



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN

DESARROLLO DE SISTEMA PARA SUPERVISIÓN,  
CONTROL Y ADQUISICIÓN DE DATOS DE CENTRAL  
HIDROELÉCTRICA EXPERIMENTAL

TESIS

QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO MECÁNICO ELECTRICISTA

PRESENTA:

LUIS DANIEL GUZMÁN GUILLÉN

ASESOR:

ING. ALBINO ARTEAGA ESCAMILLA

CUAUTITLÁN IZCALLI, ESTADO DE MÉXICO, 2018



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN  
UNIDAD DE ADMINISTRACIÓN ESCOLAR  
DEPARTAMENTO DE EXÁMENES PROFESIONALES

U. N. A. M.  
FACULTAD DE ESTUDIOS

ASUNTO: VOTO APROBATORIO

M. en C. JORGE ALFREDO CUÉLLAR ORDAZ  
DIRECTOR DE LA FES CUAUTITLÁN  
PRESENTE



ATN: I.A. LAURA MARGARITA CORTAZAR Y EL CERRA  
Jefa del Departamento de Exámenes Profesionales

DEPARTAMENTO DE EXÁMENES PROFESIONALES  
CUAUTITLÁN

PROFESOR

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: **Trabajo de Tesis**

Desarrollo de sistema para supervisión, control y adquisición de datos de central hidroeléctrica experimental

Que presenta el pasante: LUIS DANIEL GUZMÁN GUILLÉN

Con número de cuenta: 41310871-9 para obtener el Título de la carrera: Ingeniería Mecánica Eléctrica

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

ATENTAMENTE

"POR MI RAZA HABLARÁ EL ESPÍRITU"

Cuautitlán Izcalli, Méx. a 08 de enero de 2018.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	Ing. Nicolás Calva Tapia	
VOCAL	M. en T.I. Jorge Buendía Gómez	
SECRETARIO	Ing. Albino Arteaga Escamilla	
1er. SUPLENTE	M. en C. Leopoldo Martín Del Campo Ramírez	
2do. SUPLENTE	Ing. Victor Martínez Tovar	

NOTA: los sindonales suplentes están obligados a presentarse el día y hora del Examen Profesional (art. 127)

## **Dedicatorias**

A:

Papá y mamá... por su gran apoyo, esfuerzo y confianza en mí, doy gracias a ese ente creador de todo por ponerlos en mi vida, por todas sus enseñanzas intencionadas y no, porque han hecho de mí una persona de fe, coraje y amor y todo lo que soy. Gracias por ayudarme a triunfar, a cumplir mis sueños, por darme valor cuando más lo necesito, por estar conmigo, por enseñarme que en amor lo puedo todo y porque no importa la distancia “un día llegare”, no importa lo que haya que pasar, ese día es hoy y es gracias a ustedes.

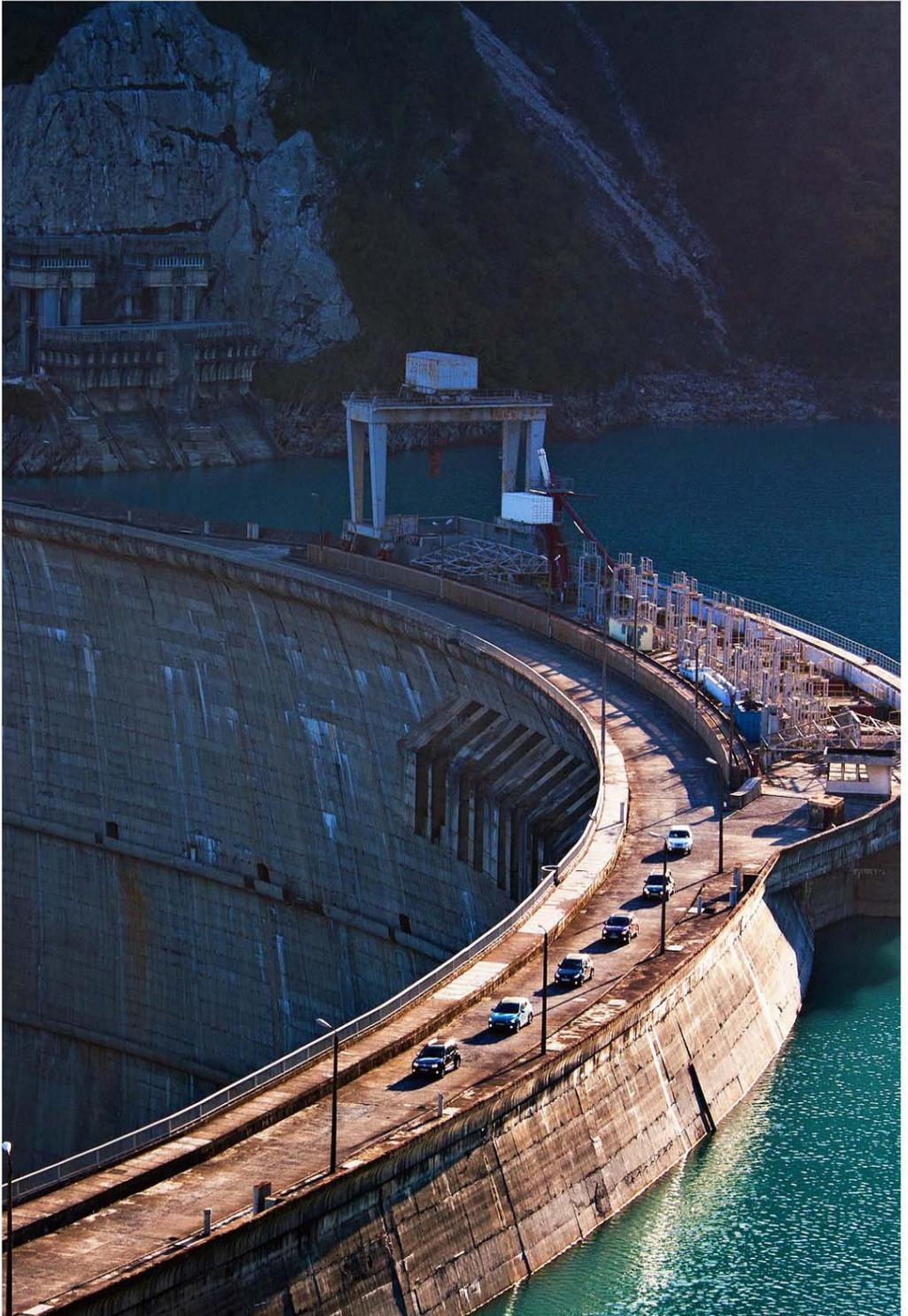
Erick y Silvana... por estar conmigo, por hacer de mi vida una aventura con todas nuestras experiencias juntos, gracias por dejarme cuidar de ustedes y ustedes de mí, por todos esos ratos jugando al max steel y con sónica. A veces cometemos errores, pero somos hermanos y siempre estaremos juntos. Como dije antes, alcanzar mis sueños y metas implica recorrer un largo camino, pero esté en donde esté yo siempre los llevo en mi corazón.

Ibeth... por ser de verdad un pilar muy fuerte en mi vida y por hacer de ella una nueva aventura junto a ti y por todo lo que nos espera, gracias por tu inteligencia, por tu cariño y amor, por darme ánimo y fuerza para lograr todo lo que me propongo, gracias por estar conmigo en este camino que juntos recorreremos hasta alcanzar nuestra meta. Aún nos queda mucho por recorrer pero estoy muy seguro que juntos venceremos cualquier obstáculo.

A mis amigos... gracias por todos esos momentos de diversión y locura, por sus consejos y gracias por haber llegado a mi vida. Erick Santos, Dante, Néstor, Marco, Eduardo, Fernando, Isaac, Didier, Mario, Diego J., Isaac V, Víctor, Miguel y todos... Gracias por su amistad.

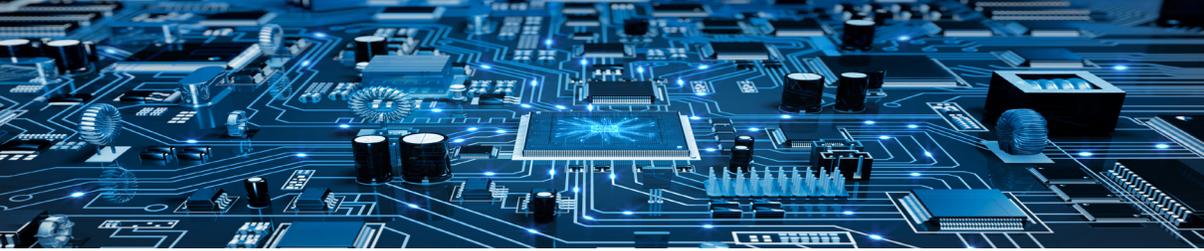
A mi asesor... Ing. Albino Arteaga Escamilla por su conocimiento, intelecto, dedicación y consejo durante el desarrollo de este trabajo.

Quiero hacer mención de los ingenieros José Antonio Sánchez y Ricardo Delgado por su ayuda y cooperación en el laboratorio de Termofluidos para la realización del proyecto, así como de los ingenieros Leopoldo Martín del Campo, Nicolás Calva, Jorge Buendía y Víctor Martínez por su excelente crítica y consejo durante la revisión del trabajo.



<b>1</b>	<b>INTRODUCCIÓN</b>	<b>1</b>
	1.1 Objetivo	4
	1.2 Objetivos específicos	5
	1.3 Justificación del trabajo	5
<b>2</b>	<b>¿QUÉ ES SCADA?</b>	<b>7</b>
	2.1 Objetivos de un sistema SCADA	7
	2.2 Funciones	9
	2.3 Ventajas y desventajas	11
	2.4 Realización del proyecto SCADA para central hidroeléctrica	13
<b>3</b>	<b>INTEGRACIÓN DE SCADA</b>	<b>15</b>
	3.1 Componentes de un SCADA	17
	3.2 El hardware	19
	3.2.1 Placas ARDUINO (UNO & NANO)	21
	3.3 El software (ARDUINO IDE & LabVIEW)	25
	3.4 Sensores y drivers	33
	3.5 Interfaz humano máquina (HMI)	46
	3.6 Control de acceso	52
	3.7 Protocolo de comunicación	56
	3.8 Alarmas	71
	3.9 Registro de datos	76
<b>4</b>	<b>PLANTA HIDROELÉCTRICA</b>	<b>81</b>
	4.1 Instalación actual	88
	4.2 Modificaciones de la planta	90
	4.3 El arternador	92
	4.4 Indicadores de planta y parámetros	98

<b>5</b>	<b>DISEÑO Y DESARROLLO DE HMI</b>	<b>101</b>
5.1	Sensor de nivel	102
5.2	Sensor de velocidad (RPM)	108
5.3	Sensor de corriente	112
5.4	Sensor de voltaje	120
5.5	Indicador de caudal	124
5.6	Indicador de velocidad de flujo	126
5.7	Indicador presión hidráulica	130
5.8	Indicador de carga eléctrica	131
5.9	Indicador de factor de potencia	131
5.10	Indicador de frecuencia	133
5.11	Indicador de fuerza	135
5.12	Indicador de carga neta	136
5.13	Indicadores de parámetros de planta	137
5.14	Control motor-bomba	142
5.15	Control de motor a pasos (inyector)	149
5.16	Control de cargas	152
5.17	Video vigilancia	155
5.18	Animación SCADA	157
<b>6</b>	<b>PRUEBAS Y RESULTADOS</b>	<b>160</b>
6.1	Análisis de resultados	164
<b>7</b>	<b>CONCLUSIONES</b>	<b>170</b>
	Bibliografía	178
	Sitios Web	180
	Anexo	184



# Capítulo 1

## Introducción

Un conjunto de conocimientos técnicos ordenados, que nos permiten crear bienes, nos facilitan la adaptación a cualquier medio y que nos ayuda a satisfacer las necesidades y deseos de la humanidad es a lo que se le conoce como tecnología; pues bien, hay diferentes tipos de tecnología y para este trabajo se comenzará clasificando el tipo de tecnología que se está desarrollando.

Según el sociólogo estadounidense James D. Thompson, en su tipología de tecnologías por cómo influye en las empresas, instituciones y organizaciones, éste proyecto entra en las siguientes clasificaciones.

- Tecnología flexible: Es toda aquella que puede utilizarse en muchas áreas de aplicación. Los sistemas SCADA son utilizados en áreas de producción automotriz, energética, aeronáutica, minera, farmacéutica, etc.
- Tecnología dura: Son los objetos, máquinas y dispositivos que tienen que ver con el proceso industrial, es el hardware y los objetos tangibles

creados por el hombre los cuales hacen el trabajo más eficaz y propician la generación de productos y servicios con mejor calidad, novedad e integridad.

Al utilizar computadoras, chips, motores y otras máquinas, éste trabajo cumple con la clasificación.

- Tecnología adecuada: También conocida como apropiada, es aquella que requiere pocos recursos, es más barata y genera un bajo impacto ambiental en comparación con tecnologías de punta que son más complicadas, caras y en ocasiones dañinas.
- Tecnología energética y limpia: energética porque es destinada a su producción en centrales hidroeléctricas, eólicas, solares, etc., y limpia porque no se utilizan combustibles fósiles que dañen el medio ambiente para su obtención.
- Tecnología educativa: Se entiende como el acercamiento científico basado en la teoría de sistemas que proporciona al educador las herramientas de desarrollo a través de recursos tecnológicos con el fin de mejorar el proceso de la enseñanza y aprendizaje maximizando el progreso educativo y buscando la efectividad del aprendizaje.

Ahora que se sabe el tipo de tecnología que se está aplicando y desarrollando se hablará y analizará hacia dónde se dirigen los sistemas SCADA.

El concepto de INDUSTRIA 4.0 o también conocido como SMART FACTORIES es relativamente nuevo, mencionado por primera vez en 2011 en la feria de Hannover, Alemania, y presentado un concepto más detallado de éste en la misma feria en 2013.

La INDUSTRIA 4.0 se refiere a una nueva manera de organizar los medios de producción, a una mayor adaptabilidad a las necesidades y procesos de producción, así como una mejor gestión de los recursos, lo que lleva a una vía de la nueva revolución industrial o cuarta revolución industrial.

Este nuevo tipo de industria implica un grado mayor de automatización y de digitalización de fábricas, haciendo uso intensivo de internet y los sistemas Ciber-físicos que son redes virtuales con la capacidad de controlar objetos físicos. La modernización de las plantas de fabricación hasta el punto de volverlas inteligentes es caracterizada por la intercomunicación continua e instantánea entre todas las estaciones de trabajo que componen la cadena de producción.

La INDUSTRIA 4.0 se centra en 9 pilares tecnológicos:

1. Big Data y análisis de los datos (Datos a gran escala)
2. Robots autónomos

3. Simulación
4. Sistemas para la integración vertical y horizontal
5. IoT (Internet de las Cosas)
6. Ciber seguridad
7. Cloud computing (Servicios de computación en la nube)
8. Fabricación aditiva
9. Realidad aumentada

Pues bien, aquí es en donde entran al juego de las SMART FACTORIES los sistemas SCADA, ya que varios de los pilares que conforman la INDUSTRIA 4.0 son incluidos dentro de estos.

## **1.1 Objetivo**

Incorporar un sistema de supervisión, control y adquisición de datos (SCADA) de bajo costo, a la instalación de la central hidroeléctrica experimental del laboratorio de Termofluidos para mejorar su funcionamiento, con el fin de proporcionar un control unificado mediante el desarrollo de una interfaz amigable y sencilla para cualquier usuario/operador, que facilite el diagnóstico en tiempo real sobre el estado general de la instalación, para ser analizado e interpretado, haciendo que la toma de decisiones sea más eficiente.



## 1.2 Objetivos Específicos

- Examinar y evaluar la instalación actual de la planta.
- Diseñar la interfaz entre los dispositivos de campo y el software.
- Desarrollar el código para el microcontrolador que permitirá la adquisición y el control.
- Desarrollar el software que enlazará al microcontrolador con la instalación para la supervisión y monitoreo de la planta.
- Realizar las modificaciones y ajustes a la instalación que permitirán la implementación del sistema SCADA.
- Comprobar el correcto funcionamiento de la planta con el sistema SCADA.
- Generar un reporte de los datos adquiridos de la central hidroeléctrica.

## 1.3 Justificación del Trabajo

El presente trabajo de tesis se enfoca principalmente a la modernización de la instalación de pruebas de la central hidroeléctrica del laboratorio de Termofluidos mediante la incorporación del sistema SCADA utilizando equipo de bajo costo y fácil operación.

Como consecuencia de la implementación de este sistema de supervisión,

control y adquisición de datos, los alumnos que realicen prácticas de laboratorio estarán en contacto con un ambiente de aprendizaje moderno que ayudará a la familiarización con las nuevas tecnologías en cuanto a la visualización de datos y obtención de la información.

En cuanto al hardware, la decisión de utilizar la tarjeta ARDUINO se debe a la relación calidad-costos, así como la variedad de sensores disponibles en el mercado, haciendo de este equipo el ideal para este proyecto.

Por tanto, éste sistema de supervisión y control se concibe de forma segura, confiable y económicamente accesible y se ajusta a las necesidades de una planta hidroeléctrica experimental.



## Capítulo 2

### ¿Qué es SCADA?

Un sistema de supervisión, control y adquisición de datos (SCADA) es cualquier aplicación de software que se comunica con los dispositivos de campo destinados al control de actuadores y/o adquisición de datos en forma automática mediante algún protocolo de comunicación que proporciona información sobre algún proceso industrial.

Por sus siglas en inglés (Supervisory Control And Data Acquisition) un sistema SCADA permite realizar acciones de monitoreo y supervisión sobre las variables de un sistema a controlar que servirán para ser estudiadas y comparadas al ser adquiridas por el sistema y almacenadas en una base de datos.

#### **2.1 Objetivos de un sistema SCADA**

Entre los principales y más comunes objetivos que estos sistemas son capaces de realizar encontramos los siguientes:

- **Economía:** Contar con un SCADA facilita la visualización y percepción de lo que ocurre en la instalación desde una oficina a diferencia de enviar un operador a realizar esta tarea. Determinadas inspecciones se vuelven innecesarias.
- **Accesibilidad:** Configurar parámetros de funcionamiento se vuelve tan fácil como dar un clic o un “touch” al panel, así como visualizar datos en forma gráfica.
- **Mantenimiento:** La adquisición de datos materializa la posibilidad de obtener datos de un proceso, almacenarlos y presentarlos de manera inteligible para un usuario no especializado. Pueden programarse fechas de revisión o visualizar comportamientos anormales en su funcionamiento y realizar maniobras correctivas y preventivas.
- **Gestión:** Los datos recopilados pueden ser valorados de múltiples maneras mediante herramientas estadísticas, gráficas, tablas, etc., que permitirán explotar el sistema con mayor rendimiento.
- **Flexibilidad:** Modificar alguna característica del sistema de visualización del interfaz humano máquina (HMI) no representa un gasto en tiempo o medios ya que no hay modificaciones físicas que requieran la instalación de algún cableado etc.

- **Conectividad:** Permite el enlace con diferentes servidores, usuarios, bases de datos o controladores, agilizando la transferencia de datos y ampliándola para un mejor manejo de la información, a su vez permite la realización de acciones remotas de forma ágil.

## **2.2 Funciones**

- **Monitorización:**  
Representación de datos en tiempo real a los operadores o usuarios de planta. Se leen datos de las variables a controlar por los autómatas para poder ser vigilados incluso a grandes distancias, cualquier anomalía puede ser identificada por el usuario.
- **Supervisión:**  
Monitoreo, mando, y adquisición de datos de un proceso y herramientas de gestión para la toma de decisiones como la aplicación de un mantenimiento predictivo. Además, se pueden ejecutar programas que bajo ciertas condiciones anulen o modifiquen tareas asociadas a los autómatas. Evita la continua supervisión humana.
- **Adquisición de datos de los procesos:**  
Esto permite que los datos sean registrados en una base para obtener algún valor con algún significado para después ser evaluados y posteriormente procesados, esto agiliza la toma de decisiones de manera eficaz y exacta.

- Mando:

Posibilidad de que los operadores puedan cambiar consignas u otros datos claves del proceso directamente desde la PC como marcha, paro o modificación de parámetros. Se escriben datos sobre los elementos de control.

- Memorizar acciones:

Permite grabar acciones programadas, es decir, grabar una configuración determinada para una máquina que opera bajo condiciones exactas.

- Seguridad de los datos:

Tanto el envío como la recepción de datos están protegidos de influencias no deseadas, intencionadas o no, como fallos en la programación, intrusos, situaciones inesperadas, etc. Esto debido a que el código no está al alcance de cualquier persona para que pueda modificar algún parámetro o diseño de la configuración.

- Seguridad en los accesos:

Permite la restricción de zonas del programa comprometidas a usuarios no autorizados, o la restricción total de todos los accesos y acciones llevadas por cualquier operador.

## 2.3 Ventajas y Desventajas

Las ventajas más evidentes de los sistemas de control automatizado y supervisado se enumeran a continuación:

- Creación de aplicaciones funcionales sin necesidad de conocimiento profundo de la materia.
- Los sistemas autómatas que son dispositivos electrónicos programables con la finalidad de controlar en tiempo real en aplicaciones industriales están diseñados para trabajar en condiciones adversas, proporcionando robustez y fiabilidad al sistema que controla.
- Adaptación de los autómatas a las necesidades actuales y ampliación posterior en caso de ser necesario.
- Sensores y actuadores pueden ser integrados al sistema de PLC (Controlador Lógico Programable) mediante tarjetas de adquisición de datos, estos dispositivos PLC son computadoras utilizadas en ingeniería para la automatización de procesos electromecánicos, electroneumáticos y robótica.
- Herramientas de diagnóstico que localizan más rápido las fallas y errores en el sistema, esto permite minimizar los periodos de paro en las instalaciones.
- Puede definirse una unidad terminal remota (RTU) de modo que funcione de forma autónoma aún sin comunicaciones con la terminal maestra.

- Los sistemas de diagnóstico implementados en los elementos de control informan continuamente de cualquier incidencia en los equipos.
- Los programas de visualización pueden presentar todo tipo de ayuda al usuario, desde la aparición de una alarma hasta la localización de una falla. Esto permite reducir los tiempos de localización de averías al proporcionarse información sobre el origen y las causas de los fallos.
- El sistema de visualización puede recoger los datos del autómatas y presentarlos en formatos fáciles de exportar a otras aplicaciones, como documentos en Excel o Word.
- Posibilidad de mantenimiento por parte de suministradores locales de servicios mediante informes vía internet o GSM a los suministradores.
- Los protocolos de seguridad permiten una gestión segura y eficiente de los datos limitando el acceso a personas no autorizadas.
- Aumento de la calidad del producto mediante las herramientas de diagnóstico. El operador es notificado en el momento en que se detecta una incidencia.
- Reducción de supervisión humana.
- La distribución de recursos y control sobre la red permite una mejor coordinación entre las estaciones remotas en caso de fallos en una de ellas.
- Mediante las redes de comunicación, el sistema SCADA se integra a la red industrial, permitiendo la integración entre los niveles de campo y gestión y completa así la estructura CIM (Computer Integrated Manufacturing).



Desventajas:

- Programación compleja debido a diferentes plataformas y lenguajes.
- Fallos independientes del sistema debido a la programación.
- Inseguridad por parte del operador debido al cambio de controles analógicos por digitales.
- Susceptibilidad al hackeo y robo cibernético.

## **2.4 Realización del proyecto SCADA para central hidroeléctrica**

En cuanto al proyecto a desarrollar, la planta de generación eléctrica ubicada en el laboratorio de Termofluidos de la Facultad de Estudios Superiores Cuautitlán es una instalación destinada al aprendizaje de los alumnos de la Universidad en el área de turbomaquinaria, la planta está conformada por un módulo de motor-bomba centrífuga, tanque de agua, turbina Pelton y generador eléctrico. Incorporando un sistema SCADA se aportaría una nueva y mejor experiencia en el aprendizaje de los alumnos de la Universidad con un banco de pruebas de generación hidroeléctrica más intuitivo, exacto y por supuesto más moderno.

Como herramientas de trabajo se utilizará el software LABVIEW para el diseño de la interfaz humano-máquina (HMI) que estará enlazada con el microcontrolador ARDUINO, que será utilizado como tarjeta de adquisición de datos y como controlador de los actuadores de la planta.

Se utilizará un protocolo de comunicaciones SERIAL PORT entre la tarjeta ARDUINO y la estación maestra, que en este caso es una laptop LENOVO T420.



## Capítulo 3

### Integración SCADA

Para poder consolidar un sistema de supervisión, control y adquisición de datos es necesario contar con los siguientes aspectos:

- Planta o sistema

La planta a la cual se realizará la adición del sistema SCADA con el fin de mejorar y facilitar su operación.

- Comunicación

Mediante algún protocolo de comunicaciones (TCP/IP, o Serial Port, por mencionar algunos), un conjunto de equipos informáticos, como cualquier computadora, se comunican entre sí para intercambiar datos y así transmitir la información.

- Sensores

Los sensores son una parte fundamental de cualquier proceso automático, ya que son capaces de detectar magnitudes físicas o químicas, llamadas

variables de instrumentación, para después transformarlas en variables eléctricas y poder así ser leídas y comparadas por algún otro dispositivo.

- Interfaces de control

El objetivo de cualquier planta o sistema es que trabaje a las condiciones y bajo los parámetros óptimos y adecuados, para obtener el mayor rendimiento. Los sensores proporcionan información del estado en el que se encuentra la planta, y para llevarla al punto de operación óptimo hay que regular algún actuador (motor, pistón, válvula, etc.) que modifique el estado de la planta o sistema, Para ello se utilizan las interfaces de control para actuadores, que al modificar su estado de trabajo modifican el estado de la variable de instrumentación (la medida por el sensor) y por tanto el de la planta en general.

- Estación Maestra

Es el dispositivo (computadora) ubicado en el cuarto de control que tiene cargado el software (HMI) responsable de comunicarse con el equipo de campo (PLCs, tarjetas DAQ, drivers, etc.).

En la terminal maestra se lleva a cabo, por medio de la interfaz, la adquisición, el procesamiento y almacenamiento de datos, la presentación de la información, alarmas, generación de reportes y envío de comandos.

En resumen, la figura 3.1 nos muestra la arquitectura funcional del sistema SCADA involucrando los aspectos antes descritos:

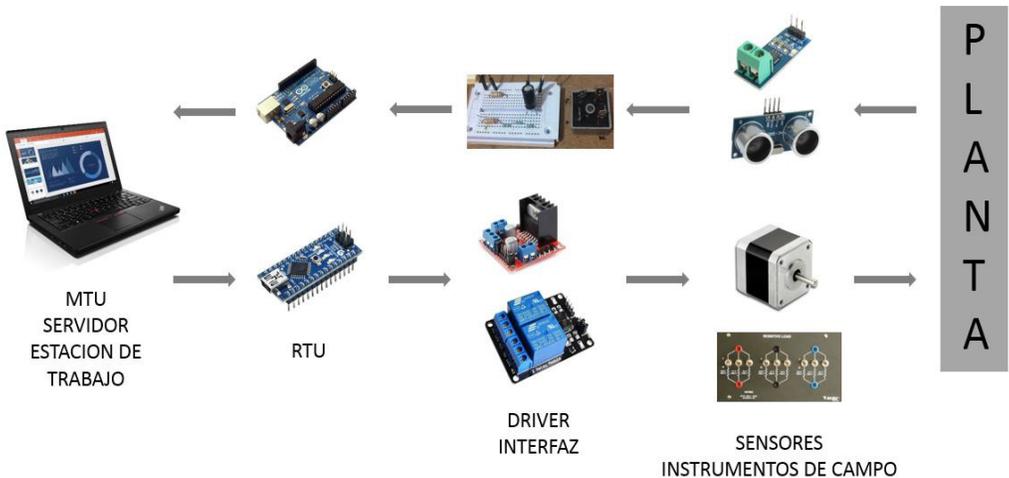


Fig. 3.1 Arquitectura del sistema SCADA aplicado a la planta.

### 3.1 Componentes de un SCADA

Los componentes necesarios para diseñar el sistema SCADA son:

- Estación maestra:

Este dispositivo recibe los datos de los instrumentos de campo, procesa la información y envía señales a las unidades terminales remotas para mantener las variables de control del proceso dentro de los parámetros establecidos.

En algunos casos, dependiendo del tipo de SCADA a implementar, la estación

maestra puede ser una PC cargada con el software de supervisión y control, y en otros casos se opta por trabajar con equipos PLC con capacidad de comunicación HMI, como algunos módulos de SIEMENS que están compuestos por el PLC y la adaptación de pantallas SIMATIC PANEL TOUCH con el software instalado para la tarea necesaria.

Para este proyecto la terminal maestra será el equipo portátil PC, una Lenovo T420 con el software realizado en LABVIEW.

- Unidades Remotas RTU:

Son los dispositivos que controlan todas las señales de entrada y salida de los instrumentos de campo, los dispositivos de campo pueden ser válvulas, motores, equipos de medición, incluidos sensores. Las RTU monitorean las condiciones de los equipos y reciben comandos de la estación maestra.

Para este trabajo se tienen dos unidades remotas encargadas una del monitoreo de señales y la otra del control de dispositivos, las cuales son las tarjetas ARDUINO UNO y ARDUINO NANO respectivamente.

- Red de telemetría:

La telemetría es un tipo de tecnología que permite la medición remota de magnitudes físicas, así como el envío de información hacia el operador del sistema



ubicado en la estación maestra. Aquí se involucran los protocolos de comunicación, que son una serie de reglas que permiten que dos o más entidades de un sistema se comuniquen entre sí para transmitir y recibir información, estas reglas definen la sintaxis, semántica y sincronización de la comunicación.

En este trabajo el tipo de comunicación será a través de SERIAL PORT, que más adelante se explicará con detalle.

### **3.2 El Hardware**

Se refiere a las partes físicas tangibles de un sistema, son los componentes eléctricos, electrónicos, electromecánicos y mecánicos, cables, gabinetes y periféricos.

Los sistemas informáticos se componen de una unidad central de procesamiento (CPU) que se encarga de procesar los datos de uno o varios periféricos de entrada, que permiten el ingreso de la información como los diferentes tipos de sensores, y uno o varios periféricos de salida, que posibilitan dar salida, como los drivers.

El hardware utilizado para el control y monitoreo del sistema utilizado en este trabajo se enlista a continuación:

- Laptop Lenovo ThinkPad T420, Intel Core i5, 2.50 GHz, 8.00 GB RAM, Sist. Op 64 bits
- Monitor SAMSUNG 18 pulgadas LCD
- Tarjeta ARDUINO UNO
- Tarjeta ARDUINO NANO
- Módulo de 4 relevadores
- Controlador L298n para motor a pasos
- Motor de pasos
- Controlador para motor-bomba
- Cámara Web
- Sensor de corriente ACS712
- Sensor de voltaje
- Sensor de RPM
- Sensor ultrasónico

De todos los dispositivos utilizados para el control y monitoreo del sistema, el hardware fundamental que será utilizado como unidades remotas (RTU), son las tarjetas ARDUINO, ya que son las unidades de procesamiento que serán vinculadas a los dispositivos de campo y a la Terminal Maestra.

Aunque el tema de los sensores se tratará más adelante, no dejan de ser parte del hardware del sistema.

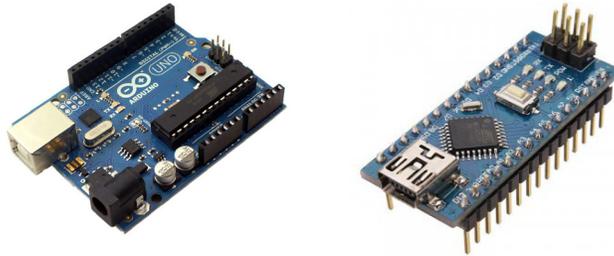


### **3.2.1 Tarjetas ARDUINO (UNO & NANO)**

Las tarjetas ARDUINO están compuestas por circuitos impresos integradas por un microcontrolador, el cual es programado mediante un entorno de desarrollo (IDE).

Las tarjetas ARDUINO tanto UNO como NANO, están basadas en el chip microcontrolador ATmega328P creado por ATMEL, estas tarjetas cuentan con pines analógicos y digitales de entrada y salida, los cuales, dependiendo del modelo, pueden ser usadas como salidas moduladas por ancho de pulso (PWM).

Se ha decidido trabajar con estos dispositivos debido a su facilidad y adaptabilidad en diferentes entornos de trabajo y por tanto en diferentes proyectos. Las tarjetas ARDUINO son dispositivos muy asequibles, relativamente baratos en comparación con otras plataformas de microcontroladores e incluso con PLC's, pero definitivamente no son lo mismo. Un PLC es un dispositivo de control programable a escala industrial, de manufactura de mayor calidad, precisión y exactitud y por tanto más costoso, pero sabiendo utilizar las tarjetas ARDUINO se puede lograr que hagan casi lo mismo a un menor costo.



*Fig. 3.2 Tarjetas ARDUINO UNO y NANO utilizadas en el proyecto.*

Otra de las razones por las que se decidió usar dos tarjetas ARDUINO es por la programación tanto en LabVIEW como el código ARDUINO que más adelante se explicará.

Empezando con la tarjeta ARDUINO UNO, esta será utilizada como receptor de datos, en la cual solo los sensores estarán conectados a la tarjeta, el objetivo de esta maniobra es tener un dispositivo para cada tarea, uno para la adquisición y otro para el control (utilizado como driver tipo RTU), a diferencia de un PLC que puede recibir datos y controlar al mismo tiempo.

Lo que hará la tarjeta ARDUINO UNO será solamente adquirir datos provenientes del estado de los instrumentos de campo del sistema para ser procesados y convertidos en señales digitales por la tarjeta ARDUINO y ser representados en la interfaz del sistema SCADA.



Algunas empresas como NATIONAL INSTRUMENTS proporcionan dispositivos para la adquisición de datos (dispositivos DAQ) como el compactDAQ, que son dispositivos bastante fieles y precisos, pero de un costo muy elevado, y uno de los objetivos de este trabajo es realizarlo al menor costo posible. En otras palabras, se adaptará esta tarjeta ARDUINO para trabajar adquirir datos de forma similar a los dispositivos DAQ de NATIONAL INSTRUMENTS.

El dispositivo ARDUINO UNO actuará como interfaz entre la PC y las señales provenientes del sistema, funcionará como un dispositivo que digitaliza las señales analógicas entrantes para que la PC pueda interpretarlas. Los componentes clave del dispositivo DAQ son el circuito de acondicionamiento de señales, el convertidor analógico-digital y un bus PC. También podrá recibir y enviar señales digitales y generar pulsos digitales como PWM para el control de actuadores.

La tarjeta ARDUINO no está diseñada para acondicionar señales, puesto que estas pueden tener ruido o ser muy peligrosas para medirse directamente, para esto, en dispositivos como el sensor de voltaje de corriente alterna, se implementa un circuito rectificador, reductor y de filtrado que acondiciona la señal para poder realizar las mediciones con la tarjeta.

La tarjeta ARDUINO tiene la capacidad de convertir señales analógicas

provenientes de los sensores en señales digitales antes de ser manipuladas por la PC. Un ADC es un dispositivo electrónico que proporciona una representación digital de una señal analógica en un instante de tiempo. En la realidad las señales analógicas varían continuamente en el tiempo y un ADC realiza un muestreo periódico de la señal, estas muestras son enviadas a la PC a través del bus donde la señal original es reconstruida desde las muestras en el software.

El dispositivo DAQ (ARDUINO UNO) se conecta a través de la ranura o puerto USB que sirve como interfaz de comunicación entre el dispositivo DAQ y la PC para pasar instrucciones y datos medidos.

Se ha descrito y explicado una parte del hardware principal conformado por las tarjetas ARDUINO UNO y NANO, la tarjeta ARDUINO UNO es la primera parte en la que se habla de la adquisición y ahora es turno de explicar la parte dos que corresponde al control, en la que se usará la tarjeta ARDUINO NANO, que realizará algo similar a la aplicación de un PLC.

A la tarjeta ARDUINO UNO se le conectarán dispositivos periféricos de entrada de señales (sensores), a la tarjeta NANO se le conectarán dispositivos periféricos de salida (actuadores) que son: Motor a pasos, encargado de la regulación de apertura del inyector de la turbina Pelton, el control de veloci-



dad del motor de la bomba para la turbina Pelton y un control por medio de relevadores para la adición de cargas al sistema de generación eléctrica.

A pesar de poder realizar este proyecto utilizando una sola tarjeta, se decidió utilizar dos para trabajos diferentes porque la programación es diferente para adquisición y control, cargando la tarjeta ARDUINO UNO con el código creado acondicionando las señales y la tarjeta ARDUINO NANO cargada con el código “LIFA Base” que permite su programación directamente desde LabVIEW para la escritura de datos en la tarjeta.

### **3.3 El Software (LabVIEW & ARDUINO IDE)**

LabVIEW es una plataforma y entorno de desarrollo para el diseño de sistemas con un lenguaje de programación gráfico. Es adecuado para la realización de sistemas hardware-software de pruebas de control y diseño, simulado o real y embebido ya que acelera su productividad. El lenguaje utilizado es “G” por graphics, llevado a cabo mediante la utilización de bloques funcionales enlazados para realizar una tarea.

Con LabVIEW la integración del hardware es muy simple y sencilla, se pueden adquirir y visualizar datos muy rápidamente desde cualquier dispositivo. La utilización del lenguaje G reduce el tiempo de programación.

La principal característica de este software es su facilidad de uso, los programas o instrumentos virtuales pueden ser realizados por programadores profesionales o por personas con pocos conocimientos en programación, la ventaja es que con LabVIEW los programas complejos hechos con lenguajes tradicionales se vuelven muy sencillos.

LabVIEW será utilizado en este proyecto para la realización de la Interfaz Humano Máquina (HMI) misma que será instalada en la PC utilizada como terminal maestra. Su propósito es mediante la interfaz tener un control total y unificado de todos los instrumentos de campo como motores, bombas, válvulas, generadores, gracias a la instrumentación virtual que permite el funcionamiento de la instalación de la central hidroeléctrica experimental en el laboratorio de Termofluidos.

El programa desarrollado que tiene por nombre “NUEVA CENTRAL”, está conformado por los siguientes partes:

- Inicialización
- Comunicación
- Driver
- DAQ
- Alarmas



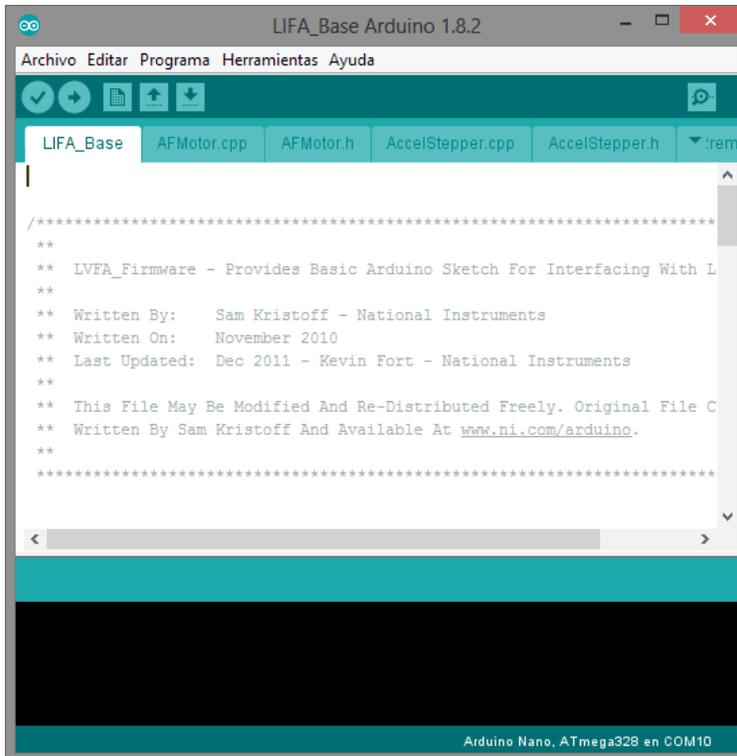
- Registro de datos
- Cámara
- Esquema animado

Cada una de estas partes principales tiene su particular programación, de estas las más importantes y críticas son la parte del Driver y la DAQ.

El resto de partes que conforman el programa serán vistos a detalle en el capítulo siguiente.

Para poder llevar a cabo el control y adquisición de los datos del sistema y visualizarlos en la terminal maestra con la HMI realizada en LabVIEW, tiene que haber una “sincronía” entre el software LabVIEW y el hardware de ARDUINO, y para eso las dos tarjetas tanto NANO como UNO tienen una programación específica para cada uno, de esta manera es posible la lectura y escritura en las tarjetas ARDUINO.

La tarjeta NANO, que es la destinada al control de dispositivos, tiene cargado un código que permite la comunicación directa con LabVIEW, el código LIFA BASE, permite que la tarjeta ARDUINO sea programada directamente desde el panel de diagrama de bloques en LabVIEW, sin tener que realizar código en ARDUINO IDE.



```
LIFA_Base Arduino 1.8.2
Archivo Editar Programa Herramientas Ayuda
LIFA_Base AFMotor.cpp AFMotor.h AccelStepper.cpp AccelStepper.h :rem

/*****
**
** LVFA_Firmware - Provides Basic Arduino Sketch For Interfacing With L
**
** Written By: Sam Kristoff - National Instruments
** Written On: November 2010
** Last Updated: Dec 2011 - Kevin Fort - National Instruments
**
** This File May Be Modified And Re-Distributed Freely. Original File C
** Written By Sam Kristoff And Available At www.ni.com/arduino.
**
*****/

Arduino Nano, ATmega328 en COM10
```

Fig. 3.3 Código “LIFA Base” para controlar la tarjeta ARDUINO NANO a través de LabVIEW.

Al momento en que la tarjeta ARDUINO NANO es cargada con este código, su programación se hace directamente por LabVIEW. Para poder establecer la comunicación, el envío y recepción de datos entre la tarjeta y LabVIEW es necesario utilizar el toolkit de ARDUINO en LabVIEW:

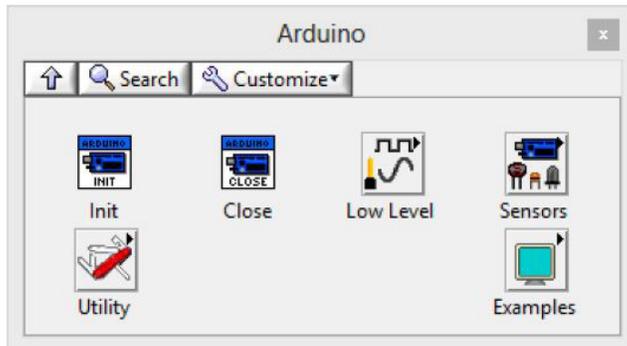


Fig. 3.4 Toolkit de ARDUINO para programación en LabVIEW.

Con estas herramientas la escritura en la tarjeta ARDUINO es posible de una manera muy sencilla y rápida.

En el caso de la tarjeta ARDUINO UNO es diferente y un poco más complejo, puesto que aquí sí hay un código en ARDUINO IDE que adapta las señales medidas por los sensores conectados a la tarjeta para posteriormente ser leídos por la interfaz en LabVIEW.

El código en ARDUINO IDE tiene por nombre "NUEVA CENTRAL", este está estructurado por funciones para cada sensor, son cuatro sensores, se tienen cuatro funciones.

Una función es un segmento de código que está separado del código principal del programa, realiza una tarea específica y regresa a la zona del

programa en donde fue llamada.

El uso de funciones en el código ARDUINO tiene las siguientes ventajas:

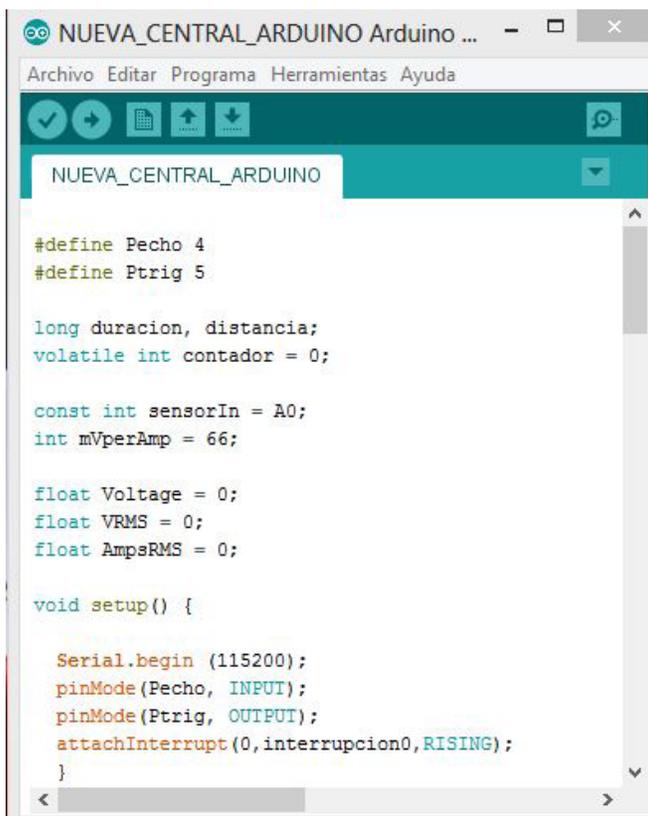
- Ayudan a organizar el programa
- Hacen el sketch (Programa de ARDUINO) más pequeño y más compacto debido a la reutilización de las secciones del código.
- Las funciones codifican una acción en un lugar, así que solo deben ser depuradas de errores una vez.
- Reducen las posibilidades de error en modificaciones en caso de ser cambiado
- Hacen más fácil la reutilización de código en otros programas.

Es necesario que con la tarjeta ARDUINO UNO utilizada para adquisición de datos, pueda “leer varios datos a la vez” provenientes de los sensores, sabemos que las computadoras son binarias (0 y 1, on y off, Low y High) y por lo tanto solo pueden estar en un estado a la vez, es decir, que solo pueden leer un dato a la vez, el problema real fue la multiplexación, esto es la combinación de dos o más canales de información en un solo medio de transmisión usando un dispositivo llamado multiplexor.

Esta multiplexación hace que a un proceso se le asigne un valor muy pequeño

de tiempo durante el cual puede ejecutar sus instrucciones antes de ceder el sitio de transmisión a otro proceso que esté esperando en cola para ser despachado por el planificador de procesos.

Es por eso que para la correcta recopilación de datos en la tarjeta ARDUINO UNO se realizó el código utilizando funciones, para que cada cierto tiempo se ejecute la instrucción de la función y posteriormente se ejecute la siguiente en cola y así sucesivamente.

The image shows a screenshot of the Arduino IDE interface. The window title is "NUEVA\_CENTRAL\_ARDUINO Arduino ...". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for checkmark, play, grid, upload, download, and settings. The file name is "NUEVA\_CENTRAL\_ARDUINO". The code editor displays the following C++ code:

```
#define Pecho 4
#define Ptrig 5

long duracion, distancia;
volatile int contador = 0;

const int sensorIn = A0;
int mVperAmp = 66;

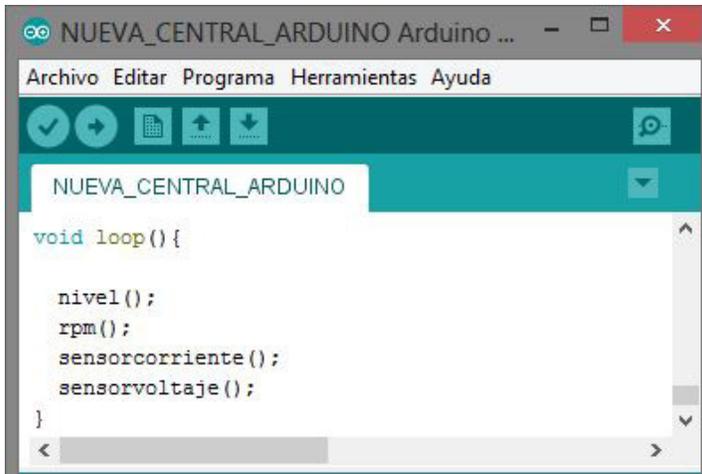
float Voltage = 0;
float VRMS = 0;
float AmpsRMS = 0;

void setup() {

  Serial.begin (115200);
  pinMode(Pecho, INPUT);
  pinMode(Ptrig, OUTPUT);
  attachInterrupt(0, interrupcion0, RISING);
}
```

Fig. 3.5 Declaración de variables utilizadas para el procesamiento de las señales.

Como se puede apreciar en la Fig. 3.5, en esta parte inicial del código se declaran todas las variables con su respectivo tipo de dato que serán utilizadas por las funciones para cada sensor, en el “void setup”, que es la función en donde se configura la tarjeta, se hacen los arreglos sobre los pines que serán utilizados por los periféricos de entrada (sensores) y en los que los datos serán introducidos para ser tratados por la función del código.

The image shows a screenshot of the Arduino IDE interface. The window title is "NUEVA\_CENTRAL\_ARDUINO Arduino ...". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with icons for checking, running, uploading, and downloading. The main editor area shows the following code:

```
void loop() {  
  
  nivel();  
  rpm();  
  sensorcorriente();  
  sensorvoltaje();  
  
}
```

Fig. 3.6 Ejecución de las funciones de los sensores en el ciclo “void loop”.

En la parte final del programa (Fig.3.6), en el “void loop”, este llama a ejecutarse la primera función, que corresponde al sensor de nivel, se leen los datos en un determinado tiempo y pasa a ejecutarse la siguiente función que corresponde al sensor de RPM y así sucesivamente.

Para que estos datos leídos por la tarjeta ARDUINO UNO puedan ser visualizados en la interfaz en LabVIEW se debe establecer una comunicación utilizando los siguientes bloques de funciones:

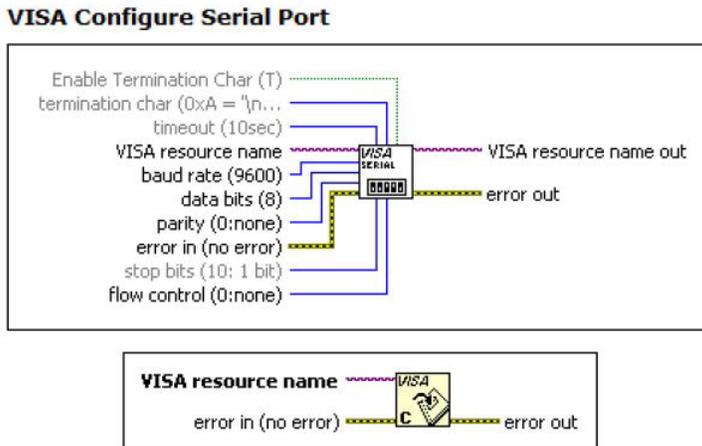


Fig. 3.7 Bloques utilizados para la comunicación serie en LabVIEW.

### 3.4 Sensores y Controladores

En la realización del proyecto SCADA para la central hidroeléctrica del laboratorio, como parte del hardware se explica el funcionamiento de los sensores y drivers que se utilizan para la adquisición de datos y el control de actuadores. Los detalles técnicos (programación, configuración e instalación) se resuelven en capítulo 5.

### Sensor de corriente ACS712

Es utilizado para medir la intensidad de la corriente eléctrica que circula por el circuito de cargas que se agregarán al generador eléctrico cuando esté en pleno funcionamiento, con el fin de poder medir, en conjunto con los datos de voltaje, la potencia eléctrica generada a diferentes cargas.

El sensor utilizado es el ACS712 de 30 amperes de efecto hall, ya que proporciona una solución económica y relativamente precisa para la medición de corriente eléctrica. Este dispositivo es fácil de implementar a cualquier sistema ya que además está fabricado específicamente para trabajar con la tarjeta ARDUINO para el procesamiento de las señales que posteriormente serán visualizadas en la interfaz gráfica (HMI).

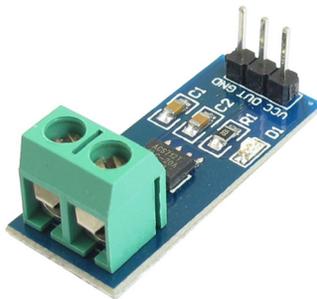
Este sensor consta de un preciso sensor de efecto Hall de bajo offset, la corriente aplicada que fluye a través de este cobre conductor genera un campo magnético que se detecta por el integrado HALL IC y es convertido en un voltaje.

La precisión del dispositivo se optimiza al aproximarse la señal magnética al transductor Hall. Un voltaje proporcional es entregado por el circuito integrado

de efecto Hall “BiCMOS chopper-stabilized” de bajo offset, en el cual es programado para tener precisión después del empaquetado.

La señal de salida del dispositivo tiene una pendiente positiva cuando una corriente creciente fluye a través del embobinado primario, en el cual la trayectoria de conducción es de los pines 1 y 2 a los pines 3 y 4, esta es la trayectoria para la detección de corriente.

La resistencia interna de esta trayectoria conductora es de  $1.2\text{ m}\Omega$ , proporcionando baja pérdida de potencia. El espesor del conductor de cobre permite el funcionamiento óptimo del dispositivo hasta cinco veces la condición de sobrecorriente. Las terminales de la trayectoria conductora están aisladas eléctricamente de los conductores del sensor, esto permite que el sensor ACS712 sea utilizado en aplicaciones que requieran aislamiento eléctrico sin el uso de opto aisladores u otras técnicas costosas de aislamiento.



*Fig. 3.8 Sensor ACS712.*

## Sensor de voltaje

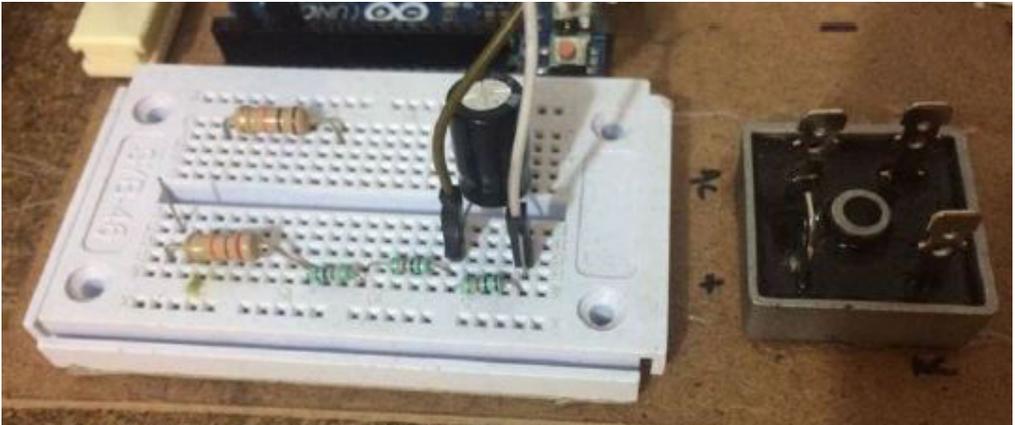
Como su nombre lo indica la función es medir el voltaje que el generador produce con el fin de poder medir la potencia eléctrica suministrada al sistema a diferentes cargas.

La selección del sensor de voltaje está en función de la capacidad de voltaje que el generador puede entregar, por tanto, se optó por construir el sensor. Puesto que la energía producida por el generador es corriente alterna, el circuito para el sensor de voltaje consta de una primera etapa en la que se rectifica la onda completa de la señal eléctrica para trabajar solo con el semiciclo positivo. La segunda etapa consiste en la reducción del voltaje a través de un divisor de tensión y filtrado de la señal para poder procesarla en la tarjeta ARDUINO.

El voltaje de corriente alterna corresponde a una señal de salida en corriente directa entre 0 y 5 V, esto es para que el microcontrolador ARDUINO pueda procesar la señal puesto que trabaja a este rango de voltaje.

Una vez que la salida del sensor entre a la tarjeta ARDUINO este realizará la lectura en el pin analógico correspondiente y convierte la señal analógica en digital por medio de su ADC que lee el voltaje de cambio como entrada analó-

gica con una resolución de 10 bits y lo convierte en un número entre 0 y 1024, correspondiente a 0 y 5 Volts.



*Fig. 3.9 Sensor de voltaje implementado.*

### Sensor de velocidad (RPM)

Este dispositivo está destinado a la medición de la velocidad angular de la turbina Pelton específicamente en revoluciones por minuto, esta medición es muy importante porque nos permite conocer el estado de la turbina, así como parámetros de funcionamiento de la planta como la eficiencia de la turbina, la eficiencia del generador y la potencia de la planta.

Se optó por implementar el sensor utilizando elementos muy sencillos y económicos como el led transmisor y fototransistor receptor, resistencias y cables para la realización del circuito.

El funcionamiento de este sensor se enfoca en la programación de la tarjeta ARDUINO, donde se usa la interrupción cero en el pin digital dos.

Se crea una variable entera de tipo volátil para que sea almacenada en la memoria RAM del microcontrolador, de esta forma se puede trabajar con interrupciones.

Con la instrucción “attachInterrupt” se habilitan las interrupciones, esto nos permite que, al ejecutar una instrucción programada, al habilitar la interrupción se salta a ejecutar otra instrucción indicada y regresa a ejecutar la instrucción anterior, es decir, cuando recibe la interrupción cero por el pin dos, se va a ejecutar la función “interrupcion0”, y después regresa al programa principal.

La instrucción “void interrupcion0” que es una función, lo único que hace es aumentar el contador cada vez que se interrumpa el infrarrojo.

El modo de disparo del interruptor es mediante RISING, es decir, durante la subida del pulso cuando cambia de cero a uno, o de cero a cinco volts.



*3.10 Sensor de velocidad (RPM).*

### Sensor ultrasónico

Este dispositivo tendrá dos usos: la medición del nivel de agua en el tanque de la instalación y la medición del caudal, estos datos son muy importantes ya que con la medición del caudal podemos calcular parámetros como la potencia de la turbina y la potencia de la planta.

El sensor utilizado es el HC-SR04, este es un módulo que incorpora un par de transductores de ultrasonido que se utilizan de manera conjunta para determinar la distancia del sensor cuando un objeto es colocado enfrente de este.

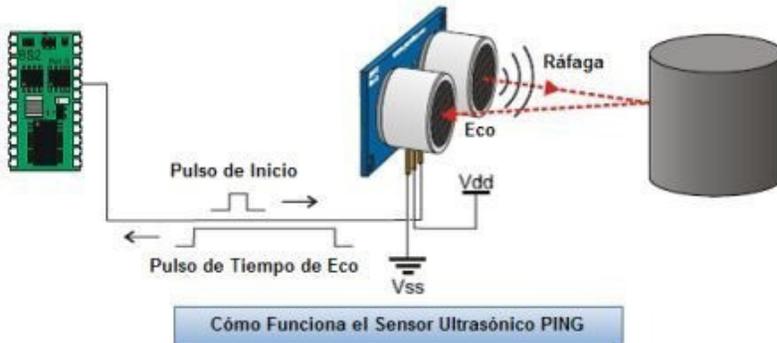


Fig. 3.11 Funcionamiento del sensor al interponerse un objeto.

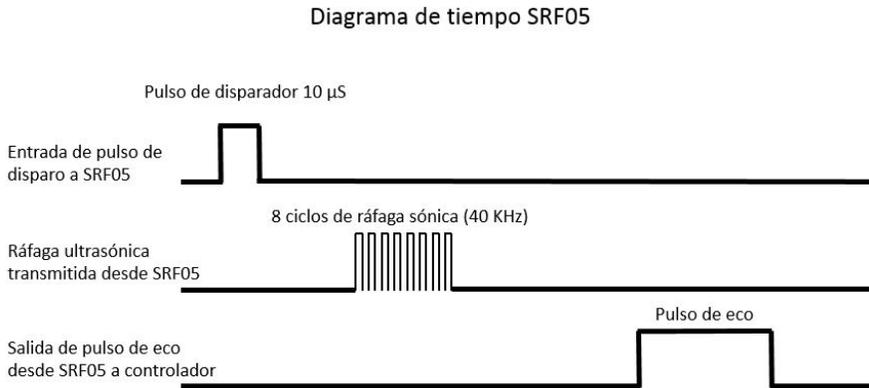
Funciona de la siguiente manera:

- 1.- Envía un pulso de al menos  $10\mu\text{s}$  por el pin trigger o disparador.
- 2.- El sensor envía ocho pulsos de 40 kHz (ultrasonido) y coloca su salida echo en High (seteo), se detecta este evento e inicia un conteo de tiempo.
- 3.- La salida de echo se mantendrá en high hasta recibir el echo reflejado por el obstáculo a lo cual el sensor pondrá el pin echo en low, es decir, termina de contar el tiempo.

La distancia es proporcional a la duración del pulso y se calcula de la siguiente forma:

$$\text{distancia en cm} = (\text{duración}/2)/29$$

Hay que tener en cuenta que la distancia que recorre la onda es dos veces la distancia del objeto al usar la fórmula anterior.



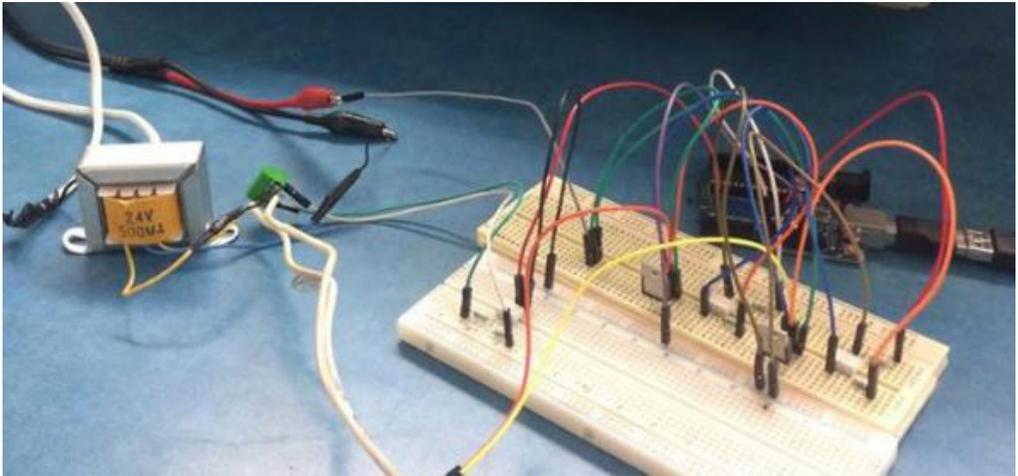
*Fig. 3.12 Funcionamiento de la señal de pulso a través del trigger y la recepción del echo*

### Driver motor-bomba

Este dispositivo servirá para controlar la velocidad del motor de corriente directa, lo que nos servirá para regular el caudal y la presión del fluido al ingresar a la tobera de inyección.

La regulación de la velocidad de giro del motor se hará a través de un control de ángulo de fase de la señal de corriente alterna durante el semiciclo positivo utilizando rectificadores controlados de silicio (SCR). Para lograr el disparo de los SCR es necesario implementar un detector de cruce por cero de la señal de CA, este detector se hará con la tarjeta ARDUINO UNO utilizando las interrupciones del pin digital 3. Cada vez que la señal cruce por cero se iniciará la

interrupción y esta activará un contador y se comparará con el ángulo de fase deseado ingresado por la interfaz HMI del sistema SCADA, al ser iguales, se dispara el SCR en el ángulo deseado, obteniendo así, una porción de la señal que corresponde a un valor de voltaje RMS rectificado.



*Fig. 3.13 Interfaz de control para motor de CD.*

### Controlador de motor a pasos

Este dispositivo será utilizado para controlar la velocidad y posición de un motor a pasos bipolar, con el fin de poder controlar la apertura y cierre del inyector de la turbina Pelton y regular el flujo de agua que por tanto regulará la velocidad de la turbina.

El driver a utilizar es el módulo L298N controlado con ARDUINO puesto que su manipulación a través de la interfaz lo hace el más adecuado.



Este módulo posee dos canales de puente H, pudiéndolos utilizar para controlar dos motores de CD o un motor a pasos, controlando el sentido de giro y velocidad.

Está conformado por un driver L298n, sus diodos de protección y un regulador de voltaje.

Posee un conector de 6 pines para ingresar las señales TTL para controlar los motores, una bornera de tres pines para la alimentación, y dos borneras de 2 pines para la salida a los motores.

EL L298 es un circuito monolítico integrado, es un controlador de alto voltaje (35 Volts CD) de puente completo dual de alta corriente (2 Amperes máximo), diseñado para aceptar los niveles de lógica TTL estándar y cargas inductivas como relés, solenoides y motores a pasos.

Proporciona dos entradas de habilitación para activar o desactivar el dispositivo independientemente de la entrada de señal, los emisores de los transistores inferiores de cada puente están conectados entre sí y los correspondientes se pueden utilizar en un terminal externo para la conexión de una resistencia de detección externa.

El motor a pasos es bipolar, es decir, cuenta con dos bobinas, la cuales se conectan a las salidas del controlador L298n, se conectan los pines de control de ARDUINO y hay que tener en cuenta que el cable negativo de la fuente de alimentación del controlador (fuente del motor 12 volts) debe unirse en un nodo al negativo de la tarjeta ARDUINO para funcionar.

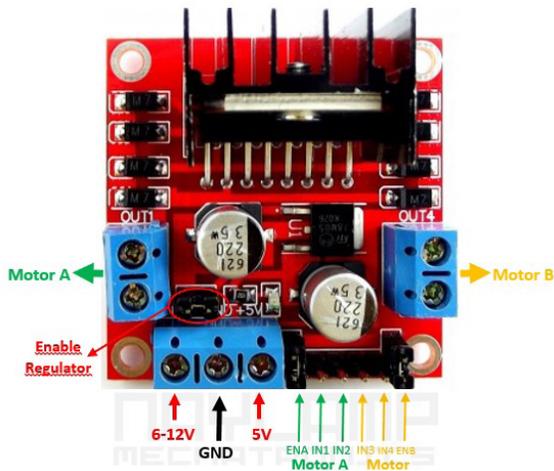
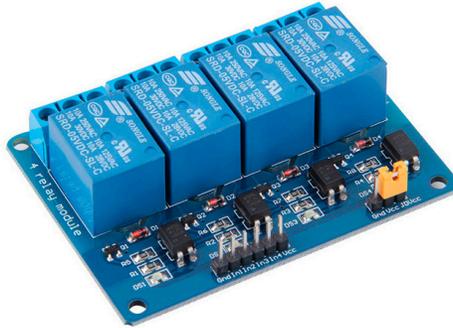


Fig. 3.14 Driver L298N para motor a pasos bipolar.

### Relevadores para cargas eléctricas

Estos relevadores serán utilizados para conectar cargas resistivas al generador eléctrico provocando caídas de voltaje al sistema para después estabilizarlo modificando la velocidad del motor de la bomba o el inyector, esto es con el fin de observar en el monitor como cambian las mediciones y el estado general de la planta.

Se utilizará el módulo de 4 canales de relé para ARDUINO con las características de 10 amperes y 250 Volts AC.



*Fig. 3.15 Módulo relevador de 4 canales para ARDUINO.*

Este relé es un interruptor que podemos activar mediante una señal eléctrica, en su versión más simple, es un pequeño electroimán que cuando lo excitamos mueve la posición de un contacto eléctrico de conectado a desconectado o viceversa.

El símbolo del relé muestra la bobina y en este caso, un accionamiento que conmuta entre dos contactos. Mediante una señal de control de poca intensidad que excite la bobina podemos conmutar circuitos de grandes voltajes o corrientes, esto es logrado mediante la señal digital proporcionada por ARDUINO que a su vez es controlada en la interfaz gráfica.

### **3.5 Interfaz Humano Máquina (HMI)**

Una interfaz humano máquina es un punto de interacción en el que el hombre entra en contacto con una máquina, es decir, no se trata ni de un humano ni de una máquina, sino algo más que permite un diálogo entre ambos, una interfaz. El ejemplo más sencillo que tenemos de una interfaz HMI es el caso de un interruptor para encender una lámpara, el interruptor es útil y significativo para el cumplimiento del objetivo, que es encender la luz, esta interfaz cumple con el requisito y está capacitada y adaptada a las necesidades, estos últimos son lo que todas las interfaces deben cumplir.

Una interfaz HMI es una interfaz de usuario asistida por computadora, esta forma parte del programa informático que se comunica con el usuario, el término interfaz se define como “todas las partes de un sistema interactivo (software y/o hardware) que proporcionan la información y el control necesarios para que el usuario lleve a cabo una tarea a través del sistema interactivo HMI”.

Las HMI se utilizan comúnmente para la realización de acciones de control sobre las máquinas y para la obtención de información del estado de algún proceso o máquina.



Fig. 3.16 Ejemplo de una HMI en un dispositivo móvil.

La HMI, al ser un software en computadora, puede ser destinada para una sola tarea correspondiente a una sola máquina, como el encendido y apagado de un motor, o puede ser una HMI multi operacional, es decir, que pueda realizar operaciones de control en distintas máquinas al mismo tiempo y a su vez obtener información de diferentes procesos y máquinas todo en un mismo software en una PC controlado por un solo usuario.

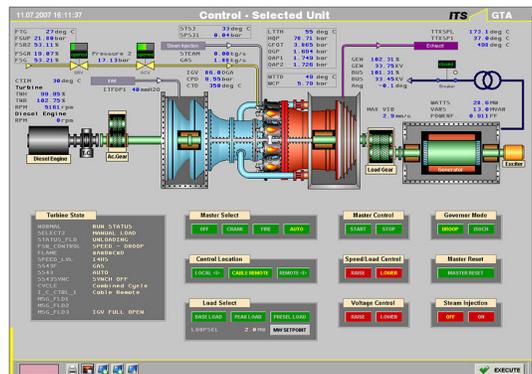


Fig. 3.17 Interfaz HMI para realizar operaciones de control en central termoeléctrica.

Las principales ventajas que se pueden tener al contar con la utilización de una HMI se basan en las funciones que la interfaz puede realizar, entre ellas:

- Puesta en marcha y apagado.
- Control de las funciones manipulables del equipo.
- Manipulación de archivos y directorios.
- Herramientas de desarrollo de aplicaciones.
- Comunicación con otros sistemas.
- Información de estado.
- Configuración de la propia interfaz y entorno.
- Intercambio de datos entre aplicaciones.
- Control de acceso.
- Sistema de ayuda interactivo.

Hay que mencionar que las interfaces de usuario se clasifican según en la forma de interactuar con el usuario:

- Interfaz de línea de comandos (CLI)
- Interfaz gráfica de usuario (GUI)
- Interfaz natural de usuario (NUI)

La interfaz que se desarrolla en este trabajo corresponde a dos de estas

clasificaciones, es del tipo GUI porque representa gráficamente los elementos de control y medición de la planta, pero a su vez podrá ser del tipo NUI, ya que mediante el uso de dispositivos portátiles la manipulación de la interfaz será de modo táctil en el panel de control.

Como se mencionó antes en la sección 3.3, la realización de la interfaz HMI para el control de la planta de generación hidroeléctrica, será realizada a través de un software llamado LABVIEW, este software nos permitirá realizar un programa ejecutable sin la necesidad de tener instalado LABVIEW, lo que facilita la instalación de la HMI en cualquier PC que cumpla con las características necesarias.

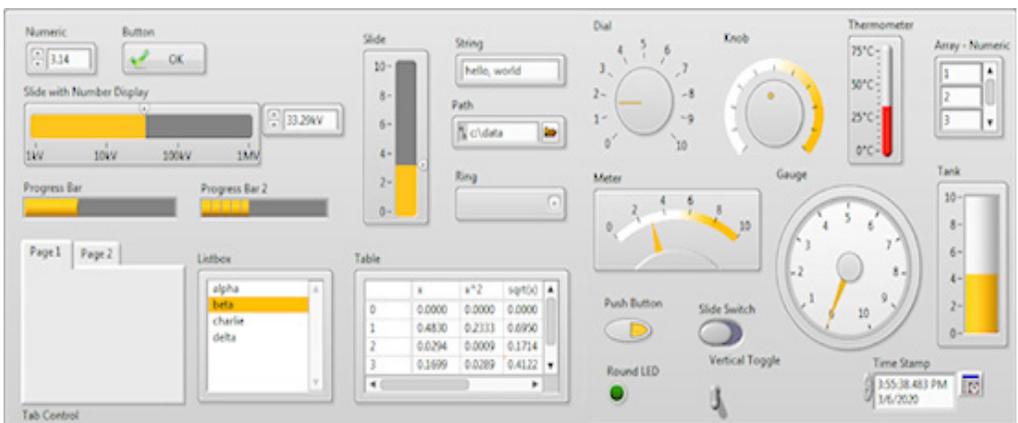


Fig. 3.18 Interfaz gráfica de instrumentación hecha en LabVIEW.

LabVIEW tiene un lenguaje de programación del tipo G de graphics, lo cual hace relativamente fácil la programación de la interfaz, puesto que se utilizan

elementos precargados para la creación de los indicadores en el panel de control y la lógica funcional de programa se realiza por medio de bloques funcionales.

La interfaz está compuesta de los siguientes elementos básicos:

- Control de acceso
- Inicialización
- Comunicación
- Etapa de control
- Etapa de adquisición
- Animación
- Alarmas
- Video vigilancia
- Registro de datos

En las figuras 3.19 y 3.20 se muestran las vistas panorámicas del panel frontal y el diagrama de bloques del sistema SCADA para central hidroeléctrica experimental (vista ampliada disponible en el anexo). Todos los detalles de programación serán detallados en el capítulo cinco.

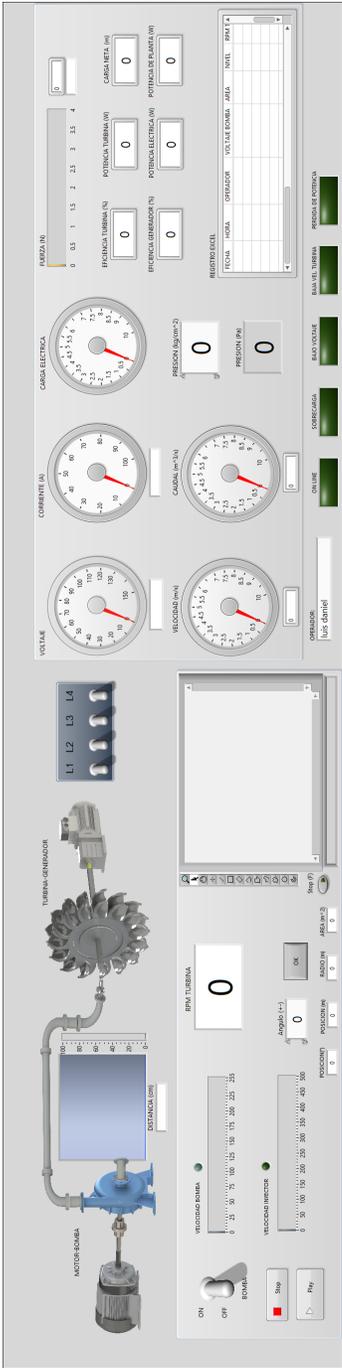


Fig. 3.19 Vista del panel frontal (HMI) del sistema SCADA.

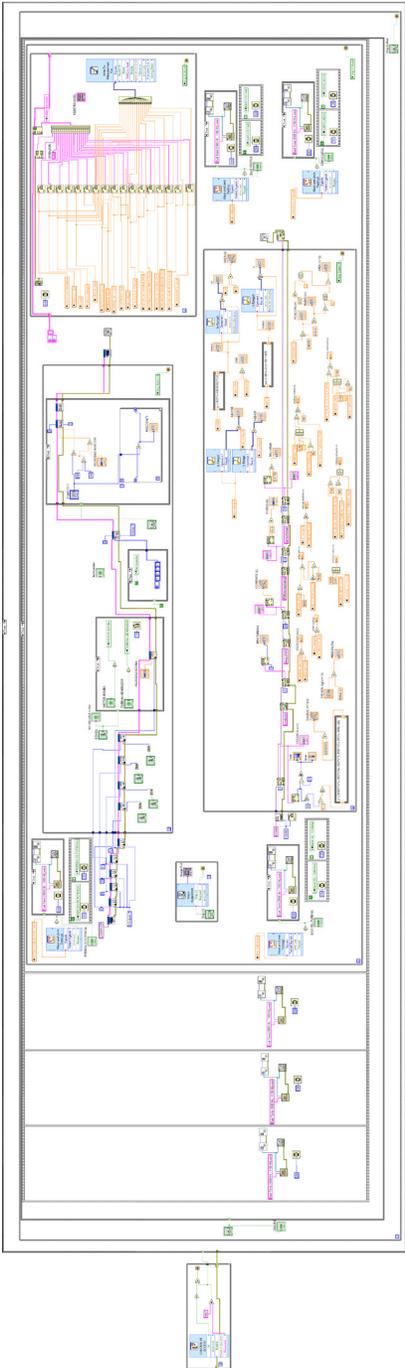


Fig. 3.20 Vista completa del diagrama de bloques de la aplicación SCADA.

### **3.6 Control de acceso**

El control de acceso consiste en la autorización y autenticación de la información almacenada en computadoras, es decir, consiste en la aprobación a la seguridad informática, por lo que el sistema adquiere la decisión de aceptar o rechazar alguna solicitud de acceso de algún usuario sobre a lo que el usuario está autorizado a acceder.

Existen muchos métodos de autenticación como tokens, contraseñas, escaneo biométrico, llaves físicas, llaves electrónicas, caminos ocultos, rompecabezas, etc.

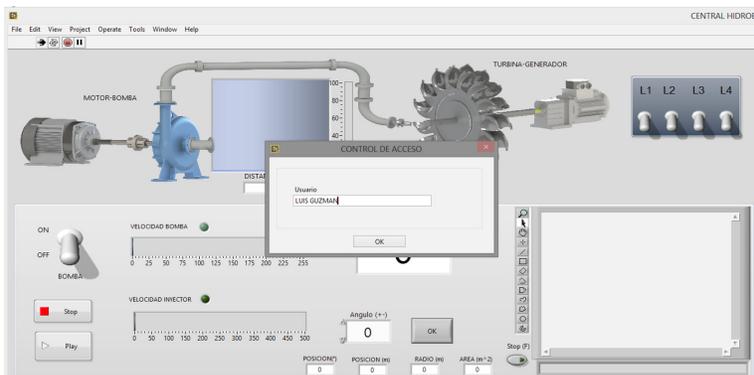
Una forma sencilla de implementar un control de acceso es programando una solicitud de contraseña que solo los usuarios autorizadas conocen.

El objetivo de implementar un control de acceso al sistema SCADA es proteger el uso y manipulación del software, así como de la planta de generación hidroeléctrica de usuarios quienes no tengan la capacidad, conocimientos e intenciones adecuadas para evitar malas operaciones, desconfiguraciones o fallos en la planta.



*Fig. 3.21 Protección y seguridad informática puede ser implementada con controles de acceso.*

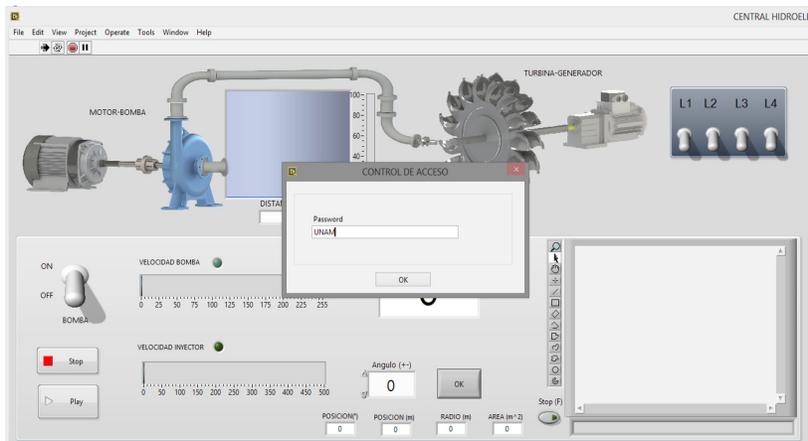
Como se mencionó antes, en los elementos básicos de los que está compuesta la interfaz del sistema SCADA, el software comienza el control de acceso pidiendo al usuario que ingrese el nombre con el cual el sistema lo registrará, esto es útil en el registro de datos, las lecturas o cambios en el sistema que se hayan registrado responden al usuario.



*Fig. 3.22 Ventana inicial de ingreso de usuario.*

Después de ingresar el usuario, el software requiere la contraseña de acceso que solo el programador o el director de la planta conocen. Esta contraseña puede ser cambiada en caso de ser necesario.

Si la contraseña no es la correcta el sistema no pasará del requerimiento de esta (sin importar el número de intentos) hasta que el usuario ingrese la contraseña correcta que le permitirá el acceso a la interfaz.



*Fig. 3.23 Ingreso de contraseña para iniciar la aplicación.*

La programación del control de acceso es bastante sencilla, comienza con una estructura “flat sequence” de dos frames o pasos a realizar. En el primer frame, marco o paso, se utiliza la herramienta “prompt user”, la cual nos permite desplegar un cuadro de texto con una casilla para procesar información, al ingresar el usuario se sigue al marco o paso siguiente, en el cual dentro de

un ciclo “while loop” tenemos la herramienta “prompt user” en el que se ingresa la contraseña, esta es comparada con la contraseña designada al programa dentro de una constante de tipo string, si la contraseña ingresada es igual a la designada el ciclo while loop termina al igual que ciclo flat sequence y se sigue a la inicialización del programa, de lo contrario seguirá desplegando la ventana de ingresar contraseña hasta tener la correcta.

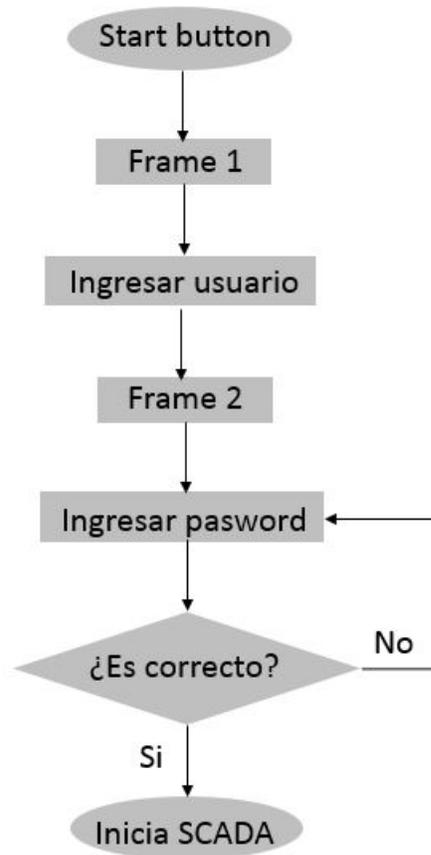


Fig. 3.24 secuencia de funcionamiento del control de acceso.

En la figura 3.24 se puede observar que en el bloque “ingresar el usuario” no se evalúa su autenticidad puesto que solo es para registrar al usuario y visualizarlo en el panel frontal de la interfaz.

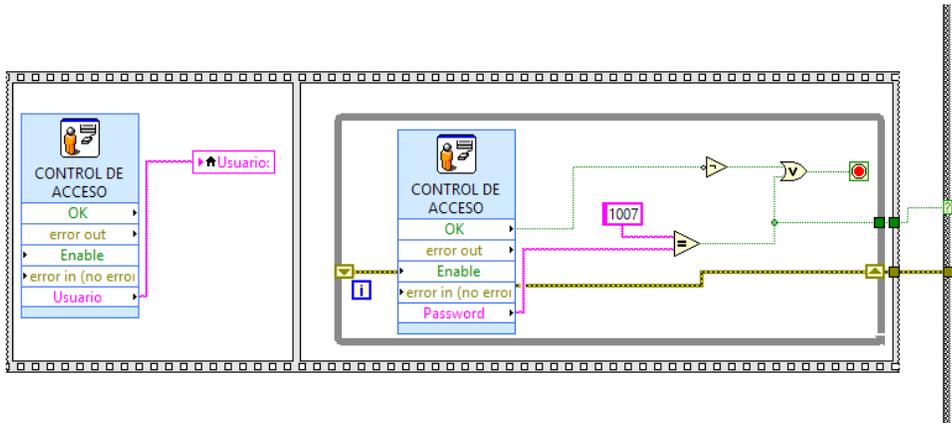


Fig. 3.25 Programación del control de acceso.

### 3.7 Protocolo de comunicación

Un protocolo de comunicaciones es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre sí para transmitir información por medio de cualquier tipo de variación de alguna magnitud física.

Los protocolos de comunicación se tratan de reglas que definen sintaxis, semántica y sincronización de la comunicación, así como los posibles métodos para la recuperación de errores.

El protocolo de comunicación que se utilizará en este trabajo para realizar el proceso de comunicación entre los dispositivos de control y adquisición con la interfaz HMI es SERIAL PORT.

Serial port es el nombre genérico con el que se denomina a algunas interfaces físicas o virtuales que permiten la comunicación entre computadoras.

Un puerto serie envía la información mediante una secuencia de bits, para ello se necesitan al menos dos conectores para realizar el envío y transmisión de datos, siendo RX (recepción) y TX (transmisión).

El puerto serial envía y recibe bytes de información un bit a la vez, aunque este protocolo es más lento que la comunicación en paralelo, la comunicación serial port permite alcanzar mayores distancias, la especificación IEEE 488 (GPIB) para la comunicación en paralelo determina que el largo del cable para el equipo no puede ser mayor a 20 metros, mientras que utilizando serial port el largo del cable puede llegar a los 1200 metros aproximadamente<sup>1</sup>.

---

1 "Comunicación Serial: Conceptos Generales - National Instruments", Digital.ni.com, 2018. [Online]. Available: <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E-0F005888D1>. [Accessed: 21- Sep- 2017].

Para realizar la comunicación serie se utilizan tres líneas:

- Tierra o referencia
- Transmisión
- Recepción

Debido a que la transmisión es asíncrona, es posible enviar datos por una línea mientras se reciben datos por otra (Full-Duplex)<sup>1</sup>.

Para que dos puertos se puedan comunicar es necesario que las siguientes características sean iguales.

- Baud rate

Es la velocidad de transmisión e indica el número de bits por segundo que se transfieren y se mide en baudios, es decir 9600 baudios representan 9600 bits por segundo. Baudio es el número de variaciones de la señal por unidad de tiempo.

El baud rate utilizado en la programación del código en ARDUINO es de

---

<sup>1</sup> "RS-232, RS-422, RS-485 Serial Communication General Concepts - National Instruments", Ni.com, 2018. [Online]. Available: <http://www.ni.com/white-paper/11390/en/>. [Accessed: 21- Sep- 2017].

115200 bits por segundo.

- Bits de datos

Se refiere a la cantidad de bits en la transmisión. Cuando la computadora envía un paquete de información, el tamaño de los datos no necesariamente tiene que ser de 8 bits. El número de bits que se envía depende del tipo de información que se transfiere. Un paquete se refiere a una transferencia de un byte (8 bits), incluyendo los bits de inicio/parada, bits de datos y paridad. Un tipo de dato es un atributo de los datos que indica a la PC sobre la clase de datos que se van a manejar, esto incluye imponer restricciones en los datos, como qué valores se pueden tomar y qué operaciones se pueden realizar.

Tipo de dato	Tamaño en memoria (bits)
char	8
byte	8
short	16
int	32
long	64
float	32
double	64
boolean	8

*Fig. 3.26 tipo de datos utilizados en la programación de ARDUINO.*

- Bits de parada

Usado para indicar el fin de la comunicación de un solo paquete. Los valores típicos son 1, 1.5 o 2 bits. Debido a la manera como se transfiere la información a través de las líneas de comunicación y que cada dispositivo tiene su propio reloj, es posible que los dos dispositivos no estén sincronizados. Por lo tanto, los bits de parada no sólo indican el fin de la transmisión sino además dan un margen de tolerancia para esa diferencia de los relojes. Mientras más bits de parada se usen, mayor será la tolerancia a la sincronía de los relojes, sin embargo, la transmisión será más lenta.

- Paridad

Es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada. La opción de no usar paridad alguna también está disponible. Para paridad par e impar, el puerto serial fijará el bit de paridad (el último bit después de los bits de datos) a un valor para asegurarse que la transmisión tenga un número par o impar de bits en estado alto lógico. Por ejemplo, si la información a transmitir es 011 y la paridad es par, el bit de paridad sería 0 para mantener el número de bits en estado alto lógico como par. Si la paridad seleccionada fuera impar, entonces el bit de paridad sería 1, para tener 3 bits en estado alto lógico. La paridad



marcada y espaciada en realidad no verifican el estado de los bits de datos; simplemente fija el bit de paridad en estado lógico alto para la marcada, y en estado lógico bajo para la espaciada. Esto permite al dispositivo receptor conocer de antemano el estado de un bit, lo que serviría para determinar si hay ruido que esté afectando de manera negativa la transmisión de los datos, o si los relojes de los dispositivos no están sincronizados.

### Comunicación serial asíncrona

Comienza con el estado de marca o reposo que se encuentra en 1, cuando va a transmitir un dato pone su salida a cero para que la computadora detecte el cambio de estado y se queda en cero el tiempo de duración de un bit, esto es conocido como bit de arranque, luego se transmiten los bits de datos comenzando desde el menos significativo, a esto se le llama carácter, después, opcionalmente sigue un bit de paridad y después se transmite el bit de parada que dura entre 1 o 2 bits, si la computadora no tiene otro dato que transmitir mantiene la salida en 1 de lo contrario vuelve a generar el bit de arranque para una nueva transmisión, a todo esto descrito se le conoce como trama de la comunicación serial asíncrona.

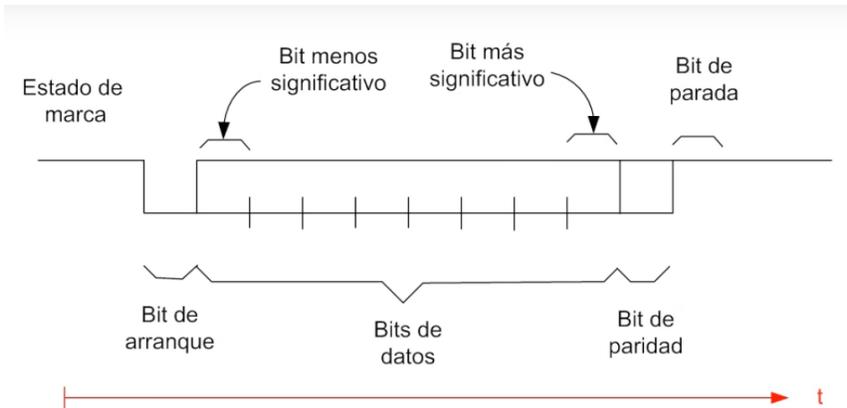


Fig. 3.27 Protocolo de comunicación serie.

### Conector puerto serie

Un puerto serie es una interfaz de comunicaciones de datos digitales, utilizado por computadoras y periféricos, en donde la información es transmitida bit a bit, enviando un solo bit a la vez.

Uno de los defectos de los puertos serie iniciales era su lentitud en comparación con los puertos paralelos -hablamos de 19.2 Kbits por segundo-. Sin embargo, con el paso del tiempo, han aparecido multitud de puertos serie de alta velocidad que los hacen muy interesantes ya que presentan las ventajas del menor cableado y solucionan el problema de la merma de velocidad usando un mayor apantallamiento, y más barato, usando la técnica del par trenzado. Por ello, el puerto RS-232, e incluso multitud de puertos paralelos, se están sustituyendo reemplazándose por los nuevos puertos serie como el USB, el

FireWire o el Serial ATA.

La transmisión de datos entre el microcontrolador de ARDUINO y la PC es como ya se dijo, por medio del protocolo SERIAL PORT, utilizando como interfaz física conectores tipo USB.

El Bus Serial Universal (BUS) (en inglés: Universal Serial Bus), más conocido por la sigla USB, es un puerto que permite conectar periféricos a una computadora para el intercambio de datos, desarrollo de operaciones y en algunos casos la carga de baterías de dispositivos.

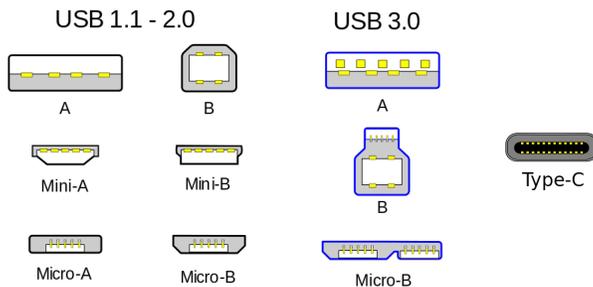


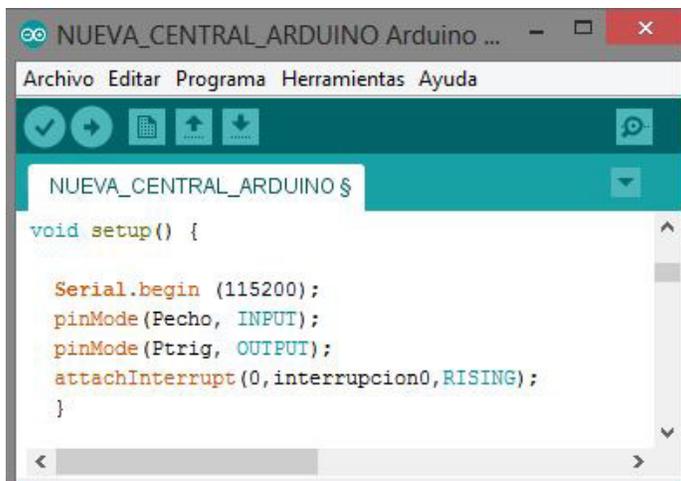
Fig. 3.28 Tipos de conectores para comunicación serie USB.

### Comunicación serial port ARDUINO-LabVIEW

La comunicación serie en ARDUINO se lleva a cabo en los pines RX Y TX (0 y 1 respectivamente), este utiliza niveles lógicos TTL (5V o 3.3 V), estos pines no deben usarse directamente como salidas o entradas.

La comunicación serial port se utiliza entre la tarjeta ARDUINO y algún dispositivo PC, estos dispositivos cuentan con puertos SERIAL PORT o UART. Se comunica con los pines digitales RX Y TX así como con la PC por medio del conector USB.

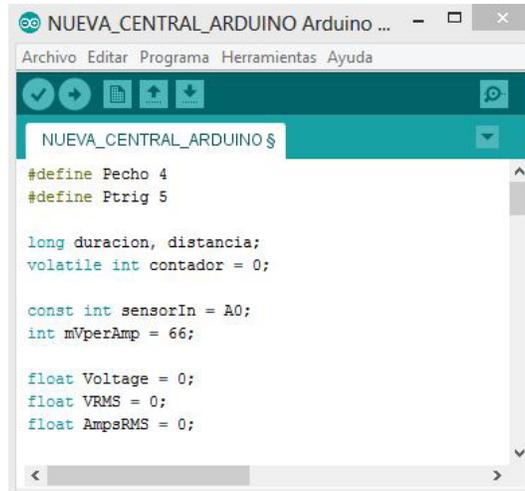
En el código en ARDUINO para el sistema SCADA, el protocolo de comunicación y el baud rate se define en el void setup:

The image shows a screenshot of the Arduino IDE interface. The window title is "NUEVA\_CENTRAL\_ARDUINO Arduino ...". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with icons for checkmark, play, upload, download, and a gear. The main editor area shows the following code:

```
NUEVA_CENTRAL_ARDUINO $  
  
void setup() {  
  
    Serial.begin (115200);  
    pinMode (Pecho, INPUT);  
    pinMode (Ptrig, OUTPUT);  
    attachInterrupt (0, interrupcion0, RISING);  
}
```

Fig. 3.29 Configuración de la velocidad de transmisión de datos.

El tipo de datos utilizado para las variables medidas utilizadas en la programación para la adquisición de datos se puede apreciar en la imagen de la figura 3.30.



```

NUEVA_CENTRAL_ARDUINO $
#define Pecho 4
#define Ptrig 5

long duracion, distancia;
volatile int contador = 0;

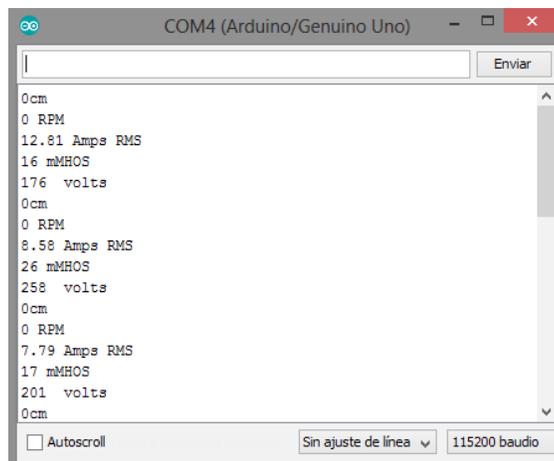
const int sensorIn = A0;
int mVperAmp = 66;

float Voltage = 0;
float VRMS = 0;
float AmpsRMS = 0;

```

Fig. 3.30 Declaración de variables con el tipo de dato.

La tarjeta ARDUINO UNO, que es la destinada para la adquisición de datos, obtiene los valores de las medidas de los sensores, estas señales son procesadas para poder mostrarse en el monitor serie como se ven a continuación:



```

COM4 (Arduino/Genuino Uno)
Enviar
0cm
0 RPM
12.81 Amps RMS
16 mMHOS
176 volts
0cm
0 RPM
8.58 Amps RMS
26 mMHOS
258 volts
0cm
0 RPM
7.79 Amps RMS
17 mMHOS
201 volts
0cm
Autoscroll Sin ajuste de línea 115200 baudio

```

Fig. 3.31 Información de los datos adquiridos por ARDUINO UNO en el monitor serie.

Para que LabVIEW pueda mostrar los datos de cada medición en su respectivo indicador diseñado en la interfaz, debe haber una comunicación SERIAL PORT entre la tarjeta ARDUINO y la PC que a su vez va dirigido al software SCADA, ya se explicó cómo funciona el envío de datos vía serial port desde la tarjeta ARDUINO hacia la PC, a continuación, se explicará la recepción de datos desde el software en LabVIEW.

El software en LabVIEW recibe información de la tarjeta ARDUINO UNO y envía información a la tarjeta ARDUINO NANO, por tanto, la programación es diferente y las herramientas de comunicación SERIAL PORT son diferentes.

Para la adquisición de datos se utilizó la herramienta VISA (NI serial communication), esta herramienta es un estándar para configurar, programar y solucionar problemas de sistemas de instrumentación que comprenden interfaces GPIB, VXI, PXI, Serial, Ethernet y / o USB. VISA proporciona la interfaz de programación entre el hardware y entornos de desarrollo como LabVIEW, con esta herramienta se logra la comunicación serial port que en este caso solo es para recibir información.

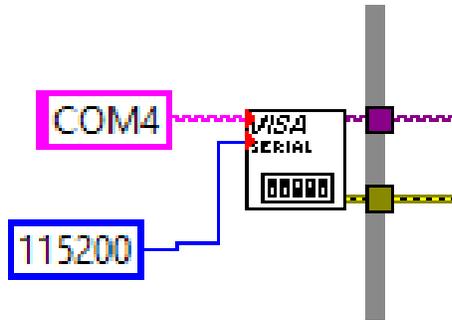


Fig. 3.32 Configuración del puerto serie y baud rate para comunicación serie en LabVIEW.

El proceso comienza indicando el puerto al que está conectada la tarjeta de adquisición de datos, que es por donde el software recibirá la información, y se indica el baud rate al que trabaja la tarjeta ARDUINO.

Se inicia un ciclo while loop y dentro se programan los sensores, su procesamiento de datos, cálculos y visualización en el panel frontal.

Como ejemplo tomaremos el caso del sensor de nivel, ARDUINO envía la información como:

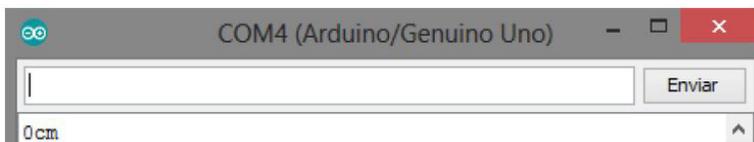


Fig. 3.33 LabVIEW interpreta la información proveniente de ARDUINO como dato tipo string y lo procesa para mostrarlo en su respectivo indicador.

En ARDUINO este dato es del tipo “string” puesto que hay un número cero y la palabra “cm”, para leer la información completa se utiliza la herramienta “VISA READ” esta nos permite leer la información proveniente de la tarjeta, se indica el número de bits de lectura, en este caso 8 que son los bits de datos tipo “string”, y se utiliza un indicador del tipo string localizado en el panel frontal que indica la información completa y en el diagrama de bloques. Para interpretar solo el valor número del estado del sensor se utiliza la herramienta “Decimal String to Number Function” esta convierte los caracteres numéricos en la cadena, empezando por el desplazamiento a un número entero decimal y lo devuelve en número para poder así utilizar indicadores numéricos, se hace el procesamiento de la señal con las operaciones necesarias y se colocan los indicadores.

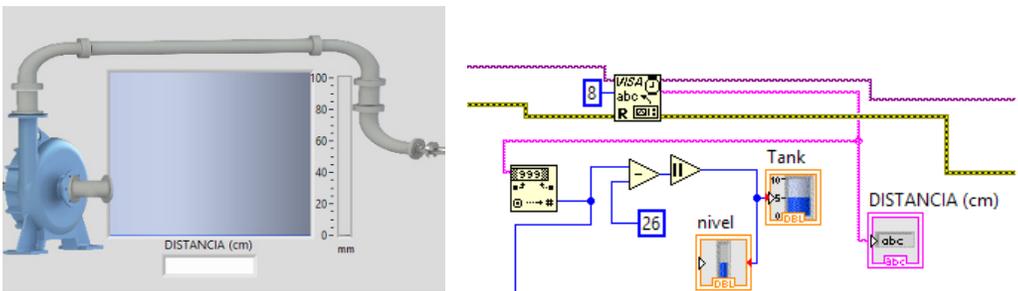


Fig. 3.34 Programación en LabVIEW para la lectura de datos provenientes de ARDUINO.

Para finalizar la comunicación de la recepción de la información se utiliza la herramienta “VISA Close Function”, esta cierra una sesión de dispositivo o un objeto de evento especificado. A continuación, se muestra la programación para la recepción de información para un sensor de nivel con lo mencionado:

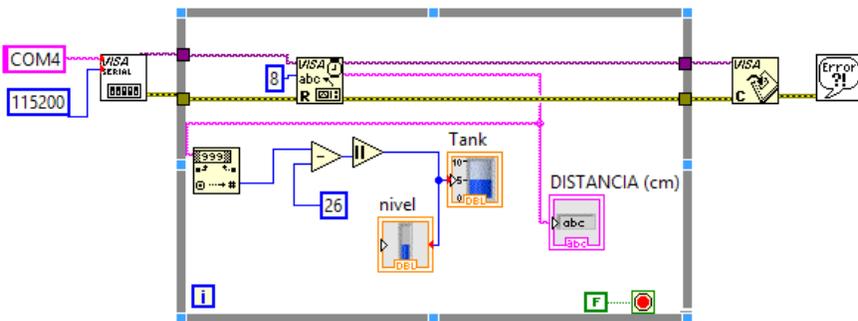


Fig. 3.35 Inicio y cierre de comunicación serie de la tarjeta ARDUINO UNO.

El proceso para recibir la información proveniente de la tarjeta ARDUINO UNO es igual para todos los sensores. En el caso de la tarjeta ARDUINO NANO la comunicación SERIAL PORT se lleva acabo de forma diferente que en la tarjeta uno, esto es porque se hace uso de la herramienta “LIFA BASE”, este es un código para ARDUINO, el cual, al ser cargado a la tarjeta, permite hacer uso de ella desde el entorno de LabVIEW usando las herramientas que el toolkit para LabVIEW de LIFA BASE proporciona.

Una vez cargado este código la programación de la tarjeta ARDUINO NANO se hace a través de LabVIEW. Ahora se pretende enviar información desde LabVIEW hacia la tarjeta ARDUINO NANO puesto que esta es destinada para el control de actuadores.

```

LIFA_Base
AFMotor.cpp
AFMotor.h
AccelStepper.cpp
AccelStepper.h
rem

/*****
**
** LVFA_Firmware - Provides Basic Arduino Sketch For Interfacing With L
**
** Written By:    Sam Kristoff - National Instruments
** Written On:   November 2010
** Last Updated: Dec 2011 - Kevin Fort - National Instruments
**
** This File May Be Modified And Re-Distributed Freely. Original File C
** Written By Sam Kristoff And Available At www.ni.com/arduino.
**
*****/

Arduino Nano, ATmega328 en COM10

```

Fig. 3.36 Código “LIFA Base” cargado en la tarjeta ARDUINO NANO.

La comunicación se inicia usando la herramienta “INIT”, esta inicializa la conexión con la tarjeta y comienza a correr el código LIFA BASE cargado en la tarjeta, se debe indicar el puerto por el que se enviará la información a la tarjeta, después se inserta la herramienta SET DIGITAL PIN MODE, esto nos permite configurar el pin de la tarjeta que se va a utilizar ya sea como salida

o entrada (INPUT, OUTPUT) así como el número de pin, después se realiza la programación de lo que se va a controlar dentro de un ciclo while loop, en este caso el switcheo de un relevador utilizando un switch virtual, para finalizar la comunicación serie se utiliza la herramienta “CLOSE”, esta termina la conexión activa con la tarjeta ARDUINO.

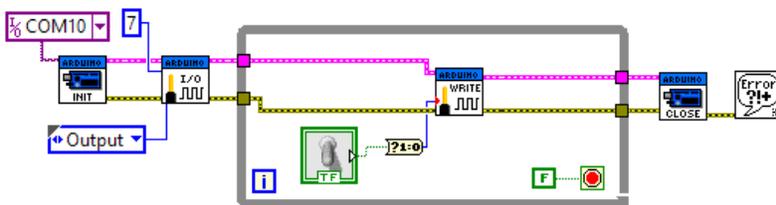


Fig. 3.37 Inicio y cierre de comunicación serie usando “LIFA Base” con ARDUINO NANO.

### 3.8 Alarmas

Son elementos de seguridad pasiva, es decir, que no evitan una situación anormal, pero si son capaces de advertir alguna de ellas.

Los avisos de alarma son señales que informan a una comunidad determinada que deben llevarse a cabo una serie de instrucciones y/o procedimientos específicos para poder conllevar la emergencia o anomalía de la mejor manera debido a la presencia inminente o real de alguna anomalía.

En el sistema SCADA para central hidroeléctrica se cuenta con las siguientes alarmas:

- Alarma de arranque de planta
- Alarma de carga eléctrica
- Alarma de voltaje
- Alarma de velocidad en la turbina
- Alarma de potencia de planta

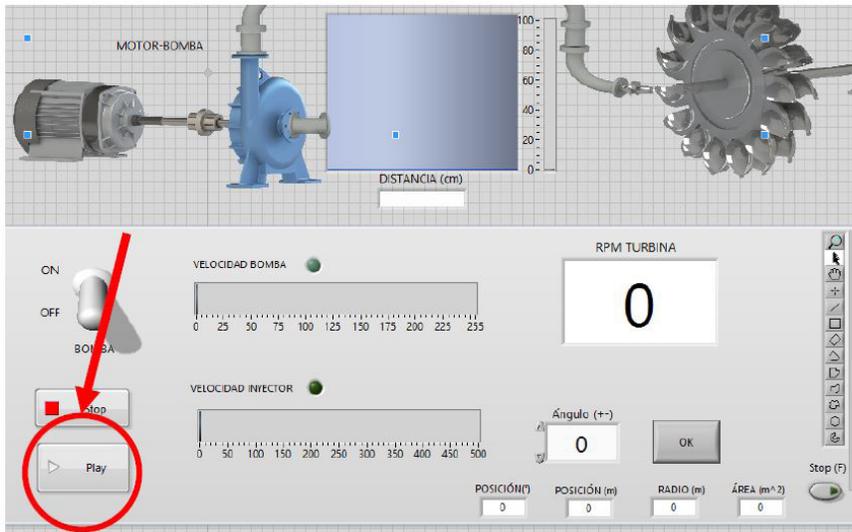


Fig. 3.38 Ubicación de los indicadores de alarmas en el panel frontal.

Alarma de arranque de planta:

Esta alarma nos indica que el sistema está arrancando e iniciando, tanto el software como el funcionamiento de todas las partes del sistema, y como este contiene elementos rotatorios y eléctricos, el personal cerca de la instalación debe mantener precaución en todo momento hasta el paro del sistema.

Después de ingresar el usuario y la contraseña, el software está inactivo, hasta oprimir el botón “PLAY” la alarma de inicio comienza y arranca la planta.



3.39 Localización del botón “play” que da inicio al programa.

La programación de la alarma de arranque inicia con el botón PLAY y un case structure, si el botón es presionado (estado true), se llevará a cabo lo programado dentro del case structure, de lo contrario no hará nada. Después de presionar el botón, se inicia una estructura flat sequence, en la que, en el primer frame se emite un sonido durante 350 mili segundos, al terminar este frame se sigue con el segundo y el tercero que tienen programada la misma tarea, después de esto el programa SCADA está en modo “on line”.

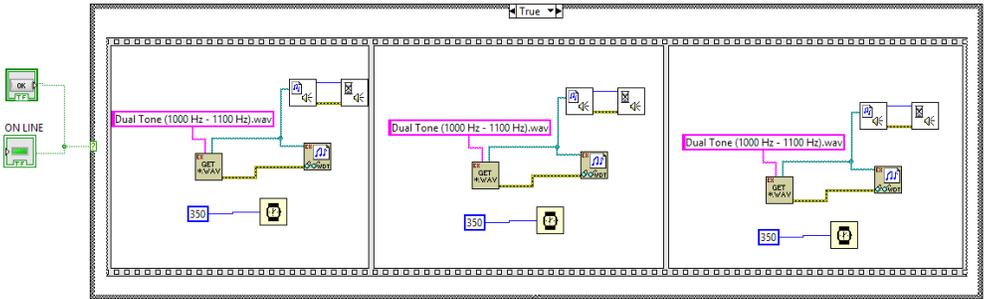


Fig. 3.40 Programación de la alarma de arranque de planta.

### Alarmas de parámetros de planta

Le llamaremos alarmas de parámetros de planta al resto de las alarmas antes mencionadas para englobar la explicación de su funcionamiento ya que es bastante similar.

Las alarmas funcionan cuando los valores de las señales sobrepasan sus límites superior e inferior, esto es para mantener el funcionamiento de la planta en niveles de tolerancia restringidos por estos límites.

La programación de estos límites se logra mediante el uso de la herramienta “Mask and limit testing”, a esta función entra la señal de la magnitud que queremos controlar y mantener dentro de ciertos límites, como en el caso del voltaje, se requiere un voltaje de entre 120 y 118 volts, un voltaje mayor o inferior producirá malfuncionamiento de la planta.

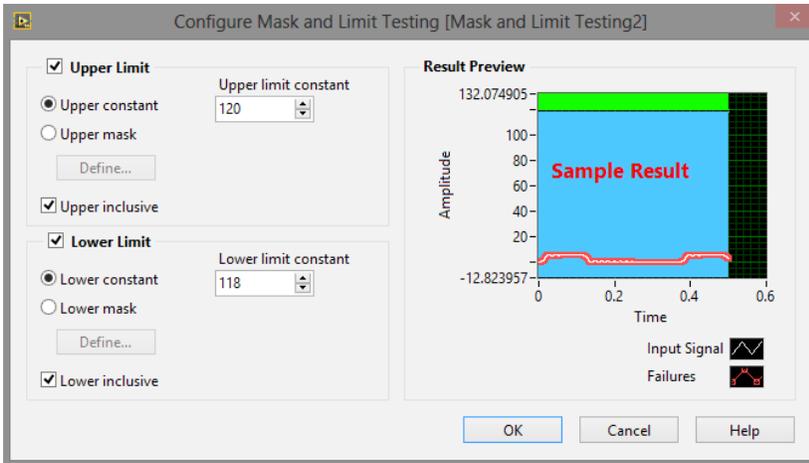


Fig. 3.41 Configuración de los límites que activan las alarmas.

Con esta herramienta, si la señal de la magnitud no se encuentra dentro de los límites establecidos comenzará a sonar un timbre intermitente de 500 milisegundos y un led comenzará a parpadear, indicando de que alarma se trata para proceder a arreglar el problema emergente, de lo contrario no habrá alarma.

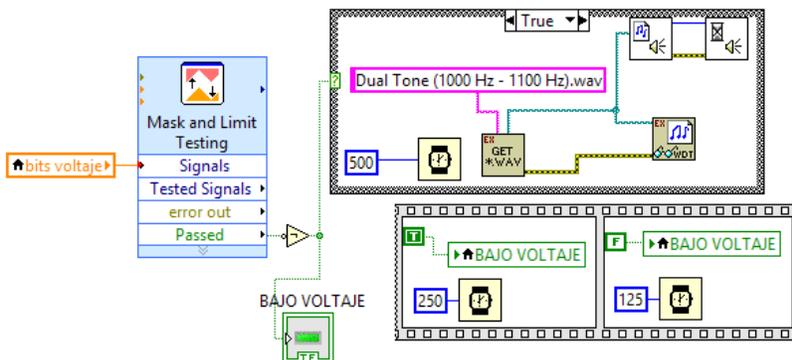


Fig. 3.42 Patrón de programación para todas las alarmas utilizadas en el programa.

ALARMA	LÍMITE SUPERIOR	LÍMITE INFERIOR
RPM Turbina (rpm)	2410	2350
Potencia de planta (watts)	8	6
Bajo voltaje (volts)	60	50
Sobrecarga (ohms)	220	190

*Fig. 3.43 límites establecidos de las alarmas*

### 3.9 Registro de datos

Se refiere al almacenamiento para dejar constancia de algún tipo de documento, en este caso se requiere guardar un historial de todas las mediciones de todas las magnitudes que conforman la planta de la central hidroeléctrica de laboratorio, esto permite estudiar y dictar conclusiones sobre el funcionamiento de ésta.

El registro de datos se lleva a cabo de dos formas, la primera en tiempo real en la interfaz HMI, y la segunda crea un reporte en formato Excel de todas las mediciones desde el inicio hasta el final de la operación del sistema SCADA.



elementos a una matriz de dimensión  $n$ , por último, las señales enlazadas se conectan a un bloque función llamado “Insert into array Function”, esto permite insertar elementos en la matriz de dimensión  $n$  que al conectar la matriz de señales, la función “insert into array function” cambia automáticamente de tamaño para mostrar las entradas para cada dimensión de la matriz (señales).

Para la creación del reporte en Excel, estas mismas variables locales de las señales de las magnitudes entran a un bloque de función llamado “NAME DBL SIGNAL”, el cual permite nombrar la columna de cada señal para reconocimiento en el reporte, esta función convierte la señal numérica en datos dinámicos para poder ser ingresados junto con el resto de las señales en una función llamada “Write to measurement file”, esta función es la que crea el reporte en Excel.

Al configurar la función se escribe la dirección del documento en el que se registrarán los datos, se escoge el formato (Excel) y se agregan configuraciones particulares.

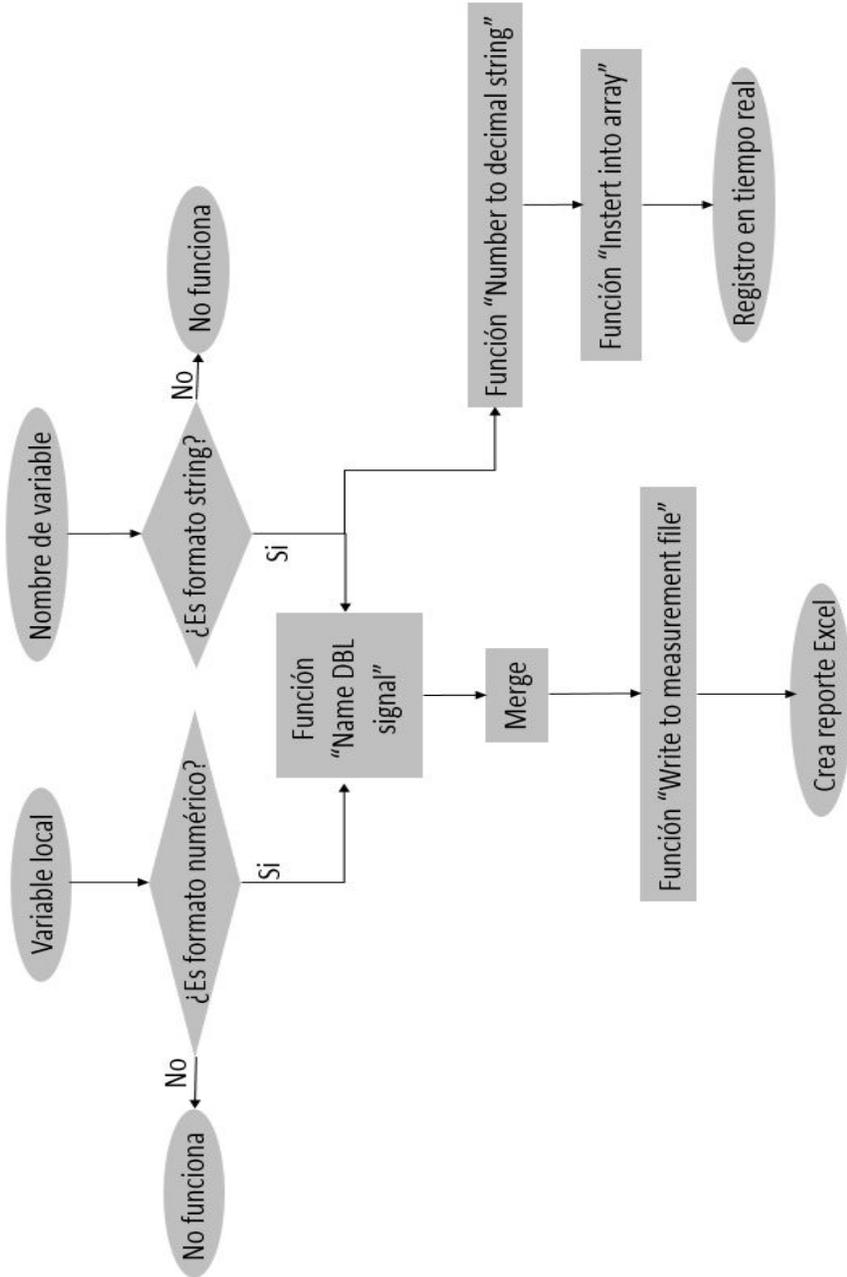


Fig. 3.45 Diagrama de flujo de creación de reporte Excel y base de datos en tiempo real.

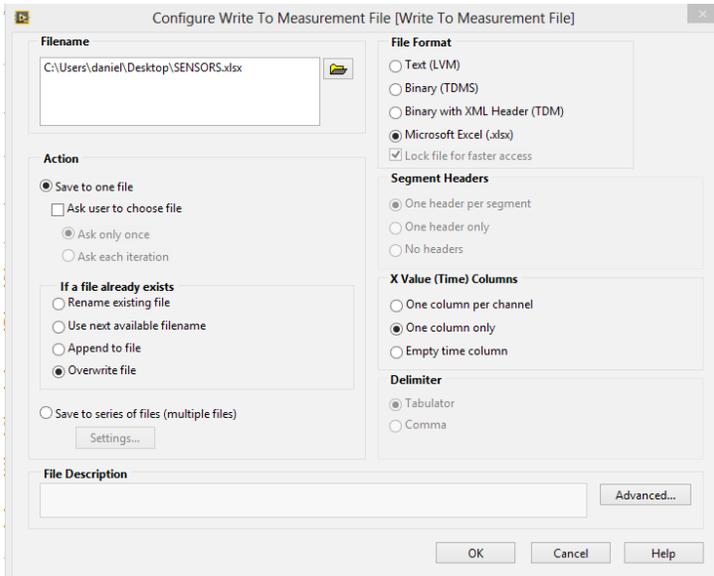


Fig. 3.46 Configuración del generador de archivo Excel para la base de datos.

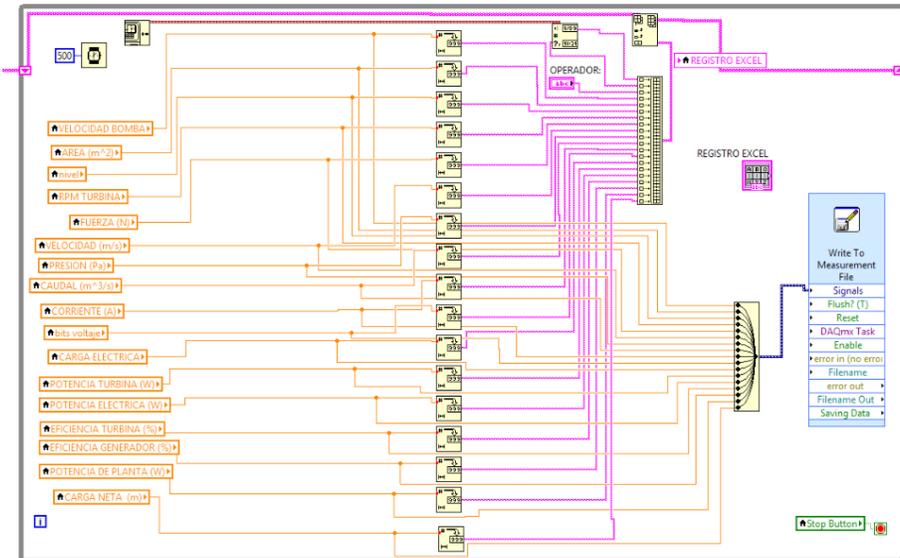


Fig. 3.47 Programación del registro y base de datos en Excel.

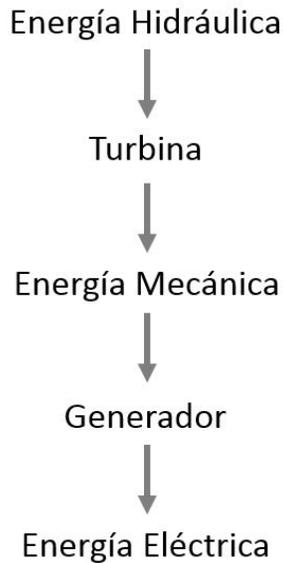


## Capítulo 4

### Planta hidroeléctrica

La planta de generación hidroeléctrica está ubicada en el laboratorio de Termodinámicos (LIME 1), esta instalación se utiliza para la enseñanza de los alumnos de la carrera de Ingeniería Mecánica Eléctrica, como ya se mencionó antes, el propósito de este trabajo es en parte, modernizar la instalación en cuanto a la forma de operarla, para que el aprendizaje sea de una forma más moderna y apegada a la realidad en cuanto a los sistemas computacionales que controlan las instalaciones.

Una central hidroeléctrica puede definirse como una instalación mediante la cual se consigue aprovechar la energía contenida en una masa de agua situada a cierta altura, transformándola en energía eléctrica. Esto se logra conduciendo el agua desde el nivel en el que se encuentra, hasta un nivel inferior en el que se sitúan una o varias turbinas hidráulicas que son accionadas por la caída del fluido.



*Fig. 4.1 Diagrama de flujo de energía en central hidroeléctrica.*

En una central hidroeléctrica se pueden distinguir tres fases principales:

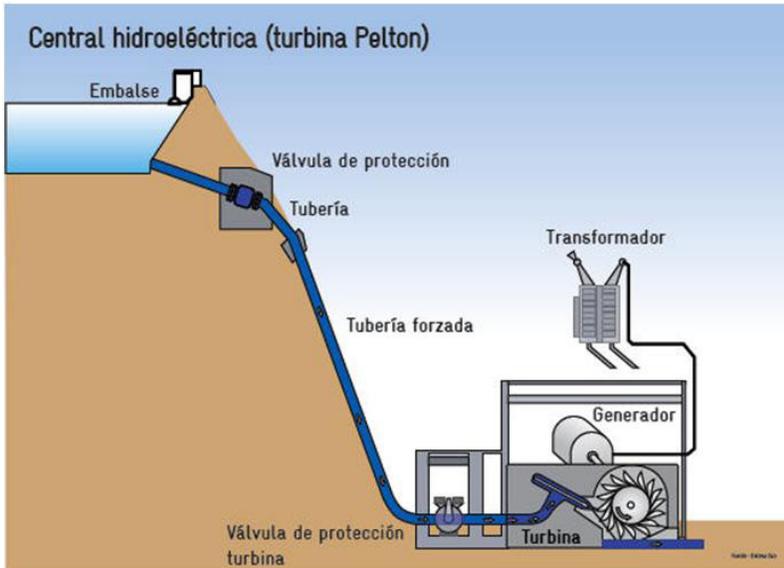
- Fase de generación:

Esta fase está compuesta de una tubería que transporta el agua hacia la turbina, en la que se produce la primera transformación de energía potencial del agua en energía mecánica. El elemento básico de la turbina es el rotor, este cuenta con hélices o canjilones alrededor de su circunferencia de tal forma que el fluido en movimiento provoca una fuerza tangencial que impulsa la rueda haciéndola girar.

Esta energía mecánica se transfiere a través de un eje para proporcionar el movimiento del rotor del generador.

- Fase de transformación:

La fase de transformación la constituyen máquinas eléctricas estáticas, cuya función es la de cambiar los parámetros de la energía eléctrica, esto nos permite transmitir la energía eléctrica desde un sistema con un voltaje dado a otro sistema con un voltaje deseado.



*Fig. 4.2 Componentes principales en una central hidroeléctrica.*

- Fase de salida:

Es la fase en la que la energía generada por la central hidroeléctrica sale del patio de maniobras con un voltaje determinado para llegar a la etapa de transmisión en la que los voltajes son reducidos o elevados por los transformadores de potencia en las subestaciones eléctricas.

El funcionamiento de una central hidroeléctrica comienza con la fuerza del agua en movimiento, la energía hidráulica que se puede obtener de una zona depende de los cauces de agua y desniveles de la misma y existe, por lo tanto, una cantidad máxima de energía que se puede obtener por este medio.

El principio de funcionamiento dice que la energía cinética del agua se convierte en energía eléctrica mediante sucesivas transformaciones de energía, esto se logra al aprovechar un desnivel para conducir el fluido hacia una instalación más abajo, en ella se hace pasar el agua a gran presión por una tubería hasta llegar a la turbina, lo que hace transformar la energía cinética del fluido en energía mecánica rotatoria, lo cual hace funcionar un generador eléctrico, que convierte la energía mecánica en energía eléctrica. Realizado este proceso el agua se devuelve al río y se normaliza su curso.

La ventaja principal de esta forma de adquirir energía eléctrica es que el caudal del agua puede ser controlado, de forma que en el momento que la de-

manda eléctrica aumente o disminuya, el fluido se deja fluir o simplemente se mantienen cerradas las compuertas de descarga.

La generación de electricidad tiene lugar en los generadores a partir de la rotación del rotor electromagnético impulsado por una turbina, se induce la tensión en las bobinas del estator para finalmente extraer la energía eléctrica de las terminales de este. Esto es posible debido al principio de inducción electromagnética descubierto por Michael Faraday en 1831, este principio dice que la magnitud de la tensión inducida es proporcional a la variación del flujo magnético, es decir, se producirá una fuerza electromotriz (voltaje) en un medio expuesto a un campo magnético variable, o en un medio móvil respecto a un campo magnético estático no uniforme, es así cuando un cuerpo conductor produce una corriente inducida.

### Composición de una planta hidroeléctrica:

Se pueden catalogar los componentes principales de una central hidroeléctrica en los siguientes cuatro grupos:

- Cortinas

Las cortinas pueden ser hechas de tierra o concreto, se construyen a través de lechos de ríos para crear embalses de almacenamiento. Las cortinas

permiten regular el flujo de agua durante todo el año para que la central eléctrica pueda funcionar a casi toda su capacidad.

Se proporcionan vertederos adyacentes a la cortina para agua cuando el nivel del embalse suba demasiado. Se ha visto que la demanda de electricidad varía considerablemente a lo largo del día y durante las estaciones, por consiguiente, el agua disponible no siempre se puede utilizar para suministrar energía al sistema, si el embalse es pequeño se tendrá que dejar fluir el agua por el vertedero sin poder utilizarla.

Las cortinas muy a menudo tienen un doble propósito, proporcionar irrigación y vías de navegación, además de su rol de generación de potencia.

- Conductos, tuberías de presión y caja espiral

En grandes instalaciones, los conductos dirigen el agua del sitio de la presa a la planta de generación, estos pueden ser canales abiertos o túneles cavados en roca. Los conductos alimentan una o más tuberías de presión, las cuales llevan el agua a las turbinas.

Las tuberías de presión canalizan el agua hacia la caja espiral que circunda el rodete o rotor de modo que el agua se distribuya uniformemente alrededor de su circunferencia. Aspas guía y compuertas de mariposa controlan el agua para que fluya uniformemente hacia las aspas del rotor.

- Tubo de aspiración y canal de descarga

El agua que ha pasado a través del rotor fluye entonces a través de un canal vertical llamado tubo de aspiración. Ésta, mejora la eficiencia hidráulica de la turbina. Conduce al canal de desfogue, el cual canaliza el agua corriente abajo del lecho del río.

- Central eléctrica

La central contiene los generadores síncronos, transformadores, cortacircuitos y todo el equipamiento electromecánico. Los instrumentos, relevadores y medidores están en un cuarto central desde donde toda la planta puede ser monitoreada y controlada.

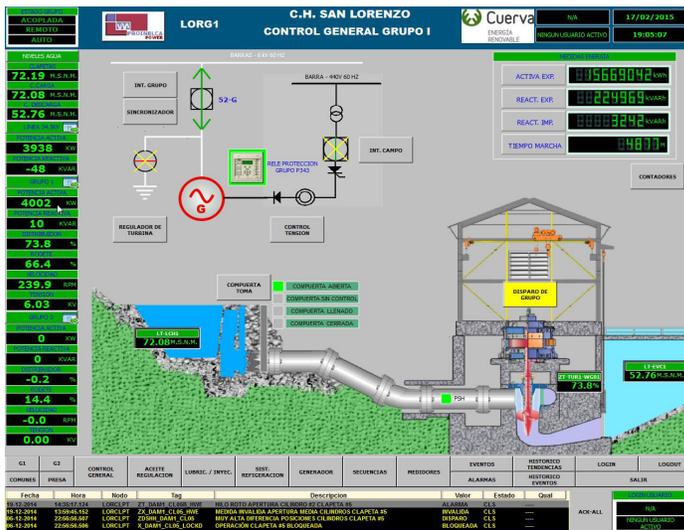


Fig. 4.3 Interfaz gráfica de un sistema SCADA en central hidroeléctrica.

## 4.1 Instalación actual

El banco de pruebas del laboratorio cuenta con el siguiente equipo:

1. Tanque de realimentación
2. Indicador de nivel en tanque
3. Indicador de presión de inyector
4. Control de motor de velocidad variable
5. Motor DC
6. Bomba centrífuga
7. Indicador de fuerza con freno Prony
8. Turbina Pelton
9. Inyector



*Fig. 4.4 Instalación con los elementos originales.*



## Procedimiento de operación actual (sin sistema SCADA)

1. El arranque del motor es mediante un switch on/off y la velocidad es regulada por un control de rotación. Se debe regresar a cero el control de rotación antes de apagar la unidad.
2. Cuando la bomba este trabajando se debe abrir la válvula reguladora de presión (perilla negra) para iniciar el flujo de agua.  
Hay que asegurarse de haber calibrado el flujómetro (vertedero), poniendo a cero el indicador de vidrio cuando la válvula reguladora este completamente abierta.
3. El nivel de agua en el canal no deberá de ser tal que rebase el flujo indicado en la escala de vidrio colocada a un lado del tanque.
4. La presión de entrada a la tobera de inyección es indicada por un manómetro de Burdon.
5. El ajuste de la presión se hace mediante la válvula reguladora o mediante la velocidad del motor, obteniendo un buen flujo de agua a través de la turbina.
6. Si la unidad de turbina esta lista podrá trabajar libremente. Puede entonces colocársele una carga apretando el tornillo negro en el freno Prony.
7. La velocidad de la turbina o de la bomba deberán ser medidas con un tacómetro de mano.
8. Continuar haciendo las pruebas variando la carga sobre la turbina, el flujo de agua, la presión de alimentación, etc.

La operación de la planta inicia encendiendo el switch on/off, después con el reóstato se controla el voltaje de la armadura del motor (motor shunt controlado por armadura), lo cual controla la velocidad de giro, al controlar la velocidad de giro se controla la presión y el caudal del fluido que va dirigido hacia el inyector, con este también se puede regular el caudal, puesto que regula el área de paso del fluido, y por tanto al variar cualquiera de estos dos reguladores se estará manipulando la velocidad de giro de la turbina Pelton y por tanto la velocidad de giro del generador eléctrico, esto se verá reflejado en la potencia eléctrica útil.

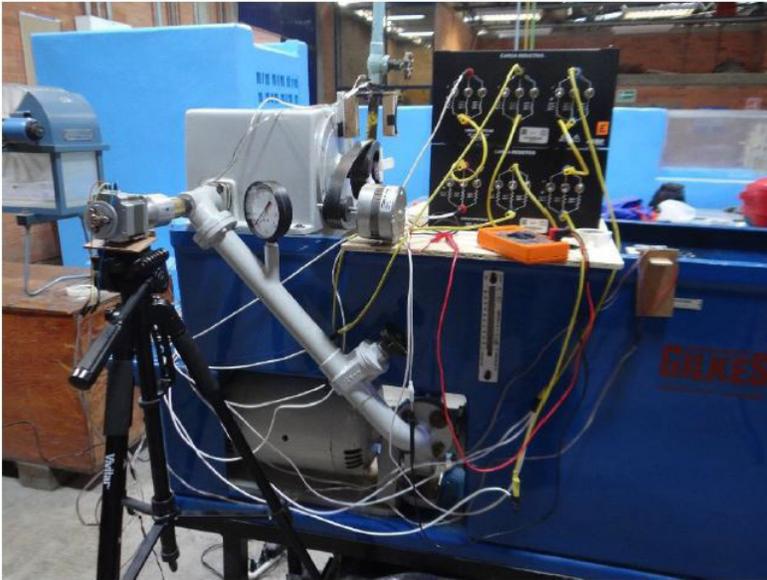
## **4.2 Modificaciones a la planta**

Se modificará el control de la apertura del inyector de la turbina Pelton, esto se hará agregando un motor a pasos con un cople para adaptarlo a la manija del inyector; usando como interfaz electrónica un driver L298n conectado a la tarjeta ARDUINO NANO, se realiza el control del motor a través de la interfaz HMI.

Los dispositivos que se agregarán son sensores, estos serán instalados y posicionados físicamente sobre la instalación actual de la turbina Pelton, así como el generador, y las cargas para este.

Los dispositivos por agregar son:

- Sensor de RPM
- Sensor de nivel
- Sensor de voltaje
- Sensor de corriente
- Relé de cargas
- Alternador
- Cargas



*Fig. 4.5 Componentes anexados a la instalación para implementación SCADA.*

Funcionamiento con el sistema SCADA:

El sistema SCADA está diseñado a modo de que todas las operaciones manuales tanto de control como de supervisión y monitoreo de indicadores se lleven a cabo desde una unidad central de control y monitoreo llamada estación maestra. Al concentrarse ambas acciones en un mismo sitio, el requerimiento físico de trasladarse a la instalación a modificar parámetros es anulado al tener un control total y unificado en la PC, esto permite la concentración completa del estado de la planta en tiempo real y remotamente.

Todo el control de la planta se maneja desde la interfaz HMI en la PC a través de controles e indicadores virtuales desde un sitio fijo.

### **4.3 El alternador**

Para una instalación de planta hidroeléctrica es indispensable contar con la máquina dinámica que transformará la energía mecánica en energía eléctrica, el alternador. En la instalación del laboratorio de Termofluidos, la turbina Pelton tiene una polea a la cual, mediante una banda, se le puede acoplar un alternador.

La instalación del banco de pruebas tiene una limitante que es el espacio de trabajo, debido a esto, adaptar un alternador del laboratorio de eléctrica no



es posible ya que tiene un tamaño mayor y las características de trabajo y funcionamiento no se adaptan a las de la turbina, es por eso que se optó por construir el alternador.

La idea para la construcción del alternador es tomar un motor asíncrono y modificar el rotor agregando un núcleo de hierro e imanes de neodimio para la creación del campo magnético.

En el rotor, que recibirá la fuerza mecánica de rotación (inductor), se adaptará un núcleo de hierro e imanes permanentes de neodimio que de acuerdo con su tamaño producirán un flujo de campo magnético, esto sería el equivalente a la excitación con corriente directa al devanado de campo de un alternador convencional. Con esta modificación al rotor se tiene un alternador con rotor de polos salientes.

El estator (inducido) cuenta con 4 polos bobinados en torno a un núcleo de material ferromagnético (por lo tanto 4 polos en el rotor), la rotación del inductor hace que el campo formado por imanes fijos sea variable en el tiempo y el paso de este campo variable por los polos del inducido genera una corriente alterna en las terminales de la máquina. En el estator no se realizó ninguna modificación, solo se agregó un puente rectificador para el sensor de voltaje en paralelo con el circuito de cargas.

En el eje del alternador se adaptó una polea torneada en material de polímero (poliamida PA6) lo que reduce su peso y costo de fabricación en comparación con aluminio, manteniendo propiedades similares.

Para tener una idea del voltaje que puede generar el alternador se tiene:

$$\phi = B * A \text{ [Wb]}$$

$$B = 3800 \text{ gauss} = 0.38 \text{ Tesla}$$

Donde:

- $\phi = \text{Flujo magnético [Wb]}$
- $B = \text{Campo magnético [T]}$
- $A = \text{Área [m}^2\text{]}$

Utilizando imanes de 20 mm x 20 mm se tiene:

$$\phi = 0.38 \text{ Tesla} * 0.0004 \text{ m}^2 = 0.000152 \text{ Wb} \text{ por imán}$$

Usando dos imanes por polo:

$$\phi = 0.000304 \text{ [Wb]} \text{ por polo}$$

Si se considera al alternador como máquina síncrona se tiene:

$$E = \phi f N * 4.44 \text{ [V]}$$

Donde:

$$E = \text{FEM inducida o voltaje [V]}$$

$$\phi = \text{Flujo magnético [Wb]}$$

$$f = \text{Frecuencia [Hz]}$$

$$N = \text{Número de espiras o vueltas}$$

De forma experimental se obtuvo que a 3100 RPM aproximadamente se generó 56 volts AC, por tanto:

$$f = \frac{nP}{120} = \frac{3100 \text{ RPM} * 4}{120} = 103.33 \text{ [Hz]}$$

Donde:

- $f$ =Frecuencia [Hz]
- $n$ =Velocidad [RPM]
- $P$ =Número total de polos

Se deduce que:

$$N = \frac{E}{\phi f * 4.44} = \frac{56 \text{ V}}{.00034 \text{ Wb} * 103.33 \text{ Hz} * 4.44} = 359.02 \text{ [Espiras]}$$

Podría decirse que si se llevara el alternador a su máxima velocidad posible (6000 RPM):

$$E = 0.00034 \text{ Wb} * 200 \text{ Hz} * 359.02 \text{ espiras} * 4.44 = 109 \text{ Volts AC}$$

Como componente final se añadió una polea al eje del alternador para acoplarse a la polea de la turbina por medio de una banda.

La relación entre diámetros de poleas y velocidades es la siguiente:

$$D1 * N1 = D2 * N2$$

$$N2 = \frac{D1 * N1}{D2} \text{ [RPM]}$$

Donde:

- D1=Diámetro de polea de turbina [m]
- D2=Diámetro de polea de alternador [m]
- N1=Velocidad de turbina [RPM]
- N2=Velocidad de alternador [RPM]

Colocando a una distancia entre centros de 15 cm el alternador de la polea de turbina, la longitud de la banda es:

$$Lb = \frac{\pi}{2} (D + d) + 2C + \frac{(D - d)^2}{4C}$$

Donde:

- Lb=Longitud de banda
- D=Diámetro de polea de turbina [m]
- d=Diámetro de polea de alternador [m]
- C=Distancia entre centros [m]

$$Lb = \frac{\pi}{2} (0.11 \text{ m} + .04\text{m}) + (2 * 0.15\text{m}) + \frac{(0.11\text{m} - 0.04\text{m})^2}{4 * 0.15\text{m}} = 54.31\text{cm} = 21.38 \text{ Pulgadas}$$



*Fig. 4.6 Vista del eje del alternador con el rotor reconstruido.*



*Fig. 4.7 Vista completa del alternador con polea para acoplar a turbina.*

## 4.4 Indicadores de planta y parámetros

Se entiende por parámetro a las variables que, incluidas en una ecuación, dan un resultado de esta, el valor que toman estos parámetros no es fijo, es decir, pueden cambiar en el tiempo cuando el estado de los elementos de control que rigen a las variables es modificado y por tanto el resultado de la ecuación.

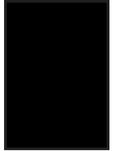
Se entiende como indicadores principales de la planta a los resultados de las ecuaciones conformadas por variables a la que se ha llamado parámetros, estos indicadores principales permiten conocer el estado de la planta, puesto que proporcionan la información principal para determinar su estado.

En el panel frontal, los indicadores principales de operación de la planta se encuentran aislados del resto de parámetros para enfatizar su importancia y fácil visualización.

A continuación se enlistan los indicadores y los parámetros de operación de la planta.



*Fig. 4.8 Indicadores de parámetros de planta ubicados en el panel frontal.*



### Indicadores:

- Potencia útil de planta [Watts]
- Potencia hidráulica [Watts]
- Potencia eléctrica [Watts]
- Eficiencia de turbina [%]
- Eficiencia de generador [%]
- Carga neta [m]
- Factor de potencia [adim]
- Potencia mecánica [Watts]

### Parámetros:

- Presión hidráulica (carga en manómetro) [ $\text{N/m}^2$ ]
- Caudal [ $\text{m}^3/\text{s}$ ]
- Carga neta [m]
- Velocidad hidráulica [m/s]
- Voltaje [Volts]
- Corriente eléctrica [Amperes]
- Velocidad angular de turbina [RPM]
- Área de paso en inyector [ $\text{m}^2$ ]
- Densidad flujo [ $\text{kg/m}^3$ ]
- Gravedad [ $\text{m/s}^2$ ]

POTENCIA ÚTIL DE LA PLANTA	$P_u = \rho * g * Q * H_n * \eta_g * \eta_t$ [Watts]
POTENCIA HIDRÁULICA	$P_h = \rho * g * Q * H_n$ [Watts]
POTENCIA ELÉCTRICA	$P_e = V * I * f_p$ [Watts]
POTENCIA MECÁNICA	$P_m = P_e + I^2 Z$ [Watts]
EFICIENCIA DE TURBINA	$\eta_t = \frac{(F * r) \left(2\pi * \frac{n}{60}\right)}{\rho * g * Q * H_n}$ [%]
EFICIENCIA DE GENERADOR	$\eta_g = \frac{V * I * f_p}{(F * r) \left(2\pi * \frac{n}{60}\right)}$ [%]
CARGA NETA	$H_n = \frac{P}{\gamma} + \frac{V^2}{2g}$ [m]
FACTOR DE POTENCIA	$f_p = \tan^{-1} \left( \frac{2\pi f L}{\sqrt{Z^2 - (2\pi f L)^2}} \right)$ [adim]

Fig. 4.9 Tabla con los indicadores principales de la planta.



## Capítulo 5

### Diseño y desarrollo de HMI

En este capítulo se explicará a detalle todo lo que conforma el sistema SCADA. Para que el sistema funcione, es necesario enlazar el software con el hardware, es por eso que, detalles como diagramas de conexión, explicación de código en C, explicación de código en G, etc., serán tratados aquí.

Como se mencionó en capítulos anteriores, el diseño de la interfaz se realizará utilizando el software LabVIEW, usando las dos herramientas principales del software que son el panel frontal y el diagrama de bloques. En el panel frontal es en donde se lleva a cabo el diseño visual de la interfaz, es decir, aquí se realiza el posicionamiento de cada indicador y cada elemento de control virtual tal y como lo verá en pantalla el usuario al momento de operar la planta. Todo lo que se encuentre en el panel frontal es con lo que el usuario va a interactuar.

El desarrollo de la interfaz se lleva a cabo en el diagrama de bloques, es en donde todos los elementos posicionados en el panel frontal se enlazan con

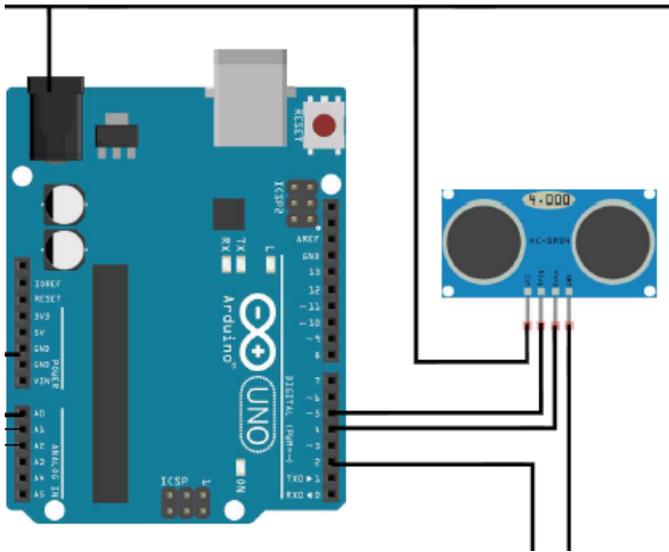
otros bloques de función para concretar el programa y que realice la tarea de control y supervisión de la planta.

En el diagrama de bloques es en donde se explica el funcionamiento del programa, para poder entender completamente el funcionamiento de la interfaz es necesario explicar todo lo que hay detrás del software y es por eso que en este capítulo se explican conceptos como la ubicación de cada dispositivo de campo, la programación en LabVIEW, tanto en bloques como su posición en el panel frontal, la programación en ARDUINO y su diagrama de conexión.

## **5.1 Sensor de nivel**

Se utiliza un sensor ultrasónico HC-SR04 compatible con la placa ARDUINO, usando este sensor es posible obtener tres indicadores: el estado del nivel del agua en el tanque (visual), en nivel del tanque en milímetros y el caudal.

La conexión de este sensor al sistema es muy sencilla, consta de cuatro pines, vcc es conectado al bus de 5 volts que alimenta a todos los sensores, GND es el pin neutro, el pin trig (trigger) o disparador es conectado al pin digital 5 de la placa ARDUINO UNO, y el pin echo es conectado al pin digital 4.



*Fig. 5.1 Diagrama de conexión del sensor de nivel.*

Como se mencionó en el capítulo 3, se debe enviar un pulso de 10 micro segundos a través del pin 5 (trigger) para que posteriormente el sensor a través del transductor piezoeléctrico envíe una señal de 8 pulsos de 40 KHz para realizar la ecolocación, la distancia medida corresponde a un pulso con un ancho que representa el doble de la distancia a medir (ida y vuelta), esta distancia es obtenida mediante el siguiente código.

El código comienza definiendo los pines digitales para el echo y el trigger y se declaran las variables duración y distancia como datos de tipo long.

En el void setup se configura el modo de los pines “Pecho” (echo) como INPUT y “Ptrig” (trigger) como OUTPUT.

En la parte principal del código “void nivel” la primera instrucción es para mantener apagado el disparador por dos micro segundos (LOW) para evitar errores de lectura, después se mantiene el disparador encendido (HIGH) por 10 micro segundos y después se apaga (LOW).

Para saber la duración del pulso que llega por el pin echo se utiliza la instrucción “pulseIn” que nos dice cuanto tiempo estuvo en alto (HIGH).

Una vez detectado el tiempo, la distancia se determina mediante la siguiente formula:

$$\text{DISTANCIA EN cm} = (\text{duración} / 2) / 29$$

La duración del pulso corresponde a la distancia de ida y de vuelta por eso es dividida entre 2 y este resultado es dividido entre 29 puesto que la velocidad del sonido es 340 m/s ó 1/29 cm/ $\mu$ s.

El resultado en centímetros es impreso por la variable “distancia” en el monitor serie.

```

void nivel(){
  digitalWrite(Ptrig, LOW);
  delayMicroseconds(2);
  digitalWrite(Ptrig, HIGH); //
  delayMicroseconds(10);|
  digitalWrite(Ptrig, LOW);

  duracion = pulseIn(Pecho, HIGH);
  distancia = (duracion/2) / 29;

  Serial.print(distancia);
  Serial.println("cm");
}

```

*Fig. 5.2 Código en ARDUINO para sensor de nivel.*

Para desplegar la información en la interfaz HMI en LabVIEW, la programación comienza con un ciclo while en el que se establece la comunicación serial a 115200 baudios a través del puerto COM4 usando la función bloque “Visa Serial”, en seguida un bloque de lectura es utilizado para leer los datos proporcionados por ARDUINO UNO, como la información es proporcionada en forma de cadena de caracteres o “String”, se utiliza el bloque función “Decimal string to Number” que convierte una cadena de caracteres numéricos en enteros decimales y devuelve un número, esto es para poder representar el valor de la variable en un indicador numérico, a la vez que se muestra la lectura original proveniente de la placa ARDUINO UNO usando un indicador string ( de color rosa en la Fig. 5.3).

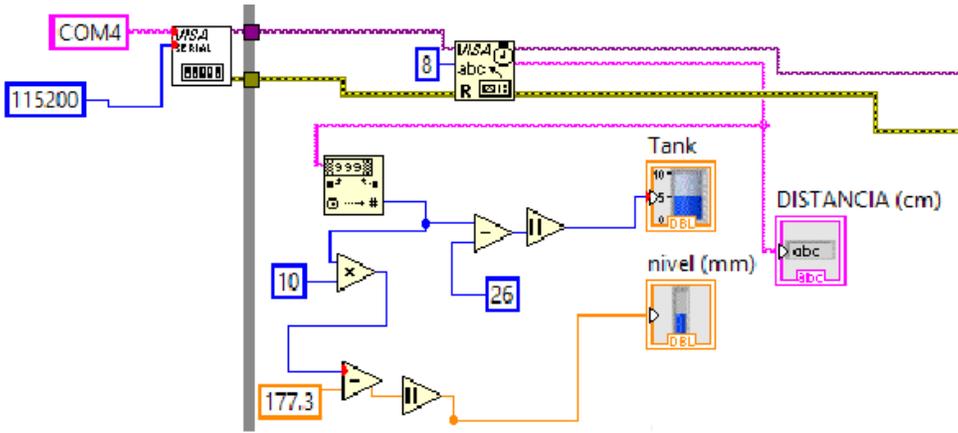


Fig. 5.3 Diagrama de bloques en LabVIEW para sensor de nivel.

Una vez que la información en “String” ha sido convertida a tipo numérica, la señal se redirige a dos indicadores: el indicador de nivel en milímetros y la representación del llenado o nivel del agua en el tanque.

Para mostrar el nivel del agua en el tanque se le resta 26 cm que es la longitud de la profundidad del tanque a la medición que proporciona en centímetros el sensor ultrasónico, después, usando la función de valor absoluto (puesto que en la resta se obtiene un número negativo), un valor numérico entero positivo es representado en el indicador “Tank”.

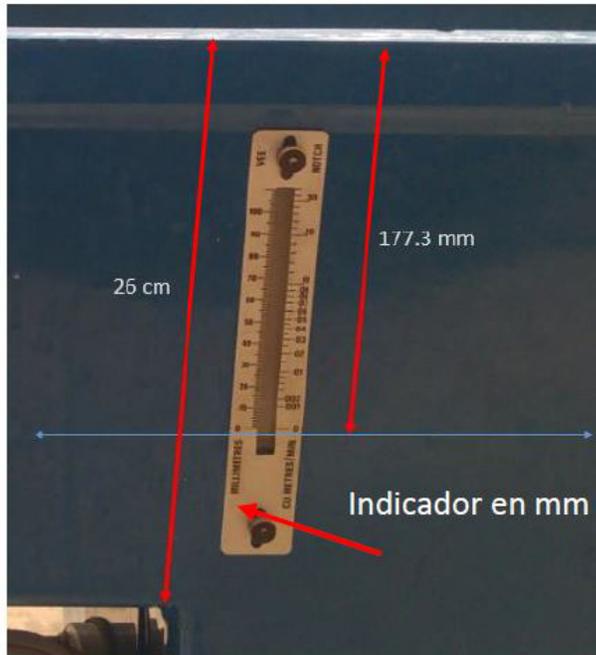


Fig. 5.4 Flujómetro de vidrio con indicador de nivel.

Para mostrar el nivel del agua en el taque en milímetros, se multiplica por 10 el valor proporcionado por sensor de nivel y se resta 177.3 milímetros, que es la distancia desde la marca cero del indicador analógico hasta la superficie en el tanque (el sensor es posicionado en la superficie), después se utiliza la función de valor absoluto para obtener un valor positivo. De esta manera, se comprueba que la medición mostrada en la interfaz HMI es la misma proporcionada por el indicador analógico del tanque.



Fig. 5.5 Indicador de nivel en panel frontal.

## 5.2 Sensor de Velocidad (RPM)

Como ya se mencionó antes, se optó por construir este sensor utilizando un led infrarrojo y un fototransistor, de modo que al interrumpirse la transmisión y recepción entre el led y el foto transistor por la turbina Pelton las interrupciones en ARDUINO se habilitan y se puede detectar la velocidad angular de la turbina.

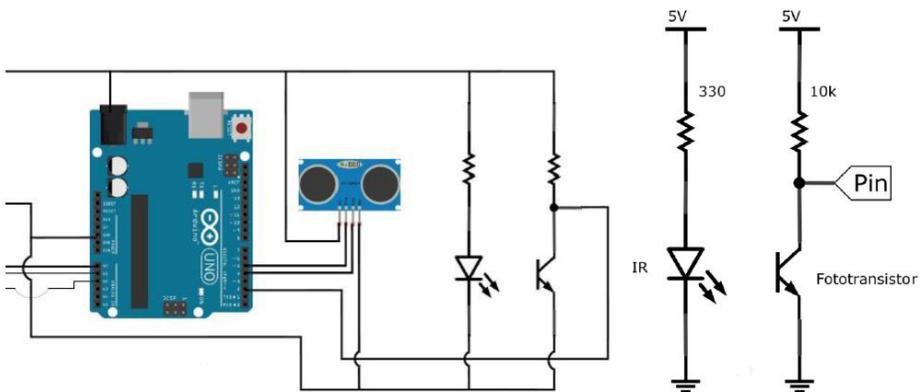


Fig. 5.6 Diagrama de conexión de sensor de RPM.



Como se puede observar en el esquema, el circuito es alimentado por los cinco volts que proporciona la placa ARDUINO, se utiliza una resistencia de 330 Ohms en led infrarrojo y una resistencia pull-up de 10 kilo Ohms, en esta configuración, la salida al pin digital en la placa ARDUINO proporciona un estado alto (HIGH) cuando el foto transistor se encuentra cerrado (saturación) y un estado bajo (LOW) cuando abierto (corte).

El código en ARDUINO para este sensor comienza con la declaración de variables, las cuales son “duración” y “distancia” con datos de tipo LONG y una variable del tipo VOLATILE INT llamada “contador” e iniciada en cero.

En el void setup se configura la velocidad de transmisión de datos a 115200 baudios y se establece la interrupción cero en el pin digital dos para que se active durante el flanco de subida o RISING.

Al ocurrir las interrupciones por el pin 2, el programa principal “void rpm” se detiene y ejecuta la función “void interrupcion0”, una vez concluida retoma el programa principal “void rpm”.

La función “void interrupcion0” aumenta el contador cada vez que ocurra la interrupción en el infrarrojo durante el flanco de subida “RISING” de la señal.

En la función principal “void rpm” se crea un retardo de casi un segundo (999 ms), que es el tiempo en el que ocurren las interrupciones y el “contador” las registra y las almacena en la memoria RAM dentro de la variable “contador”.

Las interrupciones registradas durante un segundo se multiplican por un factor 30, que consta de los segundos que tiene un minuto por el número de aspas (60 segundos / 2 aspas) =30, de esta manera se obtienen las RPM, se imprime el resultado y se reinicia el contador.



```
void rpm() {
    delay(999);
    Serial.print(contador*30);
    Serial.println(" RPM");
    contador = 0;
}

void interrupcion0() {
    contador++;
}

|
```

Fig. 5.7 Código en ARDUINO para sensor de RPM.

El código en LabVIEW es muy sencillo, después de la obtención de información del sensor de nivel, al ser la comunicación en serie, se continúa con la lectura de los datos correspondientes al sensor de nivel usando la función bloque “Visa read”, se establece el número de bytes a leer (20) y se utiliza la función “Decimal string to Number” para convertir una cadena de caracteres numéricos en enteros decimales y devolverlo en número, esto es para poder representar el valor de la variable en el indicador numérico RPM TURBINA.

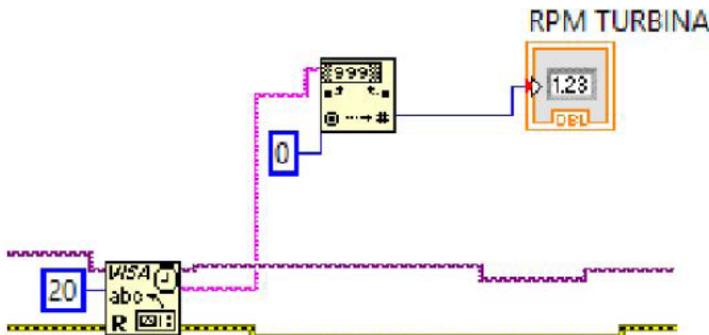


Fig. 5.8 Programación en LabVIEW para sensor de RPM.

Se observa en el panel frontal la visualización del indicador de RPM de un tamaño considerablemente mayor al resto de los indicadores puesto que es uno de los más importantes que hay que monitorear y debe ser fácil de localizar.

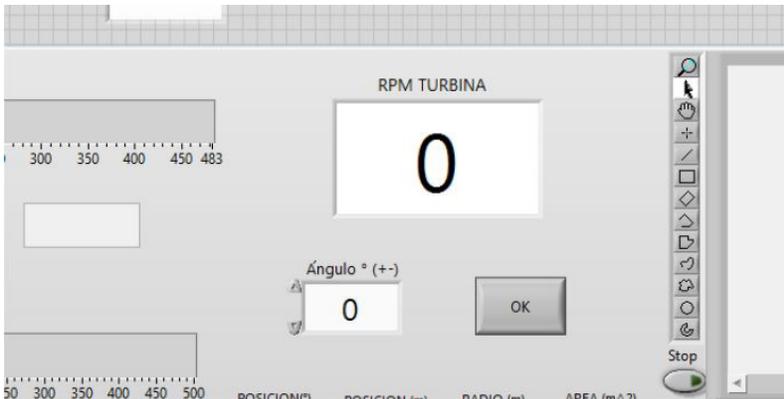
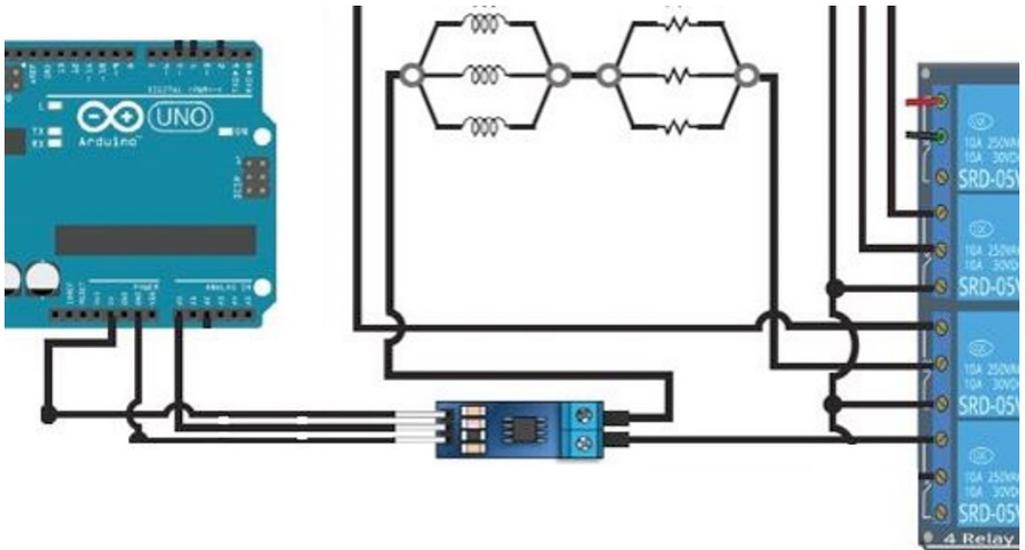


Fig. 5.9 Ubicación del indicador de RPM en el panel frontal.

### 5.3 Sensor de corriente

Comenzando con el diagrama de conexión, se usa un sensor de corriente de efecto hall ACS712 compatible con la placa ARDUINO, este sensor es alimentado por el bus de 5 volts de CD que alimenta además a todos los sensores, el pin OUT es el que proporcionará los datos del sensor y es conectado al pin analógico A0 de la placa ARDUINO UNO.

El sensor es conectado en serie al circuito al que se realizarán las mediciones, para esto el sensor debe conectarse como se muestra en el esquema, entrando la corriente por el borne situado cerca del pin GND, y la salida de la corriente por el borne situado cerca del pin VCC, de otra forma el sensor no funcionará.



*Fig. 5.10 Diagrama de conexión para sensor de corriente.*

La obtención de los valores de corriente utilizando este sensor se hace mediante la aplicación de la siguiente metodología:

El sensor ACS712 es capaz de medir la corriente en dos direcciones, es decir, es posible encontrar el valor pico en una dirección y en otra; una vez encontrado el valor de estos picos, como se sabe que la forma de onda es sinusoidal, aplicando una sencilla ecuación se puede obtener el valor RMS (valor eficaz) para la medición.

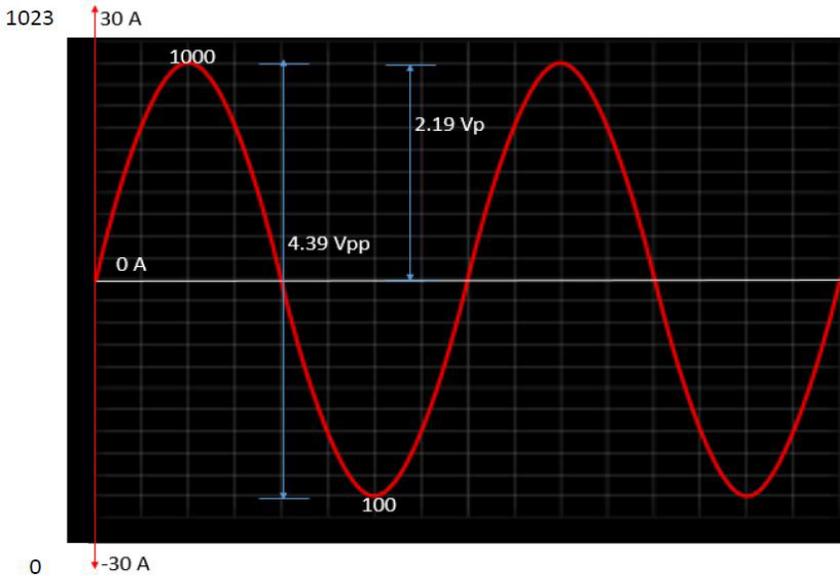


Fig. 5.11 Medición de la señal de corriente alterna por el sensor ACS712.

El código en C para la obtención de datos de corriente alterna comienza a funcionar con la subrutina `getVPP()` de la función `sensorcorriente()`, la cual usando la función "millis" nos permite medir durante un segundo los valores pico a pico de señal sinusoidal de la corriente, se lee un valor máximo y un valor mínimo, los cuales son guardados en la variable `readValue`, esto es para poder determinar la amplitud de la señal pico a pico en valor numérico entre 0 y 1023, una vez medidos estos valores, la variable `result` realiza una operación para entregar esa amplitud pico a pico en volts un valor entre 0 y 5 volts (Fig. 5.12).

```

float getVPP()
{
    float result;

    int readValue;
    int maxValue = 0;
    int minValue = 1024;

    uint32_t start_time = millis();
    while((millis()-start_time) < 1000)
    {
        readValue = analogRead(sensorIn);

        if (readValue > maxValue)
        {
            maxValue = readValue;
        }
        if (readValue < minValue)
        {
            minValue = readValue;
        }
    }
    result = ((maxValue - minValue) * 5.0)/1024.0;
    return result;
}

```

*Fig. 5.12 Subrutina para la medición de valores pico a pico de corriente alterna.*

En la función “sensorcorriente()” (ver figura 5.13) la variable “Voltaje” solicita el valor que mide la subrutina “getVPP()” para procesar ésta información y obtener el valor pico de la señal de corriente, la cual, se divide entre dos (por el semiciclo positivo y negativo) y se guarda en la variable “Voltaje”, después se multiplica por el factor 0.707 para obtener el valor RMS de la corriente y se guarda en la variable “VRMS”, de esta manera se obtiene la señal acondicionada a un valor entre 0 y 5 volts. Para expresar el valor de la corriente medida en amperes se multiplica el valor guardado en la variable “VRMS” por 1000

(mili amperes en un ampere) y se divide entre el factor de sensibilidad, en este caso 66 mili volts por ampere (dato característico del sensor).

```
void sensorcorriente() {
    Voltage = getVPP();
    VRMS = (Voltage/2.0) *0.707;
    AmpsRMS = (VRMS * 1000)/mVperAmp;
    Serial.print(AmpsRMS);
    Serial.println(" Amps RMS");
}
```

Fig. 5.13 Función principal del código de sensor de corriente.

Tomando como ejemplo los valores máximos y mínimos medidos por el sensor son 1000 y 100:

$$1000 - 100 = 900$$

$$\frac{900 * 5 \text{ volts}}{1023} = 4.39 \text{ Volts}_{pp}$$

$$\frac{900 * 5 \text{ volts}}{1023} = 4.39 \text{ Volts}_{pp}$$

$$\frac{900 * 5 \text{ volts}}{1023} = 4.39 \text{ Volts}_{pp}$$

El código para el sensor de corriente comienza con la definición de las variables a usar y el tipo de dato:

```
#define Pecho 4
#define Ptrig 5
long duracion, distancia;
volatile int contador = 0;

const int sensorIn = A0;
int mVperAmp = 66;

float Voltage = 0;
float VRMS = 0;
float AmpsRMS = 0;

int FSR=A1;
int FSRReading;
int FSRVoltage;
unsigned long FSRResistance;
unsigned long FSRConductance;
```

*Fig. 5.14 Declaración de variables para código de sensor de corriente.*

En la figura 5.14 se define el pin analógico utilizado para el sensor (A0), se define una constante entera que corresponde a la sensibilidad del dispositivo de medición “mVperAmp=66”, la variable “Voltage”, que guarda el valor de la corriente pico a pico expresada en volts, la variable “VRMS” que guarda el valor RMS de la corriente expresada en volts y la variable AmpsRMS.

gettVPP es la subrutina que realiza la medición, durante un segundo, de los valores máximos y mínimos de la corriente alterna, estos valores son almacenados en la variable entera readValue, para después realizar la operación que

entrega el valor de la corriente pico a pico en volts.

Después, la función principal void sensorcorriente() solicita el valor obtenido en la subrutina gettVPP() para procesar la información, como se mencionó antes, y obtener el valor de la corriente en amperes RMS.

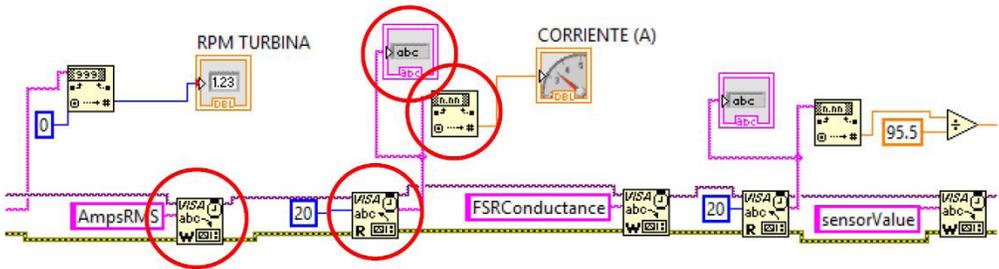


Fig. 5.15 Programación en LabVIEW del sensor de corriente.

La figura 5.15 muestra un fragmento del código en LabVIEW para la adquisición de datos, para lo cual, la parte delimitada en rojo corresponde a los bloques de función para la adquisición de los datos del sensor ACS712 de corriente alterna.

El bloque visa read function nos permite leer un número específico de bytes desde el dispositivo especificado por el bloque VISA RESOURCE NAME y devuelve el dato en un read buffer, en este caso el dispositivo especificado es el correspondiente al puerto USB “COM4” que es la tarjeta ARDUINO UNO.

Este bloque leerá información proveniente de ARDUINO UNO y será mostrada en el display del bloque string (bloque color rosa) tal cual como ARDUINO lo envía. Se utilizan 20 bytes puesto que la información es amplia como se puede observar.

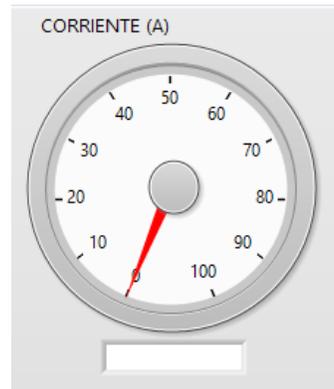
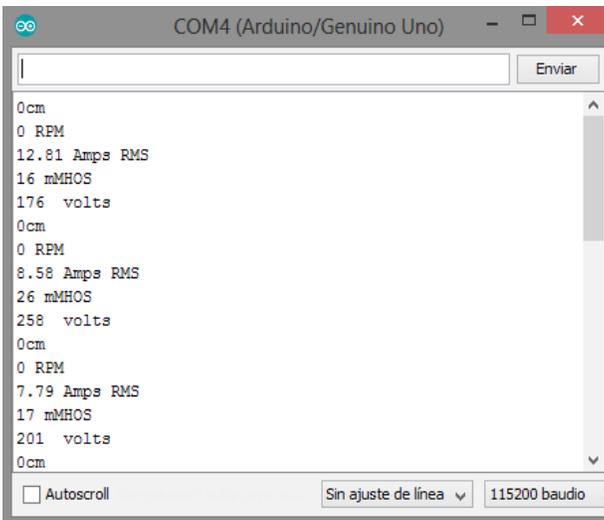
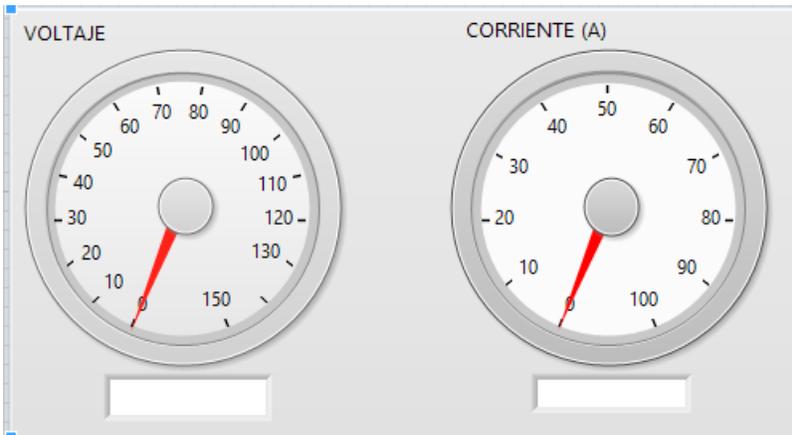


Fig. 5.16 Adquisición de datos en el monitor serie e indicador de corriente en LabVIEW.

Para poder mostrar los datos en un indicador de aguja se utiliza la función Fract/Exp String To Number Function, esta interpreta caracteres numéricos, operacionales, con punto decimal, exponencial, etc., provenientes de formato string y los regresa en formato numérico, el cual es el utilizado en los indicadores.



*Fig. 5.17 Indicador de corriente en panel frontal.*

La figura 5.17 muestra el diseño del panel frontal en donde se puede visualizar la información proveniente de la placa ARDUINO en un indicador de aguja virtual.

#### **5.4 Sensor de Voltaje**

Este sensor medirá el voltaje CA producido por el alternador, como se mencionó antes, se optó por construir este sensor haciendo uso del convertidor analógico-digital de la placa ARDUINO, un puente rectificador de diodos y un divisor de voltaje:

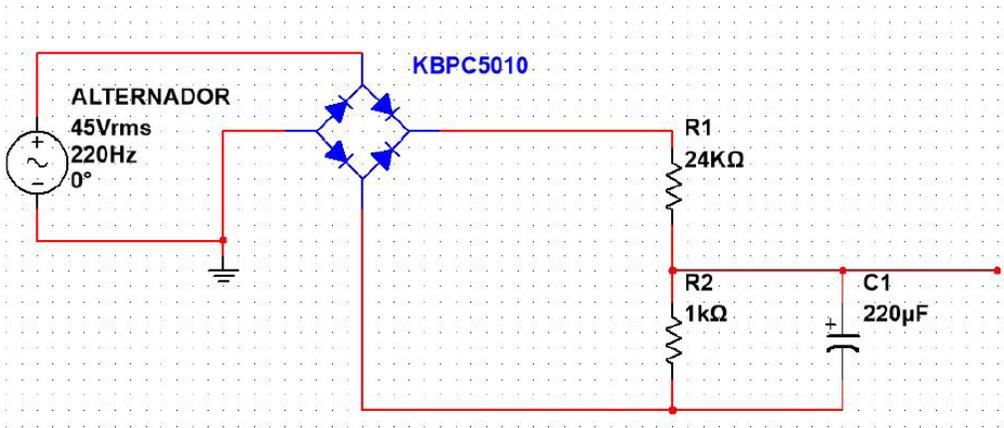


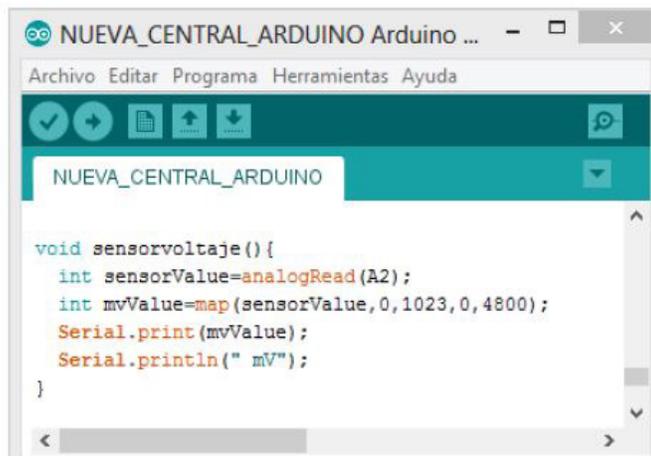
Fig. 5.18 Circuito de sensor de voltaje.

Para medir el voltaje producido por el alternador primero se debe rectificar la señal y trabajar solo con el semiciclo positivo, puesto que ARDUINO solo trabaja con corriente directa, mediante el divisor de voltaje se obtiene una expresión que relaciona el voltaje que entra a la placa ARDUINO con el voltaje producido por el alternador que es medido por el sensor:

$$\frac{900 * 5 \text{ volts}}{1023} = 4.39 \text{ Volts}_{pp}$$

Debido a las características de construcción, el sensor es capaz de medir hasta 133 Volts RMS de corriente alterna.

La programación en ARDUINO para este sensor comienza con la lectura de la variable entera “sensorValue” a través del pin analógico número dos, en el código se puede observar que se ha definido en la variable “mvValue” un rango de resolución que mapeará voltajes de entrada almacenados en la variable “sensorValue” entre 0 y 4800 mV en valores enteros entre 0 y 1023, se imprimirá el valor leído con las unidades en mili volts.

The image shows a screenshot of the Arduino IDE interface. The window title is "NUEVA\_CENTRAL\_ARDUINO Arduino ...". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with icons for checking, running, uploading, and downloading. The main editor area shows the following C++ code:

```
void sensorvoltaje(){  
  int sensorValue=analogRead(A2);  
  int mvValue=map(sensorValue, 0, 1023, 0, 4800);  
  Serial.print(mvValue);  
  Serial.println(" mV");  
}
```

Fig. 5.19 Código en ARDUINO para el sensor de voltaje.

Cuando el software SCADA esté en funcionamiento y la comunicación serie sea estable, el programa SCADA en LabVIEW leerá la información proveniente de la placa ARDUINO y la procesará de acuerdo con su programación, comenzando con la función “String to Number function”, la cual interpreta los caracteres de una cadena en enteros decimales y devolverá un número.

Para mostrar la lectura del sensor en el indicador de voltaje se debe procesar la información dividiendo entre 1000 y después realizar la operación que relaciona el voltaje de entrada y salida antes descrita.

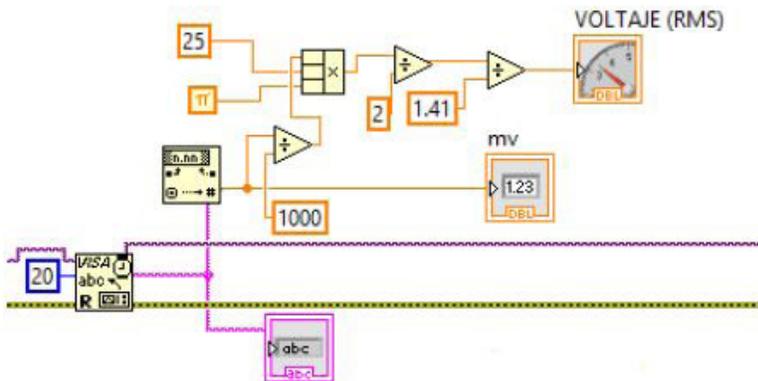


Fig. 5.20 Programación en LabVIEW para el sensor de voltaje.

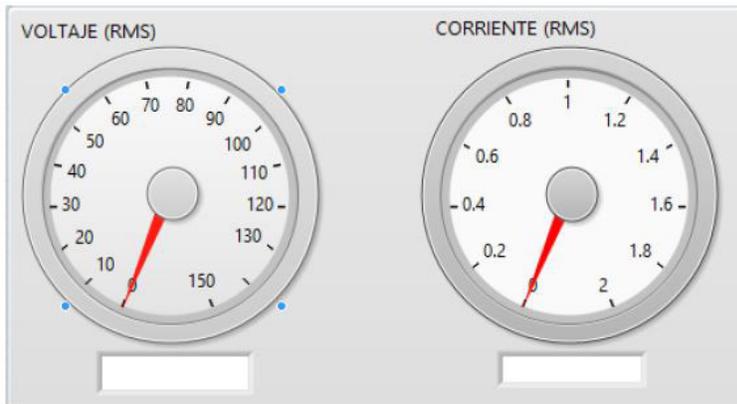


Fig. 5.21 Indicador de sensor de voltaje en panel frontal.

## 5.5 Indicador de caudal

La información proporcionada por el indicador de caudal proviene del procesamiento de las lecturas del sensor de nivel (HCS-04), este sensor se utiliza para medir el nivel de agua en el tanque y relaciona la lectura a un flujo volumétrico, esto se puede observar en el flujómetro analógico incorporado en el tanque de la instalación.

El funcionamiento del indicador de caudal es a través de la relación que hay entre el nivel del agua y el caudal descrito por la función:

$$y = (0.000035 * x - 0.0008 * x + 0.0066 * x - 0.017 * x + 0.015 * x - 0.099) - 0.048$$

Obtenida de forma experimental:

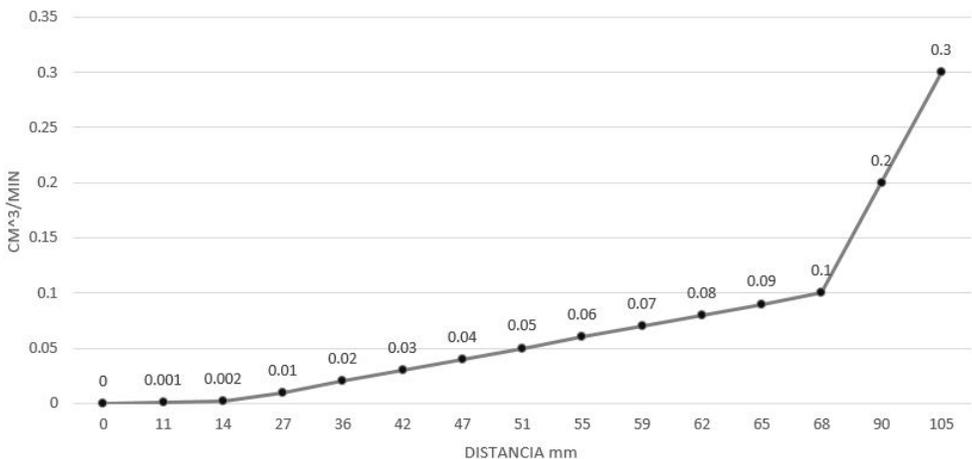


Fig. 5.22 Gráfica de la relación entre caudal y el nivel de agua en tanque.

De esta forma, la lectura del nivel de agua en el tanque es operada por la función y entrega un valor que, al ser multiplicado por una constante, se obtiene la medición del caudal en metros cúbicos por segundo, visualizados en el indicador analógico y digital.

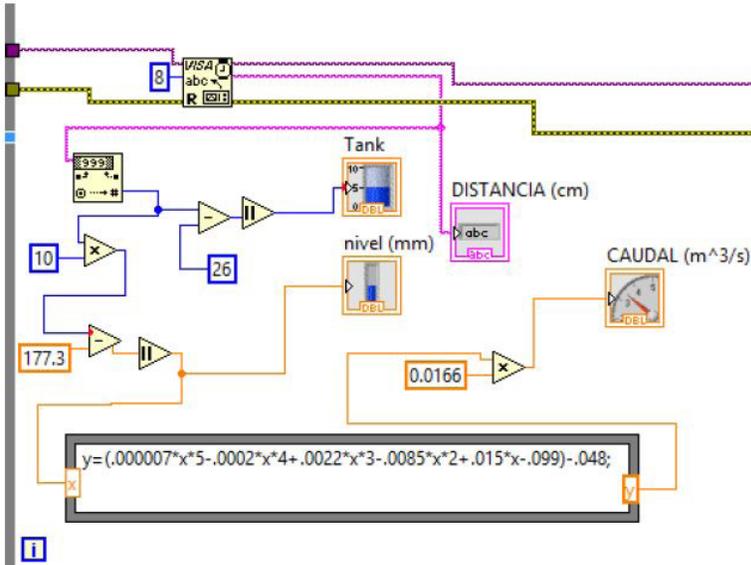


Fig. 5.23 Programación en LabVIEW del indicador de caudal.

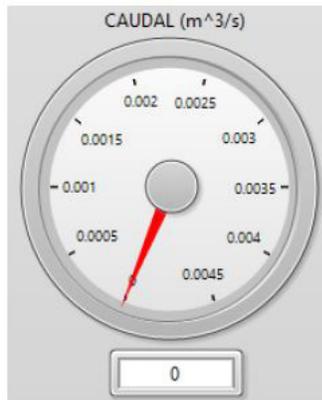


Fig. 5.24 Indicador de caudal en panel frontal.

## 5.6 Indicador de velocidad de flujo

El indicador de velocidad de flujo funciona con la relación entre el área final en la tobera de inyección a la turbina Pelton y el caudal que fluye a través de esta, el área de paso en el inyector es obtenida al procesar la información de la posición del motor a pasos que controla la válvula inyectora.

Para esto se han tomado las medidas del inyector y la distancia que recorre desde la apertura máxima hasta la mínima.

La posición inicial de la válvula inyectora es con el área de paso a su total abertura, conforme la manija de la válvula, accionada con el motor a pasos, cambie de posición, el inyector se desplazará reduciendo el área final.

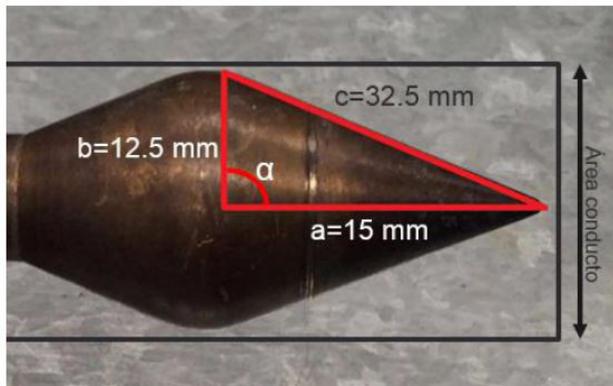
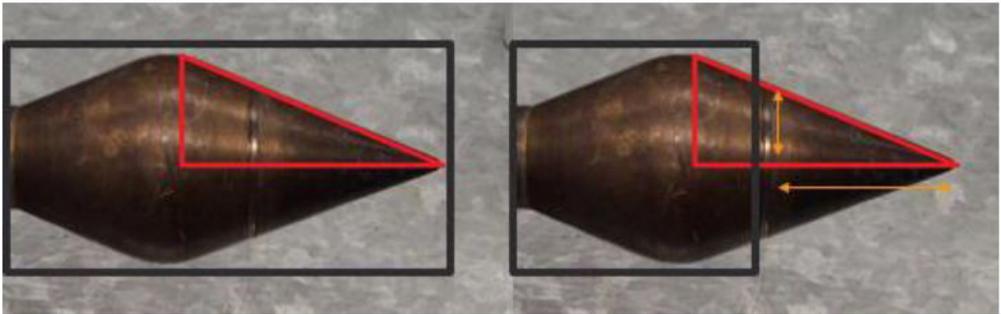


Fig. 5.25 Medidas del inyector.

Las medidas del inyector servirán para obtener la medida del cateto “b” a través del área de conducto al desplazarse linealmente debido al posicionamiento angular del motor a pasos, este cateto “b” es el radio del área del inyector que obstruye el área de paso, y la diferencia entre estas áreas es el área de circulación del fluido que varía con el desplazamiento del inyector.



*Fig. 5.26 Posición inicial y final del inyector a través de la tubería.*

Para esto:

$$\alpha = \tan^{-1}\left(\frac{b}{a}\right) = \tan^{-1}\left(\frac{12.5 \text{ mm}}{15 \text{ mm}}\right) = 39.8055^\circ$$

Se estudió el desplazamiento del inyector al variar la posición angular del motor a pasos y por tanto la variación de la medida del cateto “a” resultando que por cada 240 grados se desplaza el inyector 1 mm, esto quiere decir que, al ingresar 3600 grados, se desplaza en su totalidad el inyector obteniendo la máxima longitud del cateto “a” (15 mm) y por tanto se obtiene el área de obstrucción total.

Dicho lo anterior se puede obtener la longitud del radio “b” a determinada longitud “a”:

$$b = a \tan \alpha = a(0.8333)$$

Ahora es posible calcular el área final de paso del fluido:

$$\begin{aligned} \text{Área}_{\text{paso}} &= \text{Área}_{\text{conducto}} - \text{Área}_{\text{inyector}} \\ \text{Área}_{\text{paso}} &= \pi(0.02\text{m})^2 - \pi(bx)^2 \end{aligned}$$

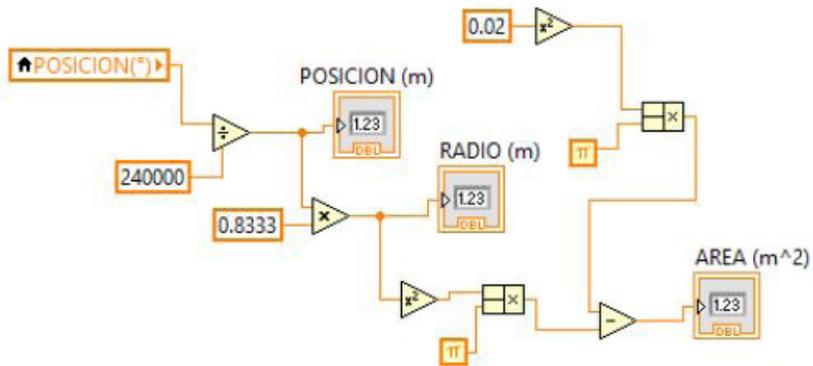


Fig. 5.27 Programación en LabVIEW para la obtención del área en función de la posición del motor a pasos.

Ahora que se tiene la información de la variación del área de paso final se calcula la velocidad del fluido a determinado caudal y se muestra en su respectivo indicador analógico y digital.

$$Vel = \frac{Q}{A} \text{ [m/s]}$$

Donde:

- $Q$ =Caudal [ $m^3/s$ ]
- $A$ =Área de paso del inyector [ $m^2$ ]

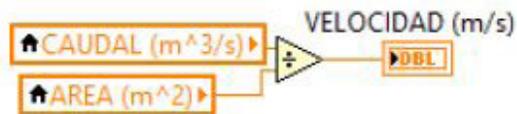


Fig. 5.28 Programación para la obtención de la velocidad de fluido.

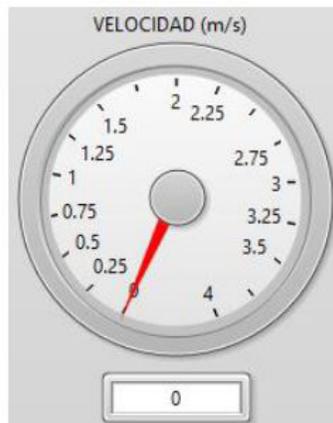


Fig. 5.29 Indicador de velocidad de fluido.

## 5.7 Indicador de presión hidráulica

Debido a la dificultad de conseguir un sensor de presión hidráulica, y con el fin de no modificar drásticamente la instalación, se optó por ingresar al SCA-DA el valor de presión que nos indica el manómetro de Burdon.

Sencillamente se ingresa la lectura de presión del manómetro de forma manual y, como esta lectura está en  $\text{kg/cm}^2$ , se multiplica por la constante 98066.52 para convertir la lectura a pascales (Pa).

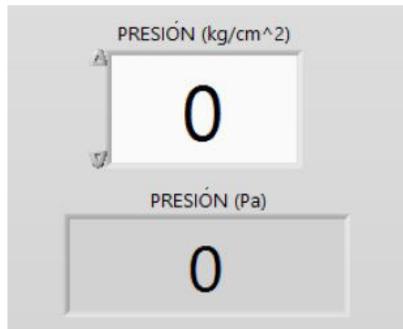


Fig. 5.30 Control e indicador de presión hidráulica en panel frontal.

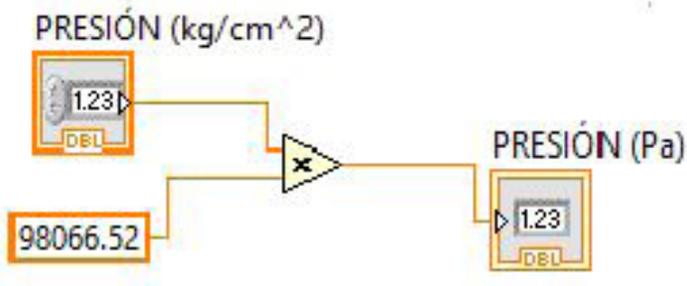


Fig. 5.31 Programación para la obtención de presión en Pascales.

## 5.8 Indicador de carga eléctrica

El indicador de carga eléctrica es sencillamente la aplicación de la ley de Ohm a las lecturas del sensor de voltaje y del sensor de corriente, este indicador nos muestra la impedancia  $Z$  que aparece en el sistema al añadir las cargas conformadas por resistencia y reactancia inductiva, a través de la interfaz, por medio de los relevadores de conmutación.



*Fig. 5.32 Obtención de impedancia e indicador en panel frontal.*

## 5.9 Indicador de factor de potencia

El indicador de factor de potencia está muy relacionado con el indicador de carga eléctrica y la frecuencia de la señal del voltaje producido, hay que aclarar que, aunque las cargas sean fijas ( $Z=200+57.1j \Omega$ ), estas variarán de acuerdo a la frecuencia y al variar las cargas el factor de potencia también.

Este indicador funciona calculando la reactancia inductiva por medio de la frecuencia y la inductancia que es fija (0.15 H) y calculando la componente

real (resistencia) de la impedancia proporcionada por el indicador de carga y el indicador de frecuencia.

$$X_L = 2\pi Lf = 0.94828(f) \text{ } [\Omega]$$

$$R = \sqrt{Z^2 - X_L^2} \text{ } [\Omega]$$

Donde:

- $X_L$  = Reactancia inductiva
- $f$  = frecuencia del Voltaje generado [Hz]
- $L$  = Inductancia [H]
- $R$  = Resistencia [ $\Omega$ ]
- $Z$  = Impedancia [ $\Omega$ ]

Una vez obtenidos estos valores se puede calcular el ángulo entre la componente de la impedancia y la componente resistiva:

$$\theta = \text{atan}\left(\frac{X_L}{R}\right)$$

$$fp = \cos\left(\left|\frac{\theta * 360}{2\pi}\right|\right)$$

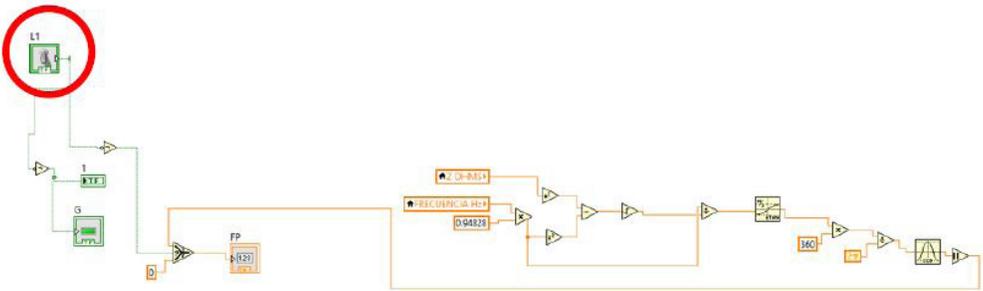


Fig. 5.33 Programación para el cálculo de factor de potencia.

El indicador de factor de potencia en la Fig. 5.35, entra en funcionamiento al activar la primera carga con el relevador L1.



Fig. 5.34 Indicador de factor de potencia en panel frontal.

## 5.10 Indicador de frecuencia

El indicador de frecuencia está ligado a la velocidad angular de la turbina Pelton y a la construcción del alternador, para determinar la frecuencia a la que se genera el voltaje tomamos la ecuación:

$$f = \frac{N2 * P}{120} [Hz]$$

Siendo  $N_2$  la velocidad angular del alternador y  $P$  el número de polos del alternador (4 polos), se obtiene la velocidad angular del alternador usando los diámetros de las poleas de la turbina y el alternador:

$$N_2 = \frac{N_1 D_1}{D_2} = \frac{N_1 * (0.11m)}{(0.04m)} \text{ [RPM]}$$

Donde:

- $D_1$ =Diámetro de polea de turbina [m]
- $D_2$ =Diámetro polea alternador [m]
- $N_1$ =Velocidad angular de turbina pelton [RPM]

$N_1$  corresponde a la velocidad angular de la turbina proporcionada por el sensor de RPM en su correspondiente indicador.

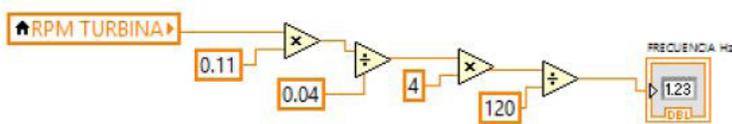


Fig. 5.35 Programación para el cálculo de la frecuencia del voltaje generado.



Fig. 5.36 Indicador de frecuencia en panel frontal.

## 5.11 Indicador de fuerza

Este indicador muestra la fuerza con la que la turbina Pelton está trabajando a determinada carga, lo que hace que aumente o disminuya la fuerza es un aumento o disminución en la corriente suministrada al sistema.

Para el cálculo de la fuerza es necesario contar con el cálculo de la potencia mecánica, el cual se hace de la siguiente manera:

$$F = \frac{Pot_{mecánica} * 60}{2r\pi N1} [N]$$

Donde:

- $F$ =fuerza [N]
- $r$ =Radio de polea de turbina (0.055 m)
- $N1$ =Velocidad angular de turbina [RPM]

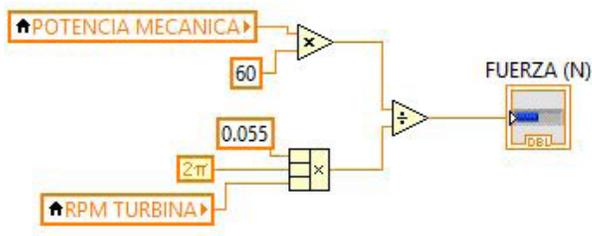


Fig. 5.37 Programación para el cálculo de la fuerza aplicada por la turbina.

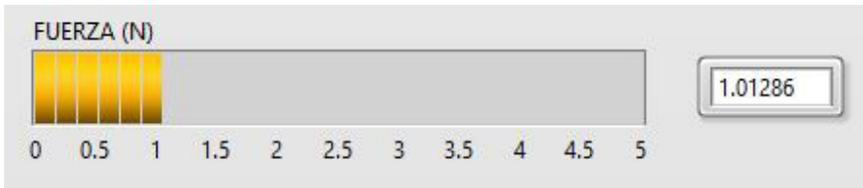


Fig. 5.38 Indicador de fuerza en panel frontal.

## 5.12 Indicador de carga neta

Este indicador proporciona la información sobre la carga neta disponible para realizar trabajo sobre la turbina, en él están involucradas las mediciones y lecturas de los indicadores de presión hidráulica y de velocidad de fluido.

La carga neta se calcula mediante la ecuación:

$$H_n = \frac{P}{\gamma} + \frac{Vel^2}{2g} = \frac{P}{9810 \frac{N}{m^3}} + \frac{Vel^2}{2 \left(9.81 \frac{m}{s^2}\right)} \quad [m]$$

Donde:

- $H_n$ =Carga neta
- $P$ =Carga en manometro  $[N/m^2]$
- $\gamma$ =Peso específico  $[N/m^3]$
- $Vel$ =velocidad de flujo  $[m/s]$
- $g$ =Aceleración de gravedad  $[m/s^2]$

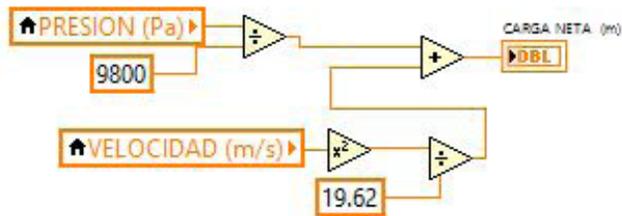


Fig. 5.39 Programación para el cálculo de carga neta.



Fig. 5.40 Indicador de carga neta.

## 5.13 Indicadores de parámetros de planta

### Indicador de potencia eléctrica

Interpretado como el producto de la corriente eléctrica total del sistema por el voltaje de alimentación y el factor de potencia.

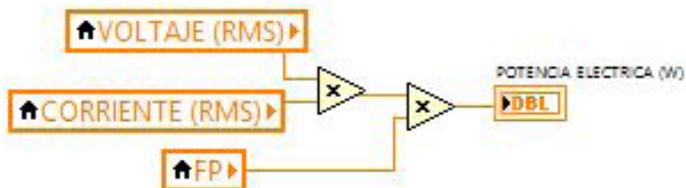


Fig. 5.41 Programación para el cálculo de potencia eléctrica.

### Indicador de potencia mecánica

La potencia mecánica se obtiene de la suma de la potencia eléctrica más las pérdidas eléctricas causadas por efecto Joule:

$$P_m = P_e + I^2 Z \text{ [Watts]}$$

Donde:

- $P_m$ =Potencia mecánica [Watts]
- $P_e$ =Potencia eléctrica [Watts]
- $I$ =Corriente en el circuito [Amperes]
- $Z$ =Impedancia de cargas [ $\Omega$ ]

En las pérdidas por efecto Joule se toma en cuenta la impedancia y no la resistencia porque en este caso al variar la frecuencia del voltaje generado al aumentar la velocidad de la turbina, modifica el valor tanto de la resistencia como el de la reactancia inductiva, incrementando el valor de la reactancia y por tanto las pérdidas son más significativas debido a la parte inductiva por la resistencia de los hilos conductores.

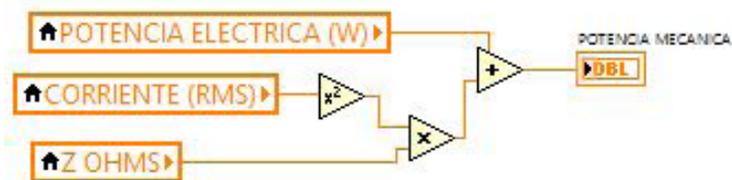


Fig. 5.42 Programación para el cálculo de potencia mecánica.

### Indicador de potencia hidráulica

Es interpretado como el producto de la carga neta, el caudal y el peso específico del agua:

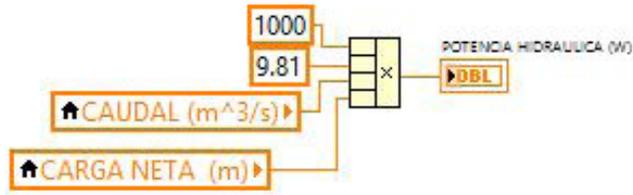


Fig. 5.43 Programación para el cálculo de potencia hidráulica.

### Indicador de eficiencia en turbina

La eficiencia de la turbina está definida como la relación entre la potencia mecánica y la potencia hidráulica:

$$\eta_{turbina} = \frac{P_m}{P_h} * 100 [Adim]$$

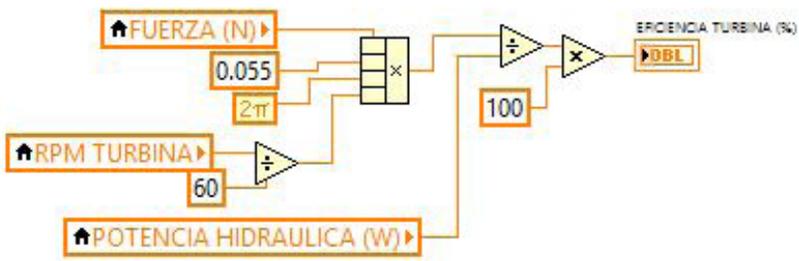


Fig. 5.44 Programación para el cálculo de eficiencia en turbina.

### Indicador de eficiencia del alternador

La eficiencia del alternador se define como la relación entre la potencia eléctrica y la potencia mecánica:

$$\eta_{\text{alternador}} = \frac{P_e}{P_m} * 100 \text{ [Adim]}$$

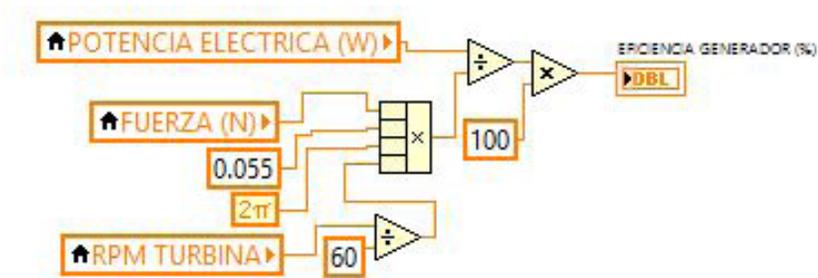


Fig. 5.45 Programación para el cálculo de eficiencia en alternador.

### Indicador de potencia de planta

La potencia de la central hidroeléctrica experimental del laboratorio se define por la siguiente ecuación:

$$P_{\text{planta}} = \rho * g * Q * H_n * \eta_n * \eta_g \text{ [Watts]}$$

Donde:

- $\rho$ =Densidad del fluido [Kg/m<sup>3</sup>]
- $g$ =Aceleración de gravedad [m/s<sup>2</sup>]
- $Q$ =Caudal [m<sup>3</sup>/s]
- $H_n$ =Carga neta [m]
- $\eta_h$ =Eficiencia hidráulica [Adim]
- $\eta_g$ =Eficiencia generador [Adim]

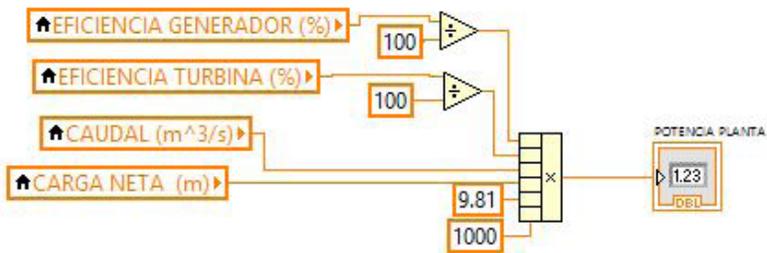


Fig. 5.46 Programación para el cálculo de potencia de planta.

### Indicador de eficiencia de planta

Se define por la relación de la potencia de salida y potencia de entrada, que corresponde a la potencia hidráulica y potencia eléctrica:

$$\eta_{planta} = \frac{P_e}{P_h} * 100 \text{ [Adim]}$$

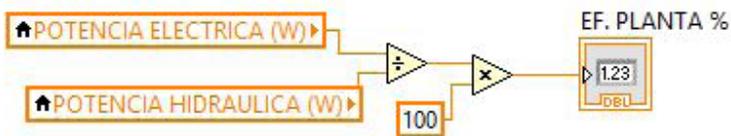


Fig. 5.47 Programación para el cálculo de eficiencia de planta.

Los indicadores de los parámetros de planta están distribuidos en el panel frontal de la interfaz del sistema SCADA de la siguiente manera:

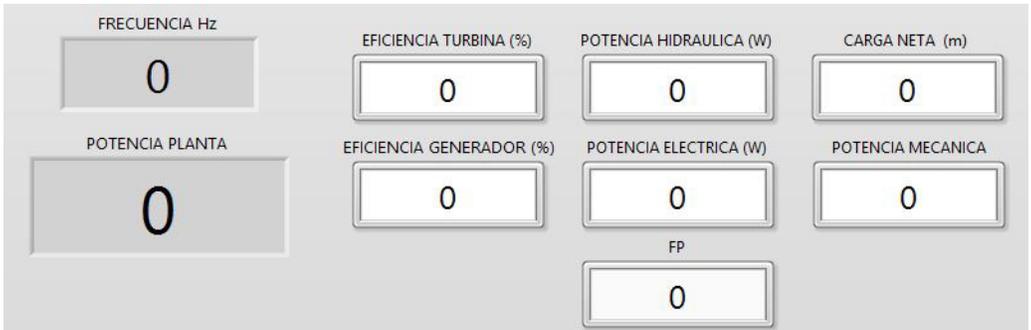


Fig. 5.48 Distribución de los indicadores de parámetros de planta en el panel frontal.

## 5.14 Control motor-bomba

Aunque este control no se haya implementado en el sistema SCADA para no modificar la instalación de la central hidroeléctrica experimental del laboratorio, sí se llevó a cabo el desarrollo del control del motor-bomba mediante la interfaz HMI.

Cabe mencionar que actualmente el control de velocidad en la interfaz es solo indicativo y por tanto no controla nada.

El control de la velocidad del motor de CD se realiza mediante un control de fase, haciendo uso de los rectificadores controlados de silicio (SCR) disparados con ARDUINO.



Para la prueba del control de ángulo de fase, en el circuito de la Fig. 5.51, podemos ver que se usa un transformador con tap central, con el cual obtenemos 12 V CA conectándolo a la red de 120 Vca. En paralelo se conecta el CI H11AA1 que es un optoacoplador con salida de foto transistor de tipo bidireccional, al contar con dos diodos en anti paralelo, su funcionamiento consiste en que, al emitir luz cualquiera de los dos diodos led, el foto transistor estará en conducción y estará en corte cuando ambos leds estén apagados, de esta forma se puede conectar una corriente alterna en el lado primario del optoacoplador, el secundario (foto transistor) se comportará como un interruptor cerrado excepto cuando el voltaje en el primario sea cercana a 0 Volts, por lo que se comportará como un interruptor abierto, así se puede generar un pulso cuando el voltaje en el lado primario del circuito sea cercano a 0 volts, es decir, cuando la señal “pasa por cero”, cambiando del semiciclo positivo al negativo.

El pulso de la señal proporcionado por el foto transistor será lo que propicie las interrupciones en la placa ARDUINO a través del pin digital 2, que corresponde a la interrupción cero.

Una vez iniciada la interrupción por la detección de la señal al cruzar por cero, se inicia un contador de tiempo que se compara con el tiempo corres-

pendiente al del ángulo de fase deseado, cuando estos tiempos sean iguales, el microcontrolador encenderá los SCR manteniendo un voltaje en GATE, el SCR se apagará si el voltaje en GATE es cero la próxima vez que la señal CA cruce por cero.

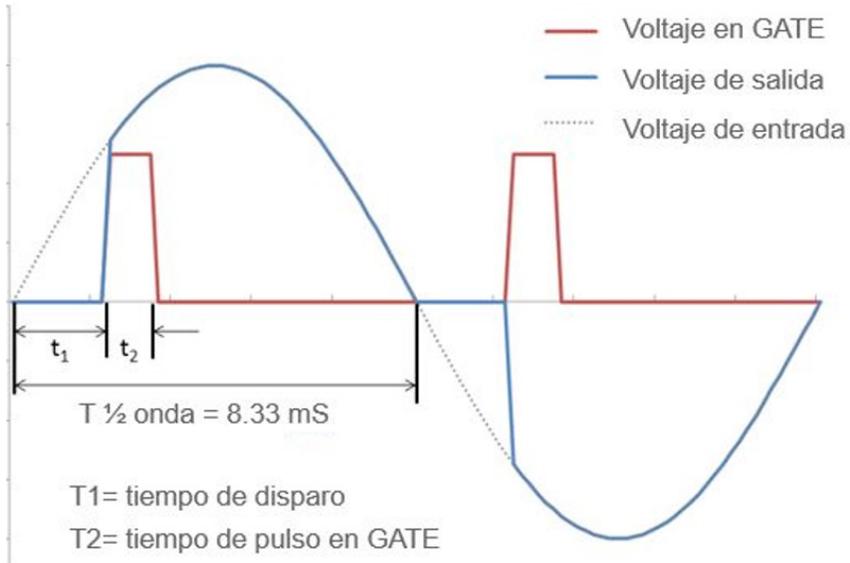


Fig. 5.49 Disparo de SCR utilizando detector de cruce por cero.

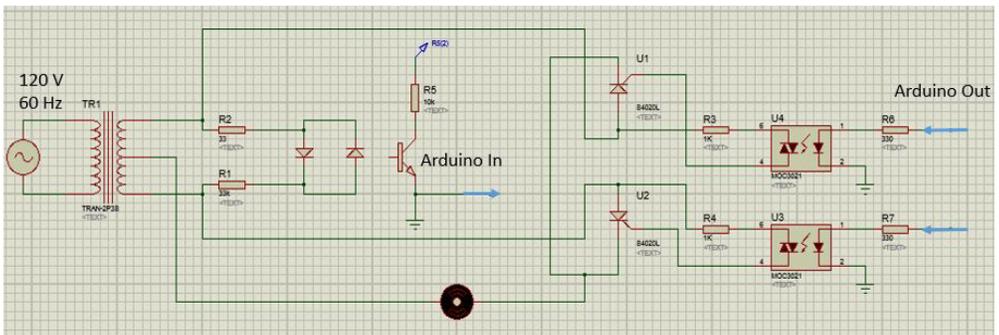


Fig. 5.50 Circuito del control de ángulo de fase con detector de cruce por cero con ARDUINO.



El control de ángulo se realiza por medio de la placa ARDUINO UNO, ya que la tarjeta NANO tiene cargado el programa “LIFA Base” para la comunicación entre la placa y el software LabVIEW.

El programa en ARDUINO UNO comienza con la verificación de la comunicación serie, si la comunicación es viable convierte en entero el valor del primer número del buffer serial con formato string y lo almacena en la variable “a”, los valores a ingresar para el ángulo de control de fase son entre 483 para menor potencia y 64 para mayor potencia, se imprime el valor con la abreviación de unidades “uni”.

Al ocurrir las interrupciones se ejecuta la función “void zeroCrossingInterrupt” en el que compara el valor de ángulo de fase ingresado por el monitor serie y al ser iguales pone en alto (HIGH) los pines digitales 9 y 10, que corresponden al GATE de los SCR 1 Y 2, y se establece el ancho del pulso, después se desactiva la señal GATE de ambos SCR al pasar por cero la señal de alterna y se deshabilita el temporizador.

```

#include <avr/io.h>
#include <avr/interrupt.h>
#define DETECT 2
#define GATE1 9
#define GATE2 10
#define PULSE 4
int a;

void setup() {
  pinMode(DETECT, INPUT);
  digitalWrite(DETECT, HIGH);
  pinMode(GATE1, OUTPUT);
  pinMode(GATE2, OUTPUT);
  OCR1A = 100;
  TIMSK1 = 0x03;
  TCCR1A = 0x00;
  TCCR1B = 0x00;
  attachInterrupt(0, zeroCrossingInterrupt, RISING);
}

void motordc() {
  if(Serial.available()){
    a=Serial.parseInt();
    OCR1A=a; //escribe entre 483 y 64 para ajustar
  }
  Serial.print(a);
  Serial.println(" uni");
}

void zeroCrossingInterrupt(){
  TCCR1B=0x04;
  TCNT1 = 0;
}

ISR(TIMER1_COMPA_vect){
  digitalWrite(GATE1,HIGH);
  digitalWrite(GATE2,HIGH);
  TCNT1 = 65536-PULSE;
}

ISR(TIMER1_OVF_vect){
  digitalWrite(GATE1,LOW);
  digitalWrite(GATE2,LOW);
  TCCR1B = 0x00;
}

```

*Fig. 5.51 Código en ARDUINO para la implementación de detector de cruce por cero y disparo de SCR.*

La programación del control de fase en LabVIEW funciona con un slide con escala entre 64 y 483 para indicar la potencia, este número entra a la función “Number to String” y el botón “to pump” confirma la ejecución del ciclo “case” que realiza la escritura de la información a la placa ARDUINO, la función “Visa Read” lee la información que se le indicó a través de la interfaz con sus unidades, este valor se convierte en un número entero a través de la función “Number to String”.

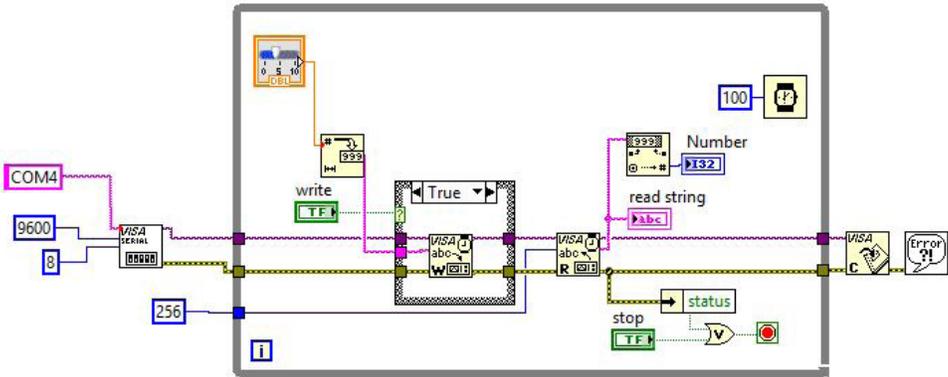


Fig. 5.52 Programación del control de velocidad de motor de CD.

El diseño de la interfaz del control de la bomba es el siguiente:

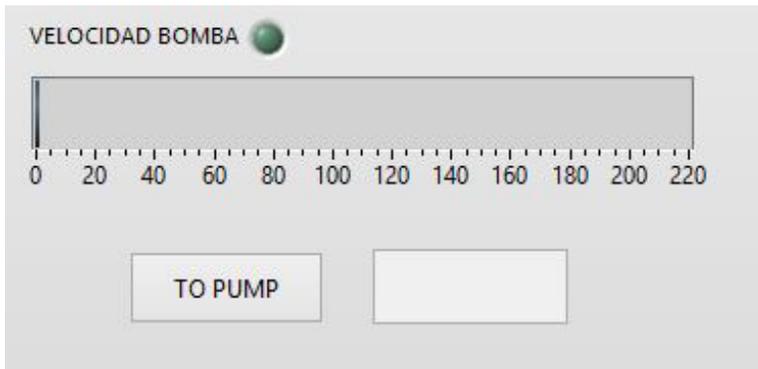


Fig. 5.53 Control de velocidad de motor de DC en el panel frontal.

### Prueba del control de ángulo de fase.

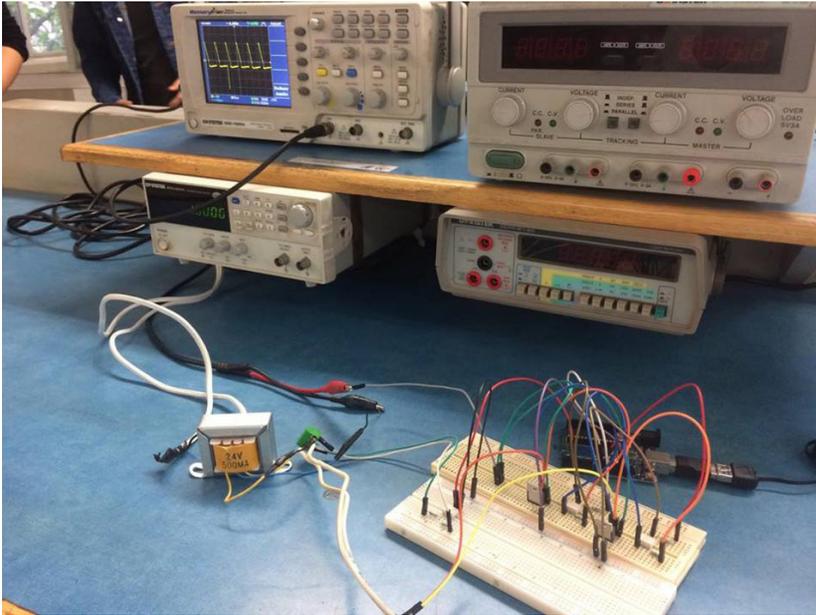


Fig. 5.54 Prueba de control de ángulo de fase en laboratorio de Facultad de Ingeniería C.U.

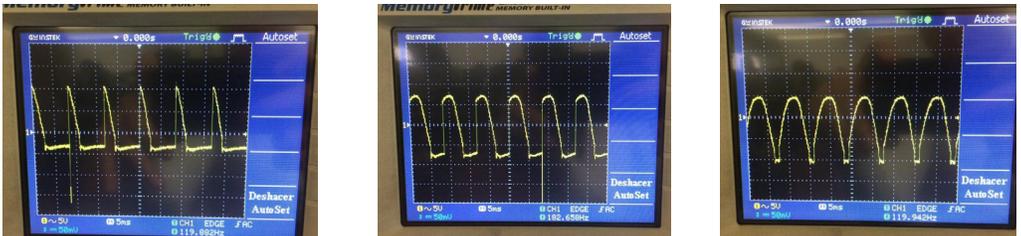
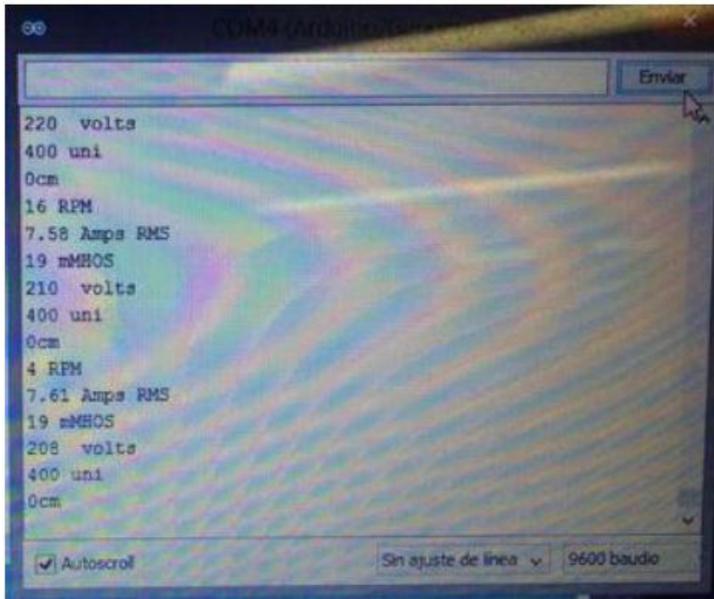


Fig. 5.55 Señales obtenidas con el control implementado.



*Fig. 5.56 Lectura y escritura para el control de ángulo de fase.*

## 5.15 Control motor a pasos (inyector)

El control del motor a pasos comienza con un bloque de función que pertenece a la librería de ARDUINO llamado “steper configure” en que se configuran los pines destinados al control del motor y el método de control del motor.

En este trabajo se utiliza un motor a pasos bipolar que cuenta con 4 cables de control y por tanto el método de control en el bloque “steper configure” será por “4wire”, y los pines de control de ARDUINO corresponden del 4 a las 7 digitales.

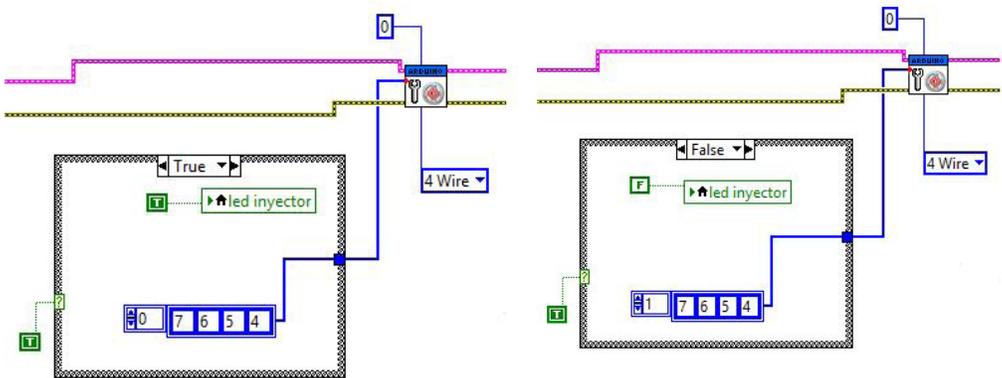


Fig. 5.57 Configuración de pines de control de motor a pasos.

Podemos observar en la Fig. 5.58 que los pines de control de ARDUINO están dentro de un “case structure”, este ciclo permite controlar el sentido de giro del motor, en el “case selector” se observa una constante booleana en “true” para ambos casos, en lugar de esta constante se puede utilizar un switch que cambie el sentido de giro, pero para evitar poner demasiados controles se optó por la constante booleana, lo que nos permite hacer el control de giro al agregar el signo menos (-) a la cantidad de grados que girará el motor.

El control de la posición y la velocidad del motor a pasos se realiza mediante la utilización de otro case structure, en el cual se puede ver el bloque función que escribe en ARDUINO el número de pasos a avanzar, la velocidad de giro en pasos por segundo y un bloque delay.

La posición se controla a través del indicador de ángulo, se ingresan los grados que se quiere que se desplace el motor y este número es multiplicado con la constante 1.111 que corresponde a los pasos por grado que avanza el motor, debido a que tenemos un motor de 400 pasos por vuelta.

El dato de la multiplicación de los pasos por grados también es multiplicado por el valor ingresado en el slide de control de velocidad (pasos por segundo) y la información es ingresada a la función "steper Write".

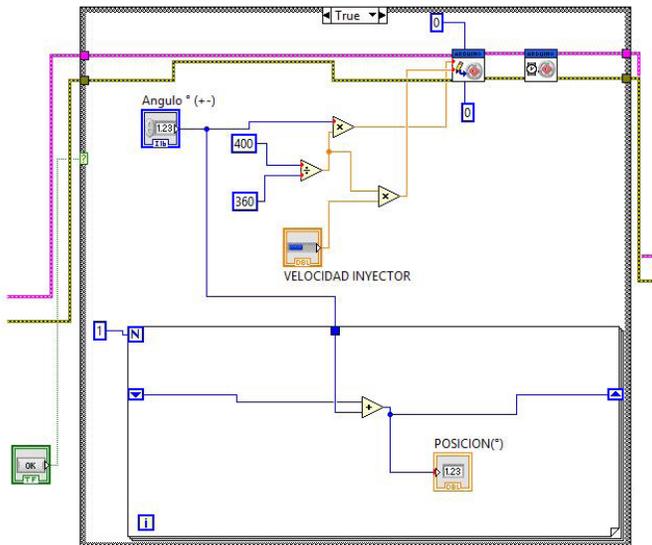
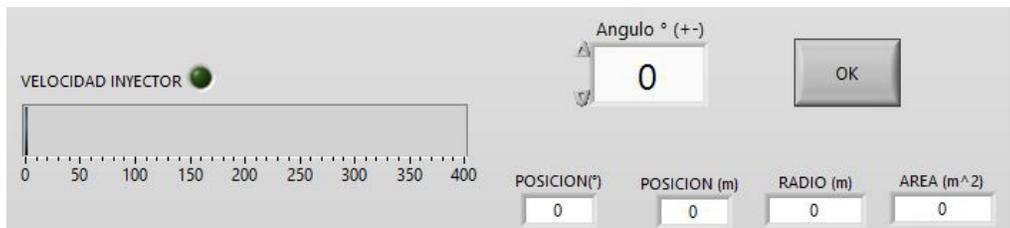


Fig. 5.58 Programación para el control de posición y velocidad del motor a pasos.

Por último, el indicador de posición, al tener la opción de giro en ambos sentidos, es necesario llevar un contador sumador de grados que avanza y retrocede el motor, por esta razón el control de giro utilizando el signo menos (-) es más útil que un switch, ya que permite realizar la suma de números con diferente signo y llevar la cuenta de la posición del motor, que se realiza dentro de un ciclo “for”.

La interfaz gráfica para el control del motor a pasos es de la siguiente manera:



*Fig. 5.59 Control de velocidad y posición del motor a pasos en el panel frontal.*

## 5.16 Control de cargas

El control de las cargas que se acoplarán al alternador se lleva a cabo mediante el uso de relés de conmutación activados y desactivados por el microcontrolador ARDUINO NANO mediante la interfaz HMI del sistema SCADA.

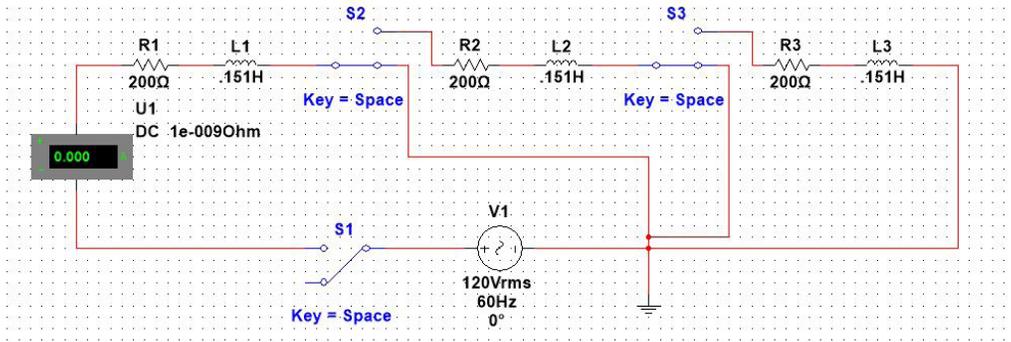


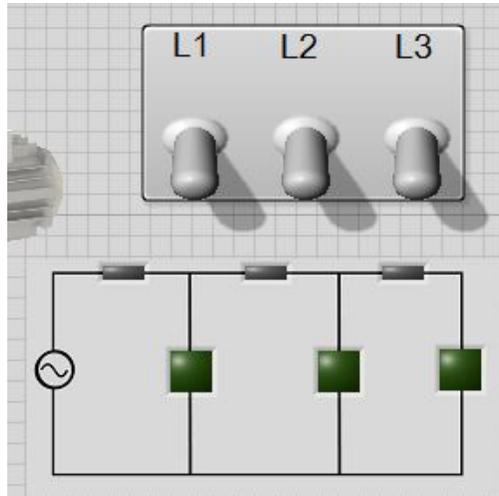
Fig. 5.60 Circuito de conexión de relevadores de conmutación y cargas al alternador.

Al activar el relé S1 se cierra el circuito formando una malla con una impedancia compuesta de carga resistiva e inductiva, así como del sensor de corriente, en el relé S1 el conector común está conectado a la línea del alternador, el conector normalmente cerrado no se conecta a nada, lo que por default deja abierto el circuito hasta que se energice la bobina del relé y cambie a la posición del conector normalmente abierto que cerraría el circuito.

En los relevadores S2 y S3 el conector normalmente cerrado está conectado al neutro del alternador para cerrar los circuitos al activar cada relevador, es decir, al activar el relé S1 el relevador S2 está normalmente cerrado al neutro del alternador y se tiene una malla de una carga, al activar el relevador S2 el selector cambia de posición a normalmente abierto y ahora se tiene una malla de dos cargas al estar el relevador S3 normalmente cerrado al neutro del alternador.



La interfaz para el control de las cargas y el esquema de estas es representada de la siguiente manera:



*Fig. 5.62 Control de cargas eléctricas e indicador.*

## 5.17 Video vigilancia

Para el monitoreo visual en tiempo real de la instalación se acopla una cámara web, al tener conexión USB facilita su uso y adaptación al sistema SCADA.

La programación para la adquisición de video se puede realizar instalando la herramienta "Vision and Motion" para LabVIEW, una vez instalada se utiliza la herramienta "visión Acquisition" y se configura el puerto USB de la cámara y el tamaño de la imagen, así como el tipo de adquisición de imagen que en este caso es continuo a través del ciclo "while loop".

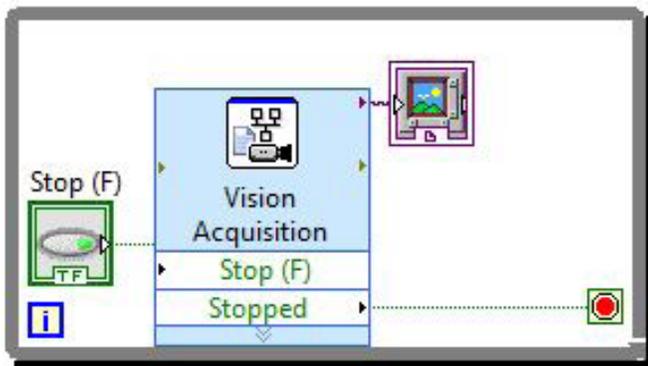
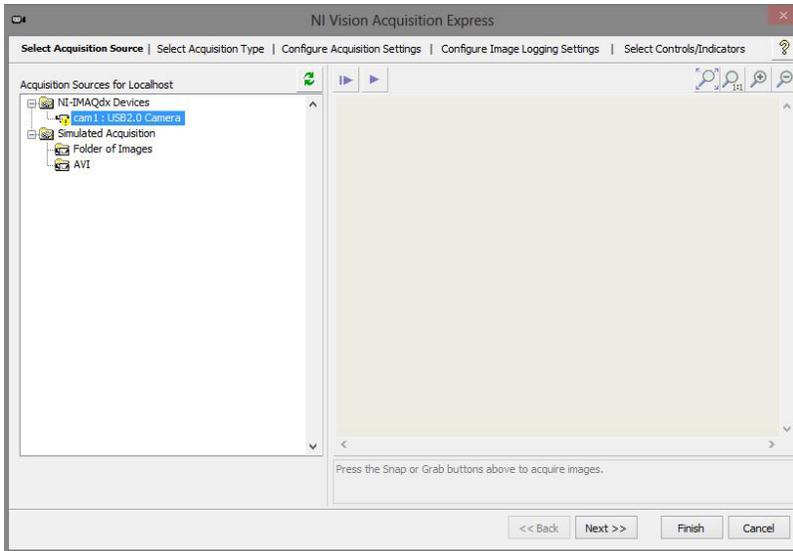


Fig. 5.63 Configuración y diagrama de bloques para la adquisición de video.



*Fig. 5.64 Monitor de video en panel frontal.*

## **5.18 Animación de SCADA**

La animación del esquema de los elementos principales de los que se compone la instalación de la central hidroeléctrica experimental del laboratorio es meramente estético ya que brinda una mejor experiencia visual.

Las imágenes para la animación fueron realizadas en un programa 3D (SOLIDWORKS) para darle mayor calidad, ya que los elementos disponibles en las herramientas DSC MODULE (imagen navigator) son estéticamente inadecuadas y de menor calidad.

Para realizar la animación del motor y la turbina se crearon dos controles CUSTOM:

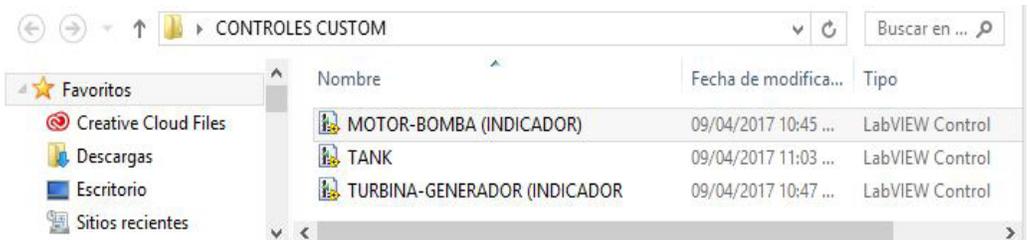


Fig. 5.65 Archivos de controles custom.

Una vez creados los dos controles custom se pueden usar para crear una intermitencia ajustada a determinado tiempo para crear la percepción de movimiento (animación).

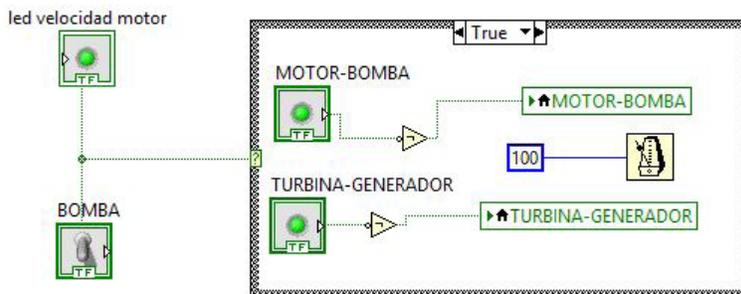
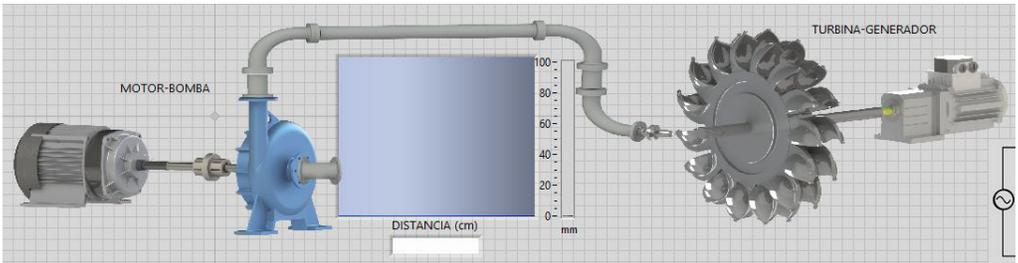


Fig. 5.66 Programación de animación en el esquema de central hidroeléctrica.

La animación dará inicio al encender la bomba con el switch control (BOMBA) que controla el ciclo "case structure".



*Fig. 5.67 Esquema animado de la instalación en el panel frontal.*



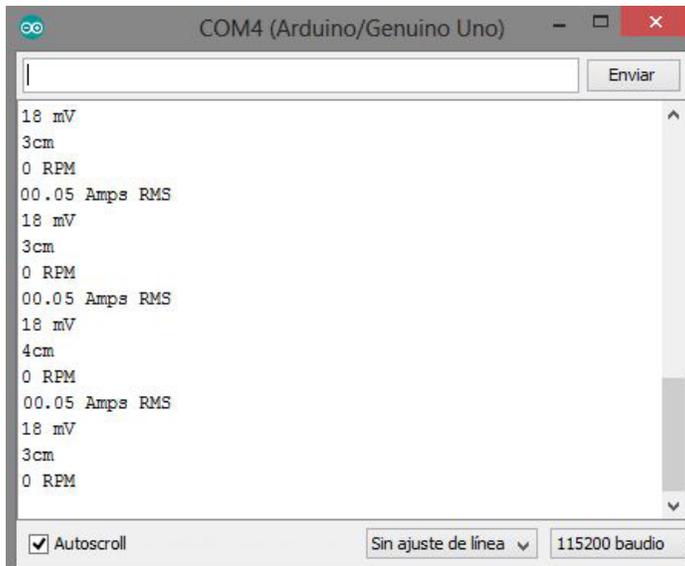
## Capítulo 6

### Prueba y resultados

La prueba de verificación del funcionamiento del sistema SCADA para la instalación de la central hidroeléctrica experimental consta de los siguientes pasos:

1. Verificación de la programación de los sensores de la tarjeta ARDUINO UNO.
2. Conexión del hardware a la instalación.
3. Inicialización del sistema SCADA.
4. Análisis de resultados.

En el paso número uno, es necesario confirmar el funcionamiento de los sensores (sin conectarse a la planta) y verificar que están captando información y no arrojen datos anormales. Esto se verifica ejecutando el programa en ARDUINO y observando a través del monitor serie los datos que obtiene la placa a través de los sensores:



*Fig. 6.1 Prueba de adquisición por monitor serie de todos los sensores.*

Como era de esperarse, las lecturas de los sensores al no estar recibiendo información de la planta muestran valores muy bajos, normales y constantes. Ahora que se ha confirmado el correcto funcionamiento de los sensores se procede a montar el resto de la instalación con ayuda del siguiente esquema:

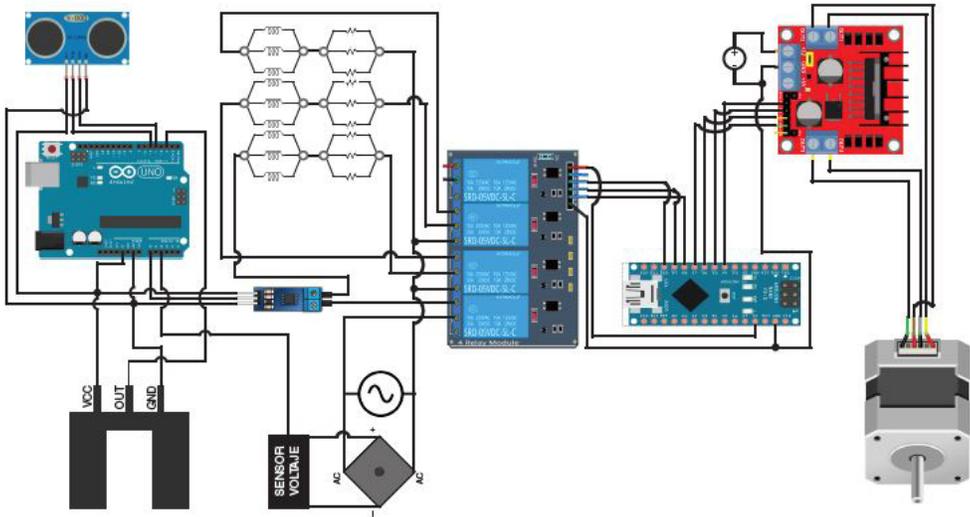


Fig. 6.2 Esquema de conexión de interfaces del sistema SCADA (Véase anexo).

La instalación queda de la siguiente manera:

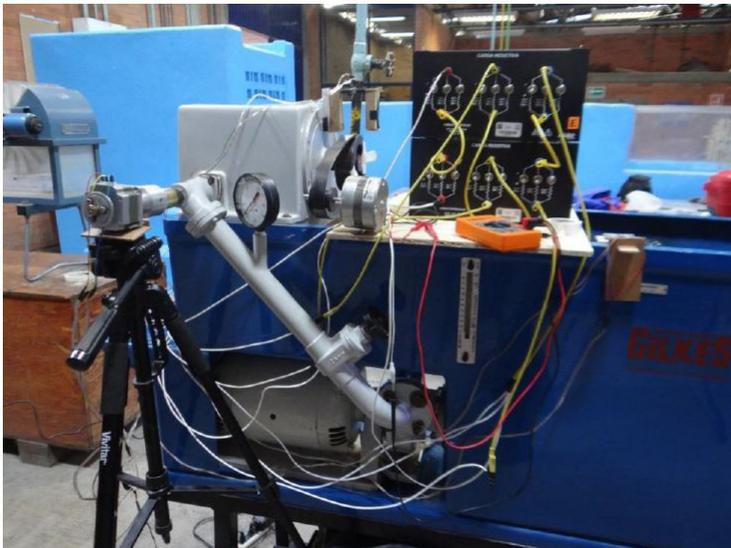


Fig. 6.3 Instalación con elementos e interfaces para la aplicación de sistema SCADA.



*Fig. 6.4 Vista completa de la planta y estación maestra.*



*Fig. 6.5 Vista trasera de la planta con Unidad Terminal Remota.*

Ahora se procede con el acoplamiento del sistema SCADA a la planta hidroeléctrica:



Fig. 6.6 Captura de la interfaz gráfica durante la prueba.

Con ayuda del reporte en Excel creado por el sistema SCADA durante la prueba se pueden analizar los datos y compararlos para determinar la precisión del software.

## 6.1 Análisis de resultados

Con las mediciones reales promedio al momento de realizar las lecturas del sistema SCADA en determinado instante de tiempo se pueden calcular todos los parámetros reales de la planta para compararlos con los de la prueba.

MAGNITUD	VALOR
VOLTAJE	44 V
CORRIENTE	.21 A
RPM TURBINA	2300 RPM
CAUDAL	.00333 m <sup>3</sup> /s
CARGA Z	210 Ohms
Fp	.99

$$Vel = \frac{Q}{A} = \frac{0.0033 \frac{m^3}{s}}{0.001257 m^2} = 2.62 m/s$$

$$H_n = \frac{P}{\gamma} + \frac{Vel^2}{2g} = \frac{58839.912 \frac{N}{m^2}}{9810 \frac{N}{m^3}} + \frac{\left(2.62 \frac{m}{s}\right)^2}{2 * 9.81 \frac{m}{s^2}} = 6.33 m$$

$$P_h = \gamma * Q * H_n = 9810 \frac{N}{m^3} * 0.00333 \frac{m^3}{s} * 6.33 m = 206.97 W$$

$$P_e = V * I * Fp = 44V * 21A * .99 = 9.21 W$$

$$P_m = P_e + I^2 Z = 9.21 W + (.21A)^2 (210 \Omega) = 18.471 W$$

$$F = \frac{P_m * 60}{r2\pi n} = \frac{18.471W * 60}{0.055m * 6.28 * 2300RPM} = 1.395 N$$

$$\eta_t = \frac{P_m}{P_h} = \frac{18.471 W}{206.97 W} * 100 = 8.92\%$$

$$\eta_g = \frac{P_e}{P_m} = \frac{9.21W}{18.471W} * 100 = 49.86\%$$

$$Pot_{planta} = \gamma * Q * H_n * \eta_t * \eta_g = 9810 \frac{N}{m^3} * 0.0033 \frac{m^3}{s} * 6.33 m * 0.0892 * 0.4986 = 10.12W$$

$$\eta_{planta} = \frac{P_e}{P_h} = \frac{9.21W}{206.97W} * 100 = 4.44\%$$

A continuación se presenta una tabla con la comparación de las propiedades de cada variable, el análisis evalúa la medición del valor real contra el valor promedio y contra el valor a una lectura teniendo en cuenta que:

$$\text{Precisión} = \text{desviación estándar} = \sqrt{\frac{\sum_i (X_i - \bar{X})^2}{n - 1}}$$

$$\text{Exactitud} = \text{error absoluto} = \text{valor medido} - \text{valor real}$$

$$\% \text{ Exactitud} = 100 - \text{error relativo} = 100 - \left( \frac{|\text{error absoluto}|}{\text{valor real}} * 100 \right) [\%]$$

$$\% \text{ Desviación} = 100 - \text{error relativo} = 100 - \left( \frac{\text{Precisión}}{\text{valor real}} * 100 \right) [\%]$$

Basado en los datos del reporte y los cálculos de los de los valores reales:

VARIABLE	Valor real	Desviación (s)	% Precisión	Promedio	Error absoluto	% Exactitud	Unal lectura	Error absoluto	% Exactitud
ÁREA (m <sup>2</sup> )	0.001257	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00
NIVEL (mm)	97.3	5.07	94.79	91.99	-5.31	94.54	97.30	0.00	100.00
RPM TURBINA	2300	59.73	97.40	2315.63	15.63	99.32	2400.00	100.00	95.65
FUERZA (N)	1.395	0.76	45.57	1.87	0.47	65.96	1.21	-0.18	86.86
VELOCIDAD (m/s)	2.62	0.28	89.46	2.65	0.03	98.67	2.99	0.37	85.98
PRESIÓN (Pa)	58839.912	0.00	100.00	58839.91	0.00	100.00	58839.91	0.00	100.00
CAUDAL (m <sup>3</sup> /s)	0.00333	0.00	90.30	0.00	0.00	97.43	0.00	0.00	87.27
CORRIENTE (A)	0.21	0.03	87.70	0.23	0.02	90.03	0.21	0.00	100.00
VOLTAJE (V)	44	13.94	68.31	60.91	16.91	61.56	46.34	2.34	94.68
FRECUENCIA Hz	210	5.65	97.31	212.61	2.61	98.76	220.00	10.00	95.24
CARGA ELÉCTRICA	210	35.40	83.14	261.47	51.47	75.49	220.48	10.48	95.01
POTENCIA TURBINA (W)	206.97	21.24	89.74	207.63	0.66	99.68	232.66	25.69	87.59
POTENCIA ELÉCTRICA (W)	9.21	3.39	63.17	10.06	0.85	90.74	7.02	-2.19	76.19
EFICIENCIA TURBINA (%)	8.92	4.31	51.71	12.45	3.53	60.43	7.20	-1.72	80.66
EFICIENCIA GENERADOR (%)	49.86	5.58	88.81	38.88	-10.98	77.98	41.92	-7.94	84.07
POTENCIA MECÁNICA (W)	18.471	8.13	55.99	24.66	6.19	66.50	16.74	-1.73	90.63
CARGA NETA (m)	6.33	0.08	98.69	6.38	0.05	99.26	6.32	-0.01	99.81
FACTOR POTENCIA	0.99	0.74	25.66	0.08	-0.91	8.43	0.72	-0.27	72.90
EFICIENCIA PLANTA %	4.44	1.81	59.12	4.79	0.35	92.02	3.02	-1.42	67.93
POTENCIA PLANTA (W)	10.12	4.20	58.49	10.28	0.16	98.43	7.00	-3.12	69.19

Fig. 6.7 Análisis de datos obtenidos durante la prueba.

De la tabla de la figura 6.7 se distinguen varios aspectos de cada variable que compone el sistema SCADA, en la tabla se observa el valor real de cada variable, contra este valor se comparan la exactitud y porcentaje de exactitud del valor promedio de todas las lecturas de cada variable durante toda la prueba así como la exactitud y porcentaje de exactitud del valor de cada variable en un instante de tiempo.

La precisión del valor leído de cada variable se define como la dispersión del conjunto de valores obtenidos de mediciones repetidas de la variable en cuestión, cuanto menor es la dispersión mayor es su precisión. En la figura 6.7 se presenta la desviación de las variables con sus correspondientes unidades y su porcentaje de precisión.

La exactitud de los valores leídos de cada variable se define como la cercanía al valor real medido, cuanto menos se aleje del valor real más exacta es la medición. Para determinar la exactitud se calcula el error absoluto, que refleja la lejanía o cercanía al valor real. El porcentaje de exactitud de la variable en cuestión se determina realizando la resta de 100% menos el error relativo, que es el porcentaje de lejanía o cercanía del valor real.

VARIABLE	% Precisión	% Exactitud en promedio	% Exactitud a una lectura
ÁREA (m <sup>2</sup> )	100.00	100.00	100.00
NIVEL (mm)	94.79	94.54	100.00
RPMTURBINA	97.40	99.32	95.65
FUERZA (N)	45.57	65.96	86.86
VELOCIDAD (m/s)	89.46	98.67	85.98
PRESIÓN (Pa)	100.00	100.00	100.00
CAUDAL (m <sup>3</sup> /s)	90.30	97.43	87.27
CORRIENTE (A)	87.70	90.03	100.00
VOLTAJE (V)	68.31	61.56	94.68
FRECUENCIA Hz	97.31	98.76	95.24
CARGA ELÉCTRICA	83.14	75.49	95.01
POTENCIA TURBINA (W)	89.74	99.68	87.59
POTENCIA ELÉCTRICA (W)	63.17	90.74	76.19
EFICIENCIA TURBINA (%)	51.71	60.43	80.66
EFICIENCIA GENERADOR (%)	88.81	77.98	84.07
POTENCIA MECÁNICA (W)	55.99	66.50	90.63
CARGA NETA (m)	98.69	99.26	99.81
FACTOR POTENCIA	25.66	8.43	72.90
EFICIENCIA PLANTA %	59.12	92.02	67.93
POTENCIA DE PLANTA (W)	58.49	98.43	69.19
	77.27	83.76	88.48
		86.12	

80-100 %	ACEPTABLE
50-79 %	MEDIANAMENTE ACEPTABLE
0-49 %	NO ACEPTABLE

Fig. 6.8 Porcentaje de aceptación en precisión y exactitud de cada variable



## Capítulo 7

### Conclusiones

Para dictar el grado de aptitud del sistema SCADA desarrollado para la central hidroeléctrica experimental del laboratorio de Termofluidos se toman en cuenta los siguientes aspectos:

- Precisión
- Exactitud
- Costo del equipo
- Instalación
- Seguridad

Con base a esto, se obtendrá un resultado en porcentaje de la evaluación de cada aspecto, y así obtener un porcentaje promedio que dicte la aptitud del sistema con el criterio antes usado (80-100% aceptable, 50-79 % medianamente aceptable, 0 50 % no aceptable).

Los dos aspectos más importantes son la precisión y exactitud del sistema, en la figura 6.8 se muestra el porcentaje de aceptación en cuanto a precisión y exactitud de cada variable del sistema, utilizando el criterio antes mencionado.

El sistema SCADA refleja una precisión del 77.27 %, esto se debe a dos variables con problemas de mediciones muy desviadas (Fuerza y factor de potencia), en el caso de la variable fuerza, rastreando el error en el programa, concluimos que se debe a la imprecisión de la variable de potencia mecánica al fallar el indicador de carga eléctrica, y por tanto a la falla del indicador de potencia eléctrica debido a la imprecisión de la variable Voltaje, la cual es la falla de origen, esto se refleja en la tabla 6.8 al notar su precisión de 68.31%, esto se debe a la construcción del sensor ya que no es de manufactura especializada.

En el caso de la variable de Factor de potencia su falla se debe a un retraso en el sistema a la hora de realizar los cálculos y por tanto no indica el resultado correcto, sino el perteneciente a otros datos de otro instante de tiempo. Esto se comprobó al extraer la programación del factor de potencia del programa SCADA haciéndolo un único programa, ingresando los datos de una lectura en un instante de tiempo se obtuvo un valor diferente del registrado en el reporte Excel. La programación aislada del factor de potencia presenta una respuesta acertada, lo que indica que en el programa completo del sistema SCADA no trabaja con la suficiente rapidez como para arrojar datos precisos y exactos.

La evaluación de la exactitud del sistema SCADA se determinó tomando la

exactitud de las lecturas promedio y la exactitud de una lectura y realizando un promedio entre estas dos.

Usando el mismo criterio, y tomando como indeseables los indicadores rojos mostrados en la figura 6.8, (por debajo del 50 %) nos damos cuenta de que la exactitud del sistema SCADA tiene un resultado del 86.12% que es bastante aceptado y que sería mayor de no ser por el resultado de la variable de factor de potencia, que como se mencionó antes en la evaluación de precisión su falla se debe a la velocidad del programa para realizar los cálculos. Por tanto, el resultado de su exactitud (8.43%) afecta a la evaluación de exactitud del sistema SCADA.

La evaluación del costo del equipo está relacionada con su desempeño en cuanto a precisión y exactitud así como la versatilidad para modificar los programas para proyectos diferentes y la amplia gama de sensores que se le pueden adaptar.

Al observar la tabla de la figura 6.8 podemos notar que predomina el estado aceptable de la lectura de las variables, lo que se traduce como un desempeño aceptable al relacionarlo con el costo del equipo y teniendo en cuenta que no es un equipo profesional ni especializado, sino un equipo didáctico. El costo del equipo no supera \$500.00.

El montaje del equipo y acoplamiento a la instalación del laboratorio toma un tiempo prolongado, lo que requiere mucha anticipación si se quiere realizar una práctica o demostración, ya que toma alrededor de 1 hora montar y adaptar todos los instrumentos.

Si se observa la imagen de la figura 6.2, en el esquema del circuito de sensores y las figuras 6.3 a 6.5 (cableado en instalación), se aprecia la cantidad de conexiones que hay que realizar y la probabilidad de hacer conexiones erróneas es elevada, esto implica que la instalación deba ser realizada por personal con conocimiento especializado o con experiencia en circuitos eléctricos-electrónicos. Por lo antes mencionado, la evaluación en instalación tendrá un puntaje medio-alto.

En cuanto a seguridad, se refiere al daño que pueda sufrir la instalación o alguno de sus elementos debido al diseño de esta haciéndola más o menos segura. Dos aspectos que se deben considerar a la hora de ejecutar el sistema SCADA es la cantidad de cables que cuelgan en diferentes partes de la instalación, esto puede provocar que alguien se tropiece y jale algún cable desconectando o tirando algún sensor o la RTU, el otro aspecto es que la base que soporta las cargas y al alternador no es muy estable, ya que al momento de arrancar la planta, la vibración del alternador hace que se deslice la base y pueda desacoplarse el alternador de la turbina, esto debido a que

aún es un prototipo y las medidas de seguridad no fueron tomadas en cuenta en ese momento. Una actualización en cuanto al procedimiento de montaje, medidas de seguridad y operación de la planta son un área de oportunidad disponible para la continuación del proyecto ya que no se abordará durante éste. Con lo antes mencionado la seguridad del sistema SCADA obtendrá una puntuación media-alta.

Resumiendo, la evaluación de los aspectos a considerar en la aptitud del sistema SCADA tenemos:

PRECISIÓN	77.27%
EXACTITUD	86.12%
COSTO	100%
INSTALACIÓN	70%
SEGURIDAD	80%
<b>APTITUD</b>	<b>82.70%</b>

Una vez determinado el grado de aptitud del sistema SCADA desarrollado, se realiza una inspección de los objetivos propuestos:



- Examinar y evaluar la instalación actual de la planta. ✓

Durante el capítulo 4.1 “Instalación actual” se hizo una revisión de los componentes de la instalación para posteriormente en el capítulo 4.2 “Modificaciones a la planta” determinar los componentes nuevos necesarios para el correcto funcionamiento del sistema SCADA.

- Diseñar la interfaz entre los dispositivos de campo y el software. ✓

Durante el capítulo 5 se profundizó acerca de las particularidades de cada sensor y variable y la interfaz necesaria (circuito electrónico) que facilitan el envío, recepción y procesamiento de información en las placas ARDUINO para ser visualizadas en la HMI del software. Tal es el caso del sensor de voltaje que requiere un circuito electrónico, el control del motor a pasos o el sensor de velocidad angular.

- Desarrollar el código para el microcontrolador que permitirá el control y la adquisición. ✓

En el capítulo 5 se desglosa el código en ARDUINO para cada sensor y control utilizado en el proyecto del sistema SCADA, este código se encuentra completo al final del trabajo en el anexo.

- Desarrollo del software que estará enlazado al microcontrolador y este a la instalación para la supervisión y monitoreo de la planta. ✓

En el mismo capítulo del punto anterior se desglosa la programación del código en LabVIEW para cada sensor, control e indicador que conforma la interfaz HMI del software del sistema SCADA.

- Realizar las modificaciones y ajustes a la instalación que permitirán la implementación del sistema SCADA. ✓

Durante la realización de la prueba de la incorporación del sistema SCADA a la planta hidroeléctrica experimental se llevó a cabo la instalación de los nuevos componentes que conforman la planta para el correcto funcionamiento del sistema SCADA, esto está registrado en capítulo 6 y se puede visualizar en las imágenes de las figuras 6.2 a 6.5.

- Comprobar el correcto funcionamiento de la planta con el sistema SCADA. ✓

En el capítulo 6.1 “Análisis de resultados” se comprobó el correcto funcionamiento del sistema SCADA al evaluar la precisión y exactitud de las mediciones y en la documentación en video de la prueba se puede apreciar la planta en sincronía con el sistema SCADA en pleno funcionamiento.

- Generar un reporte de los datos adquiridos de la central hidroeléctrica.



Como se observa en el capítulo 3.9 “Registro de datos” se realizó la programación en LabVIEW para la generación de un reporte Excel que registra las mediciones de las variables seleccionadas durante el tiempo de ejecución del sistema SCADA (fig. 3.45 Programación del registro y base de datos en Excel).

Al haber analizado los puntos anteriores se concluye que la incorporación del proyecto SCADA para la central hidroeléctrica experimental del laboratorio de Termofluidos puede ser aplicada sin problema alguno al haber obtenido un 82.70% de aptitud cumpliendo satisfactoriamente sus objetivos para el control unificado y supervisión de la central hidroeléctrica.

## **Bibliografía**

- [1]A. Rodríguez Penin, Sistemas SCADA, 1st ed. Barcelona: Marcombo, 2007.
- [2]R. Bitter, T. Mohiuddin and M. Nawrocki, LabView advanced programming techniques, 1st ed.
- [3]J. Lajara Vizcaíno and J. Pelegrí Sebastiá, LabVIEW, 1st ed. México: Alfaomega, 2011.
- [4]J. Rodríguez Fernández, L. Cerda Filiu and R. Bezos Sánchez-Horneros, Automatismos industriales, 1st ed. Madrid: Paraninfo, 2014.
- [5]J. Blum, Exploring Arduino, 1st ed. Indianapolis, IN: John Wiley & Sons, 2013.
- [6]J. Trashorras Montecelos, Configuración de instalaciones eléctricas, 1st ed. Madrid: Paraninfo, 2013.
- [7]J. Roldán Viloría, Motores eléctricos, 1st ed. Madrid, España: Paraninfo, 2008.
- [8]M. Margolis, Arduino cookbook, 1st ed. [Place of publication not identified]: O'Reilly Media, 2016.
- [9]S. Sumathi and P. Surekha, LabVIEW based Advanced Instrumentation Sys-

tems, 1st ed. Berlín, Heidelberg: Springer-Verlag Berlin Heidelberg, 2007.

[10]C. Mataix and A. Arenas, Turbomáquinas hidráulicas, 1st ed. Madrid: ICAI, 2009.

[11]T. Wildi and R. Navarro Salas, Máquinas eléctricas y sistemas de potencia, 1st ed. México: Pearson/Educación, 2007.

[12]L. Corona Ramírez, G. Abarca Jiménez and J. Mares Carreño, Sensores y actuadores. .

[13]P. Tippens, Física. McGraw-Hill Interamericana, 2000.

[14]Y. Çengel and J. Cimbala, Mecánica de fluidos. México: McGraw Hill, 2012.

## Sitios Web

[1] "What is SCADA? Supervisory Control and Data Acquisition", Inductiveautomation.com, 2017. [Online]. Available: <https://inductiveautomation.com/what-is-scada>. [Accessed: 16- Oct- 2017].

[2] "SCADA", Es.wikipedia.org, 2017. [Online]. Available: <https://es.wikipedia.org/wiki/SCADA>. [Accessed: 16- Oct- 2017].

[3] "Interfaz Hombre-máquina (IHM) - Human-Machine Interface (HMI) - COPA-DATA", Copadata.com, 2017. [Online]. Available: <https://www.copadata.com/es-mx/soluciones-hmi-scada/interfaz-hombre-maquina-hmi/>. [Accessed: 16- Oct- 2017].

[4] "Interfaz de usuario", Es.wikipedia.org, 2017. [Online]. Available: [https://es.wikipedia.org/wiki/Interfaz\\_de\\_usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario). [Accessed: 16- Oct- 2017].

[5] J. Adriano, "Los tipos de tecnología con ejemplos sencillos", Como Funciona Que, 2017. [Online]. Available: <http://comofuncionaque.com/tipos-de-tecnologia/>. [Accessed: 16- Oct- 2017].

[6] R. Archanco, +. Archanco and +. Archanco, "Los 9 pilares tecnológicos de la Industria 4.0", Papeles de Inteligencia Competitiva, 2017. [Online]. Available: <http://papelesdeinteligencia.com/pilares-tecnologicos-de-la-industria-4-0/>. [Accessed: 16- Oct- 2017].

[7]F. maquinas, “Los beneficios de utilizar el sistema de supervisión SCADA - Noticias fabricantes maquinaria industrial, automatización y máquinas”, Blog de los fabricantes de maquinaria Industrial, 2017. [Online]. Available: <http://www.fabricantes-maquinaria-industrial.es/los-beneficios-de-utilizar-el-sistema-de-supervision-scada>. [Accessed: 16- Oct- 2017].

[8]”Protocolo de comunicaciones”, Es.wikipedia.org, 2017. [Online]. Available: [https://es.wikipedia.org/wiki/Protocolo\\_de\\_comunicaciones](https://es.wikipedia.org/wiki/Protocolo_de_comunicaciones). [Accessed: 16- Oct- 2017].

[9]”Sensor”, Es.wikipedia.org, 2017. [Online]. Available: <https://es.wikipedia.org/wiki/Sensor>. [Accessed: 16- Oct- 2017].

[10]”Puerto serie”, Es.wikipedia.org, 2017. [Online]. Available: [https://es.wikipedia.org/wiki/Puerto\\_serie](https://es.wikipedia.org/wiki/Puerto_serie). [Accessed: 16- Oct- 2017].

[11]”Hardware”, Es.wikipedia.org, 2017. [Online]. Available: <https://es.wikipedia.org/wiki/Hardware>. [Accessed: 16- Oct- 2017].

[12]”CompactDAQ - National Instruments”, Ni.com, 2017. [Online]. Available: <http://www.ni.com/data-acquisition/compactdaq/esa/>. [Accessed: 16- Oct- 2017].

[13]”What Is Data Acquisition? - National Instruments”, Ni.com, 2017. [Online]. Available: <http://www.ni.com/data-acquisition/what-is/esa/>. [Accessed: 16- Oct- 2017].

[14]"Multiplexación", Es.wikipedia.org, 2017. [Online]. Available: <https://es.wikipedia.org/wiki/Multiplexaci%C3%B3n>. [Accessed: 16- Oct- 2017].

[15]"Using an FSR | Force Sensitive Resistor (FSR) | Adafruit Learning System", Learn.adafruit.com, 2017. [Online]. Available: <https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr>. [Accessed: 16- Oct- 2017].

[16]2017. [Online]. Available: [http://www.datalights.com.ec/site2/images/stories/robotica/nap/nap\\_fsr.pdf](http://www.datalights.com.ec/site2/images/stories/robotica/nap/nap_fsr.pdf). [Accessed: 16- Oct- 2017].

[17]2017. [Online]. Available: [https://www.sparkfun.com/datasheets/Robotics/L298\\_H\\_Bridge.pdf](https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf). [Accessed: 16- Oct- 2017].

[18]"Arduino y los Relés | Tutoriales Arduino", Prometec.net, 2017. [Online]. Available: <http://www.prometec.net/relés/>. [Accessed: 16- Oct- 2017].

[19]"Interfaz Hombre-máquina (IHM) - Human-Machine Interface (HMI) - COPA-DATA", Copadata.com, 2017. [Online]. Available: <https://www.copadata.com/es/soluciones-hmi-scada/interfaz-hombre-maquina-hmi/>. [Accessed: 16- Oct- 2017].

[20]"Interfaz de usuario", Es.wikipedia.org, 2017. [Online]. Available: [https://es.wikipedia.org/wiki/Interfaz\\_de\\_usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario). [Accessed: 16- Oct- 2017].

[21]"Control de acceso informático", Es.wikipedia.org, 2017. [Online]. Available: [https://es.wikipedia.org/wiki/Control\\_de\\_acceso\\_inform%C3%A1tico](https://es.wikipedia.org/wiki/Control_de_acceso_inform%C3%A1tico). [Accessed: 16- Oct- 2017].

[22]"Comunicación de Arduino con puerto serie", Luis Llamas, 2017. [Online]. Available: <https://www.luisllamas.es/arduino-puerto-serie/>. [Accessed: 16-Oct- 2017].

[23]"Puerto serie", Es.wikipedia.org, 2017. [Online]. Available: [https://es.wikipedia.org/wiki/Puerto\\_serie](https://es.wikipedia.org/wiki/Puerto_serie). [Accessed: 16- Oct- 2017].

[24]"Sistema de alarma", Es.wikipedia.org, 2017. [Online]. Available: [https://es.wikipedia.org/wiki/Sistema\\_de\\_alarma](https://es.wikipedia.org/wiki/Sistema_de_alarma). [Accessed: 16- Oct- 2017].

[25]"Aviso de alarma", Es.wikipedia.org, 2017. [Online]. Available: [https://es.wikipedia.org/wiki/Aviso\\_de\\_alarma](https://es.wikipedia.org/wiki/Aviso_de_alarma). [Accessed: 16- Oct- 2017].

[26]2017. [Online]. Available: [http://www.idae.es/uploads/documentos/documentos\\_10374\\_Minicentrales\\_hidroelectricas\\_06\\_a8275b5d.pdf](http://www.idae.es/uploads/documentos/documentos_10374_Minicentrales_hidroelectricas_06_a8275b5d.pdf). [Accessed: 16- Oct- 2017].

[27]2017. [Online]. Available: <https://ingenierosenapuros.files.wordpress.com/2013/06/centrales-hidroelc3a9ctricas-041.pdf>. [Accessed: 16- Oct- 2017].

[28]2017. [Online]. Available: <http://ocw.unican.es/enseanzas-tecnicas/centrales-de-generacion-de-energia-electrica/materiales/bloque-energia-III.pdf>. [Accessed: 16- Oct- 2017].

[29]"parámetro - Definición - WordReference.com", Wordreference.com, 2017. [Online]. Available: <http://www.wordreference.com/definicion/par%C3%A1metro>. [Accessed: 16- Oct- 2017].

## Anexo

codigo:

```
#define Pecho 4
```

```
#define Ptrig 5
```

```
long duracion, distancia;
```

```
volatile int contador = 0;
```

```
const int sensorIn = A0;
```

```
int mVperAmp = 66;
```

```
float Voltage = 0;
```

```
float VRMS = 0;
```

```
float AmpsRMS = 0;
```

```
void setup() {
```

```
    Serial.begin (115200);
```

```
    pinMode(Pecho, INPUT);
```

```
    pinMode(Ptrig, OUTPUT);
```

```
    attachInterrupt(0,interrupcion0,RISING);
```

```
}
```

```
void nivel(){

    digitalWrite(Ptrig, LOW);

    delayMicroseconds(2);

    digitalWrite(Ptrig, HIGH); // genera el pulso de trigger por 10ms

    delayMicroseconds(10);

    digitalWrite(Ptrig, LOW);

    duracion = pulseIn(Pecho, HIGH);

    distancia = (duracion/2) / 29;    // calcula la distancia en centimetros

    Serial.print(distancia);    // envia el valor de la distancia por el puerto serial

    Serial.println("cm");    // le coloca a la distancia los centimetros "cm"

}

void rpm(){

    delay(999);    // retardo de casi 1 segundo

    Serial.print(contador*30); // Como son dos interrupciones por vuelta (contador *

(60/2))

    Serial.println(" RPM"); // El numero 2 depende del numero aspas de la helise del

motor en prueba(2 ASPAS)
```

```
contador = 0;

}

void interrupcion0(){

    contador++;    // Se incrementa en uno el contador

}

void sensorcorriente(){

    Voltage = getVPP();

    VRMS = (Voltage/2.0) *0.707;

    AmpsRMS = (VRMS * 1000)/mVperAmp;

    Serial.print(AmpsRMS);

    Serial.println(" Amps RMS");

}

float getVPP()

{

    float result;

    int readValue;

    int maxValue = 0;
```



```
int minValue = 1024;

uint32_t start_time = millis();

while((millis()-start_time) < 1000)

{

    readValue = analogRead(sensorIn);

    if (readValue > maxValue)

{

    maxValue = readValue;

}

    if (readValue < minValue)

{

    minValue = readValue;

}

}

result = ((maxValue - minValue) * 5.0)/1024.0;

return result;
```

```
}
```

```
void sensorvoltaje(){
```

```
int sensorValue=analogRead(A2);
```

```
int mvValue=map(sensorValue,0,1023,0,4800);
```

```
Serial.print(mvValue);
```

```
Serial.println(" mV");
```

```
}
```

```
void loop(){
```

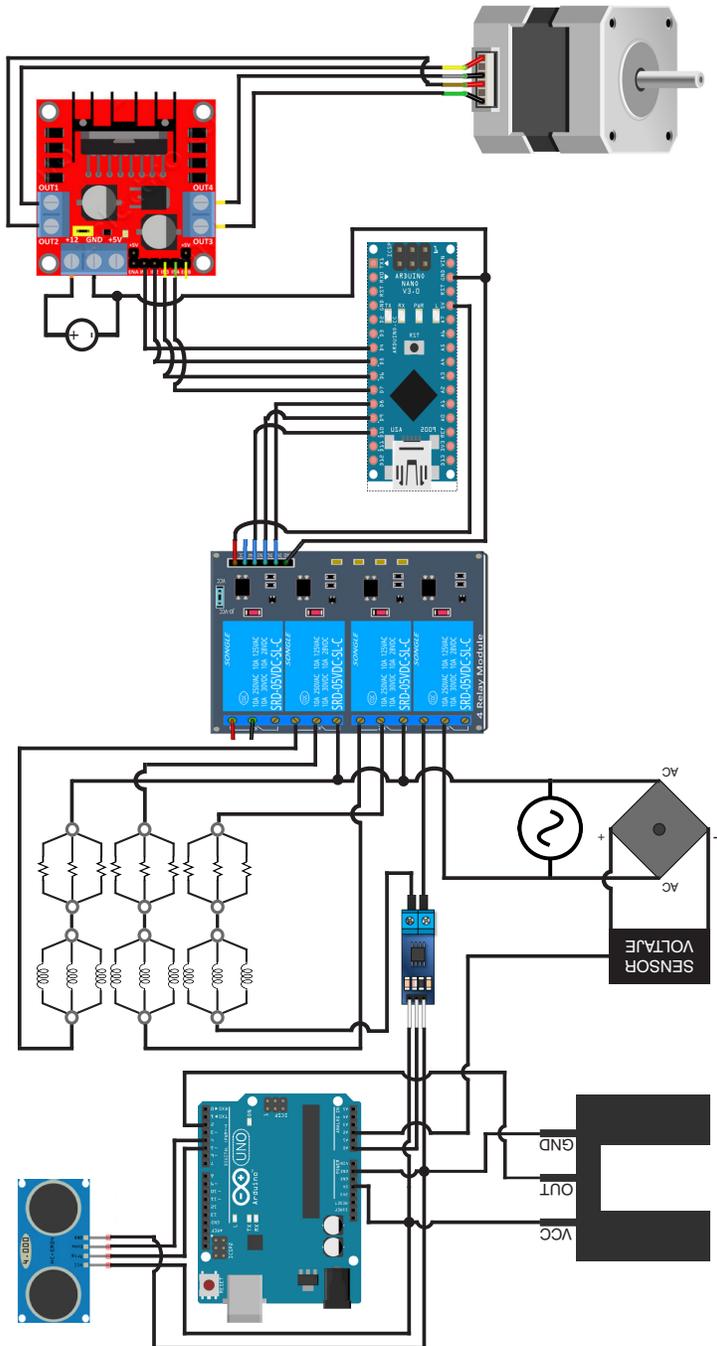
```
nivel();
```

```
rpm();
```

```
sensorcorriente();
```

```
sensorvoltaje();
```

```
}
```



**MOTOR-BOMBA**

**TURBINA-GENERADOR**

L1 L2 L3

VOLTAJE (RMS)

CORRIENTE (RMS)

Z OHMS

FUERZA (N)

FRECUENCIA Hz

POTENCIA PLANTA

PRESION (kg/cm<sup>2</sup>)

PRESION (Pa)

EFICIENCIA TURBINA (%)

POTENCIA HIDRAULICA (W)

CARGA NETA (m)

EFICIENCIA GENERADOR (%)

POTENCIA ELECTRICA (W)

POTENCIA MECANICA

FP

FECHA	HORA	OPERADOR	VOLTAJE BOMBA	AREA

VELOCIDAD BOMBA

ON

OFF

BOMBA

TO PUMP

Stop

VELOCIDAD INECTOR

Play

RPM TURBINA

Angulo ° (+)

OK

USUARIO

ON LINE

CARGA

VOLTAJE

VELOCIDAD TURBINA

POTENCIA DE PLANTA

EF. PLANTA %

0

