.5. Instalación del sensor de temperatura

Para realizar esta conexión, es necesario considerar que este módulo; el sensor KY- 028 tiene que instalarse, lo más cerca posible a la carcasa del motor eléctrico, se considera que la parte más conveniente a monitorear del mismo es su debandado interno, puesto que este se encuentra en la periferia del estator, por lo tanto, este trasfiere el calor a su carcasa, razón por la que se requiere instalación del sensor en esta misma.

Según su fabricante el sensor, está capacitado para soportar hasta 125 °C, si se considera que el sensor estará expuesto a una alta temperatura por un tiempo muy prolongado, es posible modificarlo, tratando de extender mediante cables el termistor y alejar así la placa de este módulo del calor generado por el funcionamiento del motor.

La modificación del módulo implica extender los conductores del termistor NTC, sabiendo que los motores tienden a calentarse durante su funcionamiento se aconseja utilizar cable adecuado para soportar altas temperaturas, por ejemplo, cable para termorresistencia; cable asilado con fibra de vidrio con malla metálica, este cable se fabrica para soportar temperaturas elevadas, para su instalación es necesario sujetarlo mediante abrazaderas de metal, atornilladas a la carcasa del motor, sujetando permanentemente el cable a la carcasa del motor y en consecuencia el termistor a la misma, considérese la figura 44.

Figura 44. **Instalación del sensor de temperatura KY-028**



Fuente: elaboración propia, utilizando Adobe Photoshop CS6.

4.7. Distancia de instalación

La distancia recomendada para la longitud del cable PLTC depende de diferentes factores, siendo el más influyente el calibre del conductor, la siguiente tabla muestra la distancia recomendada para los conductores con blindaje para el par de conductores.

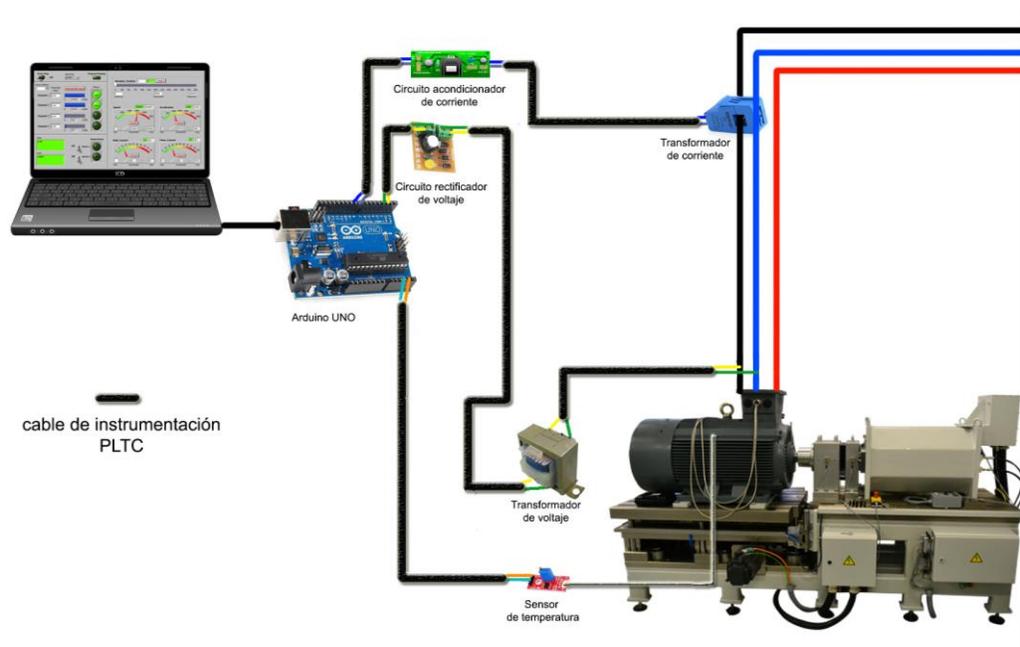
Tabla VI. **Distancia recomendada para cable con par blindado**

Sección nominal	Espesor nominal de aislamiento	Número de pares	Diámetro externo del cable	Masa	Longitud de acondicionamiento
AWG	mm		mm	kg/km	m
20	0,30	1	5,5	38	500
18	0,38	1	6,2	49	500
16	0,38	1	6,8	61	500
14	0,51	1	8,4	90	1000
12	0,51	1	9,3	120	1000

Fuente: PRYSMIAN GROUP. *Cables para Instrumentación Electrónica*.
www.ar.prysmiangroup.com. Consulta: 23 de junio de 2020.

La distancia máxima recomendada para realizar la instalación del sistema de monitoreo implica no exceder la distancia del cable de instrumentación, dado la tala IV se concluye que esta no debe ser mayor a 500 m, sin embargo, es necesario realizar pruebas de calibración en cada circuito en conjunto con LabVIEW, en el caso de requerirse más distancia para dicha instalación se recomienda la utilización de amplificadores para las señales analógicas.

Figura 45. Diagrama de conexión del sistema de monitoreo



Representación de la conexión entre motor, sensores, circuitos, Arduino y LabVIEW.

Fuente: elaboración propia, utilizando Adobe Photoshop CS.

CONCLUSIONES

1. Los motores eléctricos trifásicos de inducción con rotor de tipo jaula de ardilla, son los más utilizados en la industria en general, por su robustez, bajo mantenimiento y versatilidad por operar con corriente alterna según lo visto en el capítulo uno; además, el monitorear su comportamiento permite analizar el estado de éste, previniendo fallas o imprevistos y reduciendo pérdidas económicas producidas.
2. Arduino es una plataforma de hardware y software versátil, considerando su disponibilidad en el mercado a nivel mundial, capaz de operar incluso en aplicaciones industriales, ofreciendo diversidad de tarjetas en función de las necesidades del proyecto o trabajo según lo descrito en el capítulo dos; una ventaja considerable es su software nativo IDE, siendo este de libre acceso y permite la fácil programación del microprocesador.
3. Según el capítulo tres, se puede determinar que LabVIEW permite desarrollar aplicaciones sin necesidad de amplio conocimiento de programación basada en texto, y también permite la capacidad de trabajar en conjunto con Arduino UNO R3 por medio de conexión USB, permitiendo lectura de señales de manera rápida y eficiente.
4. El programa desarrollado en el capítulo cuatro, permite la lectura de parámetros eléctricos como la corriente, el voltaje y la temperatura de un motor trifásico, capaz de registrar el comportamiento de estos, brindando así la capacidad de visualizar su variación, mostrándolos de manera

gráfica en una interfaz amigable con el usuario mediante LabVIEW ejecutado en un ordenador.

RECOMENDACIONES

1. Ampliar la cantidad de parámetros a monitorear, por ejemplo: velocidad angular, vibración y factor de potencia. Contribuye a obtener datos más precisos acerca del funcionamiento de la máquina, mejorando la calidad del monitoreo. Esto permite hacer énfasis en el mantenimiento hacia cierta parte de la máquina.
2. Si se utiliza una placa diferente al Arduino UNO R3, debe realizarse una investigación específica acerca de las capacidades de las diferentes placas existentes en el mercado.
3. Utilizar un ordenador dedicado para el sistema de monitoreo, con sistema de alimentación ininterrumpida, con al menos 4Gb de RAM, procesador Core i3 con disco duro de 100 Gb ejecutando Windows 10 professional de 64 bits, esto ayuda a garantizar el funcionamiento continuo del programa y conexión con Arduino.
4. Es necesario establecer un tiempo de muestreo de 0,1 s para la programación de LabVIEW SP1 2015, principalmente durante el registro de los datos, estas funciones consumen considerablemente la capacidad de procesamiento del Arduino.

BIBLIOGRAFÍA

1. Arduino. *Proyectos con Arduino*. [en línea]. <www.proyectosconarduino.com/placas>. [Consulta: 20 de septiembre de 2019].
2. ARTEAGA NOGUERA, Luis Enrique. *Normas aplicables a la electrotecnología de motores*. [en línea]. <www.1library.com>. [Consulta: 20 de diciembre de 2019].
3. CASTELLI, Marcelo. *Metodología de monitoreo, detección y diagnóstico de fallos en motores asíncronos de inducción*. [en línea]. <www.core.ac.uk>. [Consulta: 10 de diciembre de 2019].
4. CHAPMAN, Stephen j. *Máquinas Eléctricas* 5a ed. México DF: McGraw-Hill, 2012. 502 p.
5. EVANS, Brian W. *Arduino programing notebook: A Beginner's Reference*. [en línea]. <www.academia.edu>. [Consulta: 2 de diciembre de 2019].
6. FRAILE MORA, Jesús. *Principios generales de las máquinas eléctricas*. 6a ed. Madrid: McGraw-Hill, 2008. 809 p.
7. JULIAN LAIME, Edgar Rafael. *Manual de programación LabVIEW 9.0*. [en línea]. <www.researchgate.net>. [Consulta: 16 de diciembre de 2019].

8. National Instruments Corp. *LabVIEW user manual*. [en línea]. <www.ni.usermanual.com>. [Consulta: 20 de octubre de 2019].
9. _____. *LabVIEW™ Core 1 Course Manual*. [en línea]. <www.ni.core1coursemanual.com>. [Consulta: 14 de enero de 2020].
10. WILDI, Theodore, *Máquinas eléctricas y sistemas de potencia* 6a ed. México: PEARSON Educación, 2007. 960 p.