



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID  
ESCUELA DE INGENIERÍAS INDUSTRIALES**

**Grado en Ingeniería Eléctrica**

**DISEÑO E IMPLEMENTACIÓN DE UN SCADA PARA LA  
MONITORIZACIÓN DE UNA CENTRAL TÉRMICA**

**Autor:**

**Madrid Ortega, Raúl**

**Tutor:**

**San Martín Ojeda, Moisés Luis**

**Serrano Sanz, José Andrés**

**Departamento de Ingeniería Eléctrica**

**Valladolid, Octubre de 2017**



## RESUMEN

El presente trabajo se centra en la monitorización de una pequeña parte de un proceso industrial de una Central Térmica Convencional.

Para ello se diseñará un sistema SCADA mediante el software LabVIEW, con su correspondiente programación, el cual estará interconexionado mediante Ethernet a un PLC de National Instruments. El programa creado será fácilmente configurable por el usuario sin necesidad de tener conocimientos de LabVIEW.

De esta forma el sistema SCADA que estará presente, en una o varias pantallas, será de gran ayuda para ver de forma rápida e intuitiva el estado de la Central. En estas pantallas se reflejarán los parámetros más significativos y poder así, seguir en todo momento su funcionamiento aun sin estar en la propia central.

Además dicho SCADA estará preparado para enfrentarse a posibles anomalías de funcionamiento. Se controlarán los estados de temperatura y presión de las turbinas con el objetivo de mantener unos límites y crear unas alarmas si es necesario.

En el caso de que se produzca algún defecto o alarma, se registrarán los detalles de la anomalía en un archivo Excel y se enviará un correo electrónico al usuario o encargado de la instalación, con el fin de agilizar lo máximo posible su resolución o su posible estudio.

## PALABRAS CLAVE

SCADA, PLC, LabVIEW, Automatización, Central Térmica.

## AGRADECIMENTOS

A mi familia, novia y amigos, que me han ayudado durante la carrera.

A mi tutor Moisés Luis San Martín Ojeda por su disposición y constantes consejos a lo largo de la elaboración del Trabajo de Fin de Grado.



## ÍNDICE DE CONTENIDO

RESUMEN.....	3
PALABRAS CLAVE.....	3
AGRADECIMIENTOS.....	3
ÍNDICE DE CONTENIDO.....	5
ÍNDICE DE ILUSTRACIONES.....	9
INTRODUCCIÓN .....	11
OBJETIVOS GENERALES .....	13
CAPITULO 1: SISTEMAS SCADA .....	15
1.1 INTRODUCCIÓN .....	15
1.2 INTERFAZ HOMBRE – MÁQUINA (HMI).....	16
1.2.1 DISEÑO INTERFAZ DE USUARIO .....	17
1.2.2 INTERFAZ DE USUARIO SEGÚN LA FORMA DE INTERACTUAR .....	18
1.2.3 INTERFAZ DE USUARIO SEGÚN SU CONSTRUCCIÓN .....	19
1.2.3.1 Interfaces hardware .....	19
1.2.3.2 Interfaces software .....	19
1.3 SISTEMAS SCADA.....	20
1.3.1 COMPONENTES DE UN SISTEMA SCADA.....	21
1.3.1.1 Módulos de un sistema SCADA y aplicación en LabVIEW .....	24
1.3.1.2 Filosofía operacional .....	25
1.3.1.3 Infraestructura y Métodos de comunicación .....	26
1.3.1.3.1 Transmisión de datos.....	27
1.3.1.3.2 El estándar OPC .....	28
1.3.2 CARACTERÍSTICAS DE UN SISTEMA HMI/SCADA .....	30
1.3.3 SOFTWARES SCADA.....	30
CAPITULO 2: SOFTWARE LABVIEW .....	31
2.1 INTRODUCCIÓN AL ENTORNO DEL SOFTWARE LabVIEW .....	31
2.2 MÓDULOS EMPLEADOS .....	32
2.2.1 MÓDULO DSC (Datalogging and Supervisory Control).....	32
2.2.2 MÓDULO FPGA .....	33

2.2.3	MÓDULO REAL TIME .....	34
2.2.4	MÓDULO VISION DEVELOPMENT .....	34
2.3	FUNDAMENTOS labVIEW .....	35
2.3.1	PANEL FRONTAL.....	35
2.3.2	PALETA DE CONTROLES.....	36
2.3.3	DIAGRAMA DE BLOQUES .....	36
2.3.4	TERMINALES.....	37
2.3.5	NODOS DE DIAGRAMA DE BLOQUES .....	37
2.3.6	PALETA DE FUNCIONES .....	39
2.4	PROGRAMACIÓN GRÁFICA.....	40
2.4.1	CABLES.....	41
2.5	ESTRUCTURAS DE DATOS.....	43
2.5.1	TIPO DE DATOS CADENA DE CARACTERES.....	43
2.5.2	TIPO DE DATO NUMÉRICO .....	44
2.5.3	TIPO DE DATO BOOLEANO.....	44
2.5.4	MATRICES.....	45
2.5.5	CLUSTERS.....	46
2.6	ESTRUCTURAS DE EJECUCIÓN .....	47
2.6.1	CICLOS.....	47
2.6.1.1	Ciclos While.....	47
2.6.1.1.1	Túneles de Estructura .....	48
2.6.1.2	Ciclos for.....	49
2.6.1.2.1	Añadir temporización.....	50
2.6.1.3	Ciclos Case Structure .....	51
2.6.1.4	Ciclos Flat Sequence .....	52
2.7	VARIABLES LOCALES .....	53
2.8	VARIABLES COMPARTIDAS (SHARED VARIABLES).....	54
2.9	AVISO CORREO ELECTRÓNICO.....	55
2.10	ALMACENAMIENTO DE DATOS .....	57
2.11	LECTURA DE DATOS.....	59
2.12	DIAGRAMA DE FLUJO CENTRAL TÉRMICA .....	61
2.13	DIAGRAMA DE BLOQUES EN LabVIEW .....	65

CAPITULO 3: DISEÑO PANTALLA SCADA.....	67
3.1    DISEÑO INTERFAZ GRÁFICA .....	67
3.1.1    CONTROLES E INDICADORES.....	69
3.2    VISUALIZACIÓN DEL SCADA VÍA WEB.....	72
3.3    GENERACIÓN APLICACIÓN SCADA .....	78
 CAPITULO 4: PLC NATIONAL INSTRUMENTS NI cRIO-9022 .....	 81
4.1    PLC COMPACT RIO REAL-TIME CONTROLLER .....	81
4.2    MÓDULOS NI cRIO-9022.....	82
4.2.1    MÓDULO NI 9201 .....	83
4.2.2    MÓDULO NI 9472 .....	84
4.3    CHASIS COMPACT RIO NI 9114 .....	85
4.4    FUENTE DE ALIMENTACIÓN NI PS-15.....	86
4.5    CONEXIÓN PC-PLC .....	87
 CAPITULO 5: DISEÑO, CONSTRUCCIÓN DE LA MAQUETA Y HARDWARE ADICIONAL.....	 91
5.1    INTRODUCCIÓN .....	91
5.2    DISEÑO DE LA MAQUETA.....	92
5.3    HARDWARE ADICIONAL.....	93
5.3.1    SIMULACIÓN ENTRADAS DEL SISTEMA.....	93
5.3.2    SIMULACIÓN SALIDAS DEL SISTEMA .....	95
5.4    CONEXIONADO DE LOS ELEMENTOS.....	97
 CAPITULO 6: PRUEBAS DE FUNCIONAMIENTO.....	 99
6.1    INTRODUCCIÓN .....	99
6.2    PRUEBAS DE FUNCIONAMIENTO .....	101
6.2.1    PRUEBA CONEXIÓN PC-PLC.....	102
6.2.2    PRUEBA DE FUNCIONAMIENTO E/S.....	103
6.2.3    SIMULACIÓN FUNCIONAMIENTO TURBINAS.....	105
6.2.4    COMPROBACIÓN LISTADO CONSIGNAS .....	107

6.2.5	SIMULACIÓN ANOMALÍA FUNCIONAMIENTO .....	108
6.2.5.1	Lectura y escritura de la anomalía.....	108
6.2.5.2	Prueba envío de email vía smtp.....	109
CAPITULO 7: MANUAL DE INSTALACIÓN Y CONFIGURACIÓN .....		111
7.1	CONFIGURACIÓN DE VARIABLES E INSTALACIÓN .....	111
IMPLEMENTACIONES FUTURAS .....		115
OBJETIVOS Y CONCLUSIONES .....		117
RECOMENDACIONES .....		121
BIBLIOGRAFÍA.....		123
ANEXOS.....		125

## ÍNDICE DE ILUSTRACIONES

<b>Figura 1.1: Esquema sistema SCADA</b> ( <a href="http://www.uhu.es">http://www.uhu.es</a> ).....	20
<b>Figura 1.2: Estación Maestra</b> ( <a href="https://repositorio.espe.edu.ec">https://repositorio.espe.edu.ec</a> ).....	21
<b>Figura 1.3: Unidades Remotas (RTU)</b> ( <a href="http://www.fanox.com">http://www.fanox.com</a> ).....	22
<b>Figura 1.4: Red de Telemetría</b> ( <a href="http://www.telemetry.com">http://www.telemetry.com</a> ) .....	23
<b>Figura 1.5: Estación de Supervisión</b> ( <a href="http://www.telemetry.com">http://www.telemetry.com</a> ) .....	23
<b>Figura 1.6: Paleta entorno de trabajo LabVIEW</b> (Elaboración propia).....	24
<b>Figura 1.7: Interconexión OPC</b> ( <a href="http://docplayer.es/">http://docplayer.es/</a> ) .....	28
<b>Figura 1.8: Estructura Cliente-Servidor OPC</b> (Elaboración propia).....	29
<b>Figura 2.9: Pantalla DSC</b> (Elaboración propia) .....	32
<b>Figura 2.10: Paleta de Controles</b> (Elaboración propia).....	36
<b>Figura 2.11: Icono subVI</b> (Elaboración propia) .....	38
<b>Figura 2.12: Icono Panel Conector</b> (Elaboración propia) .....	38
<b>Figura 2.13: Paleta de Funciones</b> (Elaboración propia) .....	39
<b>Figura 2.14: Ejemplo Flujo de Datos</b> (Elaboración propia).....	40
<b>Figura 2.15: Tabla de cables</b> ( <a href="http://www.ni.com">http://www.ni.com</a> ) .....	41
<b>Figura 2.16: Cable Roto</b> (Elaboración propia).....	41
<b>Figura 2.17: Cadena de Caracteres</b> (Elaboración propia) .....	43
<b>Figura 2.18: Estructura Matriz en secuencia</b> (Elaboración propia).....	45
<b>Figura 2.19: Ciclo While</b> (Elaboración propia) .....	47
<b>Figura 2.20: Ciclo For</b> (Elaboración propia) .....	49
<b>Figura 2.21: Temporización Ciclo</b> (Elaboración propia).....	50
<b>Figura 2.22: Ciclo Case Estructure</b> (Elaboración propia).....	51
<b>Figura 2.23: Ciclo Flat Sequence</b> (Elaboración propia) .....	52
<b>Figura 2.24: Variable y Variable Local</b> (Elaboración propia).....	53
<b>Figura 2.25: Variable Compartida</b> (Elaboración propia) .....	54
<b>Figura 2.26: Programación Correo Electrónico</b> (Elaboración propia) .....	55
<b>Figura 2.27: Programación Labview último estado de ref.</b> (Elaboración propia) .....	56
<b>Figura 2.28: Programación LabVIEW último estado de ref.</b> (Elaboración propia) .....	57
<b>Figura 2.29: Programación LabVIEW último estado de ref.</b> (Elaboración propia) .....	58
<b>Figura 2.30: Archivo estado referencia</b> (Elaboración propia) .....	58
<b>Figura 2.31: Archivo lectura parámetros</b> (Elaboración propia).....	59
<b>Figura 2.32: Lectura archivo</b> (Elaboración propia).....	59
<b>Figura 2.33: Recogida de datos archivo</b> (Elaboración propia).....	60
<b>Figura 2.34: Diagrama de Flujo 1</b> (Elaboración propia) .....	63
<b>Figura 2.35: Diagrama de Flujo 2</b> (Elaboración propia) .....	64
<b>Figura 3.36: Panel Frontal (2 Pestañas)</b> (Elaboración propia) .....	68
<b>Figura 3.37: Indicadores Numéricos</b> (Elaboración propia).....	69
<b>Figura 3.38: Adaptación indicador CTL</b> (Elaboración propia).....	70
<b>Figura 3.39: CTL animado</b> (Elaboración propia) .....	70
<b>Figura 3.40: Tabla indicadora de cadena de caracteres</b> (Elaboración propia) .....	71
<b>Figura 3.41: Web Publishing Tool</b> (Elaboración propia) .....	72
<b>Figura 3.42: Opciones de visualización vía Web</b> (Elaboración propia) .....	73
<b>Figura 3.43: Opciones texto vía Web</b> (Elaboración propia).....	73
<b>Figura 3.44: Archivo HTML en el proyecto</b> (Elaboración propia).....	74
<b>Figura 3.45: Propiedades Compact RIO</b> (Elaboración propia).....	74

<b>Figura 3.46: Propiedades Compact RIO (Elaboración propia)</b> .....	75
<b>Figura 3.47: Propiedades Compact RIO (Elaboración propia)</b> .....	76
<b>Figura 3.48: Visualización SCADA vía Web (Elaboración propia)</b> .....	77
<b>Figura 3.49: Crear aplicación SCADA (Elaboración propia)</b> .....	78
<b>Figura 3.50: Propiedades aplicación SCADA (Elaboración propia)</b> .....	79
<b>Figura 3.51: Aplicación SCADA (Elaboración propia)</b> .....	79
<b>Figura 4.52: PLC NI 9022 (<a href="http://www.ni.com">http://www.ni.com</a>)</b> .....	82
<b>Figura 4.53: Módulo NI 9201 (<a href="http://www.ni.com">http://www.ni.com</a>)</b> .....	83
<b>Figura 4.54: Módulo NI 9472 (<a href="http://www.ni.com">http://www.ni.com</a>)</b> .....	84
<b>Figura 4.55: Chasis NI 9114 (<a href="http://www.ni.com">http://www.ni.com</a>)</b> .....	85
<b>Figura 4.56: Fuente alimentación NI PS-15 (<a href="http://www.ni.com">http://www.ni.com</a>)</b> .....	86
<b>Figura 4.57: Archivo nuevo Labview (Elaboración propia)</b> .....	87
<b>Figura 4.58: Proyecto en Tiempo Real (Elaboración propia)</b> .....	88
<b>Figura 4.59: Icono Panel Conector (Elaboración propia)</b> .....	88
<b>Figura 4.60: Añadir chasis al proyecto (Elaboración propia)</b> .....	89
<b>Figura 4.61: Chasis disponibles (Elaboración propia)</b> .....	89
<b>Figura 4.62: Módulos existentes en el chasis (Elaboración propia)</b> .....	90
<b>Figura 5.63: Diseño maqueta (Elaboración propia)</b> .....	92
<b>Figura 5.64: Esquema regulador (Elaboración propia)</b> .....	93
<b>Figura 5.65: Esquema 6 reguladores (Elaboración propia)</b> .....	94
<b>Figura 5.66: Caja de conexiones reguladores (Elaboración propia)</b> .....	95
<b>Figura 5.67: Esquema Relé de Estado Sólido (Elaboración propia)</b> .....	95
<b>Figura 5.68: Relé de Estado Sólido (<a href="http://www.automation24.es">http://www.automation24.es</a>)</b> .....	96
<b>Figura 5.69: Caja Pilotos (Elaboración propia)</b> .....	96
<b>Figura 6.70: Adaptación entradas (Elaboración propia)</b> .....	99
<b>Figura 6.71: Cambio de pestaña SCADA (Elaboración propia)</b> .....	100
<b>Figura 6.72: Conexión programa PLC (Elaboración propia)</b> .....	102
<b>Figura 6.73: Conexión programa SCADA (Elaboración propia)</b> .....	102
<b>Figura 6.74: Entradas proceso (Elaboración propia)</b> .....	103
<b>Figura 6.75: Valor nominal entradas (Elaboración propia)</b> .....	103
<b>Figura 6.76: Activación electroválvula turbina AP (Elaboración propia)</b> .....	105
<b>Figura 6.77: Activación electroválvulas MP y BP (Elaboración propia)</b> .....	106
<b>Figura 6.78: Alerta estado alarma (Elaboración propia)</b> .....	108
<b>Figura 6.79: Correo en bandeja de entrada (Elaboración propia)</b> .....	109
<b>Figura 6.80: Contenido correo electrónico (Elaboración propia)</b> .....	109
<b>Figura 7.81: Variables modificables (Elaboración propia)</b> .....	111
<b>Figura 7.82: Variables en el SCADA (Elaboración propia)</b> .....	112
<b>Figura 0.83: Industria 4.0 vs Humanos (<a href="https://www.taringa.net">https://www.taringa.net</a>)</b> .....	118

## INTRODUCCIÓN

Actualmente todo proceso industrial tiende a expandirse hacia un ambiente cada vez más automatizado, lo que conlleva una mayor complejidad de las instalaciones y mayor control de las mismas.

El control de dichos procesos se realiza mediante PLCs, normalmente modulares, según el tamaño de la instalación. Estos PLCs son como ordenadores, pero diseñados para múltiples señales de entrada y salida, rango de temperaturas muy amplios, inmunidad al ruido eléctrico, resistencia a la vibración y al impacto, lo que los convierte en deseables para trabajar en entornos industriales. Disponen de una pequeña batería que les sirve para almacenar los programas de control de funcionamiento en el caso que se les anule la alimentación.

Dentro de un proceso automatizado con PLCs puede ser de gran utilidad la implementación de una interfaz visual, facilitando al usuario la interpretación rápida de los aspectos más importantes del proceso industrial, y aumentando las posibilidades de seguridad y gestión. Para ello se emplean ordenadores externos que trabajaran con software dedicados al diseño de la interfaz y comunicación con el autómatas programable.

La interfaz visual debe ser fácil de interpretar y que contenga la información necesaria para poder tener un control en todo momento del proceso industrial. Cuanto menos complejidad tenga y más claro se vea, mejor y más rápido se verán los parámetros que se busquen, creando así los llamados Interfaz Hombre-Máquina, definido por las siglas HMI.

A demás dicha interfaz debe ser capaz de interactuar con el sistema de control y poder tener un control sobre este pudiendo modificar datos, introducir valores base, poder detener o iniciar el proceso, gestionar e incluso tomar decisiones. A este sistema se le conoce con el nombre de SCADA (Supervisory Control And Data Acquisition), es decir, su misión es supervisar, controlar y adquirir datos.

Hay una gran variedad de software a utilizar para el diseño de un sistema SCADA. Para este trabajo se ha empleado el software LabVIEW, que es un software de ingeniería de alto nivel. El entorno de programación de LabVIEW simplifica la integración de hardware, reduce la complejidad de la programación y permite visualizar resultados inmediatamente con la creación integrada de una interfaz de usuario.

LabVIEW permite integrar módulos adicionales, permitiendo así integrar todo dentro de un solo software y con un mismo lenguaje de programación.

Lo que se plantea realizar en este trabajo es la supervisión y control de una Central Térmica Convencional, mediante el diseño de un sistema SCADA con el software LabVIEW y con un PLC de National Instruments modular NI c-RIO 9022.

Se llevará a cabo una simulación de una parte en concreto de la Central Térmica, debido a la complejidad que puede albergar dicha instalación si se realiza entera. La supervisión y control se hará sobre las temperaturas y presiones de las turbinas de alta, media y baja presión. Se da por hecho que del mismo modo se podría controlar cualquier otra parte de la instalación del mismo modo.

El presente trabajo se reparte en 7 capítulos:

1. Capítulo 1: Sistemas SCADA.

Se introducirán conceptos teóricos sobre Sistemas SCADA.

2. Capítulo 2: Software LabVIEW.

Se introducirán conceptos teóricos sobre el Software LabVIEW.

3. Capítulo 3: Diseño pantalla SCADA

Diseño completo de la interfaz gráfica y formas de visualización.

4. Capítulo 4: PLC NI cRIO-9022.

Nociones PLC, módulos y conexionado.

5. Capítulo 5: Diseño, construcción de la maqueta y hardware adicional.

Construcción y montaje del conjunto completo.

6. Capítulo 6: Pruebas de funcionamiento.

Comprobación funcionamiento sistema de control y cumplimiento de objetivos.

7. Capítulo: Manual de instalación y configuración.

Configuración de variables e instalación.

## OBJETIVOS GENERALES

Los objetivos a cumplir en este trabajo son:

1. Crear una Interface Gráfica para poder controlar y visualizar un proceso industrial desde varios puntos.
2. Crear un sistema SCADA configurable o fácilmente configurable.
3. Empleo de un PLC de National Instruments modular, con el que se llevará a cabo el caso práctico del proceso industrial.
4. Establecer con el SCADA unos estados límites de funcionamiento, y en el caso de superar estos límites, reaccionar ante ellos de manera apropiada.
5. Almacenar en un historial y gestionar las alertas que surjan en el transcurso del proceso.
6. Avisar al encargado de la instalación o al usuario mediante correo electrónico, el estado de la central cuando sea necesario.
7. Visualización del estado de la Central vía Web.



## CAPITULO 1: SISTEMAS SCADA

### 1.1 INTRODUCCIÓN

Cuando los avances tecnológicos no estaban tan avanzados, los procesos industriales se supervisaban y se controlaban de forma manual, es decir, los encargados de que el proceso funcionase correctamente, hacían un seguimiento lo más exacto posible de parámetros, alarmas, fallos etc.

Estos operarios no dejan de ser personas que pueden tener descuidos, fallos a la hora de tomar datos y por supuesto disponen de una velocidad limitada para realizar las labores de seguimiento.

En la actualidad con los avances en los sistemas informáticos, la implementación de Software, proporcionan suficiente capacidad para recoger datos automáticamente y registrarlos, de tal modo que los operarios puedan examinarlos en cualquier momento, minimizando así la pérdida de información y la incertidumbre que puede surgir con la recogida de información del operario y con una velocidad mayor de toma de datos.

Los Software utilizados han permitido que las industrias agilicen de forma significativa su proceso productivo, aprovechando la gran capacidad que tienen los ordenadores para el procesamiento de datos.

Actualmente la tendencia de todas las empresas en el entorno industrial es de expandirse a un entorno totalmente automatizado, ya que es la clave para ser competitivos económicamente y a nivel de tiempo. Para ello se emplean Software muy potentes que poseen una gran versatilidad, con el fin de obtener los mejores resultados posibles.

Obviamente, esto supone una inversión económica importante, pero que su rentabilidad a largo plazo es muy grande debido a sus numerosas ventajas, frente a los pocos inconvenientes que poseen.

La utilización de estos sistemas requiere de personal cualificado para su diseño, no así para el manejo una vez finalizado el diseño en software, ya que están pensados para que su manejo durante su funcionamiento sea lo más simple posible, disminuyendo así el tiempo de interpretación y respuesta visual.

## 1.2 INTERFAZ HOMBRE – MÁQUINA (HMI)

En un proceso industrial, cualquier proceso que se quiera automatizar y controlar debe ser monitorizado. Todos los elementos del proceso deben estar comunicados entre sí para poder tener un control y supervisión de los mismos. Existen elementos que son encargados de obtener unos parámetros (temperatura, presión, velocidad, posición, nivel...) que nos facilitan la información relevante del objeto a estudiar. De igual forma existen elementos de control encargados de emitir una respuesta (motores, pilotos, electroválvulas...). [1].

Es lógico pensar que entre estos sensores y actuadores debe existir algo que tome decisiones y sepa cómo actuar en cada momento dependiendo del estado de cada sensor o actuador, para ello se emplean autómatas programables o controladores lógicos programables (PLC), que están diseñados para controlar, en tiempo real y en ambiente industrial, procesos secuenciales.

Con estos elementos ya se podría automatizar un proceso industrial sin ningún problema, pero en ocasiones interesa que el propio sistema establezca una información que pueda proporcionar al operario y así estar informado del funcionamiento en todo momento, y por supuesto, que el operario pueda tener un control sobre el sistema de forma fácil y rápida. Esto se denomina HMI (Interfaz Hombre-Máquina).

El HMI simplemente es una parte del programa informático que se comunica con el usuario, proporcionando la información y control necesario para que el operario lleve a cabo una tarea con el sistema interactivo.

Las interfaces más relevantes son aquellas que incluyen la máxima información necesaria pero con la menor complejidad posible. La idea es que su interpretación sea muy sencilla para facilitar al máximo el intercambio de datos o parámetros. Por esto, el diseño de una interfaz de usuario es crítico para el manejo de un equipo. Hay que tener en cuenta que probablemente no todos los operarios que manipulen la interfaz sean expertos en ese tema, por lo que se tendrá que tener en cuenta en su diseño.

### 1.2.1 DISEÑO INTERFAZ DE USUARIO

El diseño de la interfaz de usuario requiere el estudio de las personas y el conocimiento tecnológico adecuado.

¿Qué preguntas deben plantearse y responderse para el diseño de una interfaz de usuario?

1. **¿Quién es el usuario?**
2. **¿Cómo aprende a interactuar con el nuevo sistema?**
3. **¿Cómo interpreta la información que produce el sistema?**
4. **¿Qué espera del sistema?**

Existen 3 reglas para el diseño del interfaz:

- **Dar el control al usuario:** Un sistema que reaccione a las necesidades del usuario y que le ayuda a hacer las cosas.
- **Reducir la carga de memoria del usuario:** Una interfaz de usuario bien diseñada no dependerá de la memoria del usuario. Siempre que sea posible, el sistema debe recordar la información pertinente y ayudar al usuario con un escenario de interacción que facilite el uso de la memoria.
- **Lograr que la interfaz sea consistente:** Para que un sistema sea consistente implica:
  - Toda la información visual esté organizada de acuerdo con un estándar de diseño que se mantenga en todas las presentaciones de la pantalla.
  - Los mecanismos de entrada se restrinjan a un conjunto limitado que se utilice de manera consistente en toda la interface.
  - Los mecanismos para ir de una tarea a otra se hayan definido e implementado de manera consistente.

El proceso de análisis y diseño de la interfaz de usuario abarca cuatro actividades distintas de marco de trabajo.

- Análisis y modelado de usuarios, tarea y entornos.
- Diseño de la interfaz.
- Construcción de la interfaz.
- Validación de la interfaz.

### 1.2.2 INTERFAZ DE USUARIO SEGÚN LA FORMA DE INTERACTUAR

1. **Interfaz de línea de comandos (CLI):** Estas interfaces solo presentan texto, por lo que son las más fáciles de diseñar, pero su efectividad e interpretación a veces resulta confusa.
2. **Interfaz gráfica de usuario (GUI):** Permiten comunicarse con el ordenador de forma rápida e intuitiva representando gráficamente los elementos de control y medida.
3. **Interfaz natural de usuario (NUI):** Pueden ser táctiles, representando gráficamente un panel de control en una pantalla sensible al tacto que permite interactuar con el dedo de forma similar a si se accionara un control físico.

### 1.2.3 INTERFAZ DE USUARIO SEGÚN SU CONSTRUCCIÓN

#### 1.2.3.1 *Interfaces hardware*

Se trata de un conjunto de controles que permiten que el usuario intercambie datos con la máquina, ya sea introduciéndolos (pulsadores, botones, teclas manivelas...) o leyéndolos (pantallas, marcadores, diales...).

#### 1.2.3.2 *Interfaces software*

Son programas o parte de ellos, que permiten expresar las órdenes a la computadora o visualizar su respuesta.

### 1.3 SISTEMAS SCADA

#### SCADA (Supervisory Control And Data Acquisition)

Se trata de una aplicación formada por diferentes Software, diseñada para funcionar sobre ordenadores en el control y supervisión de la producción, proporcionando comunicación con los dispositivos de campo (Controladores, PLC...) y controlando el proceso de forma automática desde la pantalla del ordenador. Además, provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros supervisores dentro de la empresa [2], [3].

Normalmente existe uno o varios ordenadores, que efectúa las tareas de supervisión y gestión de alarmas, así como tratamiento de datos y control de procesos.

Un sistema SCADA (Figura 1.1) incluye un Hardware de señal de entrada y salida, controladores, HMI, comunicaciones, redes, base de datos y software.

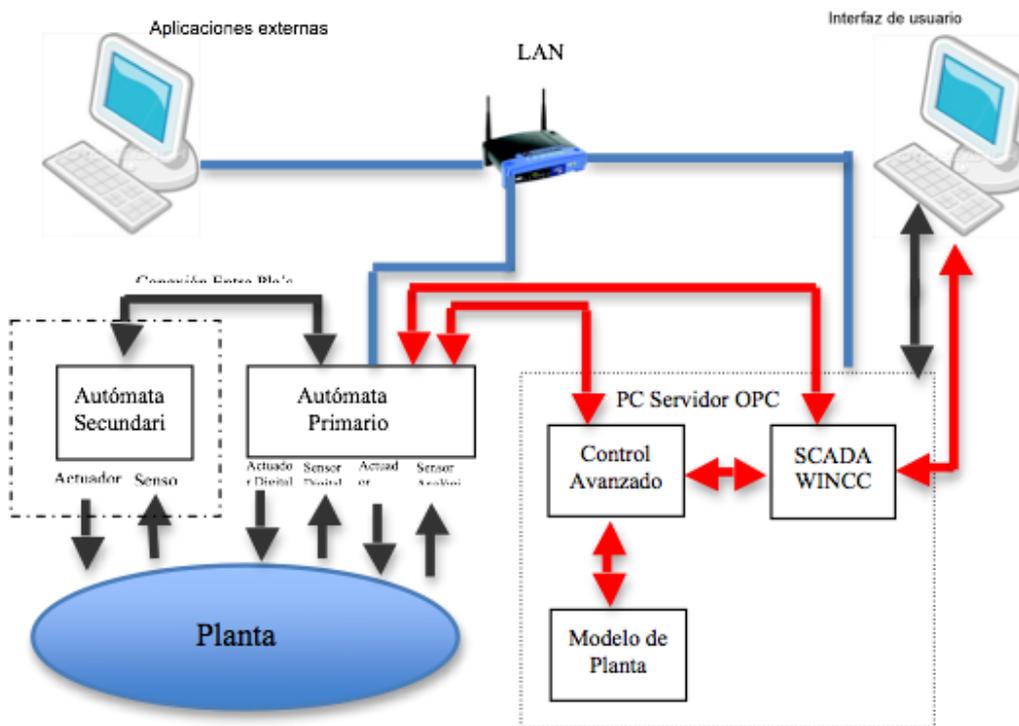


Figura 1.1: Esquema sistema SCADA (<http://www.uhu.es>)

### 1.3.1 COMPONENTES DE UN SISTEMA SCADA

- Estación Maestra:** Recibe datos del estado de los equipos en campo que es enviada por las estaciones remotas (RTU). Procesa la información y envía comandos a las estaciones remotas para mantener las variables de los procesos dentro de los parámetros establecidos [4].

Dependiendo del tipo de sistema SCADA la estación maestra, (Figura 1.2) puede ser un ordenador con un software de supervisión y control o un PLC con capacidad de comunicación que realizará la tarea de leer la información de las unidades remotas.

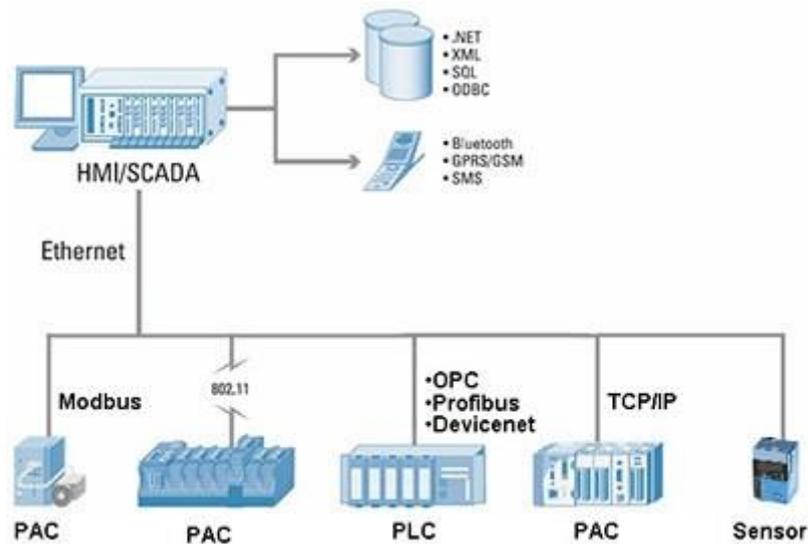


Figura 1.2: Estación Maestra (<https://repositorio.espe.edu.ec>)

- Unidades Remotas (RTU):** Controlan las señales de entrada y salida del campo, además monitorean las condiciones de los dispositivos de campo y almacenan los estados de las alarmas.

Envían los estados y las alarmas de los equipos de campo y reciben los comandos de la estación maestra.

Una consideración de RTU (Figura 1.3) puede ser dependiendo del número de señales de entrada y salida que manejen.

Hasta 100 señales se las denomina como pequeñas, entre 100-500 señales I/O como medianas y si el número de señales es mayor, se las consideran grandes.



*Figura 1.3: Unidades Remotas (RTU) (<http://www.fano.com>)*

- **Red de Telemetría:** Permiten establecer el intercambio de información entre la Estación Maestra y las Unidades Remotas (Figura 1.4).
  - **Topología usada:** Corresponde al arreglo geométrico de los nodos. Entre los principales se tiene el punto a punto, punto a multipunto, etc.
  - **Modo de transmisión:** Es la forma como viaja la información entre los distintos nodos de la red.
  - **Medio utilizado:** Corresponde al tipo de medio utilizado para enviar y recibir la información.

Componentes de la Red de Telemetría:

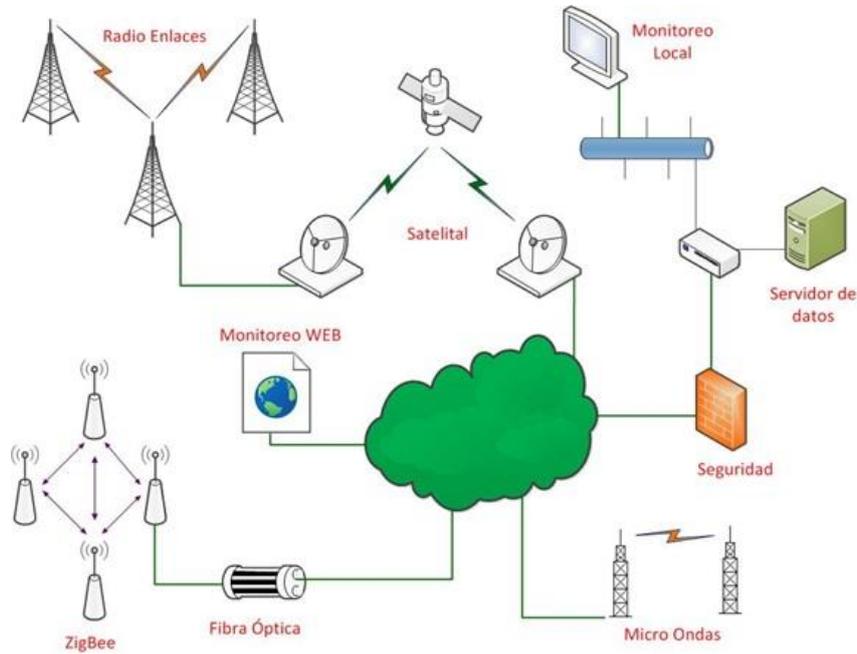


Figura 1.4: Red de Telemetría (<http://www.telemetry.com>)

- Estación de Supervisión:** Permite la visualización gráfica del estado del proceso (Figura 1.5), es decir, proporciona al operador las funciones de control y supervisión de la planta.

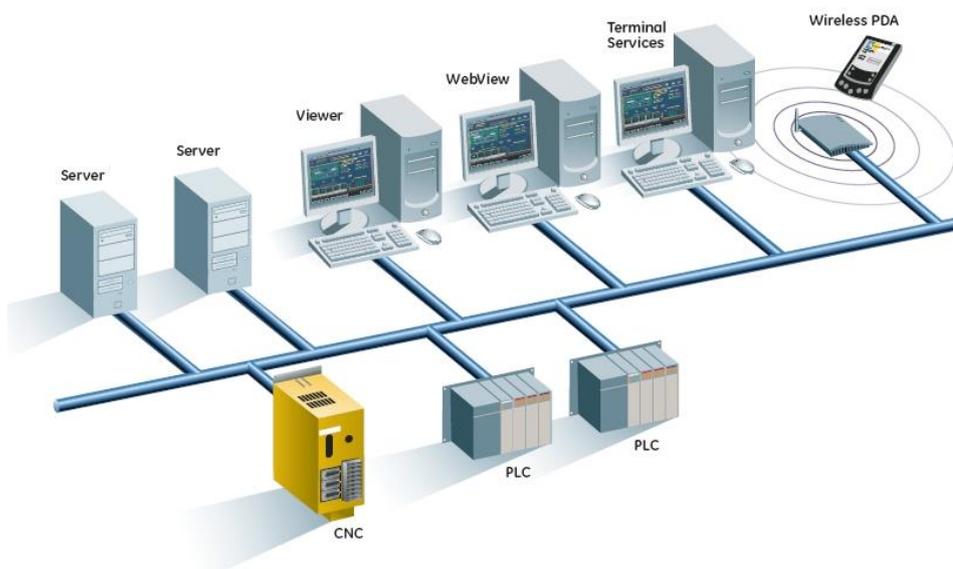


Figura 1.5: Estación de Supervisión (<http://www.telemetry.com>)

### 1.3.1.1 Módulos de un sistema SCADA y aplicación en LabVIEW

Los módulos o bloques de software que permiten las actividades de control, supervisión y adquisición de datos son [5], [6]:

- **Configuración:** Permite al operador definir el entorno de trabajo del sistema SCADA, adaptándolo a la aplicación particular que se desea desarrollar (Figura 1.6).



Figura 1.6: Paleta entorno de trabajo LabVIEW (Elaboración propia)

- **Interfaz gráfica del operador:** Proporciona al operador las funciones de control y supervisión de la planta. El proceso se representa mediante sinópticos gráficos de fácil interpretación, con representación de los parámetros más importantes del proceso. Estos parámetros estarán centrados en las presiones, temperaturas de las turbinas y caudal de aire que circula por la caldera. Con objeto de que sea más clara su interpretación y el estado de cada parte de la central, se emplearán imágenes animadas que aclararán el estado en todo momento.
- **Módulo de proceso:** Ejecuta las acciones de mando predefinidas por el operario al iniciar el proceso o leídas en el caso de que el proceso esté iniciado.

El sistema SCADA se encargará de que se cumplan unos objetivos concretados, de no ser así, impondrá las acciones necesarias. La programación se realiza por medio del lenguaje de programación en LabVIEW descrito en el siguiente capítulo.

- **Gestión y archivo de datos:** Se encarga del almacenamiento procesado ordenado de los datos (solo se guardarán datos sobre anomalías de funcionamiento), tanto en una tabla en el propio SCADA, como en un archivo aparte con extensión Excel. Además, con el fin de agilizar el tiempo de respuesta y que el operario se entere de si hay alguna anomalía, se enviará un correo en el caso de que se produzca esta, pudiendo facilitar esa información al operario sin necesidad de estar en la situación del proceso industrial, disminuyendo notablemente el tiempo de respuesta del operario, lo cual puede ser importante debido a las causas que pueden tener lugar en el caso de que no se responda inmediatamente ante el fallo en el proceso industrial.
- **Comunicaciones:** Se encarga de la transferencia de información entre la planta y la arquitectura hardware que soporta el SCADA, y entre ésta y el resto de elementos informáticos de gestión.

### 1.3.1.2 *Filosofía operacional*

El operador pasa ahora a tener menos importancia, ya que el sistema puede tener su propio control, sobre todo por temas de seguridad. El software de la estación maestra requiere hacer más análisis de datos antes de ser presentados a los operadores.

Para algunas instalaciones, los costos que pueden derivar de los fallos de un sistema de control son extremadamente altos.

El Hardware del sistema SCADA es generalmente robusto para resistir condiciones extremas como temperatura, humedad, etc.

La idea es que si hay un posible fallo, se detecte de forma rápida y esa parte pueda ser reemplazada sin necesidad de interrumpir el proceso.

### 1.3.1.3 *Infraestructura y Métodos de comunicación*

Los sistemas SCADA tienen una combinación de radios y señales directas o conexiones módem para conocer los requerimientos de comunicaciones, como es el caso de Ethernet.

Es más, los métodos de conexión entre sistemas pueden incluso que sean a través de Wireless y así no tener que emplear cables (por ejemplo si se quiere enviar la señal a una Tablet o teléfono móvil...).

Para que la instalación de un SCADA sea preferentemente aprovechada, debe cumplir varios objetivos:

- Deben ser sistemas de arquitectura abierta (capaces de adaptarse según necesidades de la empresa).
- Deben comunicarse con facilidad al usuario con el equipo de planta y resto de la empresa (redes locales y gestión).
- Deben ser programas sencillos de instalar, sin exigencias de hardware. También tienen que ser de utilización fácil. En este caso se generará un ejecutable desde el cual se podrá emplear desde cualquier ordenador sin necesidad de tener instalado el programa LabVIEW y sus correspondientes módulos, ya que suponen una exigencia bastante fuerte para el PC.

### 1.3.1.3.1 Transmisión de datos

Los escenarios tradicionales en un Sistema SCADA a nivel de transmisión de datos son básicamente [7]:

- A través de una **interfaz serial RS-232** y distintos protocolos asociados.

La comunicación por RS-232 ha sido la forma más común de un sistema SCADA con un PLC; las primeras implementaciones generaban mensajes bajo un código numérico para solicitar que el PLC ejecutará una acción o proporcionará información, posteriormente se cuentan con implantaciones en las que su programación se realizaba con lenguajes de generación (BASIC, C, etc), esta forma de comunicación se constituye en una conexión primitiva punto a punto, sin flexibilidad de comunicarse con otros equipos.

- A través de una **red protocolo propietario**.

Dada la necesidad de conexiones multipunto entre equipos de cómputo soportando Sistemas SCADA y PLCs los fabricantes se vieron en la necesidad de crear arquitecturas y topologías de redes para conectarlos, de esta forma una aplicación SCADA se puede comunicar con más de un PLC en red.

- A través de una red con **protocolo TCP/IP**.

La creciente demanda de la utilización de estándares abiertos de comunicación, llevó al diseño basado en el protocolo TCP/IP. La utilización de TCP/IP soportando aplicaciones SCADA y PLCs amplió una gama de posibilidades hacia otro tipo de equipos y esquemas tipo IPC.

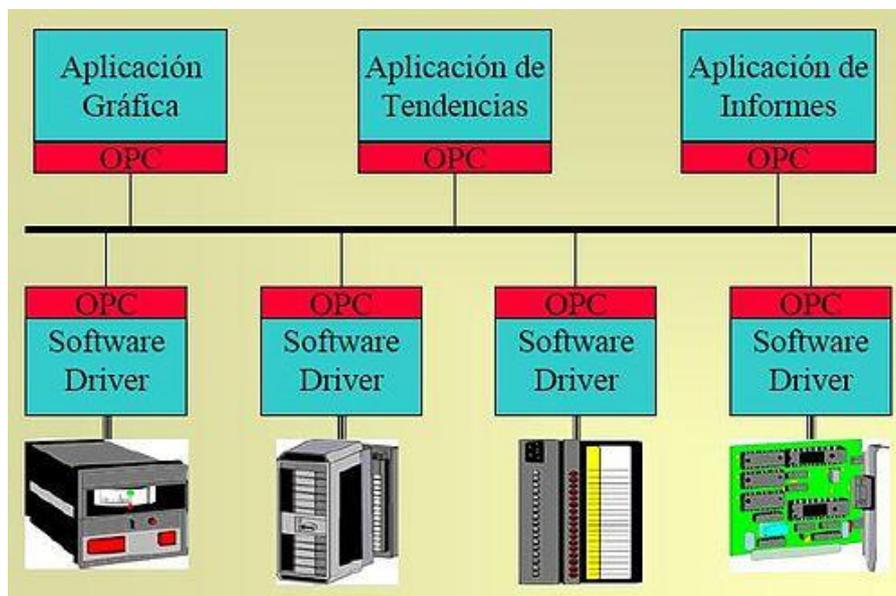
La utilización de TCP/IP, y en general, la adopción del modelo de referencia ISO/OSI, ha implicado considerar una mayor importancia del papel que desempeñan los sistemas operativos que soportan estos protocolos y esquemas de ventaja asociados, siendo importante no olvidar las características de tiempo real que deben soportar estos.

### 1.3.1.3.2

## El estándar OPC

El propósito de OPC (OLE Process Control) (Figura 1.7) es tener una infraestructura estándar para el intercambio de datos de control de procesos [7].

Es típico tener varias fuentes de información en el proceso, las cuales están contenidas en distintos dispositivos tales como controladores programables, medidores, unidades de transferencia remotas, sistemas de control centralizados, base de datos, etc.

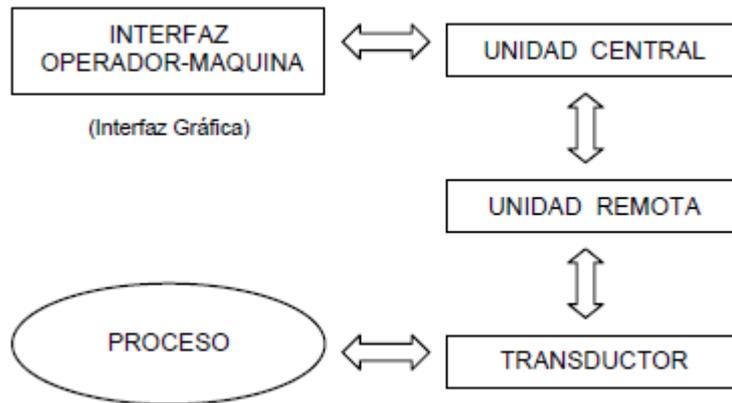


*Figura 1.7: Interconexión OPC (<http://docplayer.es/>)*

Antiguamente estos dispositivos sólo intercambiaban datos con aplicaciones provistas por el mismo fabricante, lo que representaba muchas restricciones. Sin embargo, gracias a OPC, hoy podemos intercambiar libre y fácilmente información desde estos dispositivos y aplicaciones de cualquier tipo.

La OPC es la entidad que establece y mantiene las especificaciones de este esquema de comunicación. En esta organización, sin fines de lucro, participan los principales proveedores de dispositivos para control de procesos y de aplicaciones de software, de modo que se garantiza la interconexión de todos los componentes de distintos fabricantes, sólo es necesario entonces revisar que en las especificaciones se indique que permiten comunicación OPC.

El modelo de comunicación OPC (Figura 1.8) se basa en la estructura cliente-servidor. Es decir, existen servidores que son los responsables de efectuar la comunicación con los dispositivos conforme a lo que los clientes requieran. El software servidor de comunicaciones OPC deberá contener el o los drivers necesarios para la interconexión con los distintos dispositivos, y será el que hace el enlace con las diferentes aplicaciones. Además, puede comunicarse con varios dispositivos distintos y, a su vez, proveer el enlace con diversas aplicaciones al mismo tiempo.



*Figura 1.8: Estructura Cliente-Servidor OPC (Elaboración propia)*

### 1.3.2 CARACTERÍSTICAS DE UN SISTEMA HMI/SCADA

Las características principales de un sistema HMI/SCADA se enumeran a continuación:

- Alto número de canales (y diversidad de dispositivos).
- Registro en base de datos.
- Gestión de Alarmas y Eventos.
- Seguridad.
- Trabajo en red.

### 1.3.3 SOFTWARES SCADA

En la actualidad existen muchísimos Software que permiten realizar aplicaciones SCADA. El uso de uno u otro software para realizar las diferentes aplicaciones de SCADA dependen en gran medida del fabricante del PLC, PAC, DAQ, etc., ya que cada fabricante tiene su propio protocolo de comunicación, claro que esto no es un imposible para establecer una comunicación ya que se puede usar la arquitectura OPC.

Para el presente trabajo se ha decidido emplear el software LabVIEW de National Instruments Corporation, porque han facilitado unas licencias para estudiantes y hay la posibilidad de llevar a cabo la programación a un PLC modular de National Instruments, lo cual albergará lo necesario para realizar el trabajo expuesto.

LabVIEW es un software de programación gráfico liberando a los programadores de la rigidez de las arquitecturas basadas en texto. Al ser un lenguaje de programación gráfico, no solamente va a servir para realizar tareas SCADA, sino también para implementar programas adicionales, según sea el caso.

## CAPITULO 2: SOFTWARE LABVIEW

### 2.1 INTRODUCCIÓN AL ENTORNO DEL SOFTWARE LabVIEW

LabVIEW emplea un lenguaje de propósito general, totalmente gráfico. Usa iconos que enlazados entre sí, formando aplicaciones de la dificultad y nivel que se desee o demande. Lenguajes de programación como C, matlab ó Basic, (la programación se hace mediante líneas de texto, donde las instrucciones determinan la ejecución del programa), pueden dar lugar a una complejidad mayor a la hora de interpretar el programa implementado [7].

Al ser un lenguaje gráfico, simplifica notablemente el desarrollo de los programas minimizando el tiempo de programación, facilitando el entendimiento y mejorando manejo de dicho lenguaje para el diseñador y programador

Para que sea realmente útil trabajar con un lenguaje gráfico, se tiene que tener especial cuidado en la estructuración del programa, empleando las mayores funciones posibles para que no salgan programas desmesuradamente grandes, lo cual puede dar lugar a perder totalmente las ventajas de interpretación, además de ralentizar su ejecución.

Siempre que sea posible se intentará simplificar todo lo máximo posible y ordenarlo siempre con sentido, quedando todo lo más estructurado con los bloques facilitados a la hora de realizar la programación.

LabVIEW es una herramienta diseñada para monitorizar, controlar, automatizar y realizar cálculos complejos de señales analógicas y digitales, capturadas a través de tarjetas de adquisición de datos, puertos serie...

Los programas desarrollados en LabVIEW son llamados Instrumentos Virtuales (VIs).

Un instrumento Virtual (VI) es un módulo de software que simula el Panel Frontal de un instrumento común y apoyándose en elementos de hardware accesibles por el PC. Realiza una serie de medidas como si se tratase de un instrumento real.

## 2.2 MÓDULOS EMPLEADOS

Debido al espacio ocupado de memoria del software LabVIEW, es necesario dividirlo en módulos, de tal manera que solo se instalarán en el PC los módulos que se vayan a utilizar evitando así espacio ocupado en el disco innecesario.

Para el presente trabajo se han empleado los módulos DSC, FPGA, REAL TIME, y VISIÓN DE DESARROLLO, que se detallan a continuación [8], [9].

### 2.2.1 MÓDULO DSC (Datalogging and Supervisory Control)

El Módulo LabVIEW Datalogging and Supervisory Control (DSC) es el complemento de LabVIEW ideal para desarrollar un HMI/SCADA o aplicaciones de registro de datos de muchos canales.

Con LabVIEW DSC se puede desarrollar de manera interactiva un sistema de monitoreo y control distribuido que va desde docenas hasta decenas de miles de etiquetas (Figura 2.9).

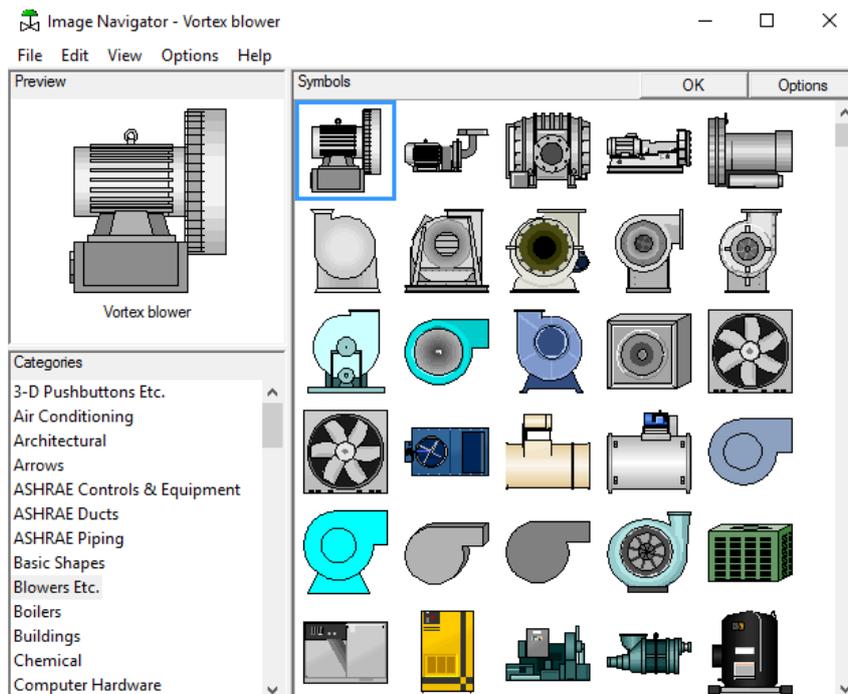


Figura 2.9: Pantalla DSC (Elaboración propia)

Incluye herramientas para registrar de datos a una base de datos histórica en red, rastrear tendencias de datos históricos y en tiempo real, crear redes de objetivos LabVIEW Real-Time y dispositivos OPC en un sistema completo y añadir seguridad a interfaces de usuarios.

Supone una comodidad bastante grande al poder importar figuras prediseñadas, las cuales se pueden emplear para el diseño de la interfaz gráfica. Estas figuras son de la calidad más simple posible, pero con la suficiente información para saber a golpe de vista de que se trata, ya que no interesan figuras o imágenes muy complejas o de mucho peso en memoria, lo cual puede dificultar el rendimiento de nuestro HMI.

Además solapando más de una figura en diversas ocasiones se consigue un GIF animado que ayuda notablemente a interpretar el interface gráfico.

### 2.2.2 MÓDULO FPGA

El Módulo NI LabVIEW FPGA extiende la plataforma de desarrollo gráfico de LabVIEW para programar FPGAs en hardware de E/S reconfigurables (RIO) de NI.

LabVIEW FPGA brinda a los desarrolladores la habilidad de diseñar de manera más eficiente y efectiva sistemas complejos al proporcionar un entorno de desarrollo altamente integrado, un gran ecosistema de bibliotecas de IP, un simulador de alta fidelidad y características de depuración.

El desarrollo tradicional para sistemas basados en FPGA requiere el uso de herramientas de software de bajo nivel y lenguajes de descripción de hardware (HDLs). Aprender y usar de manera efectiva un HDL puede ser un proceso tedioso y tardado.

LabVIEW FPGA ofrece un enfoque de programación gráfica que simplifica la tarea de conectar a E/S y comunicar datos, mejorando enormemente la productividad del diseño y reduciendo el tiempo al mercado.

### 2.2.3 MÓDULO REAL TIME

El Módulo LabVIEW Real-Time es una solución completa para crear sistemas integrados autónomos y confiables con un enfoque de programación gráfica. Como complemento al entorno de desarrollo LabVIEW, el módulo en tiempo real de LabVIEW ayuda a desarrollar y depurar aplicaciones gráficas y deterministas en tiempo real que se pueden descargar y ejecutar en dispositivos de hardware incorporados, como controladores CompactRIO.

### 2.2.4 MÓDULO VISION DEVELOPMENT

El Módulo NI Vision Development (Visión de desarrollo) está diseñado para ayudar a desarrollar e implementar aplicaciones de visión artificial. Incluye cientos de funciones para adquirir imágenes desde múltiples cámaras y para procesar imágenes al mejorarlas, verificar presencia, ubicar partes, identificar objetos y medir partes.

Cuando el procesamiento de imágenes no es suficiente para completar su aplicación, se pueden aprovechar las herramientas y las funciones para comunicarse con otros dispositivos usando una variedad de opciones de E/S y protocolos incluyendo E/S digital, Modbus, Serial RS232, TCP/IP...

## 2.3 FUNDAMENTOS labVIEW

Los programas de LabVIEW son llamados instrumentos virtuales o VIs ya que su apariencia y operación generalmente imitan a los instrumentos físicos [10], [11].

LabVIEW contiene una extensa variedad de herramientas para adquirir, analizar, visualizar y almacenar datos, así como herramientas para ayudarle a solucionar problemas en el código que escriba.

Cuando se crea un nuevo VI, se ven dos ventanas: la ventana del panel frontal y el diagrama de bloques.

### 2.3.1 PANEL FRONTAL

Cuando se abre el VI, aparece la ventana del panel frontal. En esta ventana se diseñará la interfaz gráfica o interfaz de usuario, en la cual, se verá en tiempo real los sucesos o cambios que surgen en cada momento.

Se aclara con más detalles en el Capítulo 3.

### 2.3.2 PALETA DE CONTROLES

La paleta de Controles (Figura 2.10) contiene los controles e indicadores necesarios para crear el panel frontal. Se tiene acceso a la paleta de Controles en la ventana del panel frontal al seleccionar View - Controls Palette o al dar clic con botón derecho en cualquier espacio en blanco en la ventana del panel frontal. La paleta de Controles está dividida en varias categorías; se puede exponer algunas o todas estas categorías para cumplir con sus necesidades.

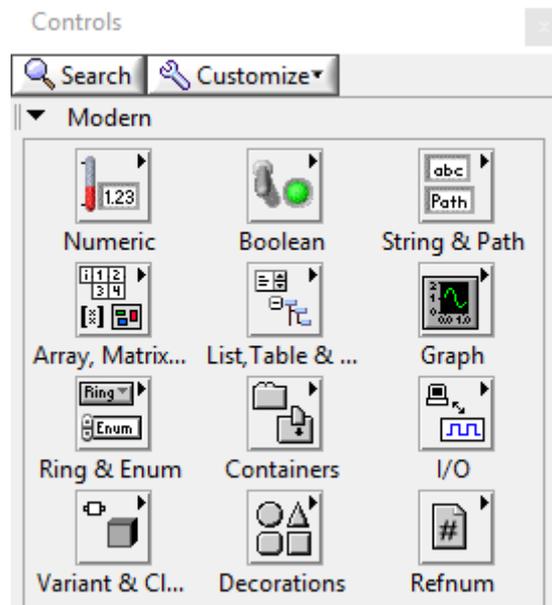


Figura 2.10: Paleta de Controles (Elaboración propia)

### 2.3.3 DIAGRAMA DE BLOQUES

Los objetos del diagrama de bloques incluyen terminales, subVIs, funciones, constantes, estructuras y cables, los cuales transfieren datos junto con otros objetos del diagrama de bloques.

Después de crear la ventana del panel frontal, añada código usando representaciones gráficas de funciones para controlar los objetos del panel frontal.

La ventana del diagrama de bloques contiene este código de fuente gráfica.

#### 2.3.4 TERMINALES

Los objetos en la ventana del panel frontal aparecen como terminales en el diagrama de bloques.

Los terminales son puertos de entrada y salida que intercambian información entre el panel frontal y diagrama de bloques. Son análogos a parámetros y constantes en lenguajes de programación basados en texto.

Los tipos de terminales incluyen terminales de control o indicador y terminales de nodo.

Los terminales de control e indicador pertenecen a los controles e indicadores del panel frontal.

#### 2.3.5 NODOS DE DIAGRAMA DE BLOQUES

Los nodos son objetos en el diagrama de bloques que tienen entradas y/o salidas y realizan operaciones cuando el VI se ejecuta. Son análogos a instrucciones, operaciones, funciones y subrutinas en lenguajes de programación basados en texto. Las estructuras son elementos de control de procesos, como Estructuras de Casos, Ciclos For o Ciclos While.

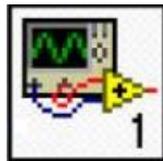
Los nodos más importantes son:

- **Funciones:** Las funciones son los elementos de operación fundamentales de LabVIEW. No tienen ventanas del panel frontal, ni ventanas del diagrama de bloques y no tienen paneles conectores. Una función tiene un fondo amarillo pálido en su ícono.
- **SubVIs:** Un VI llamado desde el diagrama de bloques de otro VI es llamado un subVI. Para crear un subVI, necesita desarrollar un panel conector y crear un ícono.

Un nodo de subVI corresponde a una llamada de subrutina en lenguajes de programación basados en texto. El nodo no es subVI, solamente una instrucción de llamada de subrutina en un programa que no es la propia subrutina.

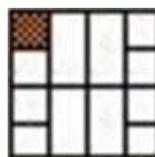
Los indicadores y controles de subVI reciben datos y regresan datos al diagrama de bloques del VI que hace el llamado. Cuando se hace doble clic en un subVI en el diagrama de bloques, aparece la ventana del panel frontal. El panel frontal incluye controles e indicadores. El diagrama de bloques incluye cables, íconos, funciones, subVIs probables u otros objetos de LabVIEW.

Cada VI muestra un ícono en la esquina superior derecha de la ventana del panel frontal (Figura 2.11) y la ventana del diagrama de bloques. El ícono puede contener texto e imágenes. El icono predeterminado contiene un número que indica cuántos nuevos VI abrió después de iniciar LabVIEW.



*Figura 2.11: Icono subVI (Elaboración propia)*

Para usar un VI como un subVI, se necesita construir un panel conector. El panel conector (Figura 2.12) es un conjunto de terminales en el icono que corresponde a los controles e indicadores de ese VI, similares a la lista de parámetros de una función llamada en lenguajes de programación basados en texto.



*Figura 2.12: Icono Panel Conector (Elaboración propia)*

### 2.3.6 PALETA DE FUNCIONES

La paleta de funciones se emplea en el diseño del diagrama de bloques.

Contiene todos los objetos que se emplean en la implementación del programa del VI, ya sean funciones aritméticas, de entrada/salida de señales, entrada/salidas de datos a fichero, adquisición de señales, temporización de la ejecución del programa...

La paleta está dividida en varias categorías (Figura 13). Se pueden mostrar y esconder categorías según necesidades.

Las funciones son los elementos de operación fundamentales de LabVIEW. No tienen ventanas del panel frontal o ventanas del diagrama de bloques pero no tienen paneles conectores. Al hacer doble clic en la función solamente selecciona la función.

Una función tiene un fondo amarillo pálido en su ícono.

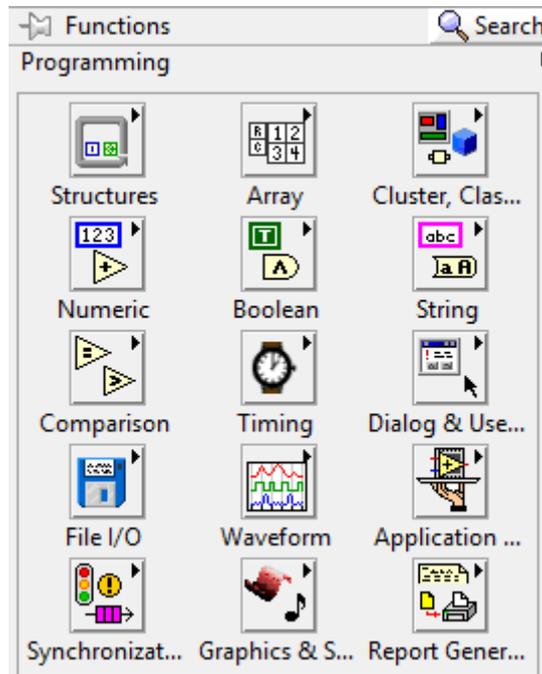


Figura 2.13: Paleta de Funciones (Elaboración propia)

## 2.4 PROGRAMACIÓN GRÁFICA

LabVIEW sigue un modelo de flujo de datos para ejecutar VIs. Un nodo de diagrama de bloques se ejecuta cuando recibe todas las entradas requeridas. Cuando el nodo se ejecuta, produce datos de salida y pasa los datos al siguiente nodo en la trayectoria del flujo de datos. El movimiento de datos a través de los nodos determina el orden de ejecución de los VIs y las funciones en el diagrama de bloques [12].

La mayoría de lenguajes de programación basados en texto siguen un modelo de flujo de control para ejecución del programa. En el flujo de control, el orden secuencial de los elementos del programa determina el orden de ejecución de un programa.

Un ejemplo de programación de flujo de datos se representa en la Figura 2.14. La señal de entrada AI3 del PLC que proviene de un termómetro, tendrá un valor numérico. Este valor se multiplicará por una constante para adecuarlo a unidades de temperatura.

Lo que se está haciendo en todo momento es comprobar que el valor real que marca el termómetro no supera un límite establecido previamente. Si este límite se supera el flujo de datos tendrá como salida un valor numérico (1), si no se supera dicho límite el flujo de datos devolverá un (0).



*Figura 2.14: Ejemplo Flujo de Datos (Elaboración propia)*

En este caso, el diagrama de bloques se ejecuta de izquierda a derecha, no porque los objetos están colocados en ese orden, sino porque la función de ( $\geq$ ) no puede ejecutarse hasta que la función de ( $\times$ ) termine de ejecutarse y pasa los datos a la función de ( $\geq$ ).

Se debe recordar que un nodo se ejecuta solamente cuando los datos están disponibles en todas sus terminales de entrada y proporciona los datos a las terminales de salida solamente cuando el nodo termina la ejecución.

## 2.4.1 CABLES

Transfiere datos entre objetos del diagrama de bloques a través de cables. En la figura 2.18, los cables conectan las terminales de control e indicador a la función de (x) y ( $\geq$ ).

Cada cable tiene una sola fuente de datos, pero se puede cablear a varios VIs o funciones que leen los datos.

Los cables son de diferentes colores, estilos y grosores dependiendo de sus tipos de datos, como se muestra en la Figura 2.15.

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

Figura 2.15: Tabla de cables (<http://www.ni.com>)

Un cable roto (Figura 2.16) aparece como una línea negra punteada con una X roja a la mitad. Los cables rotos ocurren por una variedad de razones, como cuando se intenta cablear dos objetos con tipos de datos no compatibles.



Figura 2.16: Cable Roto (Elaboración propia)

En LabVIEW, se puede usar cables para conectar múltiples terminales para pasar datos en un VI. Se deben conectar los cables a las entradas y salidas que son compatibles

con los datos transferidos con el cable. Por ejemplo, no se puede cablear una salida de tipo Booleano a una entrada numérica.

Además la dirección de los cables debe ser correcta. Se debe conectar los cables solamente a una entrada y por lo menos a una salida. Por ejemplo, no se puede cablear dos indicadores o dos controles juntos.

Los componentes que determinan la compatibilidad del cableado incluyen los tipos de datos del control y/o el indicador y los tipos de datos de la terminal.

## 2.5 ESTRUCTURAS DE DATOS

### 2.5.1 TIPO DE DATOS CADENA DE CARACTERES

Una cadena es una secuencia de caracteres ASCII visibles o no visibles. Las cadenas de caracteres ofrecen un formato independiente a la plataforma para información y datos [13].

Algunas de las aplicaciones más comunes de cadena de caracteres incluyen las siguientes:

- Crear mensajes de texto simples.
- Controlar instrumentos al enviar comandos de texto al instrumento y regresar valores de datos en la forma de ASCII o cadena de caracteres binarias, las cuales después se pueden convertir en valores numéricos.
- Almacenar datos numéricos. Para almacenar datos numéricos en un archivo ASCII, primero se deben convertir los datos numéricos en cadena de caracteres antes de escribir a un archivo de disco.
- Instruir o advertir al usuario con ventanas de diálogo.

En el panel frontal, las cadenas de caracteres aparecen como tabla, cuadros de texto y etiquetas. LabVIEW incluye VIs integrados y funciones que se pueden usar para manipular secuencias, incluyendo formateo de cadena de caracteres, análisis de cadena de caracteres y otras ediciones.

LabVIEW representa datos de cadena de caracteres en color **rosa** (Figura 2.17).



*Figura 2.17: Cadena de Caracteres (Elaboración propia)*

## 2.5.2 TIPO DE DATO NUMÉRICO

LabVIEW representa datos numéricos como números de punto flotante, números de punto fijo, números enteros, números enteros sin signo y números complejos.

La precisión puede ser doble o simple.

Los datos numéricos complejos o números con decimales son representados con el color **naranja** en LabVIEW.

Todos los datos numéricos enteros son representados con el color **azul**.

Hay algunos datos que ofrecen opciones de configuración extendida. Por ejemplo, se puede asociar unidades físicas de medida con datos de punto flotante, incluyendo números complejos.

## 2.5.3 TIPO DE DATO BOOLEANO

LabVIEW almacena datos Booleanos como valores de 8 bits. Se puede usar un Booleano en LabVIEW para representar un 0 o 1 o un TRUE o FALSE. Si el valor de 18 bits es cero, el valor Booleano es FALSE.

Cualquier valor no igual a cero representa TRUE.

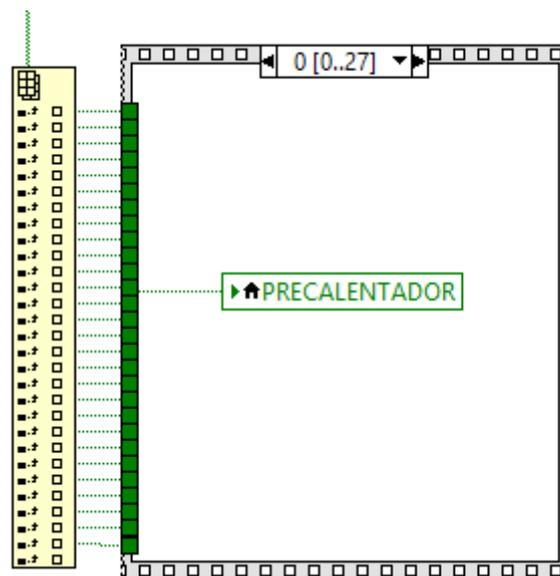
Las aplicaciones comunes para datos Booleanos incluyen representar datos digitales y servir como un control de panel frontal que actúa como un conmutador que tiene una acción mecánica generalmente usada para controlar una estructura de ejecución como una estructura de Caso.

Un control Booleano generalmente es usado como la declaración condicional para terminar un Ciclo While.

En LabVIEW el color **verde** representa datos Booleanos.

## 2.5.4 MATRICES

Cuando la cantidad de datos a manejar es elevada, es muy cómodo emplear matrices y secuencia de estructuras, para que quede mucho más estructurado y compacto, lo que hace que el código sea mucho más legible y de mucho menor tamaño como se muestra en la figura 2.18.



*Figura 2.18: Estructura Matriz en secuencia (Elaboración propia)*

Físicamente se puede trabajar con las matrices y recorrer uno a uno cada elemento de la misma, trabajando como si estuviera todos los datos separados o fuesen independientes.

De esta forma se pueden comparar valores de matrices distintas y ver cuál ha sido el cambio, variación, etc.

### 2.5.5 CLUSTERS

Los clusters agrupan elementos de datos de diferentes tipos. Un ejemplo es el cluster de error de LabVIEW, el cual combina un valor Booleano, un valor numérico y uno de cadena de caracteres.

Es similar a un registro o a una estructura en lenguajes de programación basados en texto.

Construir varios elementos de datos en clusters elimina el desorden de cables en el diagrama de bloques y reduce el número de terminales del panel conector que los subVIs necesitan.

El panel conector tiene, a lo más, 28 terminales. Si el panel frontal contiene más de 28 controles e indicadores que quiera pasar a otro VI, se puede agrupar algunos de ellos en un cluster y asignarlo a una terminal en el panel conector.

La mayoría de ellos tienen un patrón en el diagrama de bloques de cable **rosa** y terminal de tipos de datos.

Los clusters de error tienen un patrón de cable **amarillo oscuro** y terminal de tipo de datos.

## 2.6 ESTRUCTURAS DE EJECUCIÓN

Las estructuras de ejecución contienen secciones de código gráfico y controlan cómo y dónde el código dentro se ejecuta [14].

Las estructuras de ejecución más comunes son **Ciclos While**, **Ciclos For** y Estructuras de **Casos** los cuales se pueden usar para ejecutar la misma sección del código varias veces o para ejecutar una sección diferente del código basada en alguna condición.

### 2.6.1 CICLOS

#### 2.6.1.1 Ciclos While

Un Ciclo While, ejecuta el código que contiene hasta que ocurre una condición.

La Figura 2.19 muestra un Ciclo While en LabVIEW, un diagrama de flujo equivalente a la funcionalidad de Ciclo While.

El Ciclo While ejecuta el código que contiene hasta la terminal condicional, un terminal de entrada, recibe un valor Booleano específico.

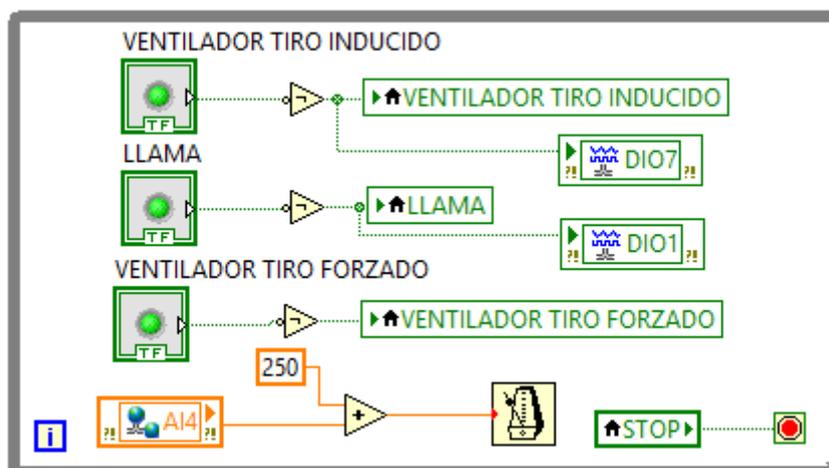


Figura 2.19: Ciclo While (Elaboración propia)

El ciclo While de la Figura 2.19, se repetirá constantemente hasta que se cumpla la condición de que (STOP) se haga.

El terminal de iteración es una terminal de salida que contiene el número de iteraciones terminadas.

La cantidad de iteraciones para el Ciclo While siempre comienza en cero.

El Ciclo While siempre se ejecuta por lo menos una vez.

#### 2.6.1.1.1 Túneles de Estructura

Los túneles alimentan datos desde y hacia estructuras.

El túnel aparece como un bloque sólido en el borde del Ciclo While.

El bloque es el color del tipo de datos cableado al túnel.

Los datos salen fuera de un ciclo después de que el ciclo termina.

Cuando un túnel pasa datos a un ciclo, el ciclo se ejecuta solamente después que los datos llegan al túnel.

Solamente el último valor de la terminal de iteración se muestra en el indicador de iteraciones.

### 2.6.1.2 Ciclos for

Un Ciclo For ejecuta un subdiagrama un número de veces establecido.

La Figura 2.20 muestra un Ciclo For en LabVIEW

El terminal de conteo es un terminal de entrada cuyos valores indican cuantas veces se repite el subdiagrama.



El terminal de iteración es un terminal de salida que contiene el número de iteraciones terminadas.



La cantidad de iteraciones para el Ciclo For siempre comienza en cero.

El Ciclo For difiere del Ciclo While en que el Ciclo For ejecuta un número de veces establecido. Un Ciclo While detiene la ejecución solamente si existe el valor en el terminal condicional.

Un ciclo For puede venir muy bien para recorrer los valores de una matriz sin necesidad de ponerle las iteraciones que debe hacer, ya que cuando termine de recorrer todos los valores de la matriz automáticamente saldrá del ciclo.

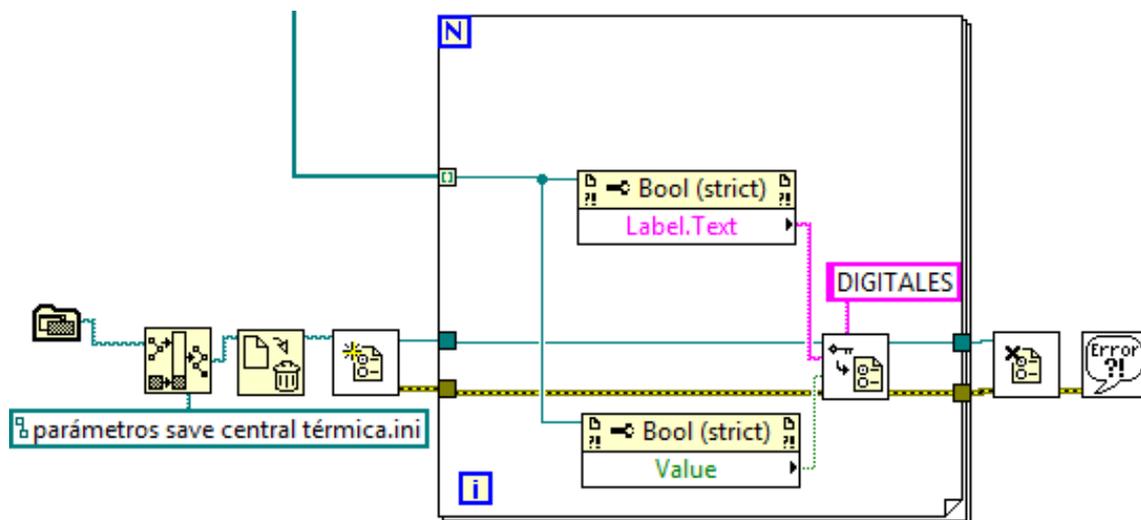


Figura 2.20: Ciclo For (Elaboración propia)

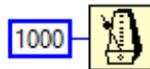
### 2.6.1.2.1

## Añadir temporización

Cuando un ciclo termina de ejecutar una iteración, inmediatamente comienza la siguiente iteración, a menos que alcance una condición de paro. A menudo se necesita controlar la frecuencia y la temporización de la iteración.

Por ejemplo, si se está adquiriendo datos y se desea adquirir los datos una vez cada 10 segundos, se necesita una manera de temporizar las iteraciones del ciclo para que ocurran una vez cada 10 segundos (Figura 2.21).

Aún si no se necesita que la ejecución ocurra a una cierta frecuencia, se necesita proporcionar al procesador el tiempo para completar otras tareas, como responder a la interfaz de usuario.



*Figura 2.21: Temporización Ciclo (Elaboración propia)*

### 2.6.1.3 Ciclos Case Structure

Una estructura de Casos tiene dos o más subdiagramas o casos.

Solamente un subdiagrama es visible a la vez y la estructura ejecuta solamente un caso a la vez. Un valor de entrada determina cual subdiagrama se ejecuta.

La etiqueta del selector de caso en la parte superior de la estructura de Caso (Figura 2.22) contiene el nombre del valor del selector que corresponde al caso en el centro y a las flechas de incremento y reducción a cada lado.

Si el tipo de datos de la terminal del selector es Booleano, la estructura tiene un caso True o un caso False.

Si la terminal del selector es un entero, secuencia o valor de tipo enumerado, la estructura tiene cualquier número de casos.

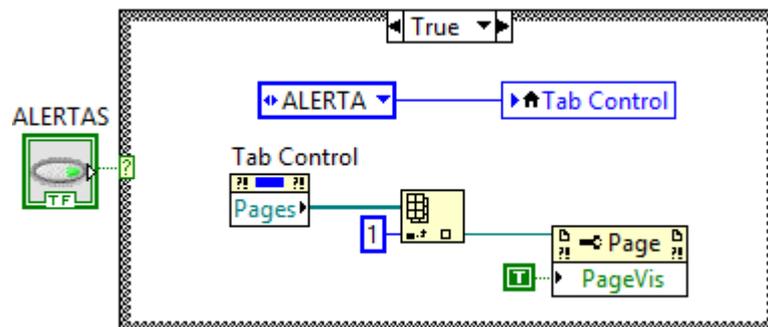


Figura 2.22: Ciclo Case Structure (Elaboración propia)

#### 2.6.1.4 Ciclos Flat Sequence

Consiste en uno o más sub-diagramas (representados como ejemplo en la Figura 2.23), o marcos, que se ejecutan secuencialmente. Se utiliza la estructura de secuencia plana para asegurar que un subdiagrama se ejecuta antes o después de otro subdiagrama.

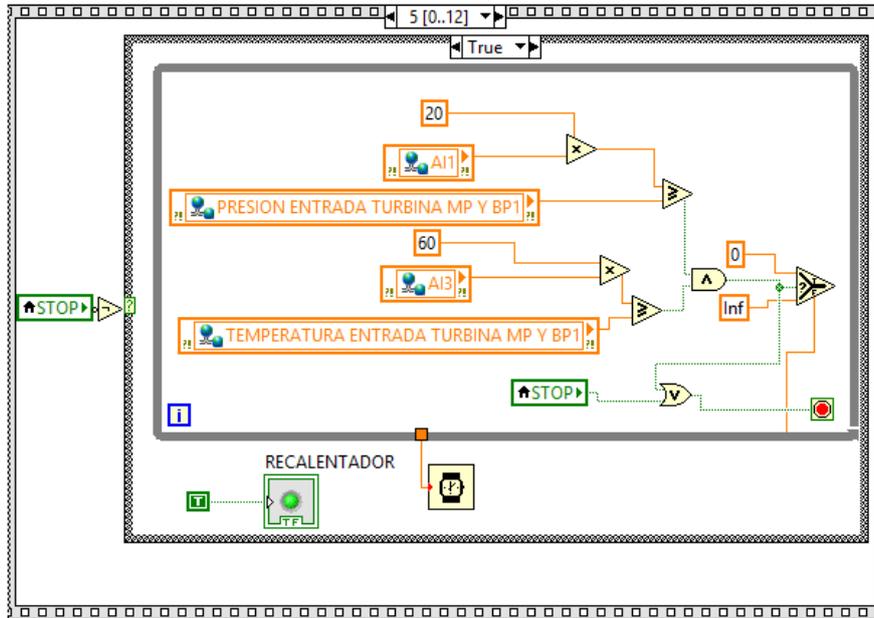


Figura 2.23: Ciclo Flat Sequence (Elaboración propia)

El flujo de datos para la estructura de secuencia plana difiere del flujo de datos para otras estructuras. Los marcos en una estructura de secuencia plana se ejecutan de izquierda a derecha y cuando todos los valores de datos conectados a un marco están disponibles. Los datos salen de cada trama cuando el marco termina de ejecutarse. Se puede ver en la Figura 2.23.

A diferencia de la estructura de secuencias apiladas, no es necesario utilizar secuencias locales para pasar datos de marco a marco en la estructura de secuencias planas. Dado que la estructura de secuencia plana muestra cada trama en el diagrama de bloques, se puede cablear de marco a cuadro sin utilizar secuencias locales y sin ocultar código.

Cuando se agrega o elimina tramas en una estructura de secuencia plana, la estructura se redimensiona automáticamente.

No se pueden arrastrar túneles a través de los marcos de una estructura de secuencia plana.

## 2.7 VARIABLES LOCALES

Uno de los elementos más usados dentro de la programación son las variables, las cuales ayudan a colocar o generar los datos para ser usados dentro de uno u otro subprograma, sin la necesidad de estar escribiendo el valor del dato cada vez, sino que solamente se coloca el nombre de la variable y se le asigna el valor una sola vez produciéndose el cambio del valor donde dicha variable se encuentre.

Las variables locales en éste caso sirven para trabajar solo dentro del mismo programa, no siendo así con las variables compartidas.

Las variables locales son utilizadas para transferir datos entre lazos o también para evitar cableados largos como se representa en la Figura 2.24, ya que dicha variable se puede usar como indicador o control desde una o más localizaciones dentro del programa.

Las variables locales ayudan de alguna manera a ahorrar y romper con el paradigma de flujo de datos.



Figura 2.24: Variable y Variable Local (Elaboración propia)

## 2.8 VARIABLES COMPARTIDAS (SHARED VARIABLES)

Existen aplicaciones en las cuales es necesario comunicar datos entre ordenadores u otros dispositivos para lo cual LabVIEW ofrece una buena forma de hacerlo, con las llamadas variables compartidas.

Con las variables compartidas, los VIs en diferentes máquinas o redes pueden leerse o escribirse sin necesidad de estar programando una red o algo más complejo para tener información una máquina de otra.

Se distinguen del resto de variables por dos esferas azules ubicadas a la izquierda del texto como se ve en la Figura 2.25.



**Figura 2.25: Variable Compartida (Elaboración propia)**

## 2.9 AVISO CORREO ELECTRÓNICO

El operario o persona encargada del control del proceso no va a estar pendiente siempre del SCADA, es más puede que no lo tenga delante, o que este en casa tranquilamente sin que tenga constancia del estado del proceso.

Para ello viene muy bien la opción que permite LabVIEW para enviar un aviso por correo electrónico en el caso de que se produzca una alarma.

De esta manera permite que el aviso pueda llegar mediante correo electrónico al móvil, ordenador, o Tablet, pudiendo dar aviso al operario en cualquier lugar y agilizar mucho la respuesta de este.

Además permite al operario más flexibilidad frente a averías dejando más responsabilidad al SCADA.

Los correos electrónicos sirven como historial de alarmas, y gracias a la descripción que se puede enviar en el correo, se puede detallar lo necesario en función de la alerta que surja.

Se muestra la programación en la Figura 2.26.

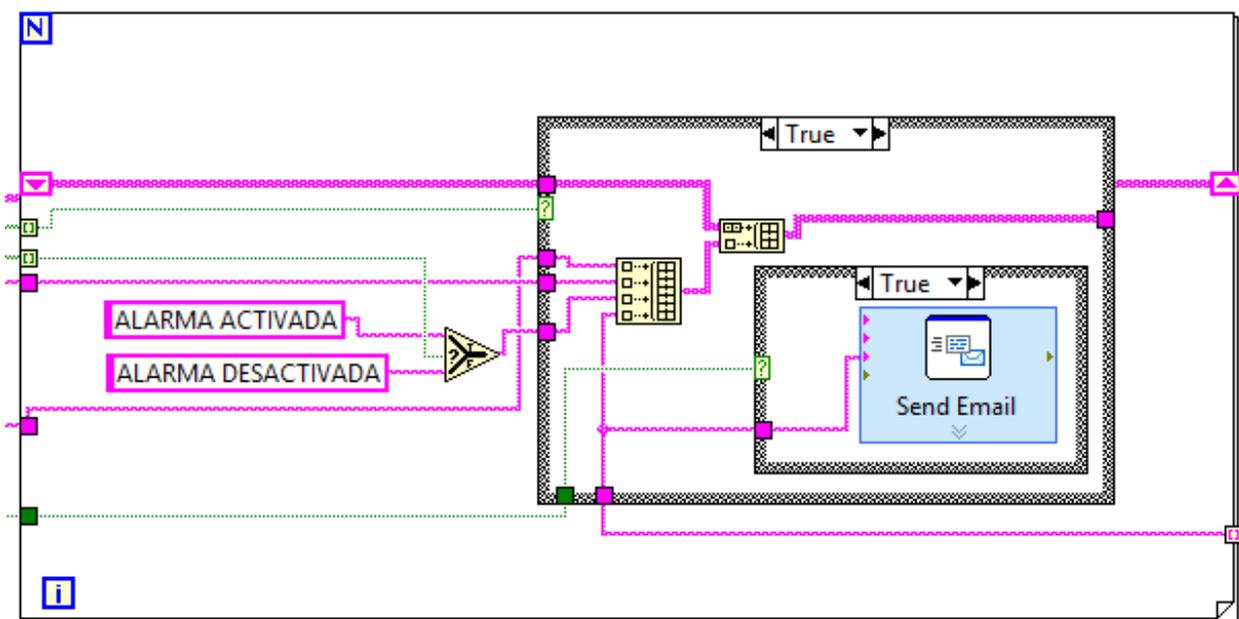


Figura 2.26: Programación Correo Electrónico (Elaboración propia)

Para que el correo electrónico llegue correctamente, se debe configurar el correo para su uso desde terceros habilitando el acceso IMAP, en los ajustes de Gmail dentro de Reenvío y correo POP/IMAP.

También se debe configurar el configurador que facilita LabVIEW, que se ejecutará cada vez que se necesite enviar un correo. Este configurador se muestra en la Figura 2.27.

Configure Send Email [Send Email]

**User Information**

Sender's Email Address  
chaul1209@gmail.com

Outgoing Mail Server (SMTP)  
smtp.gmail.com

Port  
587

Require Log-in

Username  
chaul1209@gmail.com

Password  
\*\*\*\*\*

Use Secure Connection (TLS\SSL)

**Email**

Recipients  
chaul1209@gmail.com

Subject  
Ha habido una anomalia en el funcionamiento de la Central

Message

Send Test Email OK Cancel Help

Figura 2.27: Programación Labview último estado de ref. (Elaboración propia)

Obviamente para que el correo llegue correctamente es necesario la conexión a internet.

## 2.10 ALMACENAMIENTO DE DATOS

Cuando se ejecuta el SCADA tiene una secuencia de inicio en la cual, realiza unos pasos previos como la secuencia de la entrada de aire por la caldera a través de unos ventiladores o el tratamiento de calor del agua para su posterior paso a las turbinas.

Es lógico pensar que si por cualquier circunstancia el SCADA se cierra, al abrir no debe empezar desde el inicio si no que tendrá que comenzar por el último sitio dónde lo había dejado.

En el caso de una Central Térmica, el número de horas que esta parada anualmente son muy pocas, por lo que casi siempre el proceso estará trabajando y por ello en el SCADA se almacenarán esos datos con el fin de continuar las etapas del proceso.

En el caso de que se realice una parada programada de la Central y se quieran reiniciar los parámetros para implementar el ciclo desde el inicio, se tendrán que poner los valores por defecto, situada en la pestaña « VIEW ».

En la Figura 2.28 se muestra el guardado del último estado referenciado por el SCADA.

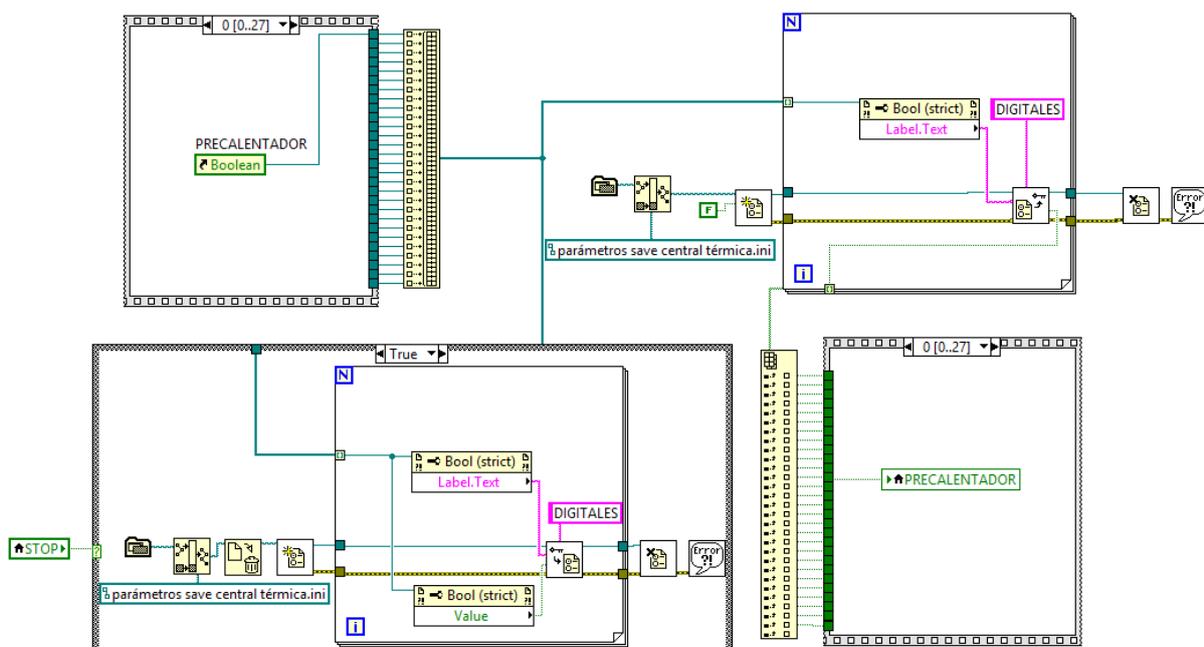
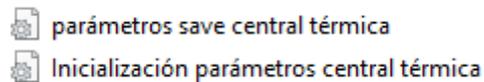


Figura 2.28: Programación LabVIEW último estado de ref. (Elaboración propia)

La forma de funcionamiento es el recorrido de cada parámetro uno a uno, gracias a un bucle For, va leyendo cada parámetro y va guardando su valor, hasta que se recorre todos los elementos posibles y sale del bucle.

Este pequeño programa genera un archivo (.ini), el cual está ubicado en la misma carpeta donde tenemos guardado todos los archivos.

Su icono es como una hoja con un engrane como se muestra en la Figura 2.29.



**Figura 2.29: Programación LabVIEW último estado de ref. (Elaboración propia)**

El archivo guardado contendrá los datos referentes al estado de cada variable como se muestra en la Figura 2.30.

```
[[DIGITALES]
PRECALENTADOR = TRUE          TURBINA MEDIA PRESIÓN = TRUE
ENTRADA AIRE CALDERA = TRUE   ENTRADA MP Y BP = FALSE
ENTRADA AIRE CALDERA 2 = TRUE TURBINA BAJA PRESION 2 = FALSE
BOMBA ALIMENTACION CALDERA = TRUE EXT 2 = FALSE
ECONOMIZADOR = TRUE          EXT 3 = FALSE
ALIMENTACION AGUA = TRUE     CONDENSADOR = FALSE
CALDERIN = TRUE              ENTRADA TORRE = FALSE
CURVA CALDERIN = TRUE        BOMBA REFREIGERACION = FALSE
SOBRECALENTADOR = TRUE      SALIDA TORRE = FALSE
TURBINA ALTA PRESION 2 = FALSE BOMBA CONDENSADOR = FALSE
ENTRADA TURBINA AP = FALSE   ENTRADA INTER MP = FALSE
EXT 1 = FALSE                INTERCAMBIADOR 2 = FALSE
PRODUCCION = FALSE          CANAL ENTR INTER AP = FALSE
RECALENTADOR = FALSE        INTERCAMBIADOR 1 = FALSE
```

**Figura 2.30: Archivo estado referencia (Elaboración propia)**

## 2.11 LECTURA DE DATOS

Dentro de los archivos generados se guardan los parámetros que hayamos indicado, y por supuesto, se pueden configurar fácilmente en cualquier momento sin necesidad de tener instalado el programa LabVIEW, ya que su apertura se realiza como un bloc de notas, como se muestra en la Figura 2.31.

```

[CONSIGNAS PROCESO CENTRAL TÉRMICA]
TEMPERATURA ENTRADA TURBINA AP = 560,000000
TEMPERATURA ENTRADA TURBINA MP Y BP = 560,000000
PRESIÓN ENTRADA TURBINA AP = 180,000000
PRESIÓN ENTRADA TURBINA MP Y BP = 180,000000

[CONSIGNAS VALORES MÁXIMOS]
TEMPERATURA MÁXIMA ENTRADA TURBINA AP = 590,000000
TEMPERATURA MÁXIMA ENTRADA TURBINA MP Y BP = 590,000000
PRESIÓN MÁXIMA ENTRADA TURBINA AP = 190,000000
PRESIÓN MÁXIMA ENTRADA TURBINA MP Y BP = 190,000000

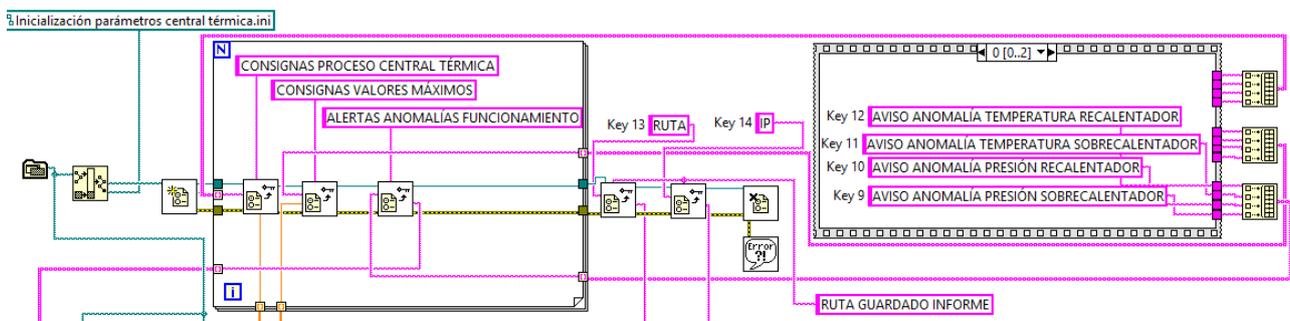
[ALERTAS ANOMALÍAS FUNCIONAMIENTO]
AVISO ANOMALÍA PRESIÓN SOBRECALENTADOR = "¡ALTA PRESIÓN SALIDA SOBRECALENTADOR!"
AVISO ANOMALÍA PRESIÓN RECALENTADOR = "¡ALTA PRESIÓN SALIDA RECALENTADOR!"
AVISO ANOMALÍA TEMPERATURA SOBRECALENTADOR = "¡ALTA TEMPERATURA SALIDA SOBRECALENTADOR!"
AVISO ANOMALÍA TEMPERATURA RECALENTADOR = "¡ALTA TEMPERATURA SALIDA RECALENTADOR!"

[RUTA GUARDADO INFORME]
RUTA = "C:\TFG\CENTRAL TÉRMICA"

[IP]
RUTA = "XXX.XX.XX.XXX"
    
```

**Figura 2.31: Archivo lectura parámetros (Elaboración propia)**

Una vez generado estos archivos, el programa debe leerlos cuando se necesite (se muestra en la Figura 2.32), para ello de forma similar a la escritura, se leen los parámetros uno por uno, y se almacenan.



**Figura 2.32: Lectura archivo (Elaboración propia)**

Los datos que se recogen en la lectura se almacenan en variables tipo DBL (Figura 2.33), las cuales van a servir para que el programa trabaje con los márgenes exigidos oportunamente.

TEMPERATURA ENTRADA TURBINA AP	TEMPERATURA MÁXIMA ENTRADA TURBINA AP
<input type="text" value="560"/>	<input type="text" value="590"/>
TEMPERATURA ENTRADA TURBINA MP Y BP	TEMPERATURA MÁXIMA ENTRADA TURBINA MP Y BP
<input type="text" value="560"/>	<input type="text" value="590"/>
PRESION ENTRADA TURBINA AP	PRESIÓN MÁXIMA ENTRADA TURBINA AP
<input type="text" value="180"/>	<input type="text" value="190"/>
PRESION ENTRADA TURBINA MP Y BP	PRESIÓN MÁXIMA ENTRADA TURBINA MP Y BP
<input type="text" value="180"/>	<input type="text" value="190"/>

**Figura 2.33: Recogida de datos archivo (Elaboración propia)**

## 2.12 DIAGRAMA DE FLUJO CENTRAL TÉRMICA

Para facilitar la comprensión del Diagrama de bloques implementado con LabVIEW, se pueden esquematizar en un Diagrama de Flujo en el que se reflejan los pasos básicos que se dan en el proceso de la Central Térmica. Dichos Diagramas se representan en las Figuras 2.34 y 2.35.

Estos diagramas utilizan símbolos con significados definidos que representan los pasos del algoritmo, y representan el flujo de ejecución mediante flechas que conectan los puntos de inicio y de fin del proceso.

El proceso empieza con la circulación de aire dentro de la Caldera y una combustión por inyección de combustible.

El aire una vez empleado en la Caldera, saldrá por la Chimenea después de ser filtrado. Esta combustión ayudará a aportar energía al circuito de agua, el cual se centrará este trabajo.

Al entrar el agua en la Caldera, pasará por el sobrecalentador y recalentador, obteniendo así una presión de 180 Bar y una temperatura de 580°. Hasta que estos datos no lleguen a conseguirse el vapor no entrará en la Turbina de AP, ya que si entra por debajo de estos valores se corre peligro de que el vapor entre sin evaporar completamente, y puede ser perjudicial para la Turbina.

De igual forma también hay que controlar que la presión o la temperatura no exceda unos límites los cuales puedan afectar al funcionamiento de la instalación y pueda ser un peligro en seguridad.

Una vez el vapor salga de la Turbina de AP, se introduce de nuevo en la Caldera para volver a recuperar la presión y la temperatura necesaria para poder entrar en la Turbina de MP Y AP.

Después de que el vapor pase por la Turbina de BP se condensará en el Condensador. Para ello hay un circuito cerrado e independiente, que servirá para disminuir la presión y temperatura del circuito general. El circuito de refrigeración está formado por agua en estado líquido y se mueve gracias a una bomba.

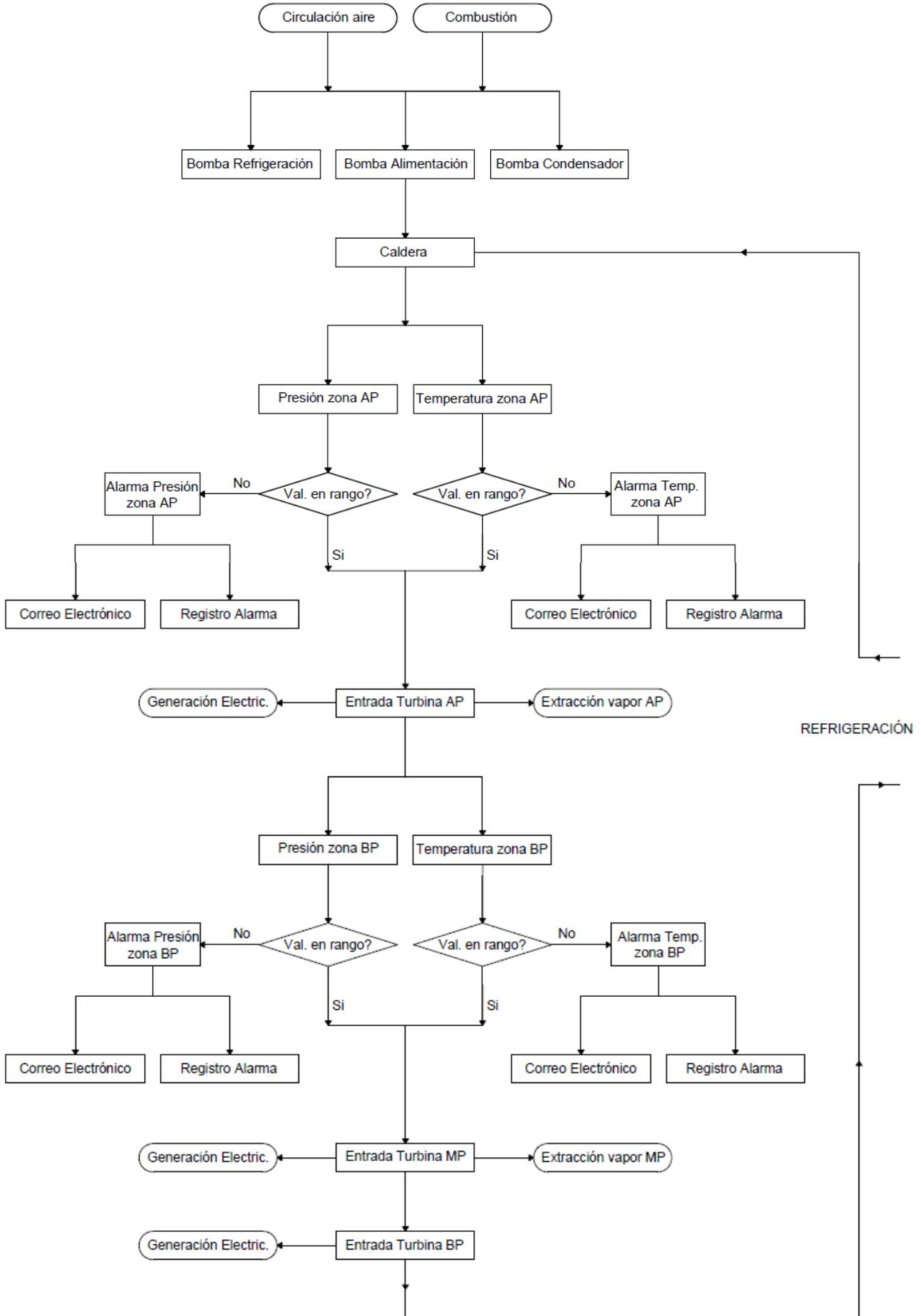
En el condensador se unen los dos circuitos, funciona exactamente igual que un intercambiador y su misión es la misma.

Una vez bajada la temperatura y la presión del fluido se introduce este en dos intercambiadores de calor, con el fin de aumentar la temperatura del fluido haciendo extracciones de vapor en la Turbina de AP y de MP. Esto se emplea para aumentar notablemente el rendimiento y no tener que aportar tanto calor en la caldera, lo que se transfiere a un ahorro de combustible importante.

El fluido vuelve a entrar en la Caldera para volver a conseguir la temperatura y presión necesarias para volver a entrar a la Turbina de AP

El ciclo es un ciclo cerrado por lo que siempre hará el mismo recorrido y solo se tendrá que aportar agua al circuito de refrigeración ya que parte del vapor se irá por la Torre de Refrigeración.

Este Diagrama de Flujo se adjunta también con más detalle en anexo 2.



**Figura 2.34: Diagrama de Flujo 1 (Elaboración propia)**

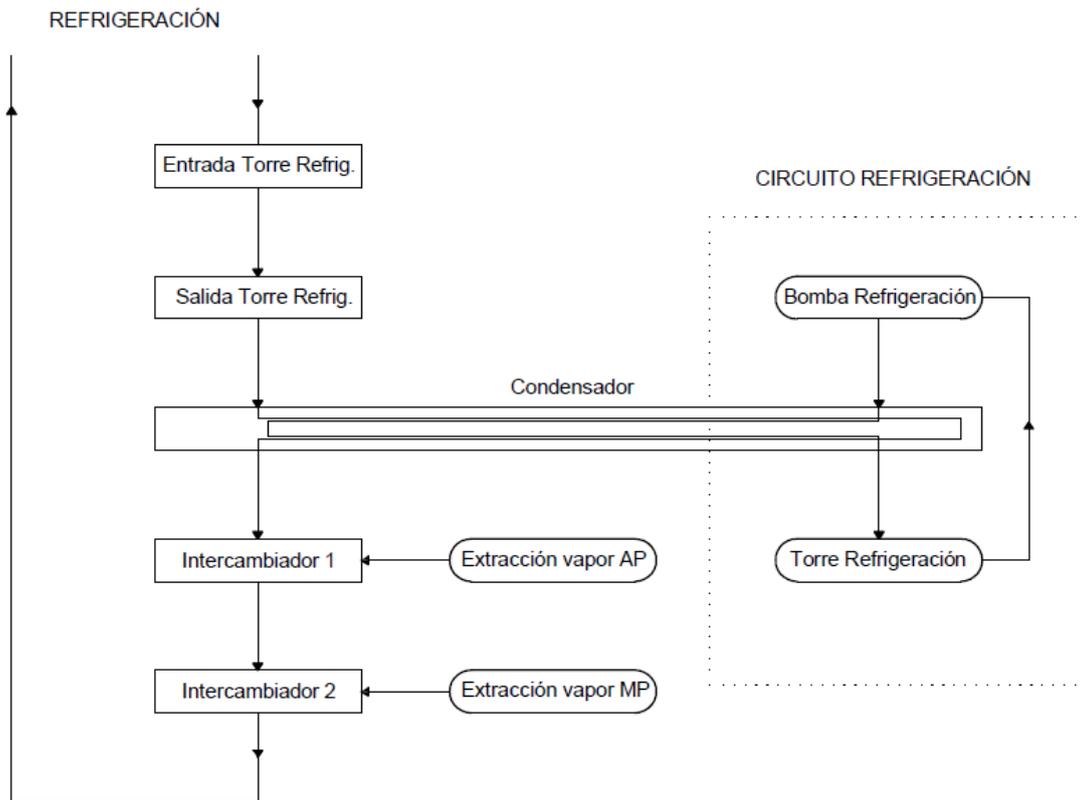


Figura 2.35: Diagrama de Flujo 2 (Elaboración propia)

## 2.13 DIAGRAMA DE BLOQUES EN LabVIEW

Constituye el código fuente del VI.

En el diagrama de bloques es donde se realiza la implementación del programa del VI para controlar o realizar cualquier procesado de las entradas y salidas que se crearon en el panel frontal.

Incluye funciones y estructuras integradas en las librerías que incorpora LabVIEW. En el lenguaje G las funciones y las estructuras son nodos elementales. Son análogas a los operadores o librerías de funciones de los lenguajes convencionales.

Los controles e indicadores que se colocaron previamente en el Panel Frontal, se materializan en el diagrama de bloques mediante los terminales.



## CAPITULO 3: DISEÑO PANTALLA SCADA

### 3.1 DISEÑO INTERFAZ GRÁFICA

Para la construcción del SCADA, se debe pasar por diseñar una interfaz de usuario o interfaz gráfica.

En ella se ve en tiempo real los sucesos o cambios que ha habido en cada momento.

Dicha ventana tiene dos pestañas (Figura 3.36), las cuales se escogerán con el botón de alertas:

- La primera pestaña constará de la interfaz gráfica central, la cual dará toda la información relevante como presiones y temperaturas. También ayuda a ver que partes está en funcionamiento o no.

Esta interfaz es la parte más delicada del trabajo, ya que debe ofrecer una gran cantidad de datos de una forma rápida y de fácil interpretación.

Normalmente la pestaña que se empleará será esta, y la pestaña de alarmas quedará como opción de aclaración o historial.

- La segunda pestaña está diseñada para poder ver una tabla, un historial sencillo de anomalías de la Central.

En este historial se puede ver:

- Fecha en la que se ha activado/desactivado alerta.
- Hora en la que se activado/desactivado alerta.
- El estado de la alerta, indicará si se ha activado o desactivado la alerta.
- Detalle de la alerta.



### 3.1.1 CONTROLES E INDICADORES

Los controles e indicadores empleados en el diseño de la interfaz gráfica han sido modificados físicamente con el único objetivo de facilitar la comprensión y disminuir el tiempo de actuación del usuario.

- Los controles generalmente son botones, deslizadores y secuencias. Simulan dispositivos de entrada y suministran datos al diagrama de bloques del VI.
- Los indicadores generalmente son gráficas, tablas, LEDs y secuencias de estado. Los indicadores simulan dispositivos de salida y muestran los datos que diagrama de bloques adquiere o genera.

Cada control o indicador tiene un tipo de datos asociado con él.

Los tipos de datos utilizados con mayor frecuencia son:

- **Controles e indicadores Numéricos:** El tipo de datos numérico pueden representar números de varios tipos como un entero o real.

En el presente trabajo se han empleado indicadores numéricos para la visualización de la temperatura y presiones de las turbinas de alta, media y baja presión (Figura 3.37).

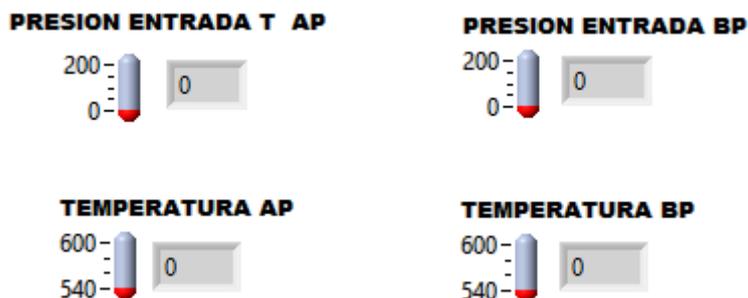


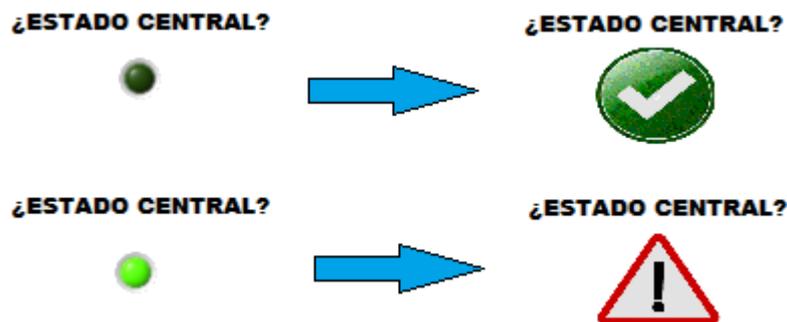
Figura 3.37: Indicadores Numéricos (Elaboración propia)

- **Controles e indicadores Booleano:** El tipo de datos Booleano representa datos que solamente tienen dos estados posibles, (ON y OFF).

Se usan los controles e indicadores Booleano para proporcionar y visualizar valores Booleano. Los objetos Booleano simulan interruptores, botones y LEDs.

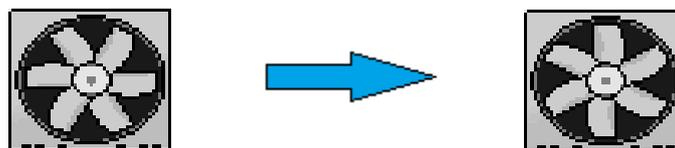
La mayoría de indicadores de la interfaz diseñados son del tipo Booleano, con el fin de ver de forma muy rápida el cambio de estado de cualquier activación/desactivación de una salida.

Además se han creado archivos en CTL que emplea como base un piloto LED y se puede modificar con las figuras ON/OFF que se deseen (Figura 3.38).



*Figura 3.38: Adaptación indicador CTL (Elaboración propia)*

También se pueden crear imágenes animadas, haciendo una secuencia ON/OFF rápida. De esta forma se crea una sensación de movimiento muy útil, como el movimiento de los ventiladores (Figura 3.39) o del fuego.



*Figura 3.39: CTL animado (Elaboración propia)*

- **Controles e indicadores de Cadena de Caracteres:** El tipo de datos de caracteres es una secuencia de caracteres ASCII.

Se usan controles en cadena para recibir texto desde el usuario.

Se usan indicadores en cadena para mostrar textos al usuario (Figura 3.40).

En el diseño de la interfaz se ha creado una sub-pestaña con una tabla en la que se dará información sobre el estado de la central.

<b>DIA</b>	<b>HORA</b>	<b>ALARMA</b>	<b>ALERTA FUNCIONAMIENTO</b>
martes, 4 de abril de 2017	13h 23m	ALARMA ACTIVADA	¡ALTA PRESIÓN SALIDA SOBRECALENTADOR!
martes, 4 de abril de 2017	14h 01m	ALARMA DESACTIVADA	¡ALTA PRESIÓN SALIDA SOBRECALENTADOR!
lunes, 17 de abril de 2017	15h 46m	ALARMA ACTIVADA	¡ALTA PRESIÓN SALIDA SOBRECALENTADOR!
lunes, 17 de abril de 2017	15h 47m	ALARMA DESACTIVADA	¡ALTA PRESIÓN SALIDA SOBRECALENTADOR!
lunes, 8 de mayo de 2017	11h 05m	ALARMA ACTIVADA	¡ALTA TEMPERATURA SALIDA SOBRECALENTADOR!
lunes, 8 de mayo de 2017	11h 05m	ALARMA DESACTIVADA	¡ALTA TEMPERATURA SALIDA SOBRECALENTADOR!
jueves, 11 de mayo de 2017	10h 52m	ALARMA ACTIVADA	¡ALTA TEMPERATURA SALIDA SOBRECALENTADOR!
jueves, 11 de mayo de 2017	10h 53m	ALARMA DESACTIVADA	¡ALTA TEMPERATURA SALIDA SOBRECALENTADOR!

*Figura 3.40: Tabla indicadora de cadena de caracteres (Elaboración propia)*

## 3.2 VISUALIZACIÓN DEL SCADA VÍA WEB

LabVIEW permite la opción de visualizar el SCADA mediante navegador Web.

Para ello en la ventana de Tools dentro del Panel Frontal, se selecciona « Web Publishing Tool » como muestra en la Figura 3.41.

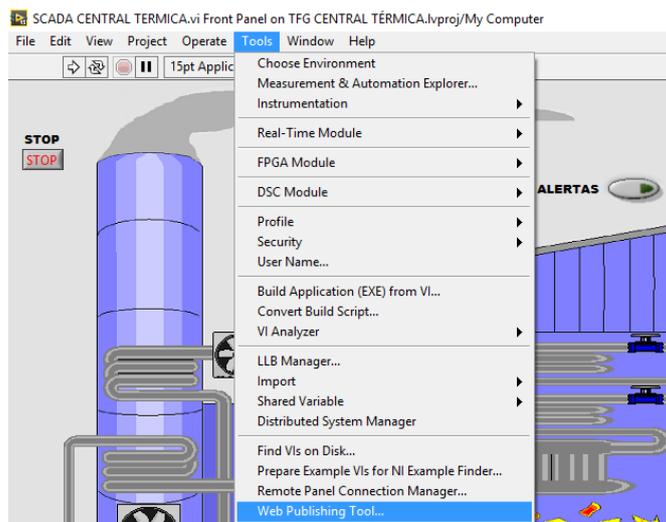
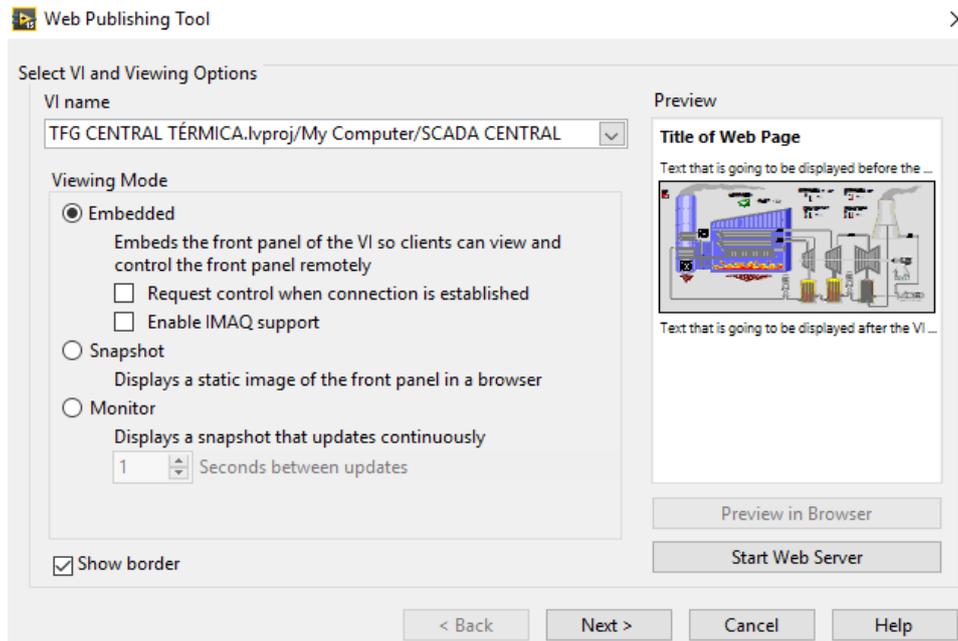


Figura 3.41: Web Publishing Tool (Elaboración propia)

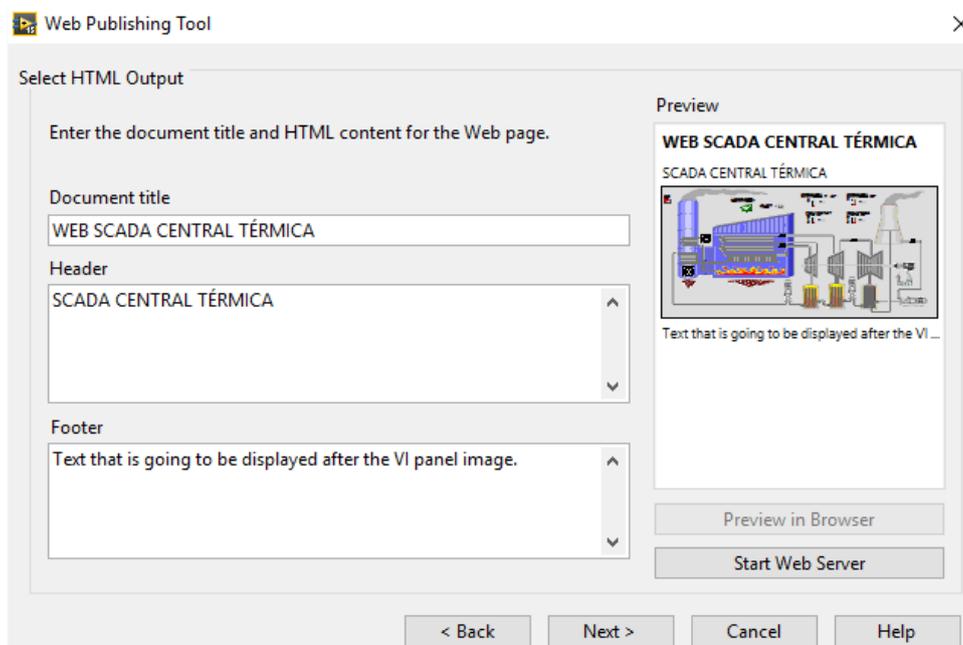
A continuación aparece una ventana como la mostrada en la Figura 3.42 en la que se debe escoger el Panel Frontal que queremos mostrar en la Web y tres opciones:

- **Embebido:** Con esta opción, permite al usuario observar y controlar el SCADA vía Web como si se tuviera abierta la aplicación de Labview.  
La opción escogida para este trabajo es esta, ya que es la más completa.
- **Instantánea:** En este caso, solo permite observar mediante una imagen el estado del SCADA sin tener control sobre él.  
Esta imagen se actualizará cada vez que se actualice la página Web.
- **Monitor:** Permite al usuario observar el SCADA mediante una imagen animada la cual se actualiza cada cierto tiempo fijado.



**Figura 3.42: Opciones de visualización vía Web (Elaboración propia)**

Una vez escogida la forma de visualización, se escoge el título y los textos superiores e inferiores que se desean visualizar (Figura 3.43).



**Figura 3.43: Opciones texto vía Web (Elaboración propia)**

Ahora en los documentos del proyecto, aparecerá el archivo HTML con el nombre elegido, como muestra la Figura 2.44.

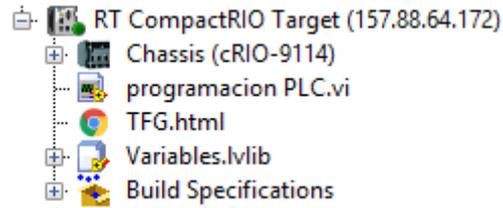


Figura 3.44: Archivo HTML en el proyecto (Elaboración propia)

Hay que configurar las propiedades del Compact RIO y seleccionar las propiedades que se muestran en la Figura 3.45.

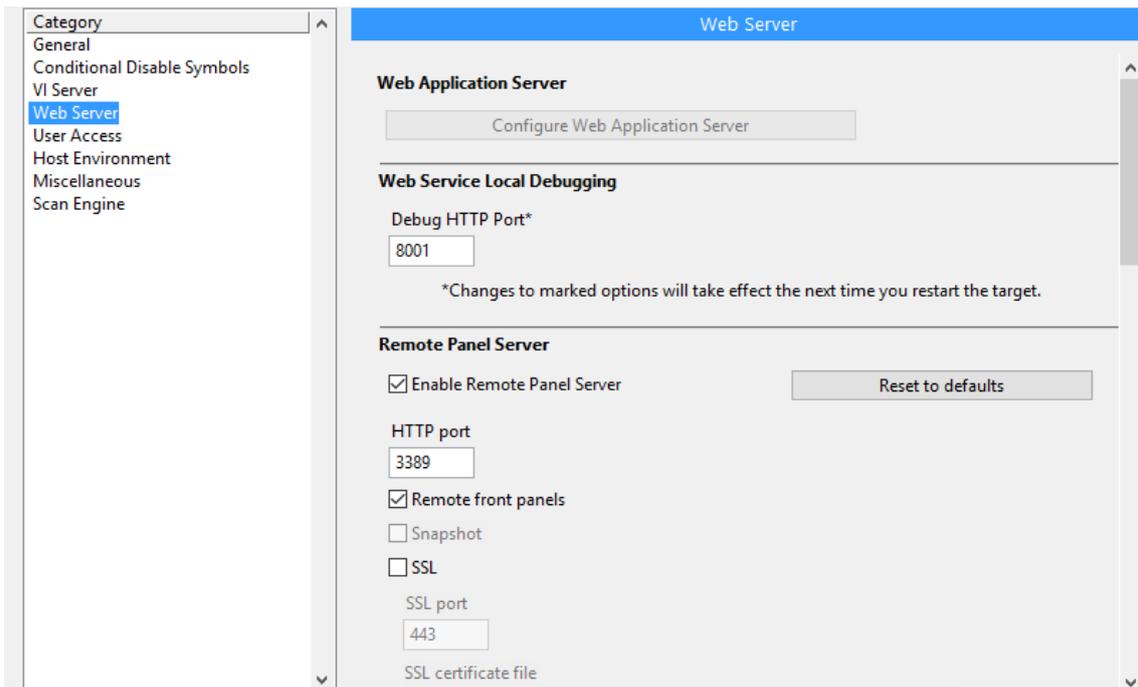
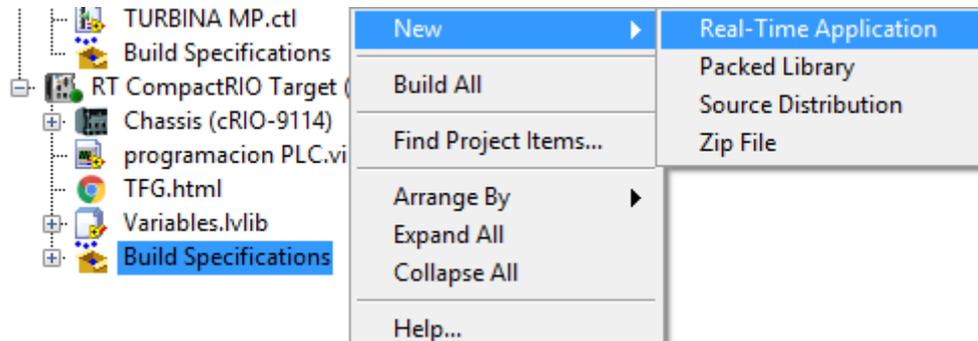


Figura 3.45: Propiedades Compact RIO (Elaboración propia)

Para terminar se construirá una aplicación en tiempo real como muestra la Figura 3.46 y se modificaran sus propiedades.



**Figura 3.46: Propiedades Compact RIO (Elaboración propia)**

- En el apartado « Source Files » hay que mover el programa del PLC a « Startup Vis » y el resto de archivos a « always included » como se muestra en la Figura 3.47.
- En el apartado « Destinations » se creará un nuevo destino llamado ‘www’ y se guardará el directorio del parche en « c:\ni-rt\system\www ».
- En el apartado « Source File Settings » se cambiar el destino del archivo HTML al www.

De esta forma se genera un enlace HTML, que se podrá emplear en cualquier ordenador, pudiendo visualizar el SCADA desde cualquier lugar y desde varios puntos distintos.

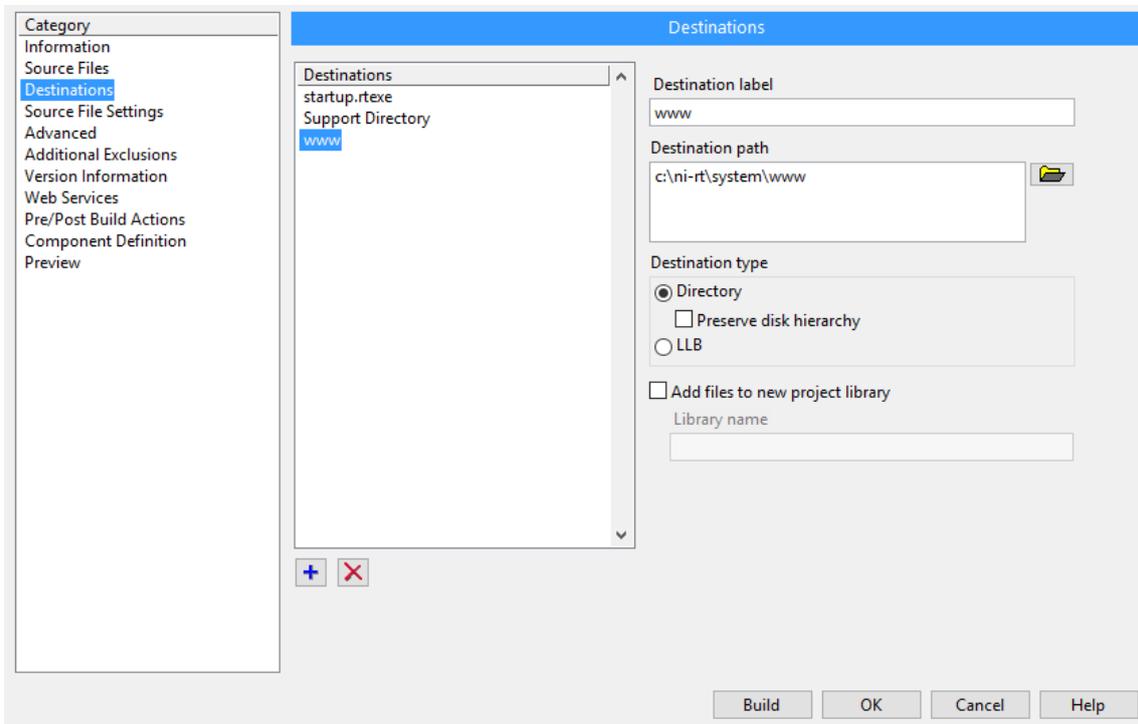
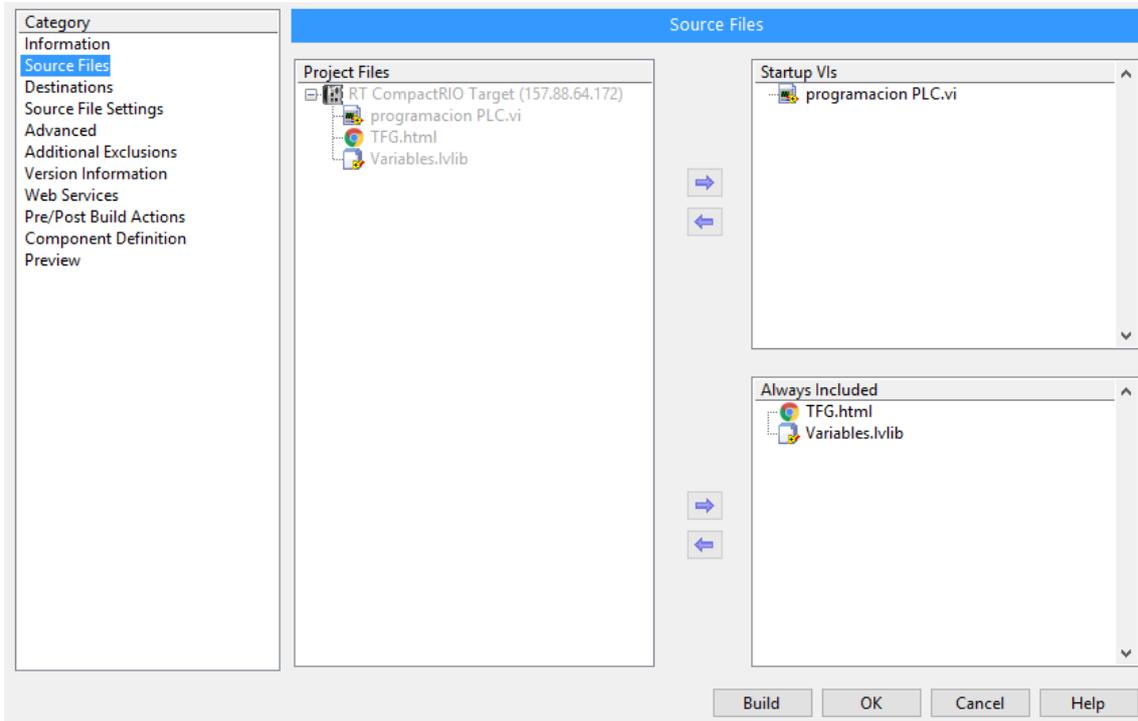


Figura 3.47: Propiedades Compact RIO (Elaboración propia)

Se puede introducir en el buscador « Internet Explorer » el enlace <http://XXX.XX.XX.XXX:3389/TFG.html>, que consta de:

- **XXX.XX.XX.XXX**: IP del PLC compact RIO.
- **3389**: HTTP port (Servidor del Panel Remoto).
- **TFG**: Título del archivo generado como HTML.

Con este enlace se puede ver perfectamente el SCADA como se muestra en la Figura 3.48.

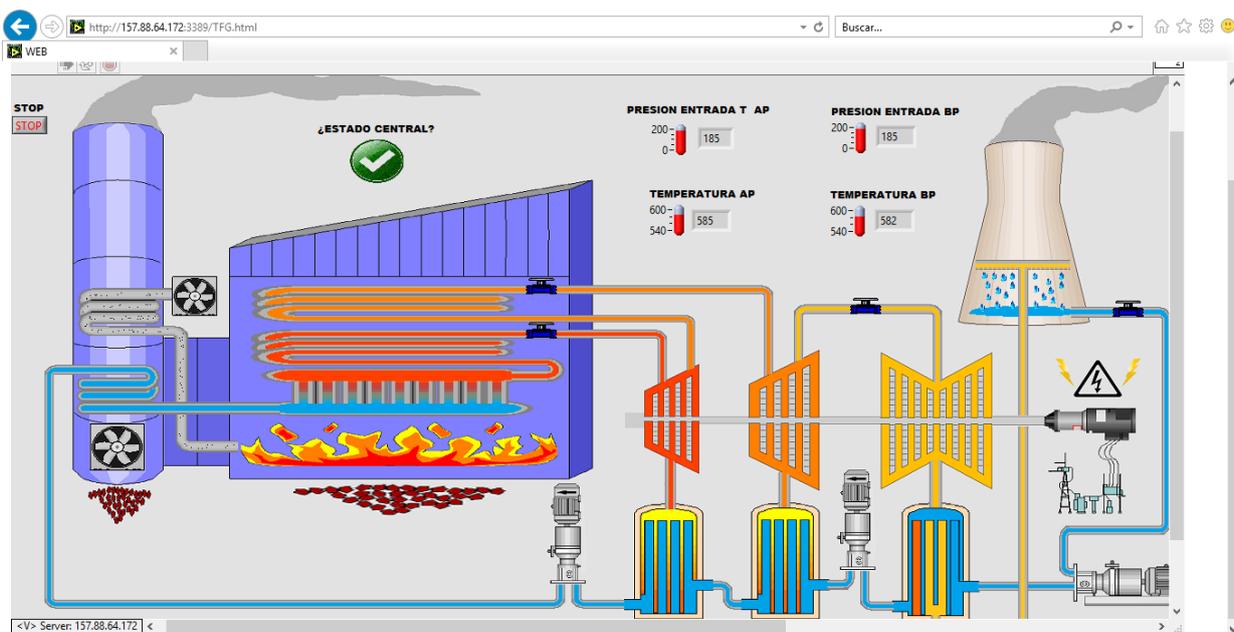


Figura 3.48: Visualización SCADA vía Web (Elaboración propia)

### 3.3 GENERACIÓN APLICACIÓN SCADA

El programa LabVIEW ocupa mucho espacio en el disco y además, requiere de licencias para poder usarse en cada ordenador. Por ello se puede generar una aplicación del SCADA que permita abrirlo sin necesidad de tener instalado LabVIEW.

Para crear la aplicación, en los archivos del proyecto se selecciona la opción « Construir Datos Específicos », « Nuevo » y « Aplicación EXE », como se muestra en la Figura 3.49.

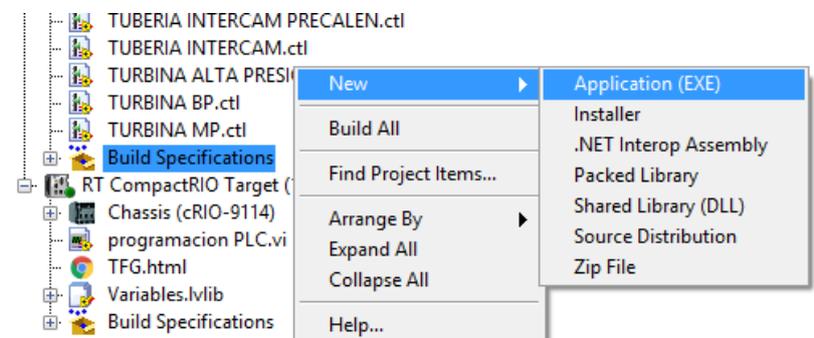


Figura 3.49: Crear aplicación SCADA (Elaboración propia)

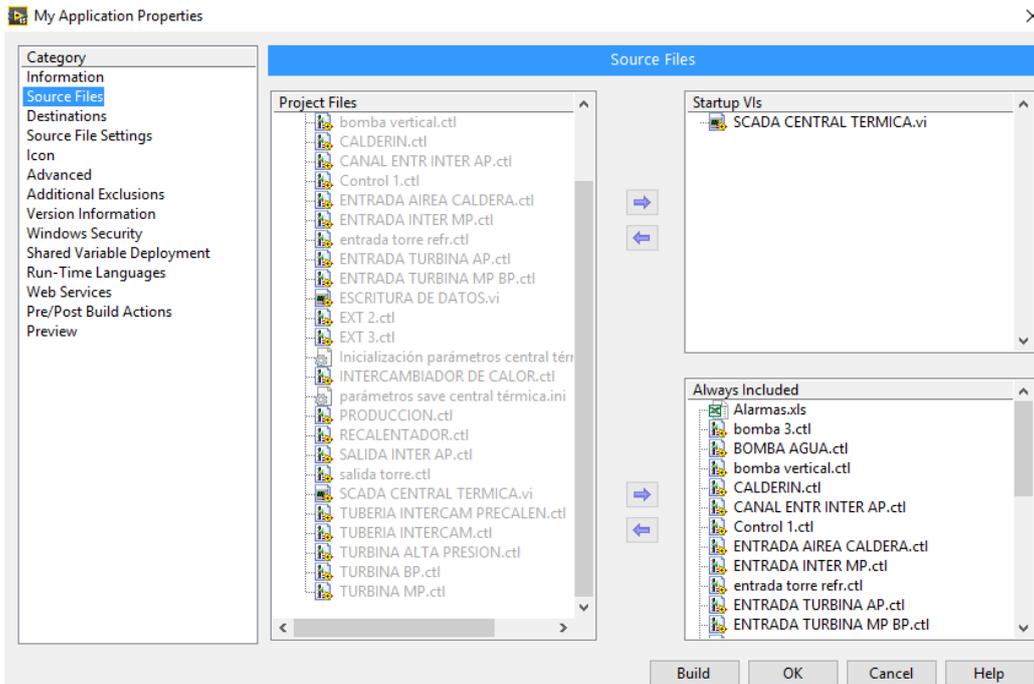
Para poder construir la aplicación lo único que se ha de tener en cuenta es que hay que ubicar los archivos fuentes en sus respectivos lugares.

En el apartado « Source Files » hay que mover el programa SCADA a « Startup Vis » y el resto de archivos a « always included » como se muestra en la Figura 3.50.

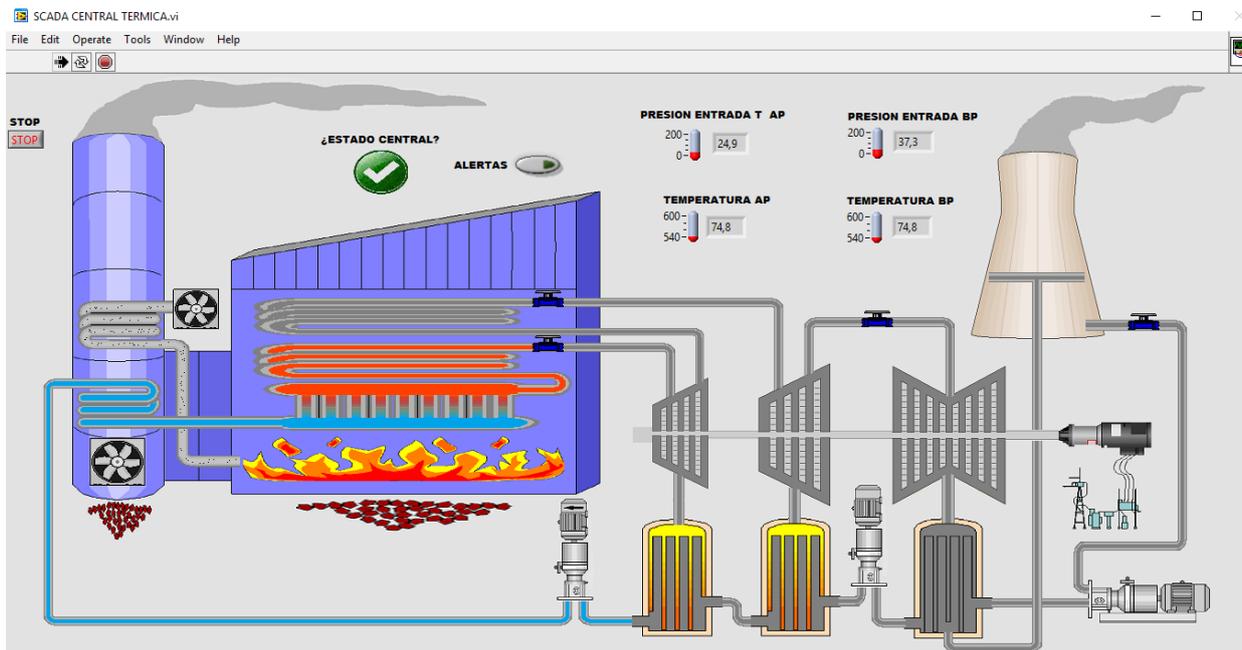
Una vez generada la aplicación se puede ejecutar el archivo con extensión .EXE y verse como en la Figura 3.51.

Esta aplicación permite emplear las dos pestañas de la Interface Gráfica por lo que la aplicación necesitará tener en la misma carpeta una copia del archivo Excel de las alarmas y de los archivos de lectura que emplea en el arranque.

La aplicación se puede transferir a cualquier ordenador y funcionar correctamente sin que contenga el programa LabVIEW.



**Figura 3.50: Propiedades aplicación SCADA (Elaboración propia)**



**Figura 3.51: Aplicación SCADA (Elaboración propia)**



## CAPITULO 4: PLC NATIONAL INSTRUMENTS NI cRIO-9022

### 4.1 PLC COMPACT RIO REAL-TIME CONTROLLER

Para el presente trabajo se ha empleado un PLC de National Instruments NI cRIO 9022 como el de la Figura 4.52. Este PLC permite una conexión con LabVIEW, el cual a su vez nos permite una configuración SCADA, objeto de este trabajo.

Sus ventajas se detallan a continuación:

- El sistema CompactRIO es una combinación de un chasis de controlador en tiempo real, módulos IO reconfigurables (RIO), un módulo FPGA y un chasis de expansión Ethernet.
- Los módulos de terceros son generalmente compatibles con los controladores de chasis producidos por NI.
- Los controladores CompactRIO en tiempo real incluyen un microprocesador para implementar algoritmos de control y soportan una amplia gama de frecuencias de reloj.
- Los controladores sólo son compatibles con los módulos de E / S de la serie C de National Instruments.
- Los módulos de E / S son intercambiables en caliente (se pueden conectar / desconectar mientras la unidad está encendida).
- El módulo FPGA puede utilizarse para implementar el procesamiento de datos de alto rendimiento en un tejido reconfigurable.
- Dicho procesamiento de datos puede realizarse en transmisión de datos desde módulos de E / S conectados.
- El FPGA se puede programar por separado y se conecta al controlador en tiempo real utilizando un bus PCI interno.
- El chasis está disponible en 4 ranuras y 8 variedades de ranura.
- Los controladores CompactRIO más nuevos y de alto rendimiento también tienen gráficos VGA incorporados que se pueden conectar a un monitor para observar la operación.



Figura 4.52: PLC NI 9022 (<http://www.ni.com>)

## 4.2 MÓDULOS NI cRIO-9022

CompactRIO de National Instruments tiene acceso directo a los circuitos de entrada/salida de cada módulo de E/S usando funciones elementales de E/S de LabVIEW FPGA.

Cada módulo de E/S contiene acondicionamiento de señales integrado y conectores de terminal de tornillo, BNC o D-Sub.

Una variedad de tipos de E/S, incluyendo entradas de termopares de  $\pm 80$  mV, entradas/salidas analógicas de muestreo simultáneo de  $\pm 10$  V, E/S digital e industrial de 24 V con capacidad de corriente de hasta 1 A, entradas digitales diferenciales con salida de suministro regulado de 5V para codificadores y entradas digitales de 250V.

Ya que los módulos contienen acondicionamiento de señales integrado para rangos de voltaje extendidos o tipos de señales industriales, se puede realizar las conexiones de cableado directamente desde el módulo CompactRIO a sus sensores/actuadores. En la mayoría de los casos los módulos CompactRIO brindan aislamiento del canal a tierra.

Los módulos CompactRIO se conectan directamente a dispositivos FPGA de E/S reconfigurables (RIO) para crear sistemas embebidos de alto rendimiento que brindan la flexibilidad y optimización de un circuito eléctrico personalizado completamente dedicado a su aplicación de entrada/salida.

El hardware RIO FPGA tiene opciones ilimitadas para temporización, disparo, sincronización y procesamiento de señales a nivel de sensor y toma de decisiones.

#### 4.2.1 MÓDULO NI 9201

El NI 9201 es un módulo de la Serie C para entradas analógicas de 8 canales.

Como con la mayoría de los módulos de la Serie C, el NI 9201 (Figura 4.53) está protegido contra picos de voltaje perjudiciales de hasta 2,300 Vrms. Esto significa que ningún voltaje dentro del rango de aislamiento puede dañar otros módulos en el sistema, el chasis o cualquier equipo de cómputo conectado.

Además, el módulo ofrece hasta 100 V de protección de sobrevoltaje para mala conexión de señal o salidas inesperadas en el módulo a los canales individuales.

Se puede escoger entre dos opciones de conectores para el NI 9201:

- Un conector de terminal de tornillo de 10 posiciones para conectividad directa.
- Un conector D-SUB de 25 posiciones.

Para este trabajo por comodidad se ha empleado un conector de terminal de tornillo de 10 posiciones para conectividad directa.



Figura 4.53: Módulo NI 9201 (<http://www.ni.com>)

#### 4.2.2 MÓDULO NI 9472

El módulo NI 9472 (Figura 4.54) incluye 8 canales de salida digital a 100  $\mu$ s, capaces de comunicar con un PC principal por USB, Ethernet o 802.11 Wi-Fi.

Se pueden emplear en cualquier chasis NI CompactDAQ o CompactRIO.

Cada canal es compatible con señales de 6 a 30 V y ofrece protección contra sobrevoltaje transitorio de 2,3 Vrms entre los canales de salida y el plano trasero.

Cada canal también tiene un LED que indica el estado de ese canal.

Con el NI 9472 se puede conectar directamente a una variedad de dispositivos industriales como lo son motores, actuadores y relés.

Se puede escoger entre dos opciones de conectores para el NI 9472:

- Un conector de terminal de tornillo de 10 posiciones.
- Un conector D-Sub de 25 posiciones. Este conector D-Sub de 25 posiciones estándar en la industria es ideal para cableado a bajo costo a una amplia variedad de accesorios de NI u otros proveedores.

Para este trabajo por comodidad se ha empleado un conector de terminal de tornillo de 10 posiciones para conectividad directa.



Figura 4.54: Módulo NI 9472 (<http://www.ni.com>)

### 4.3 CHASIS COMPACT RIO NI 9114

El chasis reconfigurable y embebido NI cRIO-9114 (Figura 4.55) de ocho ranuras es parte de la plataforma del controlador de automatización programable (PAC) de alto rendimiento CompactRIO.

Contiene un FPGA programable por el usuario Xilinx Virtex-5, lo cual le proporciona alta potencia de procesamiento y la habilidad de diseñar hardware personalizado utilizando software NI LabVIEW.

El cRIO-9114 proporciona acceso de hardware de bajo nivel a cualquier módulo de E/S NI CompactRIO, y así se puede crear esquemas de temporización, disparo, control y sincronización sin precedentes para aplicaciones embebidas e industriales.

El chasis de CompactRIO incluye orificios de montaje en panel.



*Figura 4.55: Chasis NI 9114 (<http://www.ni.com>)*

#### 4.4 FUENTE DE ALIMENTACIÓN NI PS-15

La entrada de tensión es 115/230 VAC de 1 fase, salida de 24 a 28 VDC, con una intensidad de 5 A.

Potencia de salida completa de 120 W entre -25 y +60 °C, (depreciada 3 W/°C desde 60 a 70 °C).

La reserva de potencia del 20%, para cargas dinámicas se puede usar continuamente hasta 45 °C.

Terminales tornillo-abrazadera para fácil conectividad en campo y montaje en carril DIN (Figura 4.56).

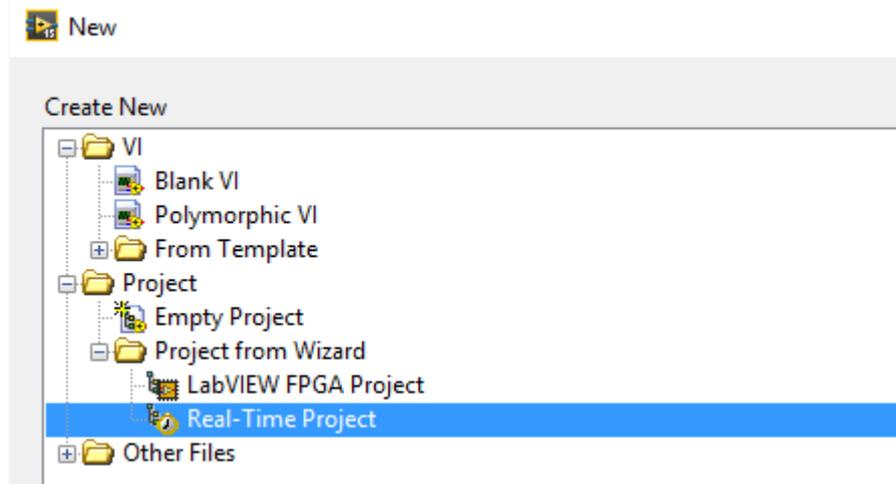


*Figura 4.56: Fuente alimentación NI PS-15 (<http://www.ni.com>)*

## 4.5 CONEXIÓN PC-PLC

Lo primero que se debe hacer, es crear un archivo de LabVIEW nuevo.

A continuación se dan varias opciones a escoger que se representan en la Figura 4.57.



*Figura 4.57: Archivo nuevo Labview (Elaboración propia)*

Entre ellas se escoge la opción de « Real-Time Project » para crear un proyecto en tiempo real.

A continuación personalizaremos las opciones de arquitectura (Figura 4.58):

- Se escogerán dos bucles con el objetivo de ejecutar dos bucles temporizados en diferentes prioridades para separar tareas deterministas y no deterministas
- Se añadirá una interfaz de usuario con panel remoto en el que se pueda emplear el navegador web.

Esto nos permitirá tener una conexión completa entre PC-PLC.

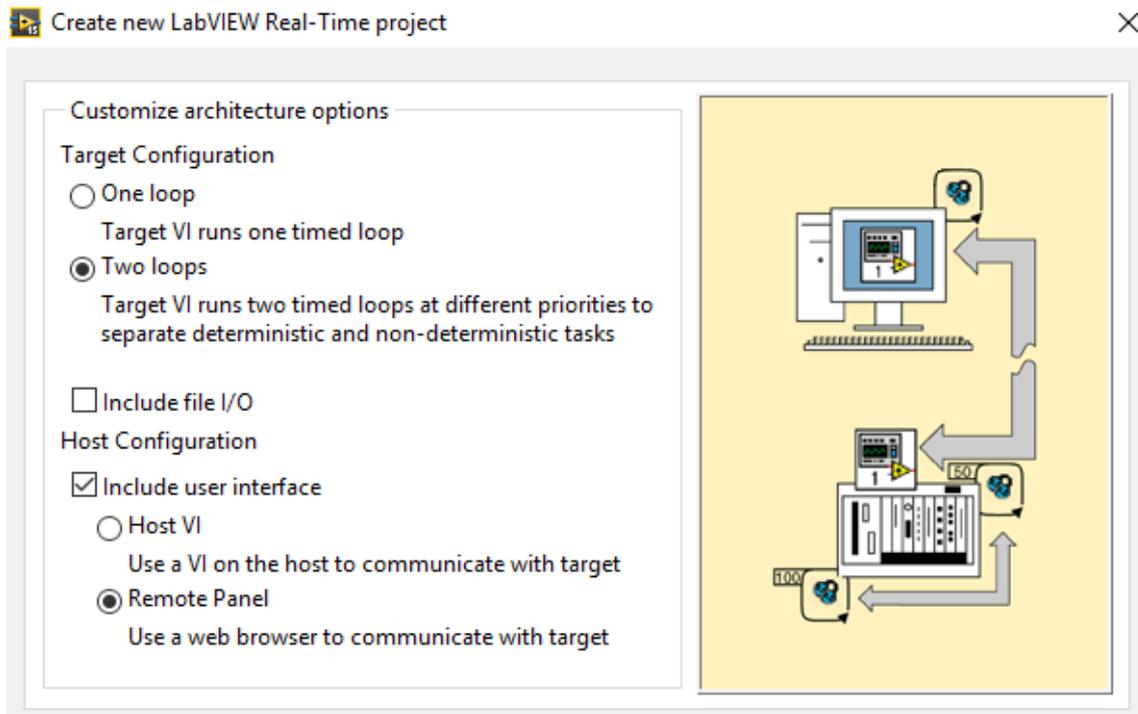


Figura 4.58: Proyecto en Tiempo Real (Elaboración propia)

Se debe tener en cuenta que se necesita saber la IP para poder conectarse al PLC. Esta se introducirá manualmente cuando se pida.

Una vez introducida la IP ya se tendrá la posibilidad de conectarse al PLC. Se puede comprobar que debe aparecer en el explorador de proyecto con el número de la IP escogida.

Mediante un piloto verde dibujado encima del icono PLC se indicará si el PC está conectado al PLC (Figura 4.59).

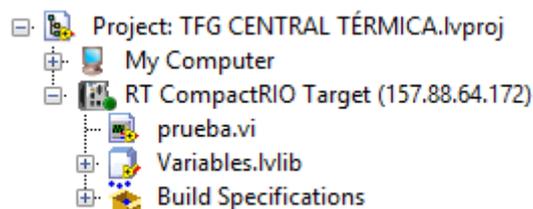
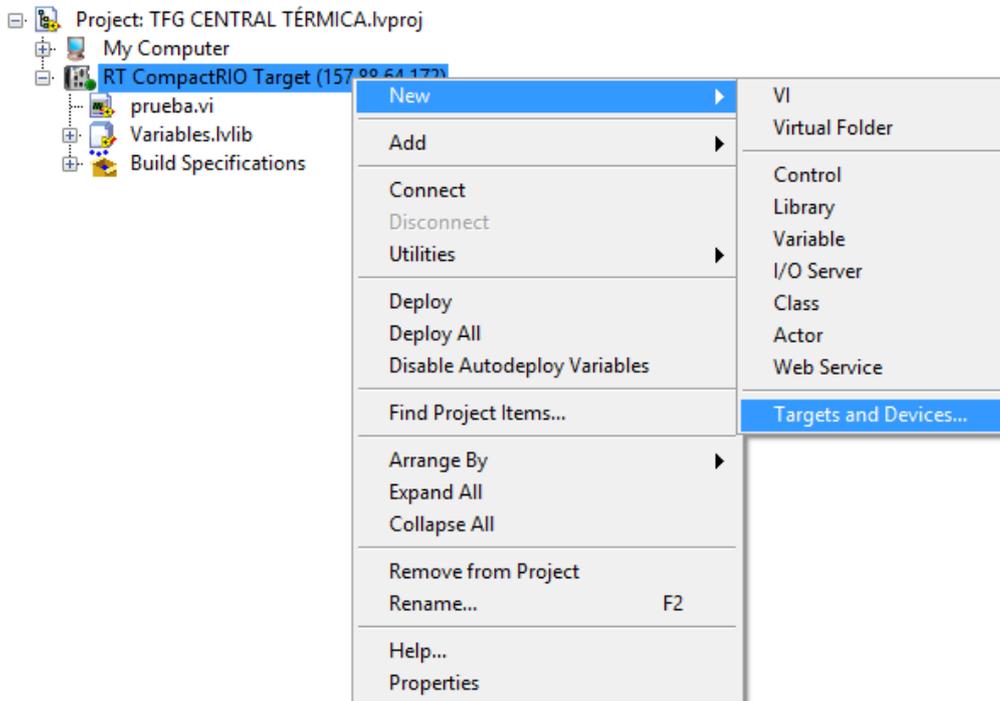


Figura 4.59: Icono Panel Conector (Elaboración propia)

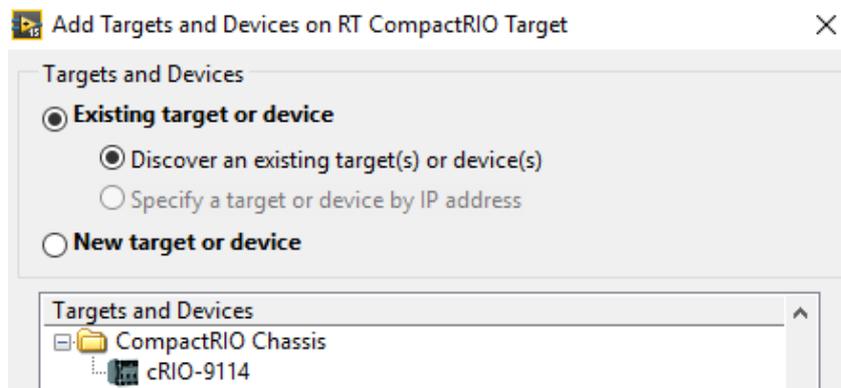
Cuando ya está comprobada su conexión, hay que introducir el chasis, que indicará los módulos que están conectados y que se serán sobre los que se pueda trabajar.

Para introducir el chasis, haciendo click derecho sobre el PLC, añadimos un nuevo chasis como aparece en la Figura 4.60.



*Figura 4.60: Añadir chasis al proyecto (Elaboración propia)*

Deberá aparecer el chasis de CompactRIO con el que se quiere trabajar como aparece en la Figura 4.61.

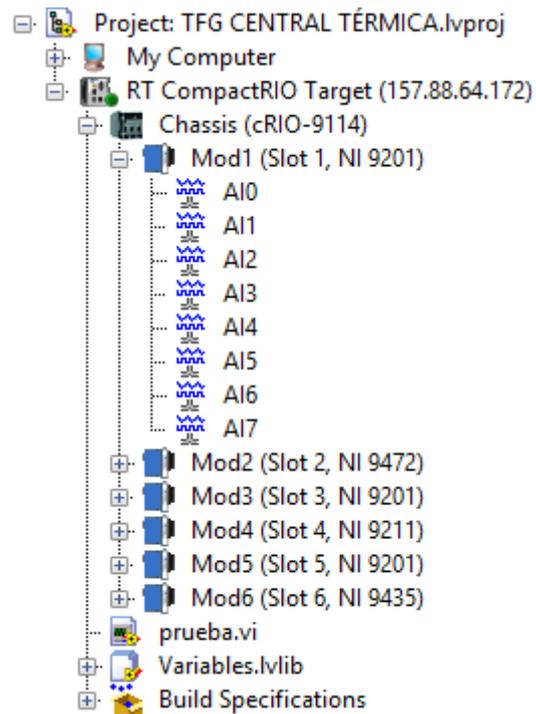


*Figura 4.61: Chasis disponibles (Elaboración propia)*

Dentro de la opción cRIO-9114 hay que seleccionar la opción « Scan interface ».

Una vez introducido el chasis, aparecen los módulos existentes con sus entradas y/o salidas, dependiendo del tipo de módulo como aparece en la figura 4.62.

Aunque en el chasis existen 6 módulos, para este trabajo solo emplearemos los módulos NI 9201 y NI 9472.



*Figura 4.62: Módulos existentes en el chasis (Elaboración propia)*

## CAPITULO 5: DISEÑO, CONSTRUCCIÓN DE LA MAQUETA Y HARDWARE ADICIONAL

### 5.1 INTRODUCCIÓN

Este trabajo está pensado para un montaje industrial a gran escala, con valores de temperatura por encima de los 500° y valores de presión por encima de los 180 bar.

Estos valores son inalcanzables en condiciones normales.

Para llevar a cabo el montaje de dicho trabajo, será necesario adaptarse a los materiales facilitados por el departamento de máquinas eléctricas, ya que se quiere simular su funcionamiento del SCADA a muy pequeña escala.

Para ello se empleará:

- Como entradas del sistema se emplearán resistencias variables, lo que nos permitirá simular la variación de temperatura y presión de las turbinas.
- Como salidas del sistema se emplearán bombillas de LED, lo que nos permitirá simular la activación y desactivación de los actuadores del sistema.

Como el proceso industrial es de gran envergadura, el número de entradas/salidas y variables a controlar es muy grande, por ello solo se van a tener en cuenta unas entradas/salidas en concreto con el fin de comprobar la posibilidad de un montaje en PLC y su funcionamiento.

Si el funcionamiento de unas pocas entradas/salidas funciona, es obvio pensar que el resto solo sería añadir módulos de entrada y salida.

Del mismo modo pasa con las alarmas, este trabajo se ha centrado en alarmas por exceso de temperatura y presión en las turbinas. Se podrían añadir infinitas alarmas con las condiciones que se necesitasen.

Todo ello se ubicará en una maqueta con la intención de ser visible, y poder ver en tiempo real, como varía cada una de las entradas/ salidas del sistema, además de variar en el SCADA.

## 5.2 DISEÑO DE LA MAQUETA

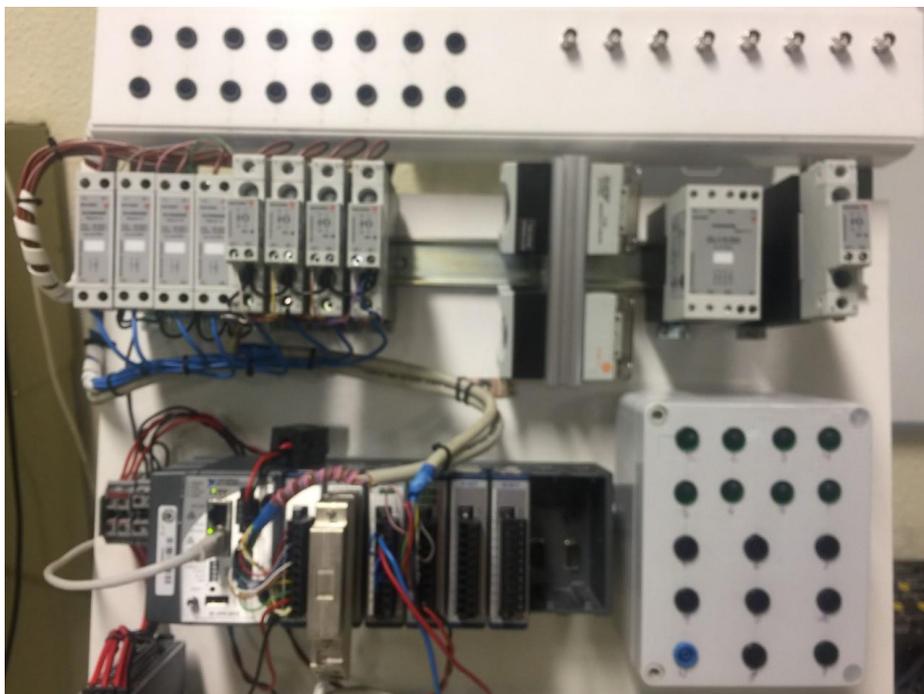
Como se ha mencionado en la introducción, la misión de la maqueta (Figura 5.63) es que los elementos sean visibles, para poder interpretar el funcionamiento de cada parte del sistema.

Debido a que todos los elementos están adaptados para su sujeción con carriles DIN, esta será la opción elegida.

Los elementos están unidos entre sí, por cables con punteras o conectores, evitando así, cualquier tipo de peligro eléctrico.

Los carriles DIN están conectados a Tierra protegiendo así la maqueta contra contactos indirectos.

Como los elementos no tienen prácticamente consumo de intensidad, debido a que son elementos de señal o actuadores con muy poca potencia, las secciones de los cables no serán importante y se empleará cable de pequeña sección siguiendo siempre la ITC-BT-19 del Reglamento de Baja Tensión.



*Figura 5.63: Diseño maqueta (Elaboración propia)*

### 5.3 HARDWARE ADICIONAL

Para completar el PLC se necesitará un tratamiento de las señales de entrada y salida para que se puedan emplear en el PLC, esto no sería necesario si se conectase directamente sensores de presión o temperatura analógicos, o contactores en el caso de las salidas.

Por este motivo el número de entradas y salidas escogidas han sido reducidas, para evitar costes en el montaje innecesarios, los cuales tampoco van a dar más información que los propuestos.

#### 5.3.1 SIMULACIÓN ENTRADAS DEL SISTEMA

El módulo de entradas trabaja con valores de tensión  $\pm 10$  Vcc.

Para conseguir una tensión continua se puede emplear una fuente de alimentación que transforme los 230V en alterna en 15 V en continua.

Ahora bien, necesitamos una tensión continua regulable de 0 a 10V, por ello será necesario realizar el circuito impreso de la Figura 5.64.

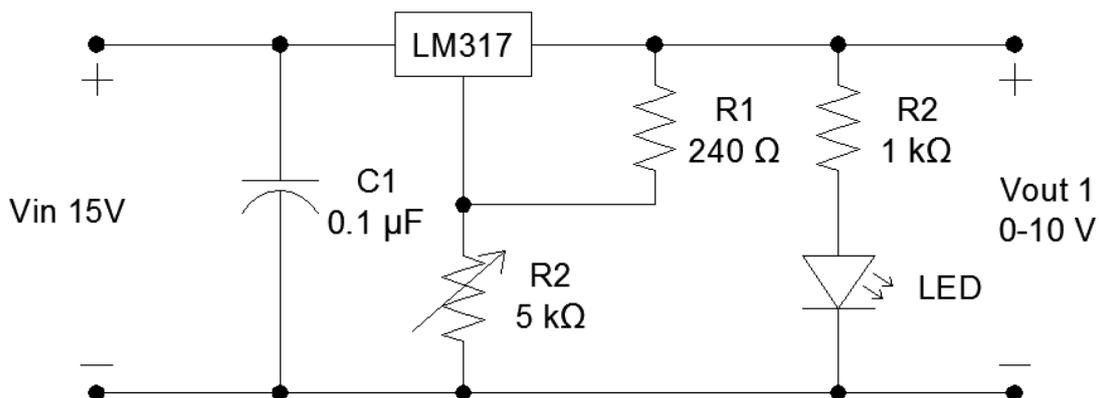


Figura 5.64: Esquema regulador (Elaboración propia)

El condensador de  $0.1 \mu\text{F}$  sirve para filtrar ruidos y que la señal continua que entre al PLC sea lo más pura posible.

El piloto LED se iluminará con una intensidad en concreto según la tensión que este alimentando al PLC.

Como el número de entradas necesarias son 5 y todas ellas son señales regulables, el circuito necesario será 5 veces el de la Figura 5.64 quedando como el de la Figura 5.65.

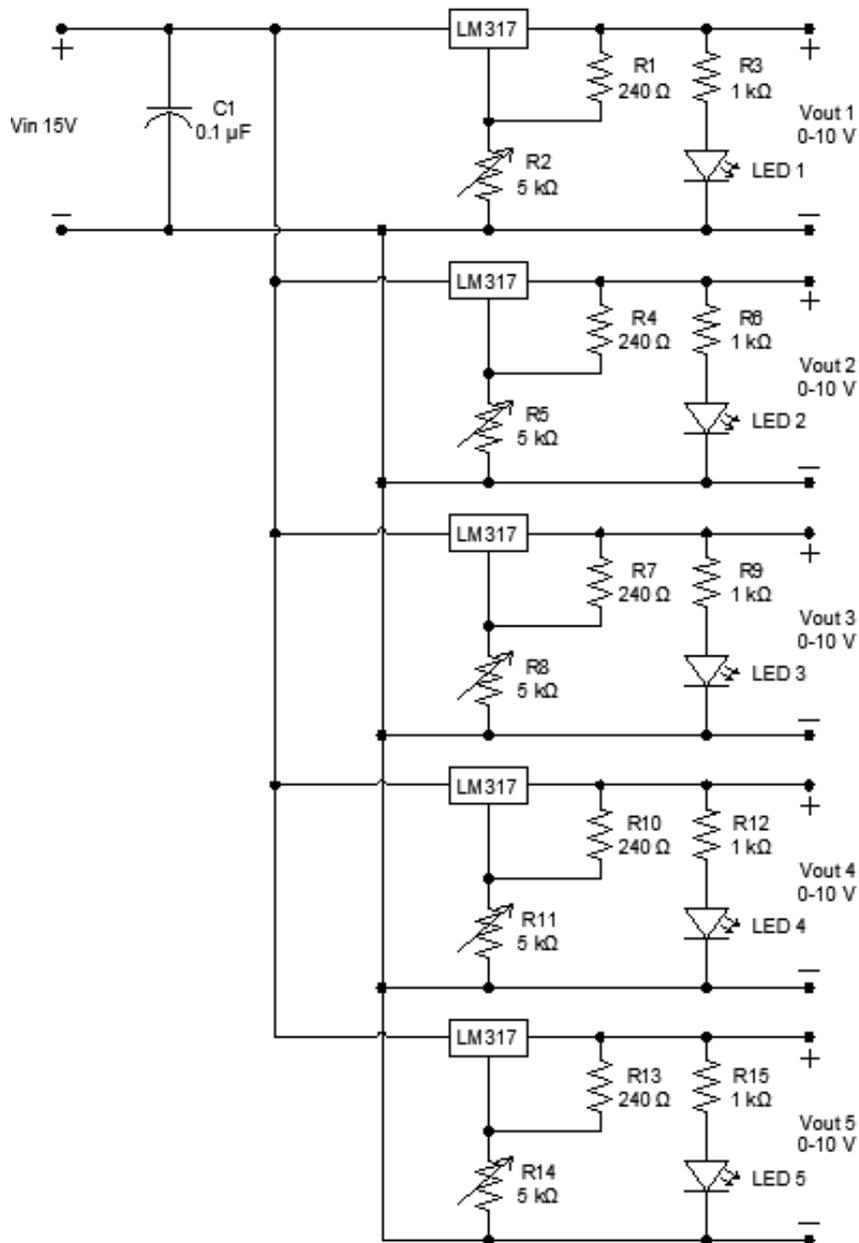


Figura 5.65: Esquema 6 reguladores (Elaboración propia)

Ese circuito se puede introducir en una caja de conexiones (Figura 5.66), para facilitar la manipulación de las resistencias variables y la interpretación de los pilotos LED.



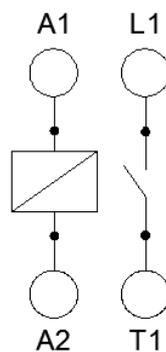
*Figura 5.66: Caja de conexiones reguladores (Elaboración propia)*

### 5.3.2 SIMULACIÓN SALIDAS DEL SISTEMA

Para las salidas se aprovecharán las señales del módulo de salidas para alimentar unas bobinas de unos relés de estado sólido, los cuales activarán o desactivarán unos pilotos.

Estos relés de estado sólido (Figura 5.68) funcionan con un rango de tensiones de 4-32Vcc, señal perfecta, ya que el módulo de salidas está alimentado por la fuente de alimentación de NI PS-15 a 24 Vcc.

Solo poseen un contacto abierto el cual es accionado cuando la bobina recibe alimentación como se puede ver en la Figura 5.67.



*Figura 5.67: Esquema Relé de Estado Sólido (Elaboración propia)*

El contacto de operación trabaja a una tensión nominal de 230 Vac, lo cual nos facilita poder conectarlo sin necesidad de una fuente de alimentación.



*Figura 5.68: Relé de Estado Sólido (<http://www.automation24.es>)*

Estos relés de estado sólido sirven para dar alimentación a unos pilotos ubicados en una caja de conexiones (Figura 5.69), los cuales nos ayudan a identificar la salida activada y por lo tanto el actuador que debe funcionar en ese momento.



*Figura 5.69: Caja Pilotos (Elaboración propia)*

## 5.4 CONEXIONADO DE LOS ELEMENTOS

El conexionado final de elementos se adjunta como anexo 3.



## CAPITULO 6: PRUEBAS DE FUNCIONAMIENTO

### 6.1 INTRODUCCIÓN

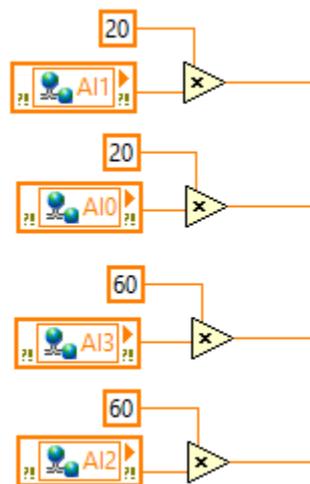
Una vez conectados todos los elementos del proceso y con el SCADA terminado, se pasa a comprobar si la programación corresponde con lo que debe hacer el sistema, depurando así los posibles errores que surgen.

Una de las primeras dificultades que se tienen, es que los valores de temperatura y presión deben oscilar entre:

- Bar: hasta 200 bares de presión.
- Temperatura: hasta 600° de temperatura.

Como los reguladores que se han implementado para el módulo de entradas dan una señal de 0-10V, se deben adaptar esos valores para conseguir lo que se necesita.

Se multiplican los valores de presión por una constante de 20 para conseguir un valor máximo de 200 en la presión, y por 60 para conseguir un valor de 600 en la temperatura como muestra la Figura 6.70.



*Figura 6.70: Adaptación entradas (Elaboración propia)*

Hay que tener en cuenta que en el SCADA, hay programadas múltiples entradas y salidas, y que en el PLC solo se emplean unas pocas, por ello, el resto de entradas se han temporizado con el fin de realizar el ciclo del proceso completo, paso por paso hasta la generación eléctrica de las 3 turbinas y refrigeración del agua empleada por las turbinas.

Otra dificultad añadida que surge, es el cambio de pestañas en el SCADA, ya que para el cambio de pestaña se necesita un pulsador llamado ALERTAS, el cual dará acceso a una u otra pestaña como se muestra en la Figura 6.71.

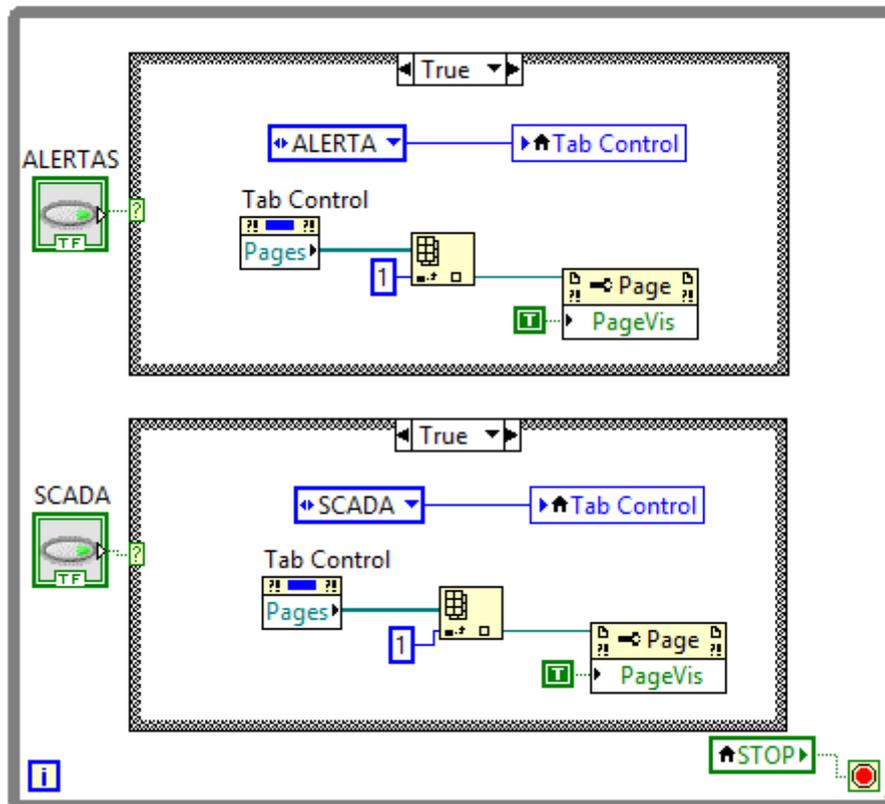


Figura 6.71: Cambio de pestaña SCADA (Elaboración propia)

## 6.2 PRUEBAS DE FUNCIONAMIENTO

Para analizar el correcto funcionamiento del SCADA se debe comprobar paso a paso el cumplimiento de las condiciones fijadas en la programación.

Cuando se programa en LabVIEW o cualquier otro método de programación, siempre queda la incertidumbre de si funciona lo que se programa y si realmente hace lo que se necesita que haga.

Además, también puede haber fallos en la conexión de equipos lo cual habrá que tener especial cuidado ya que son equipos muy sensibles.

Para las pruebas de funcionamiento y posteriores conexiones al PLC será obligatorio una conexión a internet, ya que si no dará 'error de conexión' y obviamente, no se podrá conectar al PLC ni enviar correos electrónicos.

### 6.2.1 PRUEBA CONEXIÓN PC-PLC

Para la conexión del SCADA con el PLC primero debemos conectar el PLC al ordenador y ejecutar el programa de este como se muestra en la Figura 6.72.

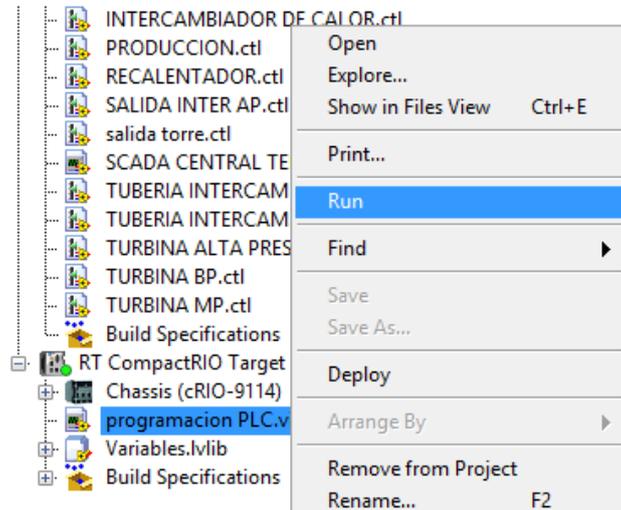


Figura 6.72: Conexión programa PLC (Elaboración propia)

A continuación se ejecuta el SCADA como se muestra en la Figura 6.73, quedando ya conectado y donde los valores y cambios se pueden observar en tiempo real.

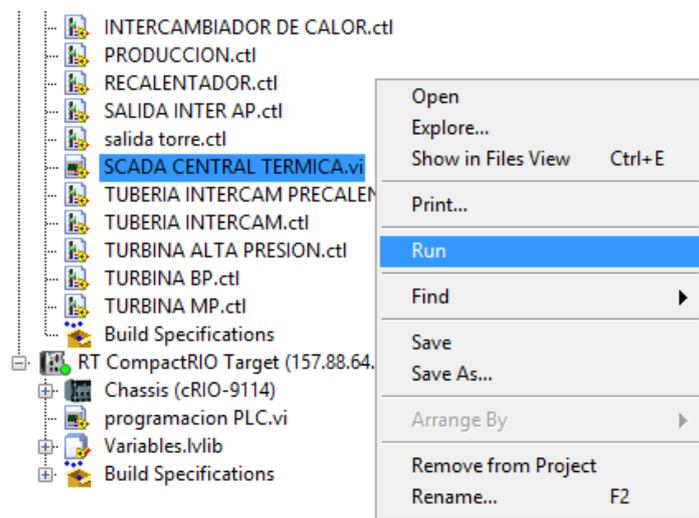
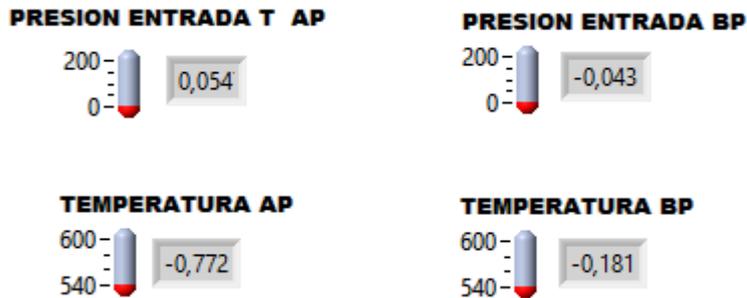


Figura 6.73: Conexión programa SCADA (Elaboración propia)

## 6.2.2 PRUEBA DE FUNCIONAMIENTO E/S

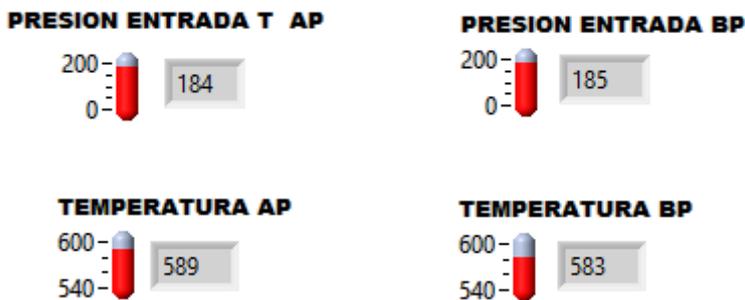
Para probar las entradas del sistema, basta con conectar el SCADA y ver en tiempo real, que el valor de la presión y temperatura varían, al variar los reguladores. Se muestra en la Figura 6.74.



*Figura 6.74: Entradas proceso (Elaboración propia)*

Se ha de tener muy en cuenta, que los valores máximos alcanzados deben rondar los 200 bar para la presión, y los 600° para la temperatura.

Se ajustan como prueba a los valores nominales como se muestra en la Figura 6.75.



*Figura 6.75: Valor nominal entradas (Elaboración propia)*

Se comprobará también que los LED de señalización que llevan acoplados los reguladores funcionan correctamente, luciendo con más intensidad en función de la tensión que tenga a la salida del regulador.

Para las salidas, los pilotos de señalización nos indicarán el estado de cada una. Además, los relés de estado sólido también llevan un pequeño LED el cual también indican el estado de los mismos.

Se deberá comprobar que las salidas activadas y desactivadas, funcionan con sentido y acorde a la programación implementada.

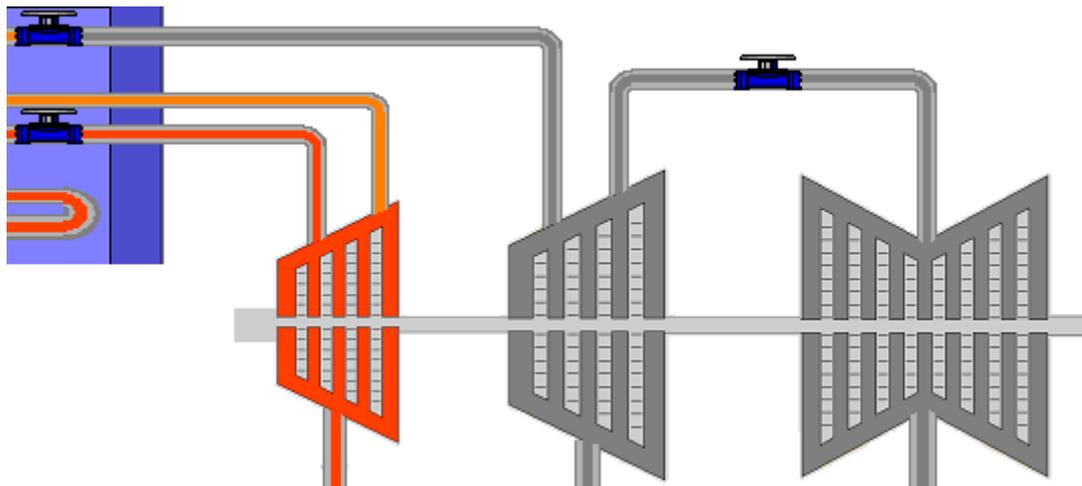
La mejor comprobación es ver, que el piloto del módulo NI 9472, el piloto del relé de estado sólido, y el piloto LED, están marcando lo mismo.

### 6.2.3 SIMULACIÓN FUNCIONAMIENTO TURBINAS

Para el funcionamiento de las turbinas se deberán cumplir una serie de hipótesis.

- **Turbina AP:** Para que la turbina de AP entre en funcionamiento, la presión de Alta Presión deberán superar, los valores establecidos. En este caso se han escogido 180 bar y 560°, por lo que hasta que no se den estas dos hipótesis la turbina no funcionará.

Una vez se cumplan las hipótesis la turbina entrará en funcionamiento gracias a una electroválvula que le permitirá el acceso de vapor. Su activación se muestra en la Figura 6.76.

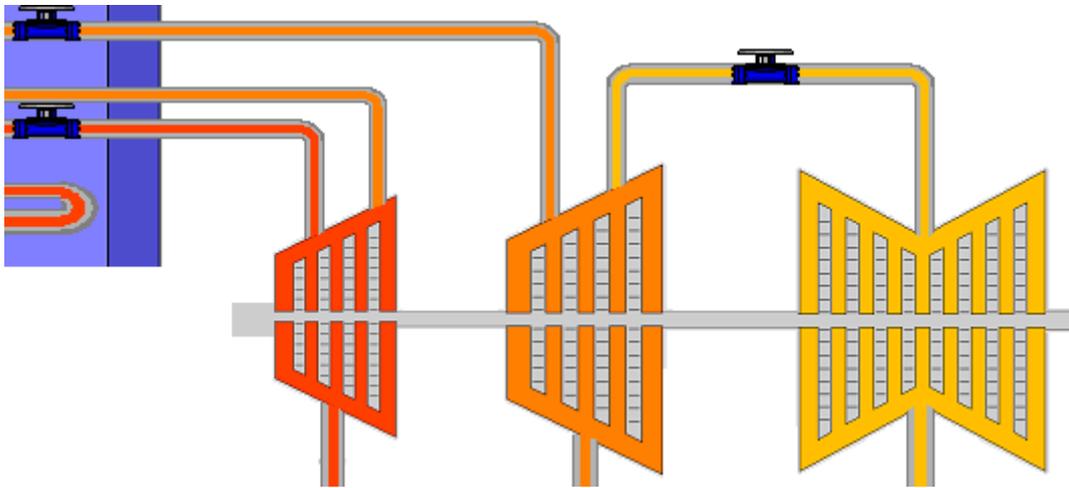


*Figura 6.76: Activación electroválvula turbina AP (Elaboración propia)*

- **Turbina MP y BP:** Para la activación de la turbina de MP y BP, se tienen que cumplir las hipótesis establecidas al igual que se cumplía en la turbina de AP.

El control de las turbinas de MP y BP también se realiza con electroválvulas y por defecto se han escogido 180 bar y 560°, por lo que hasta que no se den estas dos hipótesis la turbina no funcionará.

De la misma forma que en la turbina de AP, una vez se cumplan las hipótesis la turbina entrará en funcionamiento gracias a una electroválvula que le permitirá el acceso de vapor. Su activación se muestra en la Figura 6.77.



*Figura 6.77: Activación electroválvulas MP y BP (Elaboración propia)*

Al igual que su activación en el SCADA, también debe activarse en el módulo de salidas, y por lo siguiente activarse sus correspondientes pilotos, indicando su activación.

#### 6.2.4 COMPROBACIÓN LISTADO CONSIGNAS

Se comprueban el funcionamiento de las consignas establecidas, y se modifican manualmente desde el archivo .ini de (inicialización de parámetros central térmica) para comprobar que lee los datos cada vez que se ejecuta el SCADA, actualizando esos valores y empleándolos como nuevas consignas a cumplir.

Dichas consignas se cumplirán tanto como valores establecidos de funcionamiento, y como valores límites de funcionamiento, por lo cuales saltará una alarma si se sobrepasan con sus consiguientes consecuencias.

## 6.2.5 SIMULACIÓN ANOMALÍA FUNCIONAMIENTO

### 6.2.5.1 *Lectura y escritura de la anomalía*

Nada más ejecutar el programa, se leerá un archivo Excel con las alertas surgidas anteriormente que se guardarán en la tabla de la segunda pestaña del interface.

Si en un determinado momento, se cumple el estado de alarma, la reacción del SCADA es almacenar esa alerta en una tabla en la segunda pestaña del interface y archivar la alerta en una tabla Excel.

La escritura de esa alerta no sobrescribirá el archivo Excel existente, sino que simplemente añadirá la nueva alerta, manteniendo así, las posibles alertas anteriores. Consiguiendo así un historial completo de alertas con sus fechas y sus estados correspondientes.

De forma similar actúa cuando la alarma se ha extinguido, ya que el SCADA actuará de forma similar a cuando se ha producido. Esto es de gran utilidad para saber el tiempo que ha estado la alarma activada.

Un ejemplo se muestra en la Figura 6.78.

<b>ALERTAS ANOMALÍA FUNCIONAMIENTO</b>			
<b>DIA</b>	<b>HORA</b>	<b>ALARMA</b>	<b>ALERTA FUNCIONAMIENTO</b>
martes, 4 de abril de 2017	13h 23m	ALARMA ACTIVADA	¡ALTA PRESIÓN SALIDA SOBRECALENTADOR!
martes, 4 de abril de 2017	14h 01m	ALARMA DESACTIVADA	¡ALTA PRESIÓN SALIDA SOBRECALENTADOR!
lunes, 17 de abril de 2017	15h 46m	ALARMA ACTIVADA	¡ALTA PRESIÓN SALIDA SOBRECALENTADOR!
lunes, 17 de abril de 2017	15h 47m	ALARMA DESACTIVADA	¡ALTA PRESIÓN SALIDA SOBRECALENTADOR!
lunes, 8 de mayo de 2017	11h 05m	ALARMA ACTIVADA	¡ALTA TEMPERATURA SALIDA SOBRECALENTADOR!
lunes, 8 de mayo de 2017	11h 05m	ALARMA DESACTIVADA	¡ALTA TEMPERATURA SALIDA SOBRECALENTADOR!

*Figura 6.78: Alerta estado alarma (Elaboración propia)*

### 6.2.5.2 Prueba envío de email vía smtp

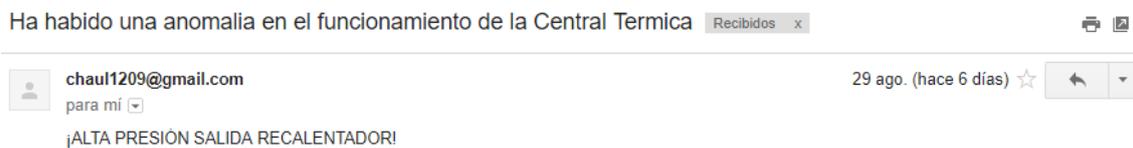
Para finalizar las pruebas del SCADA se debe comprobar que cuando surge una alerta se avisa al operario o encargado del proceso con el envío de un correo electrónico, lo cual alertará a este e impondrá las medidas oportunas en función del tipo de alerta.

En el caso de envío del correo electrónico, solo se avisará cuando la alarma se active y no se enviará cuando se desactive la alerta.

Un ejemplo de envío se muestra en las Figuras 6.79 y 6.80.



*Figura 6.79: Correo en bandeja de entrada (Elaboración propia)*



*Figura 6.80: Contenido correo electrónico (Elaboración propia)*



## CAPITULO 7: MANUAL DE INSTALACIÓN Y CONFIGURACIÓN

### 7.1 CONFIGURACIÓN DE VARIABLES E INSTALACIÓN

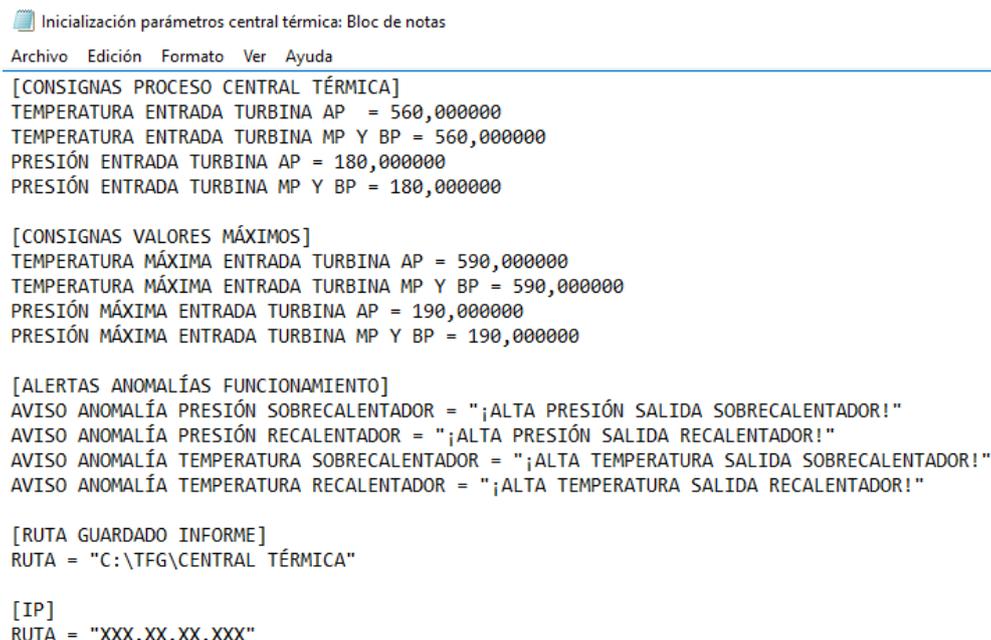
Una de las grandes ventajas, es la capacidad de poder configurar los estados de la Central en cualquier momento, sin necesidad de tener instalado el Software LabVIEW.

Como los archivos de lectura con los que trabaja el SCADA son archivos de texto (Figura 7.81), cualquier ordenador o Tablet tiene la capacidad de abrirlos y modificarlos, por lo que el usuario puede cambiar cualquier valor y guardar el archivo.

Para que el SCADA lea la nueva configuración el usuario deberá reiniciar la aplicación o el SCADA.

Es muy importante que el usuario solo modifique los valores o textos después del « igual », ya que si modifica datos anteriores a este, el SCADA en la lectura, dará un error debido a la referencia o llamada de textos.

Por la misma razón tampoco se pueden quitar las comillas dónde aparezcan.



```
Inicialización parámetros central térmica: Bloc de notas
Archivo Edición Formato Ver Ayuda
[CONSIGNAS PROCESO CENTRAL TÉRMICA]
TEMPERATURA ENTRADA TURBINA AP = 560,000000
TEMPERATURA ENTRADA TURBINA MP Y BP = 560,000000
PRESIÓN ENTRADA TURBINA AP = 180,000000
PRESIÓN ENTRADA TURBINA MP Y BP = 180,000000

[CONSIGNAS VALORES MÁXIMOS]
TEMPERATURA MÁXIMA ENTRADA TURBINA AP = 590,000000
TEMPERATURA MÁXIMA ENTRADA TURBINA MP Y BP = 590,000000
PRESIÓN MÁXIMA ENTRADA TURBINA AP = 190,000000
PRESIÓN MÁXIMA ENTRADA TURBINA MP Y BP = 190,000000

[ALERTAS ANOMALÍAS FUNCIONAMIENTO]
AVISO ANOMALÍA PRESIÓN SOBRECALENTADOR = "¡ALTA PRESIÓN SALIDA SOBRECALENTADOR!"
AVISO ANOMALÍA PRESIÓN RECALENTADOR = "¡ALTA PRESIÓN SALIDA RECALENTADOR!"
AVISO ANOMALÍA TEMPERATURA SOBRECALENTADOR = "¡ALTA TEMPERATURA SALIDA SOBRECALENTADOR!"
AVISO ANOMALÍA TEMPERATURA RECALENTADOR = "¡ALTA TEMPERATURA SALIDA RECALENTADOR!"

[RUTA GUARDADO INFORME]
RUTA = "C:\TFG\CENTRAL TÉRMICA"

[IP]
RUTA = "XXX.XX.XX.XXX"
```

*Figura 7.81: Variables modificables (Elaboración propia)*

Los parámetros que se permiten configurar son:

- **Consignas funcionamiento Central:** Con estos parámetros se puede modificar la temperatura y presión de referencia, es decir, las electroválvulas no dejarán pasar el vapor a la turbina hasta que no se llegue a unos determinados valores.

Con esto se busca mantener siempre unos parámetros en concreto con los que las turbinas darán el mejor rendimiento posible.

Estos valores aparecen en la columna de la izquierda de la Figura 7.82.

- **Consignas Alertas funcionamiento Central:** Con estos parámetros se puede modificar la temperatura y presión máxima a la que puede acceder el vapor a las turbinas.

De este modo se evitarán aumentos excesivos de presión y/o temperatura en las turbinas.

Estos valores aparecen en la columna de la derecha de la Figura 7.82.

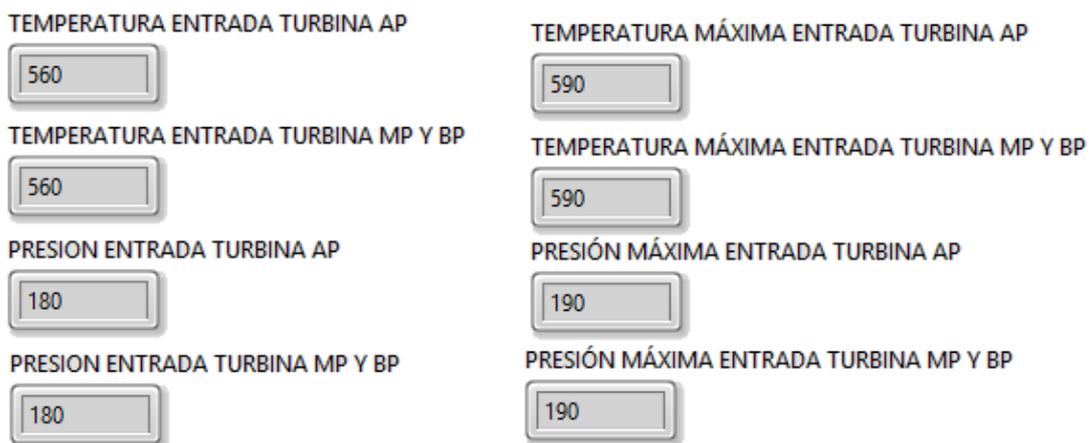


Figura 7.82: Variables en el SCADA (Elaboración propia)

- **Ruta informe:** Esta será la ruta que empleará el SCADA para guardar y leer los documentos necesarios para su correcto funcionamiento.

Si el usuario modifica la ruta, deberá reiniciar el SCADA para que este coja la nueva ruta y no haya errores.

- **IP:** Si por cualquier circunstancia el PLC cambia de IP, el usuario puede cambiar la IP vieja por la nueva.

También valdría para conectarse a otros PLC con distinta IP, pero estos deberán tener los mismos módulos y ubicados en los mismos lugares para que no haya incoherencias con el SCADA.



## IMPLEMENTACIONES FUTURAS

Este TFG se puede dividir en tres partes:

- **Diseño SCADA:** El diseño del SCADA en este TFG ha sido quizás la parte más trabajada y en la que el autor más se ha centrado.

La evolución de esta parte es bastante madura, por lo que es difícil poder ampliarla más, ya que si se amplía seguramente pierda comprensión e interpretación.

- **Programación SCADA:** Como se ha mencionado en numerosos apartados de la presente memoria, en este TFG el autor se ha centrado en conseguir implementar al SCADA un historial de alertas con envío a su vez de correo electrónico.

Este historial se almacena en un Excel y en una tabla en la segunda pestaña del SCADA.

Dicho historial se puede agrandar enormemente con infinitas alertas en diversos puntos del proceso, y además, se pueden imponer condiciones para que se implanten en el SCADA, de tal forma que si surge una alerta el programa sepa cómo responder sin necesidad de la manipulación humana.

Esto es una gran ventaja ya que independizaría aún más el SCADA dándole más responsabilidad a este y por supuesto menos al operario o encargado del proceso.

De esta forma podría conseguirse un ciclo cerrado más efectivo y con mayor rendimiento.

Además, se podría adaptar la programación para que se pudiera conectar en PLC distintos, con diferente IP, dotando el programa de más versatilidad.

- **Programación PC-PLC:** Esta parte del TFG quizás sea la parte más sencilla, basta con saber manipular PLC y cuál es su comportamiento.

La única razón por la que se ha empleado un PLC en este TFG, ha sido para no convertir el trabajo en uno completamente teórico, para poder así, transformarlo en un trabajo más visual y más fácil de interpretar.

Su continuidad y evolución también están presentes aunque quizás no sea la parte ideal, ya que una evolución más extensa supone remodelar las entradas y salidas adaptadas a la maqueta.

## OBJETIVOS Y CONCLUSIONES

Los objetivos cumplidos en este trabajo son:

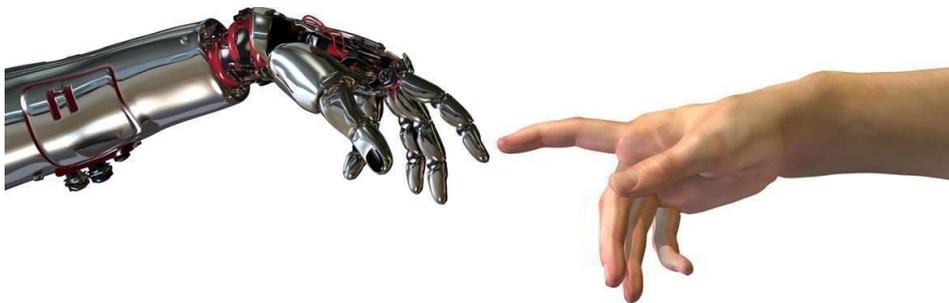
1. Crear una Interface Gráfica para poder controlar y visualizar un proceso industrial desde varios puntos.
2. Crear un sistema SCADA configurable o fácilmente configurable.
3. Empleo de un PLC de National Instruments modular, con el que se llevará a cabo el caso práctico del proceso industrial.
4. Establecer con el SCADA unos estados límites de funcionamiento, y en el caso de superar estos límites, reaccionar ante ellos de manera apropiada.
5. Almacenar en un historial y gestionar las alertas que surjan en el transcurso del proceso.
6. Avisar al encargado de la instalación o al usuario mediante correo electrónico, el estado de la central cuando sea necesario.
7. Visualización del estado de la Central vía Web.

Los sistemas SCADA ofrecen una perspectiva integrada de todos los recursos de control e información de la planta.

Una premisa fundamental en la automatización industrial es que las máquinas hagan lo que les corresponde, y que el ser humano no haga lo que una máquina o robot pueda realizar mecánicamente.

De esta forma, el ser humano dedicará su tiempo a las tareas que las máquinas no pueden hacer y no a trabajos repetitivos que fácilmente una máquina pueda realizar o supervisar.

Además de todo esto, se tiene que tener en cuenta que los humanos cometen fallos, debido a numerosos factores como el cansancio o el trabajo repetitivo. No quiere decir que los sistemas SCADA no fallen nunca, si pueden fallar, pero se pueden minimizar mucho con un buen mantenimiento y supervisión.



*Figura 0.83: Industria 4.0 vs Humanos (<https://www.taringa.net>)*

Además, es de vital importancia mantener la salud y la seguridad ocupacional en las diferentes tareas que se realizan en todo proceso productivo, lo que se ve favorecido por la automatización mediante los sistemas SCADA.

Los trabajos de actualización tecnológica y automatización implican riesgos que pueden evitarse o mitigarse por medio de una cuidadosa planificación de las actividades, adoptando la tecnología que mejor se ajusta en cada caso y realizando una ingeniería detallada y un exhaustivo conjunto de pruebas para cada sistema a implementar.

Hoy en día, poder ver el estado de cualquier elemento o sistema desde nuestra propia casa o en la terraza de un bar, facilita enormemente la eficiencia del sistema, y además se puede transmitir parte de la responsabilidad al sistema, pudiendo trabajar sin la supervisión constante del operario o encargado.

El avance y la complejidad de los nuevos procesos industriales han obligado a las empresas a buscar soluciones de integración de distintas tecnologías.



## RECOMENDACIONES

Debido a la revolución que estamos viviendo últimamente con la tecnología y la competencia que existe entre empresas, el alto desarrollo y fuerte automatización de los procesos industriales hace que sea imprescindible.

Esta fuerte automatización requiere un seguimiento y un control. Un claro ejemplo es este Trabajo de Fin de Grado, el cual da una idea clara de las cualidades que tienen los SCADA, y lo necesarios que son en la actualidad.

Actualmente todas las empresas industriales con una fuerte automatización trabajan en desarrollo de SCADA o subcontratan empresas para desarrollarlos.

Los procesos no van a ser todos iguales, pero este trabajo da una idea de cómo puede ser la base, otros procesos en vez de gestionar alertas, pueden contar por ejemplo número de piezas bien realizadas o defectuosas por turno...infinitas cosas, que ayudan siempre en la eficiencia y rendimiento del proceso.



## BIBLIOGRAFÍA

- [1] Entorno gráfico de programación (2007)  
Editorial: Marcombo
- [2] Interfaz de usuario y SCADA  
<http://www.uhu.es>  
Consultada por última vez en marzo 2017
- [3] Introducción a SCADA  
<http://www.uco.es/grupos/eatco/automatica/ihtm/descargar/scada.pdf>  
Consultada por última vez en Marzo 2017.
- [4] Instrumentación y Control (Sistemas SCADA)  
<http://www.instrumentacionycontrol.net/cursos-libres/automatizacion/cursos-sistemas-scada/item/149-componentes-en-un-sistema-scada.html>  
Consultada por última vez en Marzo 2017.
- [5] Labview For Everyone: Graphical Programming (2005)  
Autor: Jeffrey Travis / Jim Kring  
Editorial: Prentice Hall
- [6] Electro Industria. Estándar para comunicaciones OPC  
<http://www.emb.cl/electroindustria/articulo.mvc?xid=764&edi=43&xit=opc-el-estandar-para-comunicaciones-entre-dispositivos-y-sistemas-de-control-de-procesos>  
Consultada por última vez en Marzo 2017.
- [7] National Instruments. Fundamentos del entorno de labview.  
<http://www.ni.com/getting-started/labview-basics/esa/environment>  
Consultada por última vez en Junio 2017.

- [8] National Instruments.  
<http://www.ni.com/es-es.html>  
Consultada por última vez en Junio 2017.
- [9] Curso de Labview paso por paso  
<https://www.youtube.com/channel/UCBqCGO4mBpCEw1SyN1mGHJA>  
Consultada por última vez Febrero 2017.
- [10] Como hacer un ventilador con un interruptor y poder regular su velocidad  
<https://www.youtube.com/watch?v=IS2M0mLJ2Es>.  
Consultada por última vez en Enero 2017.
- [11] SCADA con tanques.  
<https://www.youtube.com/watch?v=syuK1Um4vKs>  
<https://www.youtube.com/watch?v=qAaX1jBOEio>  
<https://www.youtube.com/watch?v=Ay2182-Agp8&t=0s>  
Consultada por última vez en Enero 2017.
- [12] National Instruments. Programación Gráfica de labview.  
<http://www.ni.com/getting-started/labview-basics/esa/dataflow>  
Consultada por última vez en Junio 2017.
- [13] National Instruments. Estructuras de Datos de labview.  
<http://www.ni.com/getting-started/labview-basics/esa/data-structures>  
Consultada por última vez en Junio 2017.
- [14] National Instruments. Estructuras en ejecución de labview.  
<http://www.ni.com/getting-started/labview-basics/esa/execution-structures>  
Consultada por última vez en Junio 2017.

## ANEXOS

Se adjuntan los siguientes anexos:

- ANEXO 1: SCADA
- ANEXO 2: DIAGRAMA DE FLUJO
- ANEXO 3: ESQUEMA ELÉCTRICO