

**INFORME FINAL DE INVESTIGACIÓN**

**DISEÑO DE SIMULADOR DE CONTROL DE  
PROCESOS DE TEMPERATURA, UTILIZANDO  
CONTROLADOR PROPORCIONAL-INTEGRAL PI  
Y PROPORCIONAL-INTEGRAL-DERIVATIVO PID**

**APLICACIÓN ACADÉMICA EN ESCUELA DE  
INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

**DOCENTE INVESTIGADOR PRINCIPAL:  
TÉC. JUAN JOSÉ GUEVARA VÁSQUEZ**

**DOCENTE CO-INVESTIGADOR:  
ING. JUAN JOSÉ CÁCERES CHIQUILLO**

**ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA  
ITCA-FEPADE SEDE CENTRAL**

**ENERO 2021**



**INFORME FINAL DE INVESTIGACIÓN**

**DISEÑO DE SIMULADOR DE CONTROL DE  
PROCESOS DE TEMPERATURA, UTILIZANDO  
CONTROLADOR PROPORCIONAL-INTEGRAL PI  
Y PROPORCIONAL-INTEGRAL-DERIVATIVO PID**

**APLICACIÓN ACADÉMICA EN ESCUELA DE  
INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

**DOCENTE INVESTIGADOR PRINCIPAL:  
TÉC. JUAN JOSÉ GUEVARA VÁSQUEZ**

**DOCENTE CO-INVESTIGADOR:  
ING. JUAN JOSÉ CÁCERES CHIQUILLO**

**ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA  
ITCA-FEPADE SEDE CENTRAL**

**ENERO 2021**

### **Rectora**

Licda. Elsy Escolar SantoDomingo

### **Vicerrector Académico**

Ing. Carlos Alberto Arriola Martínez

### **Vicerrectora Técnica Administrativa**

Inga. Frineé Violeta Castillo

### **Director de Investigación y Proyección Social**

Ing. Mario W. Montes Arias

### **Dirección de Investigación y Proyección Social**

Ing. David Emmanuel Ágreda Trujillo

Inga. Ingrid Janeth Ulloa de Posada

Sra. Edith Aracely Cardoza de González

Director de Escuela de

Ingeniería Eléctrica y Electrónica

Ing. Carlos Roberto García

621.402 201 13

G939d Guevara Vásquez, Juan José, 1978 -

slv Diseño de simulador de control de procesos de temperatura utilizando Controlador Proporcional-Integral PI y Proporcional-Integral-Derivativo PID [recurso electrónico] : aplicación académica en Escuela de Ingeniería Eléctrica y Electrónica / Juan José Guevara Vásquez, Juan José Cáceres Chiquillo. -- 1ª ed. -- Santa Tecla, La Libertad, El Salv. : ITCA Editores, 2021.

1 recurso electrónico (50 p. : il. col. ; 28 cm.)

Datos electrónicos (1 archivo : pdf, 5.4 Mb). –  
<https://www.itca.edu.sv/produccion-academica/>  
ISBN : 978-99961-39-70-3 (E-Book, pdf)

1. Control de la temperatura - Automatización. 2. Simulación por computadores digitales. 3. Control Automático - Equipo. I. Cáceres Chiquillo, Juan José, 1977-, coaut. II. Título.

BINA/jnh

### **Autor**

Téc. Juan José Guevara Vásquez

### **Co Autor**

Ing. Juan José Cáceres Chiquillo

Tiraje: 13 ejemplares

Año 2021

Este documento técnico es una publicación de la Escuela Especializada en Ingeniería ITCA-FEPADE; tiene el propósito de difundir la Ciencia, la Tecnología y la Innovación CTI, entre la comunidad académica, el sector empresarial y la sociedad, como un aporte al desarrollo del país. Para referirse al contenido debe citar el nombre del autor y el título del documento. El contenido de este Informe es responsabilidad de los autores.



Atribución-No Comercial  
Compartir Igual  
4.0 Internacional

Esta obra está bajo una licencia Creative Commons. No se permite el uso comercial de la obra original ni de las posibles obras derivadas, cuya distribución debe hacerse mediante una licencia igual que la sujeta a la obra original.

Escuela Especializada en Ingeniería ITCA-FEPADE  
Km 11.5 carretera a Santa Tecla, La Libertad, El Salvador, Centro América

Sitio Web: [www.itca.edu.sv](http://www.itca.edu.sv)

TEL: (503)2132-7423

## CONTENIDO

1.	INTRODUCCIÓN.....	4
2.	PLANTEAMIENTO DEL PROBLEMA .....	5
2.1.	DEFINICIÓN DEL PROBLEMA.....	5
2.2.	ANTECEDENTES / ESTADO DE LA TÉCNICA.....	5
2.3.	JUSTIFICACIÓN .....	6
3.	OBJETIVOS.....	7
3.1.	OBJETIVO GENERAL.....	7
3.2.	OBJETIVOS ESPECÍFICOS .....	7
4.	HIPÓTESIS.....	7
5.	MARCO TEÓRICO.....	8
5.1.	EL CONTROL DE PROCESOS.....	8
5.2.	SISTEMA CONTROLADO.....	8
5.3.	TIPOS DE CONTROL .....	11
5.4.	SENSORES Y TRANSDUCTORES .....	12
5.5.	EL LABORATORIO DE CONTROL DE PROCESOS .....	14
5.6.	LABVIEW .....	16
6.	METODOLOGÍA DE INVESTIGACIÓN .....	19
7.	RESULTADOS.....	23
7.1.	ECUACIÓN DIFERENCIAL DEL ENTRENADOR DE CONTROL DE TEMPERATURA.....	23
7.2.	CÓDIGO QUE IMPLEMENTA EL MÉTODO DE EULER PARA SOLUCIONAR ECUACIONES DIFERENCIALES .....	23
7.3.	MODELO DIFERENCIAL DE LA PLANTA (ENTRENADOR DE TEMPERATURA) .....	25
7.4.	MODELO DE LA FUNCIÓN DE TRANSFERENCIA.....	26
7.5.	MODELO DE LA PLANTA + CONTROL PID .....	26
7.6.	CÓDIGO DEL CONTROL PID.....	27
7.7.	DIAGRAMA DE BLOQUES DEL SIMULADOR .....	29
7.8.	PANEL DE INSTRUMENTOS (VI) DEL SIMULADOR .....	30
8.	CONCLUSIONES.....	31
9.	RECOMENDACIONES.....	31
10.	GLOSARIO.....	32
11.	REFERENCIAS BIBLIOGRÁFICAS.....	33
12.	ANEXOS.....	34
12.1.	ANEXO 1: GUÍA DE INSTALACIÓN Y USO.....	34
12.2.	ANEXO 2. TUTORIAL DE LABVIEW .....	42

## 1. INTRODUCCIÓN

Los simuladores son ampliamente utilizados en entornos científicos y educativos para poner a prueba sistemas, circuitos y maquinaria dentro de ambientes controlados con la finalidad de reducir riesgos y daños en ellos, proporcionando además un mecanismo más rápido y eficaz para hacer ajustes en el menor tiempo posible. En el estudio de los sistemas de control de procesos, los simuladores permiten a investigadores, docentes y estudiantes evaluar el comportamiento de un sistema a partir de los parámetros de control y predecir su comportamiento a fin de reducir el tiempo de ajuste en sistemas reales.

El objetivo de este proyecto fue estudiar y analizar los elementos que forman parte de un entrenador de control de procesos de temperatura, para crear un modelo matemático que simule su comportamiento a través de un programa informático, permitiendo a docentes y estudiantes realizar simulaciones del funcionamiento en sus computadoras, con la seguridad de que los resultados serán similares a los que se obtendrían al realizar las prácticas de laboratorio en un entrenador real.

Durante el desarrollo de este proyecto, se analizó el comportamiento teórico de un entrenador de control de procesos de temperatura, así como sus características físicas para obtener un modelo matemático que permitiera predecir su comportamiento bajo diversas técnicas de control y con base en los resultados obtenidos, se diseñó un simulador PI y PID con el software LabView 2020 (versión demo) así como guías de práctica con distintos niveles de complejidad para que docentes y estudiantes puedan experimentar y simular el comportamiento del entrenador desde sus hogares, oficinas y otros espacios de manera que se pueda comprender de una forma interactiva el comportamiento del entrenador y reducir el tiempo de ajustes en las prácticas presenciales de laboratorio, así como la disminución del riesgo de daños en el equipo.

El simulador de control de procesos de temperatura diseñado permite al usuario realizar distintos ajustes de las constantes proporcional, integral y derivativa. Se elaboró una guía de usuario para la instalación, ajustes y uso del simulador. Además, se diseñaron cuatro guías de laboratorio para control de procesos de temperatura, dos de ellas de nivel intermedio para los estudiantes de carreras técnicas y otras dos de nivel avanzado para estudiantes de carreras de ingeniería.

Con este proyecto se demostró que es posible modelar matemáticamente un proceso industrial para predecir su comportamiento y sobre la base de este crear un controlador proporcional, integral y derivativo que mantenga controlada la temperatura a pesar de las variaciones a las que estaría sujeto en la realidad. Mediante el uso de simuladores como el diseñado durante esta investigación, se mejora el aprendizaje de los estudiantes en el área de control de procesos y control automático.

## **2. PLANTEAMIENTO DEL PROBLEMA**

### **2.1. DEFINICIÓN DEL PROBLEMA**

En las instituciones educativas, sucede con bastante regularidad que los equipos especializados para prácticas de laboratorio son escasos ya que son de alto costo y mantenimiento. Esto tiene como inconveniente que la cantidad de horas efectivas que los estudiantes pueden pasar frente a estos equipos es muy reducida, dificultando su proceso de aprendizaje.

En los laboratorios de control de la Escuela de Ingeniería Eléctrica y Electrónica se dispone solamente de un entrenador físico de un total de cuatro procesos industriales distintos, esto reduce considerablemente el tiempo de trabajo y experimentación de estudiantes y docentes de cursos y módulos afines, reduciendo además, el tiempo de vida útil de cada entrenador y aumentando el riesgo de fallos en los equipos. Una reducción del tiempo de acceso a los entrenadores por parte de los estudiantes limita en número de pruebas y ajustes que deben efectuar para comprender su comportamiento ante las técnicas de control PI y PID que son las más utilizadas en el control de procesos industriales.

El proyecto busca proveer una solución que permita optimizar el tiempo de acceso a los equipos para las prácticas de control de procesos industriales y registrar las actividades que los estudiantes lleven a cabo para retroalimentar al docente oportunamente.

### **2.2. ANTECEDENTES / ESTADO DE LA TÉCNICA**

El uso de simuladores de procesos industriales es hoy en día una práctica habitual sobre todo debido al auge de las computadoras, dispositivos móviles inteligentes y redes de datos. Muchas instituciones de educación públicas y privadas diseñan, implementan y mejoran simuladores no solo de control de procesos sino de otras áreas como química, electrónica, automotriz, etc. y los proporcionan como herramientas a sus docentes y estudiantes. La empresa privada, por otra parte, elabora simuladores más potentes, pero con el objetivo de capacitar a su personal y clientes que han adquirido sus productos, algunos de estos simuladores son accesibles a los usuarios, pagando una suscripción mensual o licencia permanente.

A nivel internacional ya existen laboratorios que permiten la simulación y observación de prácticas, todos ellos son resultados de proyectos de investigación. Las universidades españolas Complutense de Madrid y UNED poseen experiencia en esta área y han publicado los resultados de sus investigaciones en revistas y conferencias de ámbito científico. En Latinoamérica, el Instituto Tecnológico de Monterrey ha implementado este tipo de laboratorios en electricidad, electrónica y automatización. En El Salvador no se tienen registros sobre plataformas de este tipo en Instituciones de Educación Superior ni patentes relacionadas, y posiblemente tampoco en la región centroamericana. En los laboratorios existentes se han

utilizado diversas tecnologías, cada una de ellas acorde a las necesidades puntuales y experticia de las instituciones que las diseñaron. Sin embargo, tienen en común la integración de servicios de Internet, seguridad informática e industrial y control de acceso a los estudiantes.

Un trabajo de tesis para el grado de Master en Ciencias denominado “A Control Perspective to Adaptive Time Stepping in Reservoir Simulation” [1] consultado en la Universidad de Texas explica extensivamente el proceso de análisis y diseño teórico de un simulador con similares prestaciones al que se desarrolló en este proyecto de investigación.

En México se encontró una solicitud de patente del inventor Javier Cerón Solís que pertenece a la Dirección Nacional de Educación Superior Tecnológica de la Secretaría de Educación Pública, la patente con el título “Sistema de simulación de control proporcional integral derivativo de presión, flujo, nivel y temperatura de agua” [2]. Esta patente describe el diseño de un simulador con similares prestaciones al desarrollado en la presente investigación, pero con la incorporación de más variables físicas como la temperatura, flujo y nivel del fluido, este tipo de simulador suele proveerse junto a un entrenador de control de procesos por el fabricante a un costo significativo.

En El Salvador no se tiene conocimiento del diseño de patentes en universidades o empresa privada, una búsqueda realizada en el registro de patentes del CNR no arrojó resultados significativos.

### **2.3. JUSTIFICACIÓN**

La pandemia por COVID-19 ha sido sin duda alguna un fenómeno que ha afectado a la sociedad y nos ha obligado a cambiar los sistemas de educación tradicional por los modelos de formación a distancia, esto con el objetivo de disminuir en lo posible el contacto físico con otras personas y así reducir significativamente el riesgo de contagio.

En este contexto, con el diseño y utilización de un simulador de entrenador de control de temperatura, el instructor podrá realizar prácticas de laboratorio con la seguridad que los resultados que obtengan los estudiantes al hacer las pruebas en sus computadoras serán similares a los que se obtendrían en el entrenador real.

Los estudiantes tendrán una experiencia muy cercana a la realidad ya que, si bien no estarán físicamente frente al entrenador, el simulador provee de los instrumentos de medición y ajuste en una disposición similar a la que el entrenador real posee proporcionando información visual fácil de leer y comprender, adicionalmente permitirá interactuar con el control del proceso oportunamente y desarrollar adecuadamente sus prácticas de laboratorio.

Mediante la incorporación de herramientas de acceso y control remoto al entrenador de control de temperatura, el instructor podrá programar eficientemente el acceso a los estudiantes para el desarrollo de las prácticas de laboratorio. Los estudiantes tendrán una experiencia real ya que, si bien no estarán físicamente frente al entrenador, la incorporación de un panel de control electrónico virtual proporcionará información en tiempo real a los estudiantes y les permitirá interactuar oportunamente y desarrollar adecuadamente sus prácticas de laboratorio.

De esta manera, el proyecto de investigación contribuye no solamente a optimizar los recursos con los que cuenta la institución haciendo un uso efectivo de ellos, sino que también mejora considerablemente el aprendizaje de los estudiantes y al mismo tiempo permite disminuir el riesgo de contagio por enfermedades altamente contagiosas como el COVID-19. La Escuela Especializada en Ingeniería ITCA-FEPADE dará un salto tecnológico en educación ya que la tendencia es utilizar simuladores y hacerlos accesibles a estudiantes, docentes e investigadores que se encuentran en regiones lejanas.

### **3. OBJETIVOS**

#### **3.1. OBJETIVO GENERAL**

Desarrollar un entrenador de control de procesos de temperatura a través de un modelo matemático que permita realizar simulaciones del funcionamiento en computadoras con la seguridad que los resultados serán similares a los obtenidos al manipular el entrenador físico.

#### **3.2. OBJETIVOS ESPECÍFICOS**

1. Diseñar un simulador de control de procesos, controlando la variable de temperatura, que permita al usuario realizar distintos ajustes de las constantes proporcional, integral y derivativa.
2. Crear una guía de usuario para la instalación, ajustes y uso del simulador.
3. Crear cuatro guías de laboratorio para control de procesos de temperatura, dos de ellas de nivel intermedio para los estudiantes de carreras técnicas y dos de nivel avanzado para estudiantes de carreras de ingenierías.

### **4. HIPÓTESIS**

El diseño de un programa simulador de control de procesos de temperatura con un controlador PID que simula el comportamiento de un entrenador de control de temperatura y permite predecir el

comportamiento del equipo a partir de la manipulación de las variables de control permitirá a docentes y estudiantes de cursos y módulos de control de procesos hacer múltiples pruebas y ajustes en sus computadoras facilitando la comprensión de las técnicas de control y el cálculo de sus variables que posteriormente se verán reflejadas al configurar el entrenador físico.

## **5. MARCO TEÓRICO**

### **5.1. EL CONTROL DE PROCESOS**

El control de procesos toma en cuenta la medición y el análisis de las variables que determinan el funcionamiento de un proceso, así como la toma de decisiones y la ejecución de acciones de control para gobernar dicho proceso [3].

Las plantas industriales se diseñan para llevar a cabo procesos productivos que permitan transformar la materia prima y los insumos en bienes cuyo valor lo establece el mercado.

En una economía globalizada y de alta competencia, la mejor manera de lograr un efectivo control de procesos es mediante la incorporación de la tecnología electrónica integrada a técnicas de control automático de procesos.

Esta tecnología está basada en la adquisición de datos extraídos directamente del proceso para ser analizados con el fin de tomar decisiones de control con un mínimo de intervención humana. La ventaja de obtener información directa del proceso es que tan pronto se produce un cambio, este es registrado y utilizado para mantener el proceso bajo control.

En los controles de procesos que se utilizan actualmente, se integran recursos de hardware y software cuyo diseño y fabricación sigue estándares industriales de propósito general tales como: Controladores Lógicos Programables (PLC), Sistemas de Control Distribuidos (DCS) y Controladores Avanzados de Procesos (PAC). Adicionalmente, se pueden utilizar tecnologías que involucran especialistas en programación tales como lenguaje C, VHDL y LabView.

### **5.2. SISTEMA CONTROLADO**

Un sistema es un conjunto de elementos interrelacionados entre sí, los cuales se caracterizan por poseer parámetros inherentes que los definen y que muestran condiciones físicas asociadas, susceptibles de evolucionar con el tiempo. Estos parámetros son específicos de cada elemento y son considerados como constantes, por lo que se les llama parámetros del sistema. Por otra parte, las condiciones físicas que

cambian con el tiempo son las que determinan el estado del sistema por lo que se expresan por medio de las denominadas variables del sistema. La magnitud y evolución de dichas variables vienen regidas por las leyes físicas que las gobiernan.

En teoría de control, cada uno de los componentes elementales en los cuales puede descomponerse un sistema, recibe el nombre de bloque y puede representarse gráficamente por medio de un rectángulo al cual se le agrega una flecha de entrada y otra de salida en sus costados (fig. 1). [4]

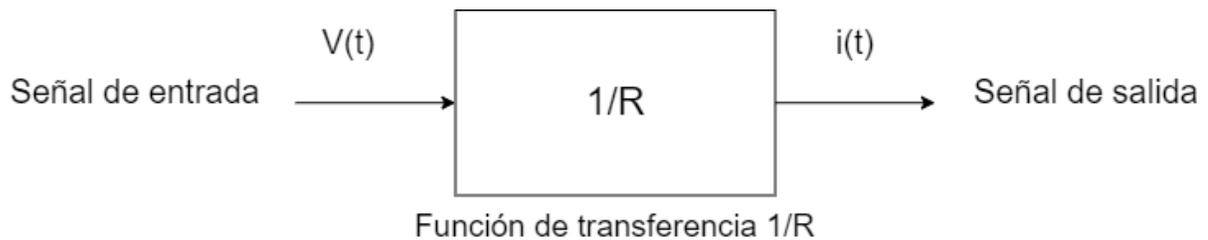


Fig. 1: Representación gráfica de un bloque

La señal de entrada corresponde a la variable física o variable de entrada, esta será convertida o manipulada por el bloque para producir una señal de salida o variable de salida. Cada uno de los bloques que forman parte de un sistema, se unen mediante líneas que con sus flechas indican la dirección del flujo de la información o señales que circulan a lo largo del sistema. El diagrama de bloques es una forma convencional de representar gráficamente las interrelaciones entre las variables significativas del sistema, así como las características de los componentes que lo forman.

Un proceso es un conjunto de equipos o dispositivos mecánicos, eléctricos, físicos, químicos, o de cualquier índole, dispuestos de tal manera que puedan en su conjunto realizar las actividades necesarias para alcanzar un objetivo. Un sistema controlado está formado por el proceso y el sistema de control.

La regulación o control automático en lazo cerrado consiste en sustituir la acción del hombre por un dispositivo llamado controlador o regulador, de manera que al conjunto de los componentes que llevarán a cabo el control de un proceso se llama sistema de control automático.

En la figura 2, se muestra un proceso que se controla automáticamente en lazo cerrado, en él puede observarse que el flujo de las señales se cierra a sí mismo, y que el sistema tiene como señal de entrada el punto de consigna (referencia o set point) y como señal de salida la variable controlada.

El dispositivo comparador entre las señales de consigna y medida suele formar parte del controlador.

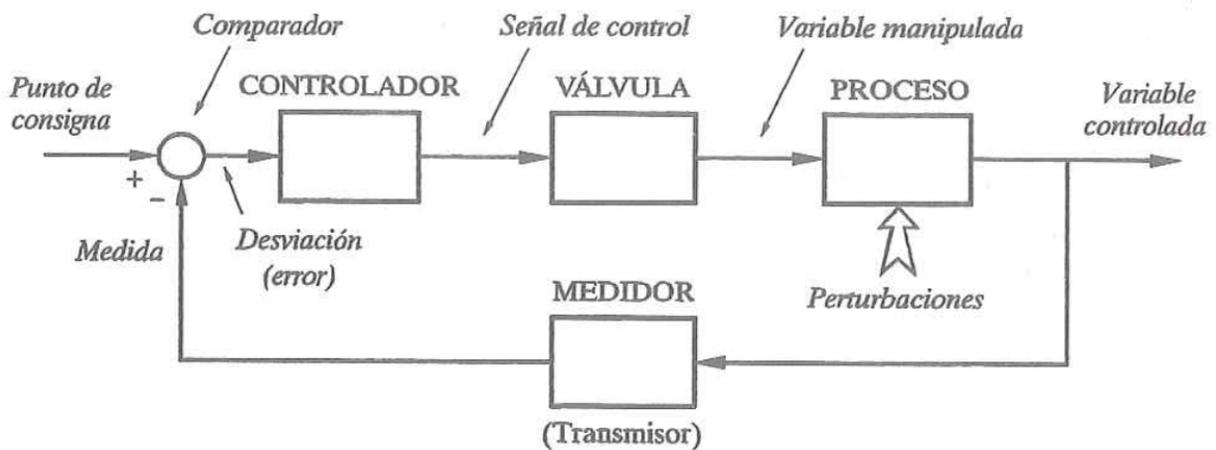


Fig. 2: Disposición básica de un proceso controlado automáticamente.

La realimentación consiste en tomar una medida de la variable controlada a través de sensores y enviarla hacia el controlador, para que genere la señal de error o desviación por medio de la comparación con la señal de referencia, a partir de la señal de error el controlador genera una señal de control o de ajuste de manera que en el ejemplo de la figura 2, se ajuste apropiadamente la posición de la válvula para que a la salida del proceso se alcancen condiciones más próximas a las deseadas.

Es entonces, objetivo del control automático la forma como, a partir de la aparición de una desviación, se efectuará el restablecimiento o recuperación del valor deseado en la variable manipulada. A esta forma de respuesta se le denomina respuesta transitoria.

Mientras que, una vez que se ha extinguido el transitorio (perturbación) y se ha alcanzado el punto deseado se genera una respuesta transitoria o estacionaria. Sin embargo, debido a las características de todo proceso real, no es técnicamente posible lograr una regulación ideal.

Así pues, la variable controlada puede efectuar la aproximación al punto de referencia de manera diferente a la ideal y de esta manera se pueden distinguir formas no satisfactorias de hacerlo:

- Aproximación demasiado lenta o errática.
- No estabilizarse en el valor de referencia.
- Rebasamiento transitorio excesivo del valor de referencia.
- Restablecimiento al valor de referencia después de excesivas oscilaciones amortiguadas alrededor del mismo.

- Presentar oscilaciones mantenidas crecientes en amplitud, lo cual puede afectar algún componente del sistema.
- Una combinación de las formas anteriores.

La recuperación del sistema ante perturbaciones debe efectuarse siempre con la máxima rapidez, el mínimo de rebasamiento y con una desviación permanentemente nula. [5]

### 5.3. TIPOS DE CONTROL

La unidad de control puede reaccionar de varias maneras ante una perturbación y proporcionar señales de salida para que actúen los elementos correctores del sistema. Los siguientes son algunos tipos de control:

- **De dos posiciones:** en este caso el controlador es en esencia un interruptor activado mediante la señal de error, este producirá únicamente una señal de corrección tipo encendido (on)-apagado(off).
- **Proporcional (P):** produce una acción de control proporcional al error. Por lo que la señal de corrección aumentará en la medida que lo haga el error. Por el contrario, si el error disminuye, también se reduce la magnitud de la corrección haciendo la respuesta más lenta.
- **Derivativo (D):** en este caso la corrección es proporcional a la rapidez con la que se cambia el error. Esto significa que, si se produce un cambio súbito en el error, la magnitud de la corrección es de gran magnitud. Por el contrario, si el cambio en el error es gradual, la señal de corrección es pequeña. Por esta característica, se considera al control derivativo como de tipo predictivo. Sin embargo, el control derivativo no se usa solo sino siempre en combinación con el proporcional y, también con el integral.
- **Integral (I):** este tipo de control produce una señal de corrección que es igual a la integral del error en función del tiempo, por lo que una señal de error constante producirá una señal de corrección creciente y seguirá aumentando progresivamente mientras el error persista. Se considera entonces, que el control integral mira hacia atrás, sumando todos los errores y responde a los cambios que ocurren.
- **Combinado:** es una combinación de los anteriores como el proporcional derivativo (PD), proporcional integral (PI) y proporcional integral derivativo (PID). [6]

## 5.4. SENSORES Y TRANSDUCTORES

El término sensor se refiere a un elemento que produce una señal relacionada con la cantidad que se está midiendo. Por ejemplo, si se está midiendo la temperatura con un sensor resistivo, la cantidad que se mide es temperatura y el sensor transforma la entrada en cambios en la resistencia eléctrica. Por lo general se utiliza el término transductor en lugar de sensor.

Un transductor se define como el elemento que al someterlo a un cambio físico experimenta un cambio relacionado, por lo que los sensores son en realidad transductores. Sin embargo, un sistema de medición puede utilizar transductores, además de sensores, en otras partes del sistema para convertir señales de una forma dada en otra distinta. Por otra parte, el término digital se utiliza cuando los sistemas ofrecen salidas que son digitales por naturaleza.

Los siguientes términos describen las características de funcionamiento de los transductores y de los sistemas de medición:

- **Intervalo y extensión:** define los límites entre los cuales puede variar la entrada.
- **Error:** es la diferencia entre el resultado de una medición y el valor verdadero de la cantidad que se mide:

$$error = valor\ medido - valor\ real$$

- **Exactitud:** es el grado hasta el cual un valor producido por un sistema de medición puede estar equivocado. Es, por lo tanto, la suma de todos los errores posibles más el error en la exactitud de la calibración del transductor.
- **Sensibilidad:** es la relación que indica cuanta salida se obtiene por unidad de entrada, es decir, salida/entrada. Por ejemplo, un termómetro de resistencia puede tener una sensibilidad de  $0.5\Omega/^{\circ}C$ .
- **Error por histéresis:** se produce cuando los transductores producen salidas distintas a partir de la misma cantidad medida ya sea por incremento o decremento continuo.
- **Error por no linealidad:** se supone que, en su intervalo de funcionamiento, la relación entre la entrada y la salida del transductor es lineal. Sin embargo, esto no es así, por lo que este error se define como desviación máxima respecto a la línea recta.
- **Repetibilidad/reproducibilidad:** describe la capacidad del transductor para producir la misma salida después de aplicar varias veces el mismo valor de entrada.

El error relacionado consiste en no obtener la misma salida después de aplicar el valor de entrada, se representa como un porcentaje del intervalo total de salida:

$$\text{repetibilidad} = \frac{\text{val. max} - \text{val. min}}{\text{intervalo total}} * 100$$

- **Estabilidad:** es la capacidad que posee un transductor para producir la misma salida cuando se usa para medir una entrada constante en un período.
- **Banda/tiempo muerto:** es el intervalo de valores de entrada para los cuales no hay salida.
- **Resolución:** es el cambio mínimo del valor de entrada capaz de producir un cambio observable en la salida.
- **Impedancia de salida:** cuando un sensor produce una salida eléctrica se conecta con otro circuito electrónico, es necesario conocer su impedancia de salida ya que esta se va a conectar en serie o en paralelo. Si no se toma en cuenta la impedancia de salida al conectar el sensor, el funcionamiento del sistema puede verse afectado.

### Sensor de temperatura RTD

Los detectores de temperatura por resistencia RTD, son elementos resistivos que adoptan la forma de bobinas de alambre hechas de platino, níquel o aleaciones de níquel-cobre; el platino es el que más se utiliza. Estos detectores de temperatura son muy estables y sus respuestas tienden a ser del orden de 0.5s a 5s, o mayores [1]. Los RTD son ampliamente utilizados en la industria ya que soportan rangos de temperatura arriba de 800°C. (fig. 3)



Fig. 3: Sensor RTD PT 100

## 5.5. EL LABORATORIO DE CONTROL DE PROCESOS

El laboratorio de control de procesos del cual dispone la Escuela de Ingeniería Eléctrica y Electrónica está conformado por entrenadores para los siguientes procesos:

- Control de temperatura.
- Control de nivel.
- Control de presión.
- Control de flujo.
- Control de mezcla.

El entrenador del control de temperatura está construido con partes que se utilizan para controlar procesos industriales de temperatura. (fig. 4)

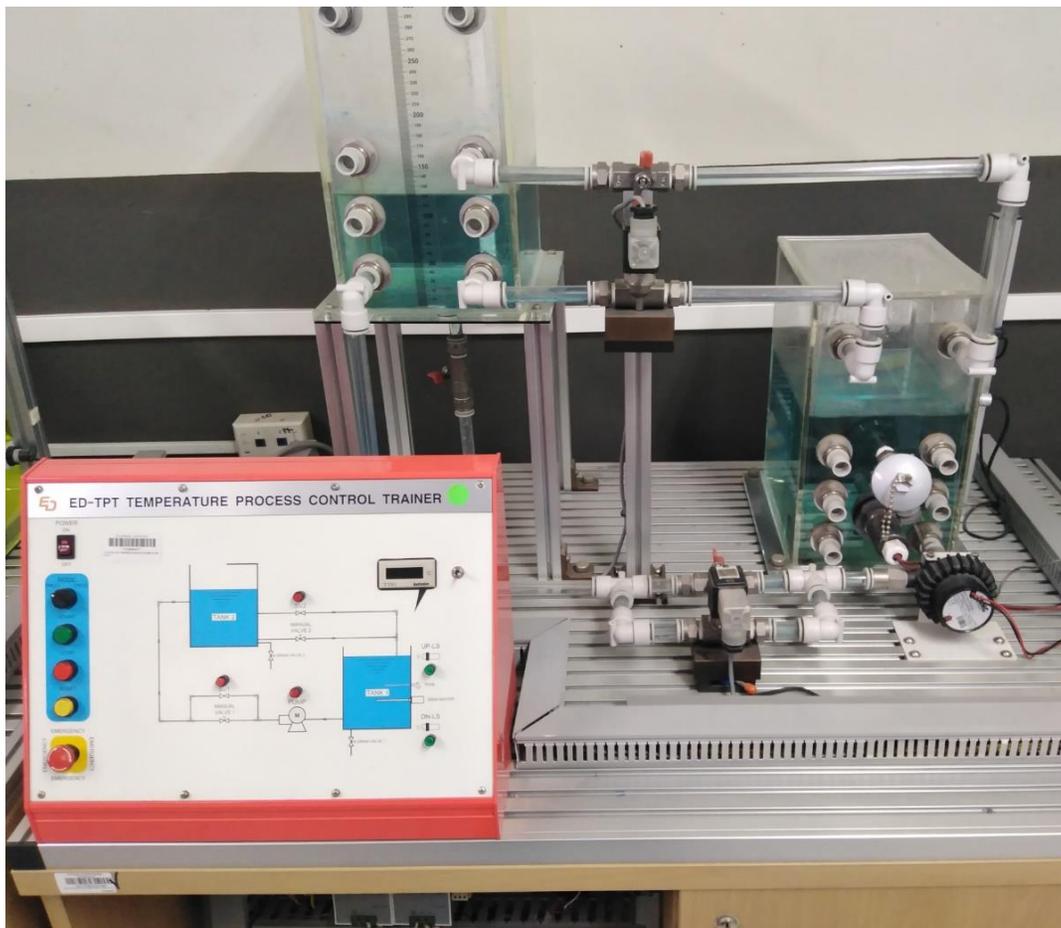


Fig. 4: Entrenador de control de temperatura del laboratorio de procesos J101.

Consiste en la adquisición de la temperatura del agua en el tanque por medio de un sensor de temperatura RTC PT 100. El agua está siendo calentada por una termo resistencia la cual a su vez es controlada por un relé de estado sólido, debido a las características del relé se puede efectuar un proceso de regulación por tiempos de conmutación del tipo PWM. En las figuras 5 y 6 puede observarse el diagrama esquemático del entrenador.

El establecimiento de un setpoint, regirá tanto el apagado de la termo resistencia así como un método de enfriamiento para poder controlar adecuadamente la temperatura. El método que se emplea para mantener la temperatura del agua en el valor predefinido, es por recirculación del líquido en el sistema. La fuerza necesaria para la circulación es proporcionada por una bomba de agua que obtiene el fluido del tanque 1 y activando a su vez las electroválvulas SV1 y SV2.

Debido a que el entrenador es realmente un banco de pruebas automatizado, es posible que los estudiantes puedan diseñar y aplicar sistemas de control automático como PD, PI y PID.

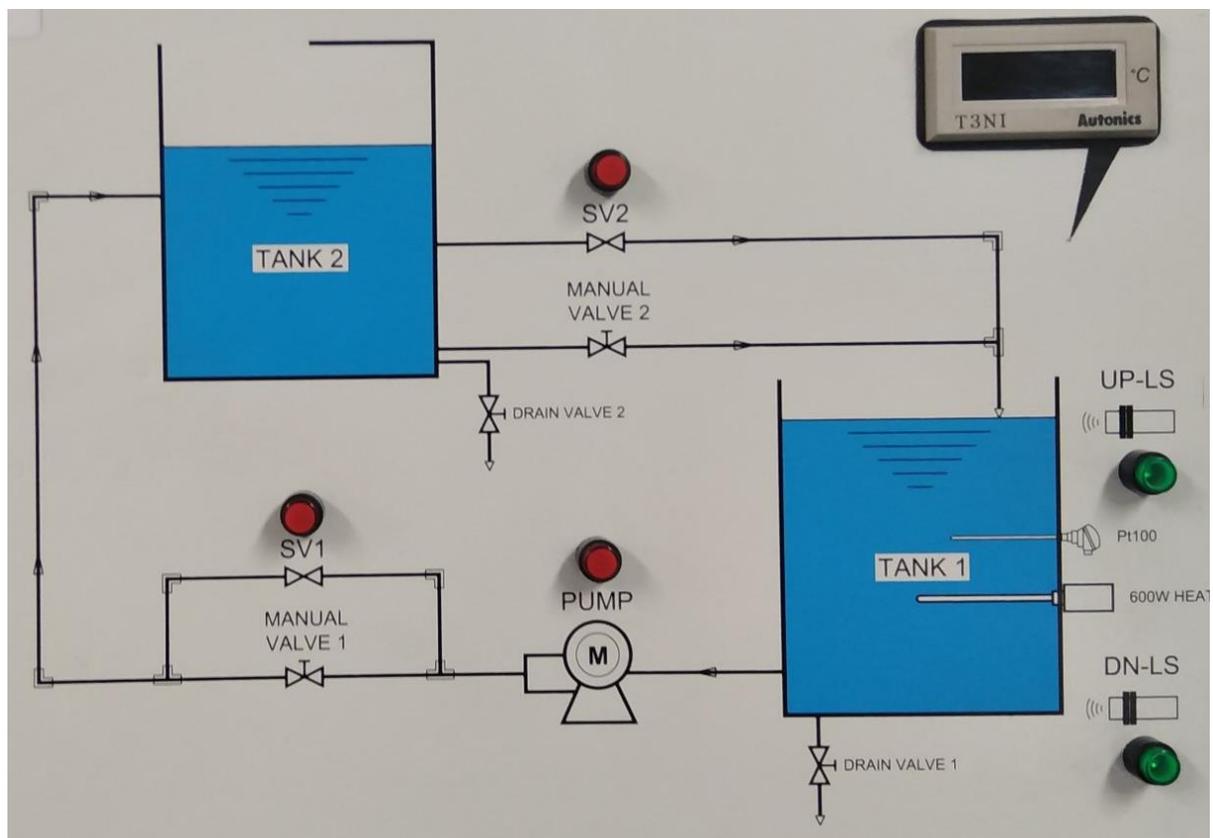


Fig. 5: Placa del entrenador de temperatura

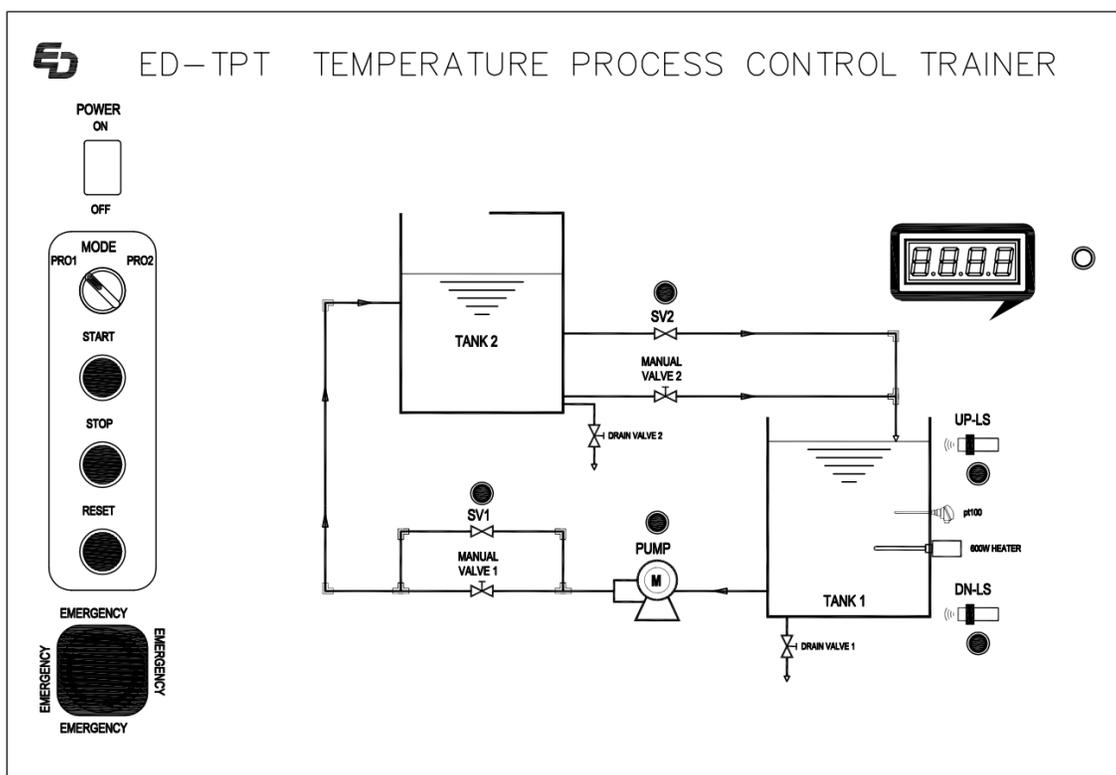


Fig. 6: Diagrama esquemático del entrenador de temperatura

## 5.6. LABVIEW

Es un entorno de programación gráfico introducido por National Instruments (NI) que simplifica el diseño de sistemas distribuidos de pruebas, medidas y control utilizando iconos intuitivos conocidos como instrumentos virtuales (VI). LabVIEW es escalable a través de múltiples sistemas operativos (OS) como Windows, Mac y Linux, de manera que actúa como una herramienta de desarrollo para multiplataforma [7].

Los programas de LabVIEW son llamados instrumentos virtuales o VIs ya que su apariencia y operación imitan a los instrumentos físicos, como osciloscopios y multímetros. 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 se escriba.

Un laboratorio remoto virtual es un ambiente de experimentación en el cual los usuarios pueden operar una serie de objetos gráficos, cada uno de ellos representa un objeto que efectúa experimentos virtuales vía internet por medio de operaciones locales realizadas con un ratón o teclado.

Con la ayuda de los laboratorios remotos virtuales, los estudiantes pueden efectuar experimentos a través de internet 24 horas al día, siete días de la semana, sin limitaciones de espacio y tiempo.

Adicionalmente, proporcionan a los estudiantes la posibilidad de efectuar prácticas y ejercicios de forma selectiva de manera que pueden seguir un proceso lógico de aprendizaje [5].

LabVIEW es un entorno de programación gráfico introducido por National Instruments (NI) que simplifica el diseño de sistemas distribuidos de pruebas, medidas y control utilizando iconos intuitivos conocidos como instrumentos virtuales (VI).

LabVIEW es escalable a través de múltiples sistemas operativos (OS) como Windows, Mac y Linux, de manera que actúa como una herramienta de desarrollo para multi plataforma [6].

Los proyectos de investigación relacionados con este tipo de laboratorios coinciden en que la arquitectura está basada en dos partes: servidor y cliente, esto significa que se aplica una arquitectura típica cliente-servidor. Sin embargo, difieren en cuanto a las prestaciones que posee la aplicación de lado del cliente.

Un enfoque sugiere utilizar un paquete LabVIEW descargable y ejecutable como requisito para acceder al sistema [7] lo cual implica delegar todo el control de seguridad, registro y monitoreo de las acciones del usuario a módulos de software creados con LabVIEW. [8].

Otro enfoque establece que el control de acceso debe ser administrado por un sistema CMS (ej. Drupal) o LMS (ej. Moodle), el cual se encarga de la validación principal, programación de acceso y administración de aulas virtuales (solo Moodle) [9], este enfoque es ideal para entornos de enseñanza y es utilizado por la UNED en sus laboratorios virtuales controlados a distancia [10].

Del lado del servidor, la arquitectura del laboratorio está compuesta por dos sistemas:

- **Sistema de Hardware.**

Este sistema esta a su vez constituido por 5 subsistemas:

1. El entrenador de control de temperatura.
2. Los dispositivos de control lógico y de potencia.
3. Las tarjetas electrónicas I/O para la lectura y escritura de datos en el entrenador.
4. El sistema de video cámaras.
5. La computadora que se utilizará como servidor.

- **Sistema de Software**

Es la parte lógica del sistema, está compuesto por los siguientes subsistemas:

1. Panel de Instrumentación Virtual.
2. Servidor Web.
3. Servidor de streaming de video.
4. Módulo de Control de Acceso.
5. Módulo de Control.
6. Registro de Prácticas de Laboratorio.

Esto puede observarse en la arquitectura empleada por la universidad UNED para su laboratorio virtual de robótica [10].

#### ARQUITECTURA DE LABORATORIO VIRTUAL DEL INSTITUTO DE TECNOLOGÍA DE HYDERABAD, INDIA

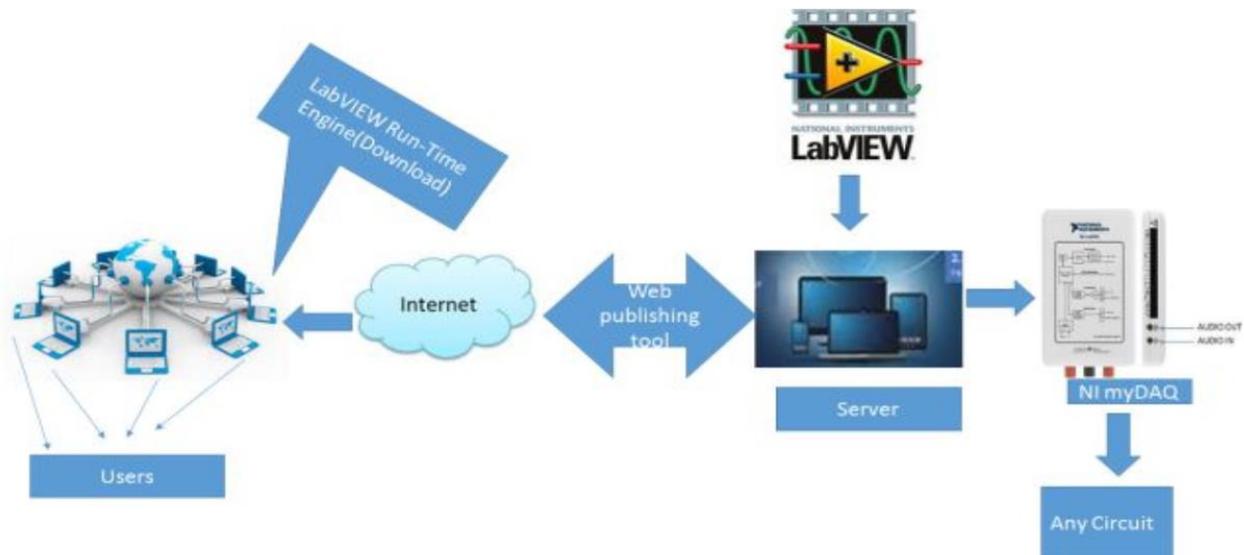


Fig. 7: Diagrama de bloques de laboratorio virtual

En esta arquitectura, LabVIEW funciona como controlador principal de acceso a través de sus herramientas de publicación Web.

## ARQUITECTURA DEL LABORATORIO VIRTUAL DE ROBÓTICA DE UNED

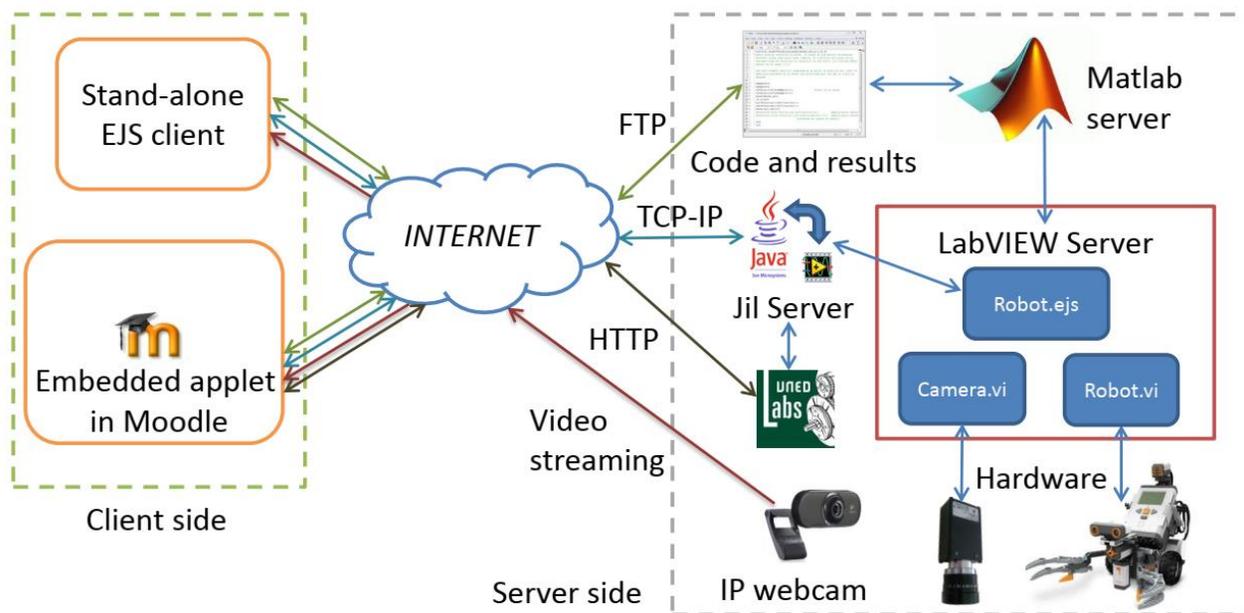


Fig. 8: Arquitectura del Laboratorio Virtual de Robótica de UNED

UNED utiliza esta arquitectura para sus laboratorios de robótica como parte de su programa a distancia de maestría en ingeniería y control automático. Se observa como el laboratorio virtual basado en LabVIEW está integrado a la plataforma LMS Moodle.

## 6. METODOLOGÍA DE INVESTIGACIÓN

En términos generales, el desarrollo de la investigación se desarrolló siguiendo secuencialmente las siguientes etapas:

### I. Diseño y simulación del funcionamiento de la planta (Entrenador de Control de Procesos de Temperatura) en Matlab.

En el diseño de todo sistema de control es indispensable crear un modelo matemático de la máquina y proceso a controlar (planta), para tal propósito, se hizo una visita al laboratorio B202 donde se encuentra el entrenador físico y se tomaron especificaciones técnicas de los dispositivos que lo conforman, así como mediciones de tuberías y reservorios para calcular el flujo volumétrico del fluido

[8]. Con la información recabada se obtuvo la ecuación diferencial del proceso y se analizó su comportamiento aplicando el método de Euler [9] para resolver la ecuación diferencial utilizando Matlab, como resultado se obtuvieron los comportamientos que se habían previsto, de manera que se pudo afirmar con certeza que el modelo obtenido es estable.

## **II. Creación y Comprobación de la Función de Transferencia en SIMULINK.**

Como siguiente paso, se procedió a crear la función de transferencia de la planta a partir de la ecuación diferencial y analizar su comportamiento mediante un controlador PID, para esta etapa se utilizó SIMULINK, la función de transferencia es la representación típica de un proceso de control, aunque dichos procesos pueden también ser representados de forma diferencial, ambos modelos fueron comprobados en esta etapa con resultados idénticos. El controlador PID de prueba utilizado es el proporcionado por SIMULINK [10], con él se puso a prueba el comportamiento de la planta con las técnicas P, PI y PID encontrando resultados satisfactorios.

## **III. Análisis del Comportamiento de la Planta en LabView 2020 (versión de prueba).**

Con los resultados obtenidos en el proceso de modelado y simulación en Matlab y SIMULINK, se procedió a crear y evaluar el comportamiento de la planta en el software LabView 2020. En la figura 9, se puede apreciar el instrumento virtual utilizado para hacer las pruebas y los valores de las variables P, I y D con los cuales se obtuvieron los resultados esperados ya que el sistema se estabiliza rápidamente en el set point definido. En la figura 10, se muestra el diagrama de bloques de la planta, cabe aclarar que se utilizó el modelo de la ecuación diferencial en lugar de la función de transferencia ya que en LabView debe pagarse para tener acceso a este tipo de funciones de control. Sin embargo, en SIMULINK ya se había comprobado con anterioridad ambos modelos (Ecuación diferencial y Función de transferencia) obteniendo los mismos resultados.

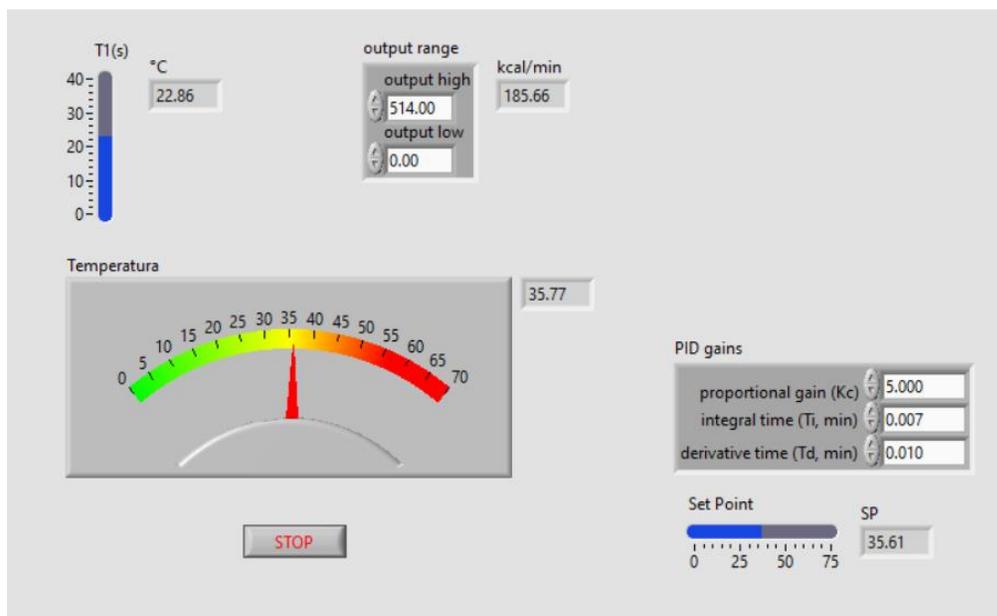


Fig. 9: Panel de Instrumentos utilizado para las pruebas de funcionamiento de la planta

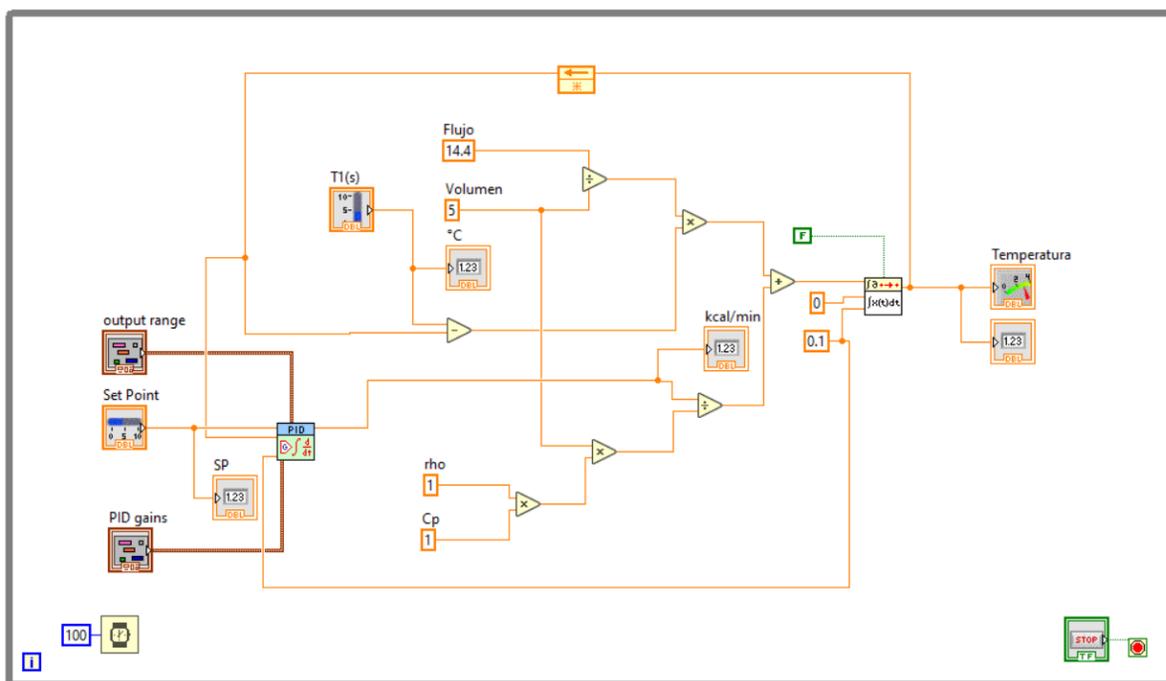


Fig. 10: Diagrama de bloques de la planta en donde se ha aplicado el modelo de la ecuación diferencial.

#### **IV. Diseño del Interface de usuario final del simulador.**

El principal objetivo en esta etapa es crear una interface similar a la presentada en el tablero de control del entrenador real. Durante esta etapa se realizaron diversos ajustes con el objetivo de crear, eliminar o modificar elementos del panel de instrumentos con la finalidad de hacer un simulador más intuitivo y sea más fácil de utilizar para los usuarios.

#### **V. Documentación.**

En esta etapa de la investigación se recopiló toda la información recolectada y elaborada durante el desarrollo de la investigación y con ella se elaboraron las guías de usuario, de práctica e informes finales de la investigación.

## 7. RESULTADOS

### 7.1. ECUACIÓN DIFERENCIAL DEL ENTRENADOR DE CONTROL DE TEMPERATURA

Como resultado del proceso de análisis de los fenómenos físicos que forman parte del proceso de control se dedujo la ecuación diferencial del proceso. (fig. 11)

$$\frac{dT}{dt} = \frac{F}{V}(T_1 - T) + \frac{Q}{\rho VC_p}$$

*Valores del Entrenador*

$$F = 14 \text{ l/min} \quad V = 5 \text{ l} \quad \rho = 1 \text{ kg/l} \quad C_p = 1 \text{ kcal/kg}^\circ$$

*Parámetros de prueba*

$$T(0) = 35^\circ\text{C}$$

$$T_1 = 25^\circ\text{C}$$

$$Q = 140 \text{ kcal/min}$$

Fig. 11: Ecuación diferencial y parámetros de prueba del entrenador de control de procesos de temperatura

### 7.2. CÓDIGO QUE IMPLEMENTA EL MÉTODO DE EULER PARA SOLUCIONAR ECUACIONES DIFERENCIALES

Fue creado en MATLAB para comprobar que el análisis matemático del funcionamiento del entrenador y la ecuación diferencial fueron correctos, la gráfica muestra la estabilidad de la ecuación diferencial (fig. 12).

A continuación se presenta el código MATLAB utilizado para solucionar las ecuaciones diferenciales con el método de EULER, se observa la utilización de los parámetros de prueba que se muestran en la figura 11.

```
% RANGO DE INTEGRACIÓN
```

```
xi = 0;
```

```
xf = 10;
```

```
% VALORES INICIALES DE "y" y "u"
```

```

T = 35;
T1 = 25;
Q = 140;

% PARÁMETROS
F = 14;
V = 5;
rho = 1;
Cp = 1;

% SECCIÓN DE INICIACIÓN
% -----
dx = 0.001;
XPRINT = 0.5; % INTERVALO DE IMPRESIÓN DE DATOS
X = xi:dx:xf;
n = length(X);
Y = T; % INICIALIZA EL VECTOR Y
% -----

for i = 1:n-1
% SECCIÓN DE ECUACIONES
% -----
%%
dT = F/V *(T1 - T) + Q/(rho*V*Cp);

% SECCIÓN DE INTEGRACIÓN
% -----
T = INT_EULER(T,dT);
Y(i+1) = T;
end

```

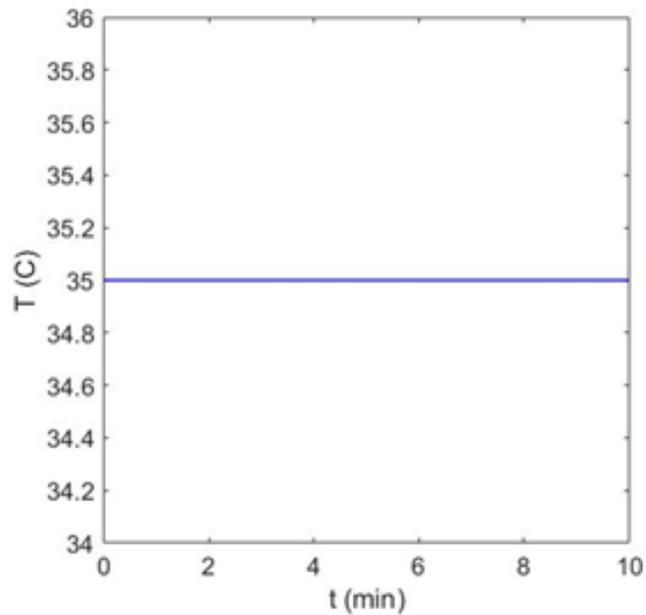


Fig. 12: La gráfica resultado de la ecuación diferencial muestra como el sistema se estabiliza a 35 °C con lo cual se comprueba que el sistema es estable.

### 7.3. MODELO DIFERENCIAL DE LA PLANTA (ENTRENADOR DE TEMPERATURA)

Este diagrama de bloques (fig. 13) representa la ecuación diferencial de todos los componentes que forman parte del entrenador de control de procesos de temperatura, fue creado con SIMULINK para comprobar la estabilidad del modelo matemático del entrenador ante perturbaciones.

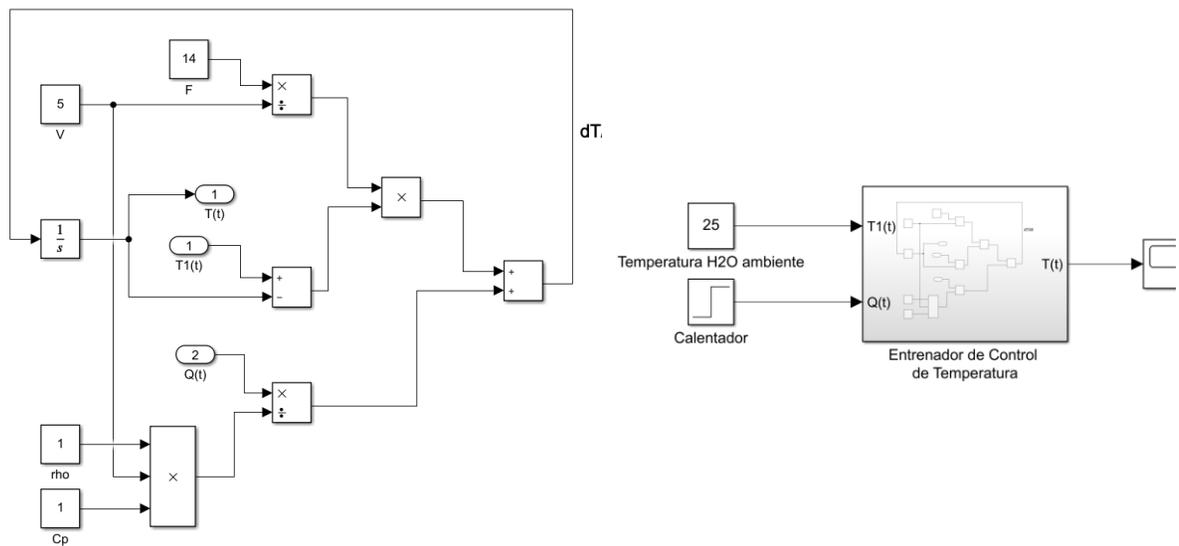


Fig. 13: Diagrama de bloques del modelo diferencial de la planta creado con SIMULINK

#### 7.4. MODELO DE LA FUNCIÓN DE TRANSFERENCIA.

Se obtuvo a partir de la ecuación diferencial para comprobar el funcionamiento del modelo diferencial anteriormente mostrado. En la figura 14, se muestra el modelo de la función de transferencia creado en SIMULINK.

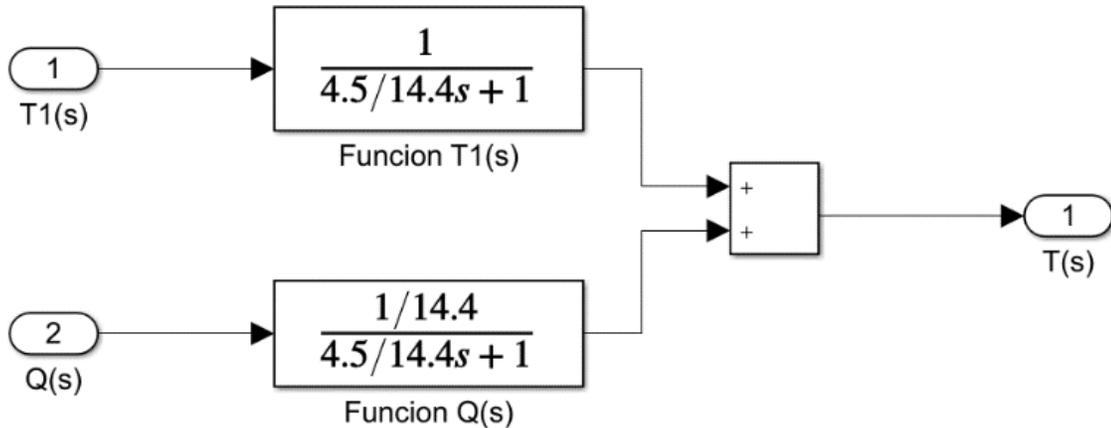


Fig. 14: Función de transferencia de la planta creada en SIMULINK

#### 7.5. MODELO DE LA PLANTA + CONTROL PID

Fue elaborado a partir del modelo de la función de transferencia por medio de SIMULINK, con el objetivo de comprobar los efectos de las variables del control PID sobre la estabilidad del proceso que se está controlando (fig. 15); se pueden apreciar que las gráficas muestran resultados satisfactorios. (fig. 16)

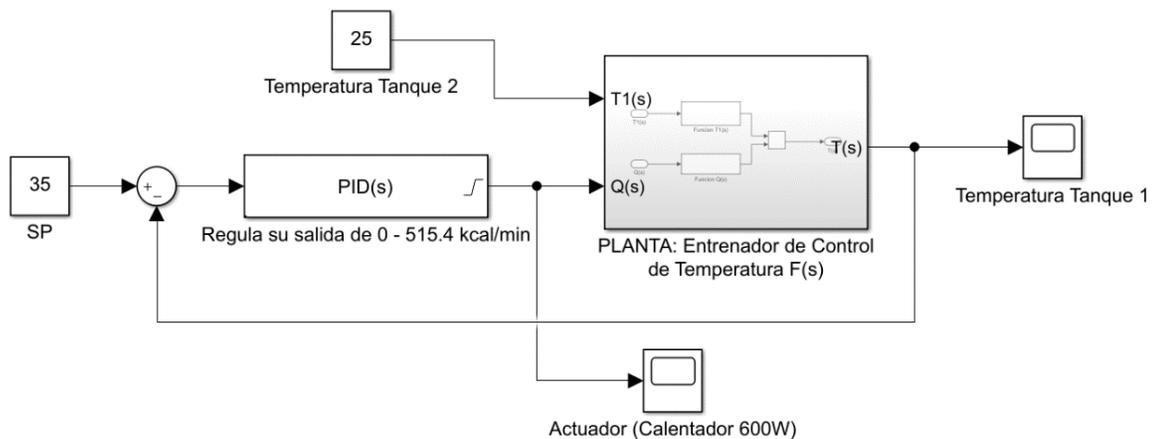


Fig. 15: Modelo de la planta + PID creada en SIMULINK

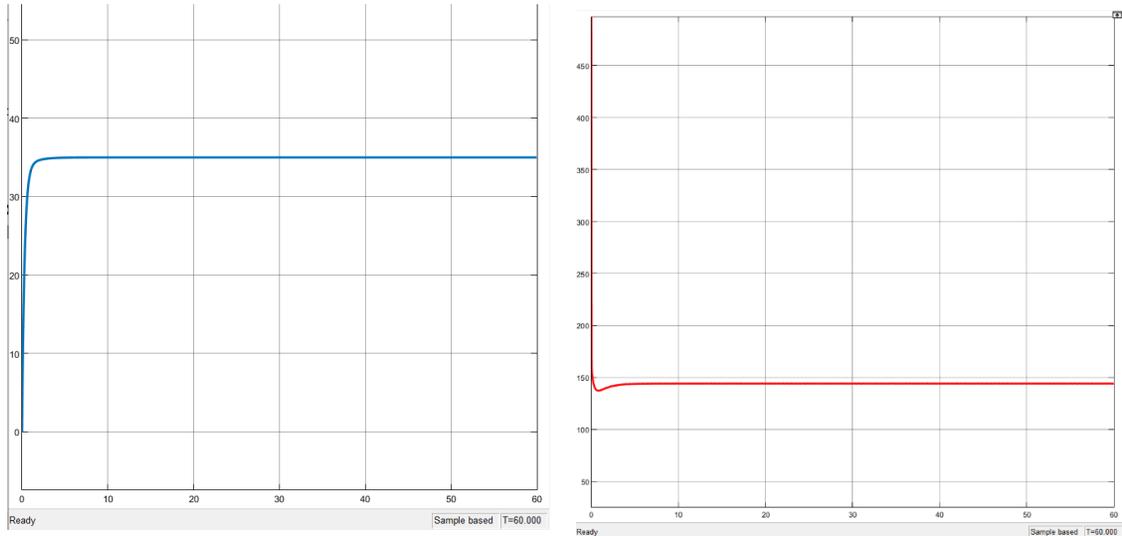


Fig. 16: En la gráfica de la izquierda se observa como el controlador estabiliza la temperatura de forma eficaz a 35°C, en la gráfica de la izquierda se puede apreciar como el controlador ajusta la potencia del calentador y el efecto de la acción derivativa.

## 7.6. CÓDIGO DEL CONTROL PID

El código del control PID Fue creado en MATLAB, se utilizó para evaluar la respuesta del controlador PID junto a la función de transferencia de la planta que representa el entrenador de control de temperatura. La gráfica resultante muestra la estabilidad del sistema. (fig. 17). A continuación se presenta el código MATLAB para comprobar funciones de transferencia junto a controles PID.

```
% Simulador de sistema de control feedback
clear
clc

% Función de transferencia de Planta
Gp = tf([1/14.3],[5/14.3 1])

% Función de transferencia de controlador PID
Kc = 5;
Tao_I = 0.15;
Tao_D = 0.0;

Gc = Kc + tf([Kc],[Tao_I 0]) + tf([Kc*Tao_D 0],[1])
```

% Función de transferencia a Lazo Cerrado

$$G_{CL} = G_p * G_c / (1 + G_p * G_c);$$

$$[y,t] = \text{step}(G_{CL});$$

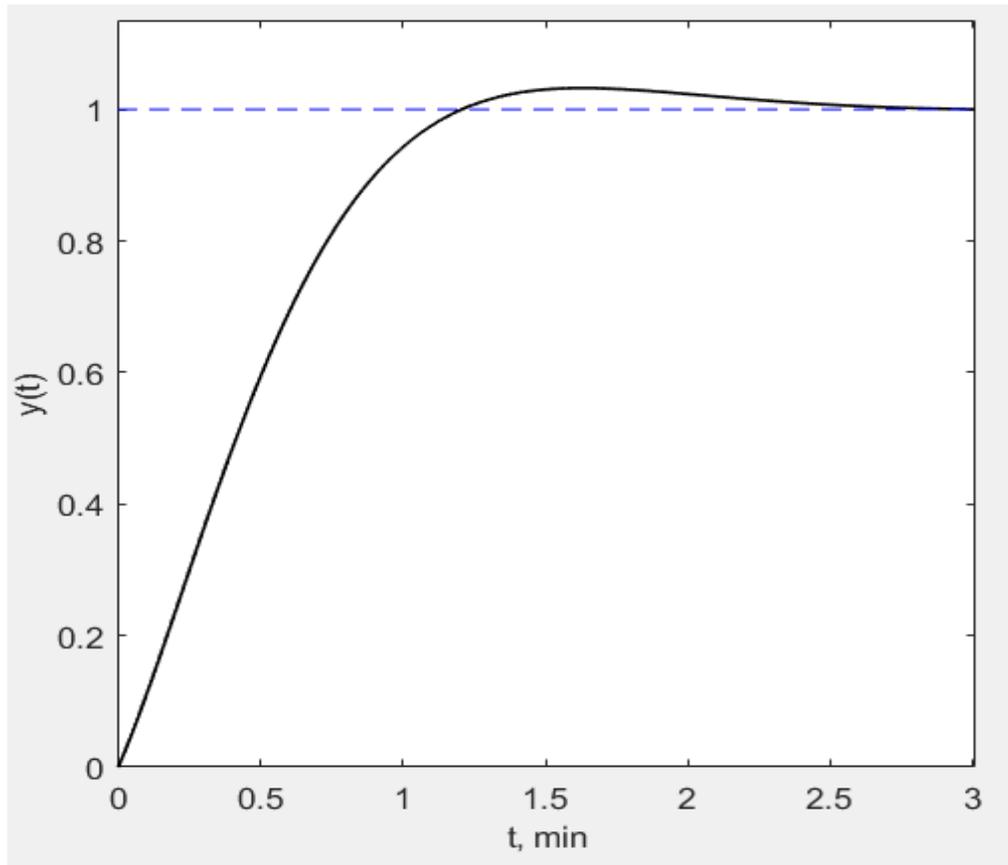


Fig. 17: Gráfica de respuesta de la planta bajo el control PID

## 7.7. DIAGRAMA DE BLOQUES DEL SIMULADOR

Ha sido creado en lenguaje G de LabVIEW versión 2020, representa el código del programa simulador, es la representación del modelo matemático del entrenador. (fig. 18)

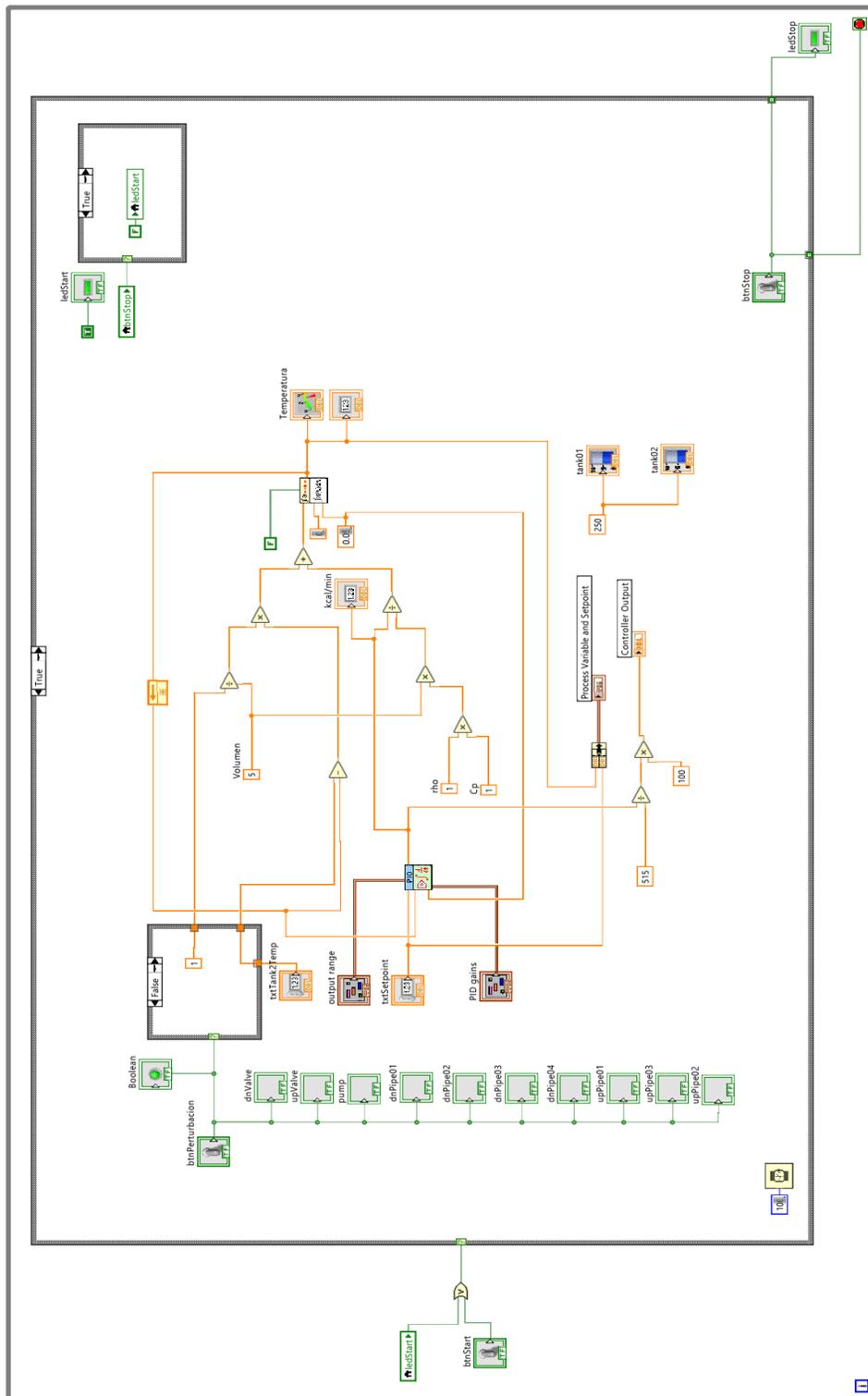


Fig. 18: Diagrama de bloques LabVIEW 2020 del simulador de control de procesos de temperatura

## 7.8. PANEL DE INSTRUMENTOS (VI) DEL SIMULADOR

Ha sido creado con la suite LabVIEW 2020 (versión de prueba), siempre que fue posible se tuvo especial cuidado en dejar una disposición de componentes similar al panel de instrumentos del entrenador real. (fig. 19)

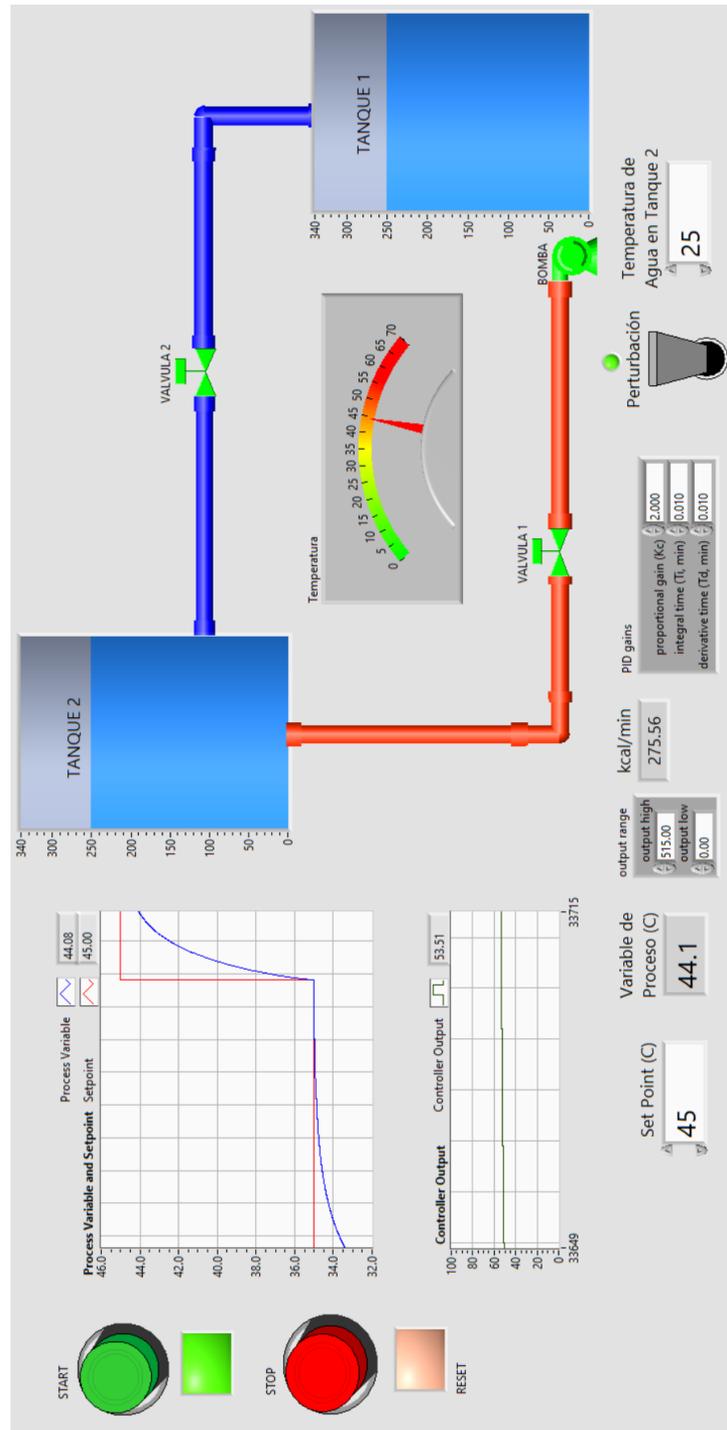


Fig. 19: Interfaz de usuario o panel de instrumentos virtuales (VI) del simulador

## 8. CONCLUSIONES

1. Es posible modelar matemáticamente un proceso industrial para predecir su comportamiento y sobre la base de este crear un controlador proporcional, integral y derivativo que mantenga controlado a pesar de las variaciones a las que estaría sujeto en la realidad.
2. Se pueden crear simuladores por software que permitan a docentes, estudiantes e investigadores evaluar el comportamiento de un sistema controlado para efectuar ajustes y corregir errores sin necesidad de arriesgarse de causar desperfectos en un equipo de alto costo y reducir tiempos de desarrollo de soluciones.
3. Se pueden utilizar los simuladores de control de procesos industriales de temperatura y utilizarlos en el aprendizaje de estudiantes ya que se pueden crear guías de práctica de niveles de técnico y de ingeniería.
4. Mediante el uso de simuladores como el diseñado durante esta investigación, se mejora el aprendizaje de los estudiantes en el área de control de procesos y control automático ya que pueden realizar muchos cálculos y experimentar con la seguridad que los resultados obtenidos serán similares a los que se producirían con los entrenadores reales.

## 9. RECOMENDACIONES

1. Como todo programa, el simulador diseñado está sujeto a mejoras con el objetivo de corregir errores y mejorar la experiencia del usuario, se recomienda utilizar el simulador en los módulos relacionados con el control de proceso y documentar las observaciones de docentes y alumnos con el fin de mejorar el programa.
2. Se cuentan con dos entrenadores de procesos físicos que pueden ser también simulados mediante un programa, se recomienda que se realice un esfuerzo para que se creen simuladores de estos equipos partiendo de la presente investigación.
3. El simulador fue diseñado con el programa LabVIEW 2020 versión demostrativa, se recomienda comprar una licencia de investigación para poder utilizar los módulos y plugins que la Suite proporciona y que facilitan mucho la creación y distribución del software.
4. Si bien el simulador facilita mucho la realización de pruebas y ajustes por parte de estudiantes y docentes, siempre será necesario acceder al entrenador real para comprobar los parámetros de control encontrados, en este sentido se debe hacer un esfuerzo para hacer llegar estos entrenadores

a los estudiantes mediante el reacondicionamiento y diseño de los entrenadores para que estos puedan utilizarse de forma remota.

## 10. GLOSARIO

**Realimentación:** propiedad por la cual se compara la salida con la entrada al sistema, de modo que se establezca una función entre ambas. También se le denomina “feedback”.

**VHDL:** es un lenguaje de programación utilizado para describir circuitos digitales y para la automatización del diseño electrónico. Es ampliamente utilizado en los FPGA.

**PWM:** Modulación por Ancho de Pulso, es una técnica que se utiliza para modificar el ciclo de trabajo de una señal periódica regulando la energía efectiva que se aplica a una carga.

**Servoválvula:** es un accionador de tipo neumático o hidráulico que conectan dos o más vías por las que circula un fluido. A diferencia de las válvulas tradicionales, estas tienen la capacidad de controlar presión o el caudal de forma proporcional.

**Planta:** una planta puede ser una parte de un equipo, tal vez un conjunto de los elementos de una máquina que funcionan juntos, y cuyo objetivo es efectuar una operación particular.

**VI:** los programas de LabVIEW son llamados instrumentos virtuales o VIs ya que su apariencia y operación imitan a los instrumentos físicos, como osciloscopios y multímetros.

**MATLAB:** es un programa que combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente. [11]

**SIMULINK:** es una herramienta de simulación de modelos o sistemas, con cierto grado de abstracción de los fenómenos físicos involucrados en los mismos. Se hace hincapié en el análisis de sucesos, a través de la concepción de sistemas (cajas negras que realizan alguna operación).

## 11. REFERENCIAS BIBLIOGRÁFICAS

- [1] D. Akakpo, "A Control Perspective to Adaptive Time Stepping in Reservoir Simulation", Texas University, Austin, 2016.
- [2] J. Solís, "Sistema de Simulación de Control Proporcional Integral Derivativo de Presión, Flujo, Nivel Y Temperatura De Agua".
- [3] A. Arbildo-López, "El Control de Procesos Industriales y su Influencia en el Mantenimiento", *Ing. Ind.*, Núm. 029, Pp. 35–49, 2011.
- [4] K. Ogata, *Ingeniería de Control Moderna*, 5a Ed., Vol. 1. Madrid: Pearson Educación, 2010.
- [5] A. R. Cusido, *Control de Procesos*. México, D.F.: Alfaomega Grupo Editor, 2003.
- [6] W. Bolton, *Mecatrónica, Sistemas de Control Electrónico en la Ingeniería Mecánica y Eléctrica*, Edición: 4 Ed. México: Alfaomega, 2009.
- [7] "1. Entorno Ni Labview - National Instruments". <https://www.ni.com/academic/students/learnlabview/esa/environment.htm#> (Consultado Ene. 18, 2021).
- [8] N. S. Nise, *Control System Engineering*, 6a Ed. Pomona, California: Wiley, 2011.
- [9] "Euler Method". <https://la.mathworks.com/matlabcentral/fileexchange/72522-euler-method> (Consultado Ene. 25, 2021).
- [10] "Systems and Models - Matlab & Simulink - Mathworks América Latina". <https://la.mathworks.com/help/simulink/slref/simulink-concepts-systems-and-models.html> (Consultado Ene. 25, 2021).
- [11] "Descripción del Producto Matlab - Matlab & Simulink - Mathworks América Latina". [https://la.mathworks.com/help/matlab/learn\\_matlab/product-description.html](https://la.mathworks.com/help/matlab/learn_matlab/product-description.html) (Consultado Ene. 25, 2021).

## 12. ANEXOS

### 12.1. ANEXO 1: GUÍA DE INSTALACIÓN Y USO



# Escuela de Ingeniería Eléctrica y Electrónica

## Guía de Instalación y Uso de

### Simulador de Control de Temperatura PID

#### I. INSTALACIÓN

**Requisitos mínimos – recomendados para instalación y utilización.**

Los siguientes requisitos necesarios que la computadora debe poseer para poder instalar y utilizar el Simulador de Control de Temperatura PID:

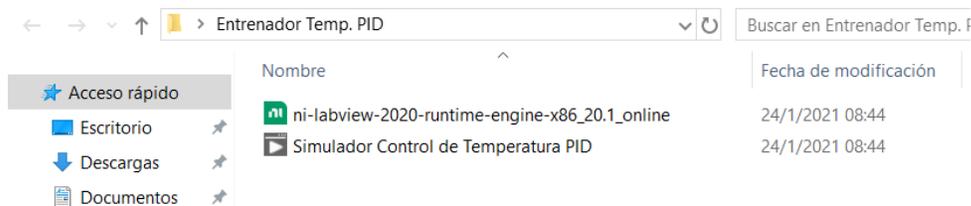
- Procesador Pentium III/Celeron 866 MHz o equivalente mínimo, Pentium 4/M o equivalente recomendado.
- RAM 1 GB recomendado.
- Resolución de Pantalla 1024 x 768 píxeles mínimo, 1920 x 1080 recomendado.
- Sistema Operativo Windows 10/8.1/8/7 SP1 recomendado.
- Espacio en Disco 620 MB.
- Acceso a Internet (se requiere para descargar paquetes del motor de ejecución de la aplicación).
- Programas:
  - ✓ ni-labview-2020-runtime-engine-x86\_20.1\_online
  - ✓ Simulador Control de Temperatura PID.exe

#### **Instalación de LabView Runtime Engine**

El programa Runtime Engine de LabVIEW es necesario para poder ejecutar cualquier programa creado con NI LabVIEW sin la necesidad de instalar todo el entorno de desarrollo. Puede omitir este procedimiento si su computadora ya cuenta con el programa Runtime.

### Procedimiento:

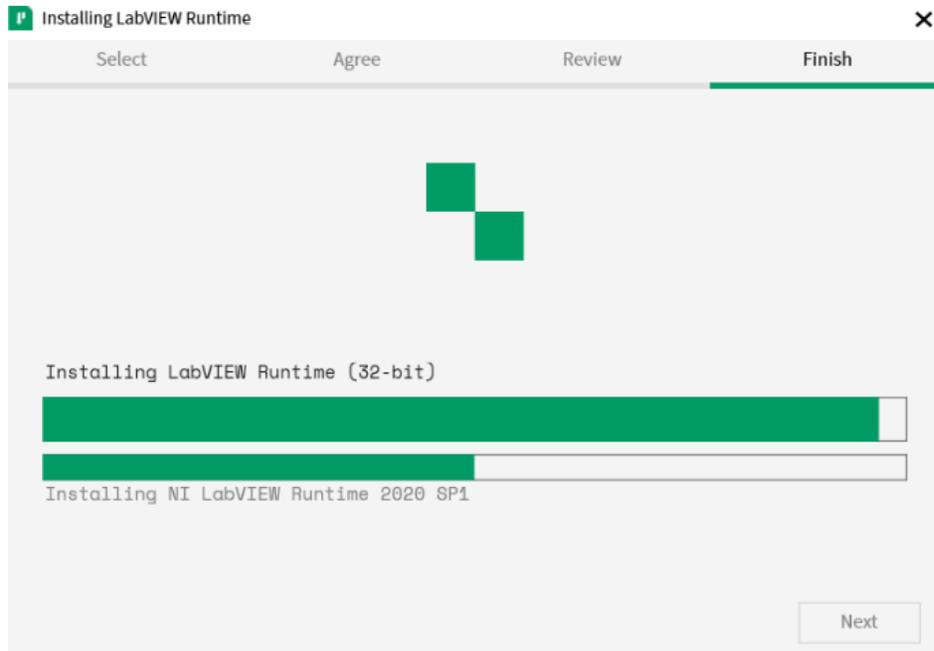
1. Descomprima la carpeta **Entrenador Temp. PID.zip** haciendo doble clic en ella. Se abrirá una nueva ventana.
2. En la ventana de clic en el botón **Extraer** todo que se encuentra en la parte superior, utilice el botón **Examinar...** y proporcione la ruta de su preferencia en donde se descomprimirán los archivos. Posteriormente de clic en el botón **Extraer** para descomprimir los archivos. Al finalizar el proceso de extracción se abrirá automáticamente una ventana mostrando los archivos descomprimidos.
3. Verifique que los archivos extraídos sean los que se muestran en la siguiente imagen:



4. Asegúrese de contar con una conexión a Internet activa y de doble clic en el archivo.

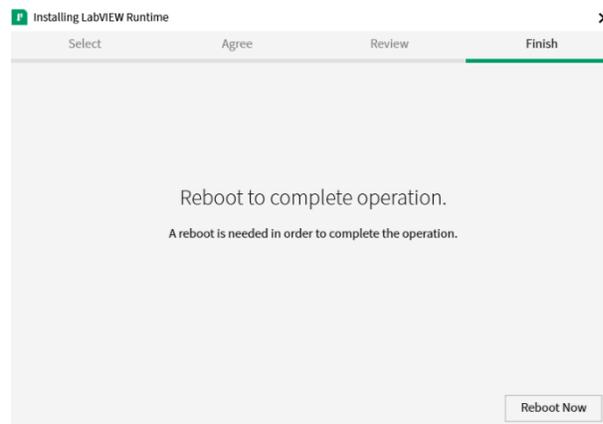
 ni-labview-2020-runtime-engine-x86\_20.1\_online

5. Se iniciará el programa de instalación del runtime y le aparecerá la ventana de confirmación de permisos de cambios en el dispositivo, de clic en **Sí**.
6. LabVIEW utiliza un programa llamado **NI Package Manager** para instalar todas sus aplicaciones y plugins por lo que observará que aparece la ventana de instalación de dicho programa a lo cual debe seleccionar **I accept the above license agreement**. Y clic en el botón Next en la ventana actual y nuevamente en la siguiente ventana. Espere a que finalice el proceso de instalación.
7. Cuando se haya finalizado la instalación de NI Package Manager, le aparecerá una nueva ventana, esta vez se le muestra el contrato de licencia de LabVIEW Runtime, seleccione **I accept the above license agreement**. Y clic en el botón Next en la ventana actual y nuevamente en la siguiente ventana. Espere a que finalice el proceso de instalación.



El tiempo de finalización dependerá del ancho de banda de su conexión a Internet y las prestaciones de su computadora.

8. Cuando el proceso de descarga e instalación haya finalizado, se solicitará reiniciar para completar el proceso, de clic en el botón Reboot Now.



9. Cuando su computadora haya reiniciado se habrá finalizado la instalación de LabVIEW Runtime Engine.

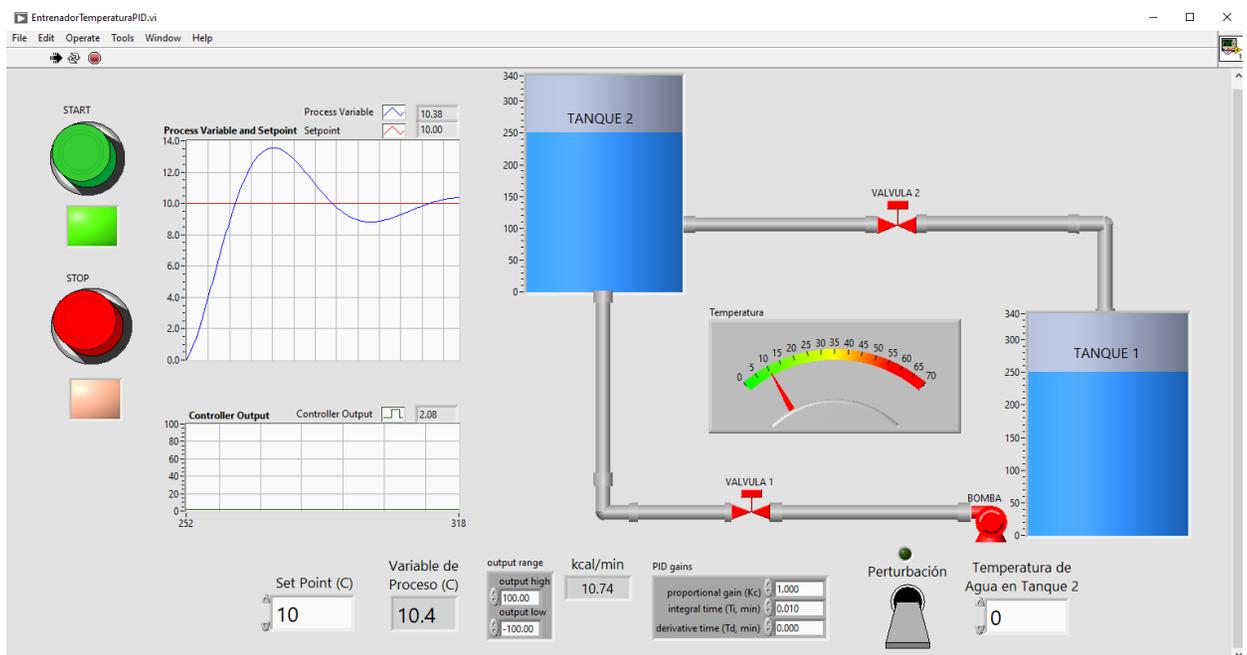
## Ejecución y uso del Simulador de Control de Temperatura PID

Ahora que se ha instalado el runtime de LabVIEW ya se puede ejecutar el simulador, este no requiere de ninguna instalación ni configuración adicional.

1. Abra la carpeta donde fueron extraídos los archivos en el paso 2 de la etapa anterior, ejecute el programa simulador  **Simulador Control de Temperatura PID** haciendo doble clic en él, se abrirá el panel de instrumentos virtual del simulador de control de temperatura PID.

**NOTA:** si la resolución de video de su computadora es baja, deberá desplazarse con las barras que aparecen abajo y a la izquierda de la ventana de la aplicación, se sugiere utilizar la resolución de video recomendada para tener una vista completa del simulador.

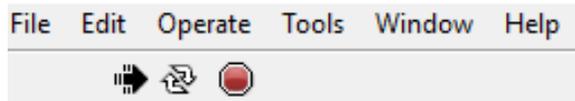
2. Para comprobar si puede ejecutarse una simulación, escriba el valor 10 en la casilla Set Point (C) y de clic en el botón verde START, observe que la variable de proceso intenta acercarse al valor Set Point, esta solo es una prueba de simulación por lo que no se preocupe porque la variable de proceso nunca se estabilice en 10 °C, observe como las gráficas y medidores también se actualizan con el tiempo:



Con esta verificación se ha comprobado que el simulador funciona y puede empezar a realizar prácticas de control PID.

## II. ELEMENTOS DEL SIMULADOR

a. Menú del programa y barra de acciones.



La barra de acciones posee tres íconos:

- **Run / Running** : si es negra indica que el programa está en ejecución (no necesariamente haciendo una simulación), si es blanca indica que el programa está detenido. Por defecto está en Running.
- **Run continuously / Running continuously** : si es negra indica que el programa está en ejecución continua, si es blanca indica que no está ejecutando continuamente. Por defecto es blanca.
- **Abort Execution** : detiene la ejecución del programa (no cierra la ventana), solo se activa cuando el programa está en modo Running, Running continuously o ambos están activos.
- En menú **Operate**, se provee de una opción para detener (stop) o ejecutar (run) el programa.

b. Controles de Simulación.

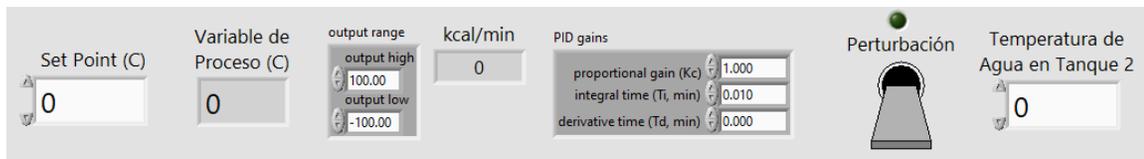


- El botón **START** se utiliza iniciar una simulación, esta inicia al momento de presionarlo. El Led verde indica si la simulación está activa o no.
- El botón **STOP** se utiliza para detener una simulación que está en progreso. La simulación se detiene al momento de presionar el botón. El led rojo se mantiene encendido mientras la simulación se encuentre detenida.

Si el programa se encuentra en modo Running con simulación activa y se presiona **STOP** el programa dejará de ejecutarse y el led rojo se encenderá. Por otra parte, si el

programa está en modo Running continuamente con simulación activa y se presiona STOP el programa solo detendrá la simulación y el led rojo se encenderá.

c. Barra de Configuración del Variables y Control PID.

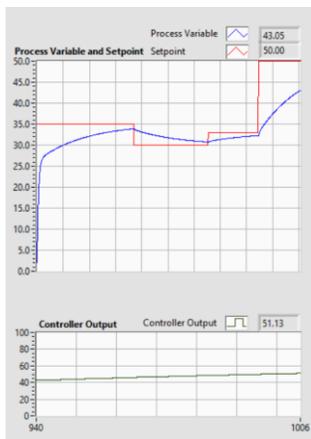


- **Set Point (C):** es una caja de texto en la que se define el valor del set point en grados centígrados (la temperatura a alcanzar y mantener).
- **Variable de Proceso (C):** es la temperatura del fluido ( $H_2O$ ), representa la temperatura real en grados centígrados del agua que se encuentra en el tanque 1.
- **Temperatura de Agua en Tanque 2:** es una caja de texto en donde se define la temperatura en grados centígrados del agua que se encuentra en el tanque 2 y que solo se toma en cuenta cuando se da un proceso de flujo entre los dos tanques del sistema. Su temperatura siempre será mas fría que la del tanque 2 y se considera una perturbación para el sistema que se está controlando.
- **Perturbación:** es un interruptor que al activarse provoca que la bomba haga circular el agua desde el tanque 1 al tanque 2 produciendo una alteración en la temperatura del agua, con esto se puede poner a prueba el funcionamiento del controlador ante perturbaciones.
- **output range:** son dos entradas en donde el usuario del simulador determina de acuerdo con sus cálculos, el rango de energía entregada al calefactor que se encuentra en el tanque 1 y que eleva la temperatura del agua. Posee un valor mínimo (output low) que siempre debe establecerse en 0.00 y un valor máximo (output high) que será establecido por el usuario. Sus unidades son kcal/min.

- **kcal/min**: representa la energía en kilo calorías por minuto que se está entregado en tiempo real al calefactor que se encuentra en el tanque 1, la energía que se está entregando es ajustada automáticamente por el controlador del simulador de acuerdo con los parámetros definidos por el usuario.
- **PID Gains**: permiten la definición y modificación de las variables Proporcional, Integral y Derivativa (PID) del controlador del simulador, sus valores deben ser especificados por el usuario a partir de sus cálculos.

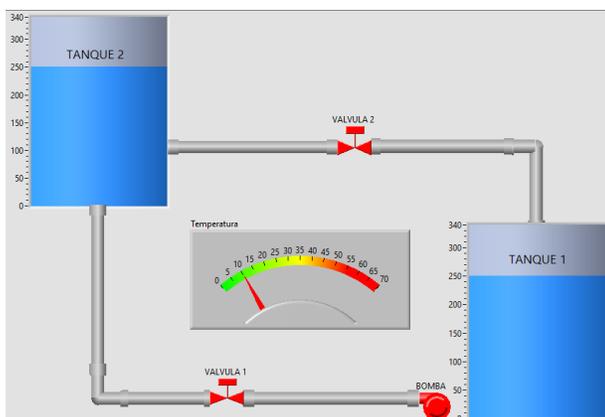
Todos los controles que son entradas pueden modificarse en tiempo real cuando el simulador está ejecutándose o está detenido.

d. Graficas de proceso.

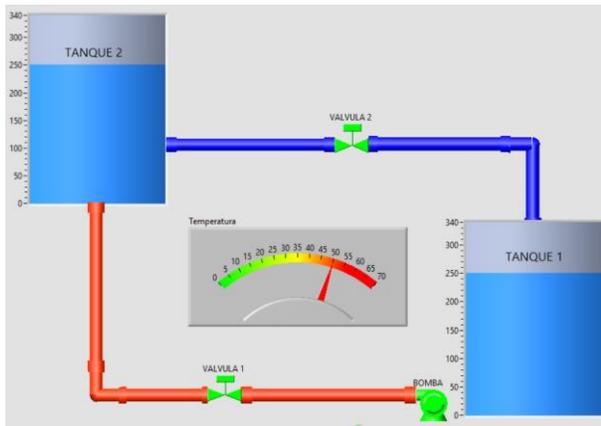


- **Process Variable and Setpoint**: muestra gráficamente la acción del controlador respecto a las variables Setpoint (rojo) y Variable de proceso (azul), las gráficas son útiles para determinar el efecto que las variables PID tienen en el controlador.
- **Controller Output**: muestra gráficamente como el controlador regula en terminos porcentuales la energía que se entrega al calefactor que se encuentran en el tanque 1 para realizar los ajustes en la temperatura del agua.

e. Esquema de estado del proceso.



Sin perturbación, sin circulación del fluido (agua): en este modo de operación los tanques 1 y 2 no intercambian agua por lo que la acción de control se limita a las condiciones físicas que afectan al tanque 1, como no hay circulación de agua las tuberías son grises, las válvulas 1, 2 y la bomba están desactivadas (en rojo).



Con perturbación, son circulación del fluido (agua): en este modo de operación los tanques 1 y 2 hacen un intercambio de agua a través de las tuberías. La tubería roja (inferior) indica que la temperatura del agua que circula por ella es superior a la temperatura del agua de la tubería de arriba (azul), las válvulas 1 y 2 están en verde indicando que están abiertas y la bomba también esta en verde indicando que hace circular el agua en el sentido de las agujas del reloj.

Este modo de operación se activa cuando el botón Perturbación está activo y se ha establecido la temperatura del agua del tanque 2.

En ambos modos de operación, el medidor de aguja indica la temperatura en °C del agua en el tanque 2.

## 12.2. ANEXO 2. TUTORIAL DE LABVIEW

### 1. Introducción

LabVIEW constituye un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos. Las ventajas que proporciona el empleo de LabVIEW se resumen en las siguientes:

- Se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.
- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

LabVIEW es un entorno de programación destinado al desarrollo de aplicaciones, similar a los sistemas de desarrollo comerciales que utilizan el *lenguaje C* o *BASIC*. Sin embargo, LabVIEW se diferencia de dichos programas en un importante aspecto: los citados lenguajes de programación se basan en líneas de texto para crear el código fuente del programa, mientras que LabVIEW emplea la programación gráfica o *lenguaje G* para crear programas basados en diagramas de bloques.

Para el empleo de LabVIEW no se requiere gran experiencia en programación, ya que se emplean iconos, términos e ideas familiares a científicos e ingenieros, y se apoya sobre símbolos gráficos en lugar de lenguaje escrito para construir las aplicaciones. Por ello resulta mucho más intuitivo que el resto de lenguajes de programación convencionales.

LabVIEW posee extensas librerías de funciones y subrutinas. Además de las funciones básicas de todo lenguaje de programación, LabVIEW incluye librerías específicas para la adquisición de datos, control de instrumentación VXI, GPIB y comunicación serie, análisis presentación y guardado de datos.

LabVIEW también proporciona potentes herramientas que facilitan la depuración de los programas.

## 2.- ¿CÓMO TRABAJA LABVIEW?

Los programas desarrollados mediante LabVIEW se denominan *Instrumentos Virtuales (VIs)*, porque su apariencia y funcionamiento imitan los de un instrumento real. Sin embargo, son análogos a las funciones creadas con los lenguajes de programación convencionales. Los *VIs* tienen una parte interactiva con el usuario y otra parte de código fuente, y aceptan parámetros procedentes de otros *VIs*.

Todos los *VIs* tienen un *panel frontal* y un *diagrama de bloques*. Las *paletas* contienen las opciones que se emplean para crear y modificar los *VIs*. A continuación, se procederá a realizar una somera descripción de estos conceptos.

### A) Panel Frontal

Se trata de la interfaz gráfica del *VI* con el usuario. Esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa. Un *panel frontal* está formado por una serie de botones, pulsadores, potenciómetros, gráficos, etc. Cada uno de ellos puede estar definido como un *control* (a) o un *indicador* (b). Los primeros sirven para introducir parámetros al *VI*, mientras que los indicadores se emplean para mostrar los resultados producidos, ya sean datos adquiridos o resultados de alguna operación.

### B) Diagrama de bloques

El *diagrama de bloques* 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*.

El *diagrama de bloques* 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*.

El *diagrama de bloques* se construye conectando los distintos objetos entre sí, como si de un circuito se tratara. Los cables unen terminales de entrada y salida con los objetos correspondientes, y por ellos fluyen los datos.

LabVIEW posee una extensa biblioteca de *funciones*, entre ellas, aritméticas, comparaciones, conversiones, funciones de entrada/salida, de análisis, etc.

Las *estructuras*, similares a las declaraciones causales y a los bucles en lenguajes convencionales, ejecutan el código que contienen de forma condicional o repetitiva (bucle *for*, *while*, *case*,...).

Los cables son las trayectorias que siguen los datos desde su origen hasta su destino, ya sea una función, una estructura, un terminal, etc. Cada cable tiene un color o un estilo diferente, lo que diferencia unos tipos de datos de otros.

### C) Paletas

Las *paletas* de LabVIEW proporcionan las herramientas que se requieren para crear y modificar tanto el *panel frontal* como el *diagrama de bloques*.

Se emplea tanto en el *panel frontal* como en el *diagrama de bloques*. Contiene las herramientas necesarias para editar y depurar los objetos tanto del *panel frontal* como del *diagrama de bloques*.

Las opciones que presenta esta paleta son las siguiente:



*Operating tool* – Cambia el valor de los controles.



*Positioning tool* – Desplaza, cambia de tamaño y selecciona los objetos.



*Labeling tool* – Edita texto y crea etiquetas.



*Wiring tool* – Une los objetos en el *diagrama de bloques*.



*Object Pop-up Menu tool* – Abre el menú desplegable de un objeto.



*Scroll tool* – Desplaza la pantalla sin necesidad de emplear las barras de desplazamiento.



*Breakpoint tool* – Fija puntos de interrupción de la ejecución del programa en *VIs*.



*Probe tool* – Crea puntos de prueba en los cables, en los que se puede visualizar el valor del dato que fluya por dicho cable en cada instante.



*Color Copy tool* – Copia el color para después establecerlo mediante la siguiente herramienta.



*Color tool* – Establece el color de fondo y el de los objetos

### Paleta de controles (Controls palette)

Se utiliza únicamente en el *panel frontal*. Contiene todos los *controles e indicadores* que se emplearán para crear la interfaz del *VI* con el usuario.



El menú *Controls* de la ventana correspondiente al panel frontal contiene las siguientes opciones:



*Numeric* – Para la introducción y visualización de cantidades numéricas.



*Boolean* – Para la entrada y visualización de valores booleanos.



*String & Table* – Para la entrada y visualización de texto.



*List & Ring* – Para visualizar y/o seleccionar una lista de opciones.



*Array & Cluster* – Para agrupar elementos.



*Graph* – Para representar gráficamente los datos.



*Path & RefNum* – Para gestión de archivos.



*ActiveX* – Para transferir datos y programas de unas aplicaciones a otras dentro de Windows.



*Select a Control* – Para seleccionar cualquier *control*.

Al seleccionar objetos desde el menú *Controls* estos aparecen sobre el *panel frontal*, pueden colocarse donde convenga, y además tienen su propio menú desplegable que permite la configuración de algunos parámetros específicos de cada tipo de *control*.

### **Paleta de funciones (functions palette)**

Se emplea en el diseño del *diagrama de bloques*. La *paleta de funciones* contiene todos los objetos que se emplean en la implementación del programa del *VI*, ya sean *funciones* aritméticas, de entrada/salida de señales, entrada/salida de datos a fichero, adquisición de señales, temporización de la ejecución del programa.

Para seleccionar una *función* o *estructura* concretas, se debe desplegar el menú

*Functions* y elegir entre las opciones que aparecen. A continuación se enumeran todas ellas, junto con una pequeña definición.



*Structures* – Muestra las *estructuras* de control del programa, junto con las variables locales y globales.



*Numeric* – Muestra *funciones* aritméticas y constantes numéricas.



*Boolean* – Muestra *funciones* y constantes lógicas.



*Comparison* – Muestra *funciones* que sirven para comparar números, valores.



*Time & Dialog* – Contiene *funciones* para trabajar con cuadros de diálogo, introducir contadores y retardos, etc.



*File I/O* – Muestra *funciones* para operar con ficheros.



*Communication* – Muestra diversas *funciones* que sirven para comunicar varios ordenadores entre sí, o para permitir la comunicación entre distintos programas.



*Instrument I/O* – Muestra un submenú de *Vis*, que facilita la comunicación con instrumentos periféricos que siguen la norma ANSI/IEEE 488.2-1987, y el control del puerto serie.



*Data Acquisition* – Contiene a su vez un submenú donde puede elegirse entre distintas librerías referentes a la adquisición de datos.



*Analysis* – Contiene un submenú en el que se puede elegir entre una amplia gama de *funciones* matemáticas de análisis.



*Tutorial* – Incluye un menú de *Vis* que se utilizan en el manual LabVIEW Tutorial.

### 3.- PROGRAMACIÓN EN LABVIEW

Con el entorno gráfico de programación de LabVIEW se comienza a programar a partir del panel frontal. En primer lugar se definirán y seleccionarán de la paleta de controles todos los *controles* (entradas que dará el usuario) e *indicadores* (salidas que presentará en pantalla el VI) que se emplearán para introducir los datos por parte del usuario y presentar en pantalla los resultados.

Una vez colocados en la ventana correspondiente al *panel frontal* todos los objetos necesarios, debe pasarse a la ventana *Diagram* (menú *Windows > Show Diagram*), que es donde se realiza la programación propiamente dicha (*diagrama de bloques*). Al abrir esta ventana, en ella se encuentran los terminales correspondientes a los objetos situados en el *panel frontal*, dispuestos automáticamente por LabVIEW.

Se deben ir situando las *funciones*, *estructuras*, etc. que se requieran para el desarrollo del programa, las cuales se unen a los terminales mediante cables.

Para facilitar la tarea de conexión de todos los terminales, en el menú "*Help*" puede elegirse la opción "*Show Help*", con lo que al colocar el cursor del ratón sobre un elemento aparece una ventana con información relativa a éste (parámetros de entrada y salida). Además, si se tiene seleccionado el cursor de cableado, al situar éste sobre un elemento se muestran los terminales de forma intermitente.

### 4.- EJECUCIÓN DE UN VI

Una vez se ha concluido la programación del VI se debe proceder a su ejecución. Para ello la ventana activa debe ser el *panel frontal* (si se está en la ventana del *diagrama de bloques*, se debe seleccionar la opción *Show Panel* del menú *Window*).

Una vez situados en el *panel frontal*, se pulsará el botón de *Run*, situado en la barra de herramientas.



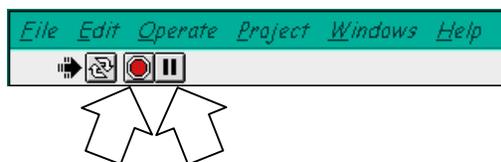
El programa comenzará a ejecutarse. Mientras dura la ejecución del mismo, la apariencia del botón de *Run* es la que se muestra a continuación:



De este modo el programa se ejecutará una sola vez. Si se desea una ejecución continua, se pulsará el botón situado a la derecha del de *Run (Continuous Run)*. Si durante el funcionamiento continuo del programa se vuelve a pulsar el citado botón, se finalizará la última ejecución del mismo, tras lo cual el programa se parará.

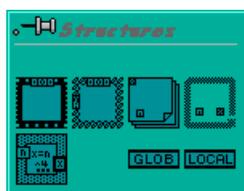


Para finalizar la ejecución de un programa se puede operar de dos formas. La primera, y la más aconsejable, es emplear un botón en el *panel frontal* del VI, cuya pulsación produzca la interrupción del bucle de ejecución de la aplicación. La segunda forma de detener la ejecución del VI es pulsando el botón de *pausa* o el de *stop*. La diferencia entre ambos es que si se pulsa *stop*, la ejecución del programa finaliza inmediatamente, mientras que si se pulsa *pausa*, se produce una detención en el funcionamiento del programa, retomándose su ejecución una vez se vuelve a pulsar el mismo botón.



## 5.- ESTRUCTURAS

En la *paleta de funciones* la primera opción es la de las *estructuras*. Éstas controlan el flujo del programa, bien sea mediante la secuenciación de acciones, ejecución de bucles, etc.



Las estructuras se comportan como cualquier otro nodo en el diagrama de bloques, ejecutando automáticamente lo que está programado en su interior una vez tiene disponibles los datos de entrada, y una vez ejecutadas las instrucciones requeridas, suministran los correspondientes valores a los cables unidos a sus salidas. Sin embargo, cada estructura ejecuta su *subdiagrama* de acuerdo con las reglas específicas que rigen su comportamiento, y que se especifican a continuación.

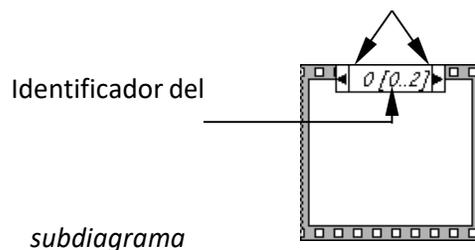
Un *subdiagrama* es una colección de nodos, cables y terminales situados en el interior del rectángulo que constituye la estructura. El *For Loop* y el *While Loop* únicamente tienen un subdiagrama. El *Case Structure* y

el *Sequence Structure*, sin embargo, pueden tener múltiples subdiagramas, superpuestos como si se tratara de cartas en una baraja, por lo que en el diagrama de bloques únicamente será posible visualizar al tiempo uno de ellos. Los subdiagramas se construyen del mismo modo que el resto del programa.

La estructura *Case* tiene al menos dos *subdiagramas* (*True* y *False*). Únicamente se ejecutará el contenido de uno de ellos, dependiendo del valor de lo que se conecte al *selector*.

De nuevo, este tipo de estructuras presenta varios *subdiagramas*, superpuestos como en una baraja de cartas, de modo que únicamente se puede visualizar una en pantalla.

También poseen un identificador del *subdiagrama* mostrado en su parte superior, con posibilidad de avanzar o retroceder a otros *subdiagramas* gracias a las flechas situadas a ambos lados del mismo.



Para pasar valores de una iteración a otra se emplean los llamados *shift registers*. Para crear uno, se pulsará el botón derecho del ratón mientras éste se halla situado sobre el borde del bucle, seleccionando la opción *Add Shift Register*. El *shift register* consta de dos terminales, situados en los bordes laterales del bloque. El terminal izquierdo almacena el valor obtenido en la iteración anterior. El terminal derecho guardará el dato correspondiente a la iteración en ejecución. dicho dato aparecerá , por tanto, en el terminal.

Se puede configurar un *shift register* para memorizar valores de varias iteraciones previas. Para ello, con el ratón situado sobre el terminal izquierdo del *shift register*, se pulsará el botón derecho, seleccionando a continuación la opción *Add Element*.

El programa comprueba el valor de lo que se halle conectado al terminal condicional al finalizar el bucle. Por lo tanto, el bucle siempre se ejecuta al menos una vez.

Con esta estructura también se pueden emplear los *shift registers* para tener disponibles los datos obtenidos en iteraciones anteriores (es decir, para memorizar valores obtenidos). su empleo es análogo al de los bucles *for*, por lo que omitiré su explicación. Para definir una fórmula mediante esta estructura, se actuará del siguiente modo:

- En primer lugar, se deben definir las variables de entrada y las de salida. Para ello, se pulsa con el botón derecho del ratón sobre el borde de la *formula node*. A continuación se seleccionará *Add Input*

o *Add Output*, según se trate de una entrada o una salida, respectivamente. Aparecerá un rectángulo, en el que se debe escribir el nombre de la variable (se distingue entre mayúsculas y minúsculas). Todas las variables que se empleen deben estar declaradas como entradas o salidas. Las que se empleen como variables intermedias se declararán como salidas, aunque posteriormente no se unan a ningún bloque posterior.

- Una vez definidas las variables a emplear, se escribirán la o las fórmulas en el interior del recuadro (para ello se emplea la *labeling tool*). Cada fórmula debe finalizar con un “;”.
- Los operadores y funciones que se pueden emplear se explican en la ayuda de LabVIEW.





## SEDE CENTRAL Y CENTROS REGIONALES EL SALVADOR



La Escuela Especializada en Ingeniería ITCA-FEPADE, fundada en 1969, es una institución estatal con administración privada, conformada actualmente por 5 campus: Sede Central Santa Tecla y cuatro centros regionales ubicados en Santa Ana, San Miguel, Zacatecoluca y La Unión.

### **1. SEDE CENTRAL SANTA TECLA**

Km. 11.5 carretera a Santa Tecla, La libertad.  
Tel.: (503) 2132-7400

### **2. CENTRO REGIONAL SANTA ANA**

Final 10a. Av. Sur, Finca Procavia.  
Tel.: (503) 2440-4348

### **3. CENTRO REGIONAL ZACATECOLUCA**

Km. 64.5, desvío Hacienda El Nilo sobre autopista a Zacatecoluca.  
Tel.: (503) 2334-0763 y 2334-0768

### **4. CENTRO REGIONAL SAN MIGUEL**

Km. 140 carretera a Santa Rosa de Lima.  
Tel.: (503) 2669-2298

### **5. CENTRO REGIONAL LA UNIÓN**

Calle Sta. María, Col. Belén, atrás del Instituto Nacional de La Unión  
Tel.: (503) 2668-4700