ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA E INSTRUMENTACIÓN

TEMA: SISTEMA DE MONITOREO Y CONTROL DE LA TEMPERATURA DE FLUJO DE AIRE MEDIANTE HARDWARE Y SOFTWARE LIBRE PARA SU USO DIDÁCTICO EN EL APRENDIZAJE DE CONTROL AUTOMÁTICO

AUTORES: GALO GEOVANNY CHACÓN GALARZA

VÍCTOR ALFONSO TAPIA TAPIA

DIRECTOR: ING. MARCO PILATÁSIG

LATACUNGA

2017

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Certifico que el trabajo de titulación, **"SISTEMA DE MONITOREO Y CONTROL DE LA TEMPERATURA DE FLUJO DE AIRE MEDIANTE HARDWARE Y SOFTWARE LIBRE PARA SU USO DIDÁCTICO EN EL APRENDIZAJE DE CONTROL AUTOMÁTICO"** realizado por los señores **GALO GEOVANNY CHACÓN GALARZA, VÍCTOR ALFONSO TAPIA TAPIA** , ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a los señores **GALO GEOVANNY CHACÓN GALARZA, VÍCTOR ALFONSO TAPIA TAPIA** para que lo sustente públicamente.

Latacunga, 28 de julio del 2017

Ing. Marco Pilatásig
DIRECTOR

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

AUTORÍA DE RESPONSABILIDAD

Nosotros, **GALO GEOVANNY CHACÓN GALARZA, VÍCTOR ALFONSO TAPIA TAPIA**, con cédulas de ciudadanía N°0503106692, 0503285421 respectivamente, declaramos que este trabajo de titulación **"SISTEMA DE MONITOREO Y CONTROL DE LA TEMPERATURA DE FLUJO DE AIRE MEDIANTE HARDWARE Y SOFTWARE LIBRE PARA SU USO DIDÁCTICO EN EL APRENDIZAJE DE CONTROL AUTOMÁTICO"** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaramos que este trabajo es de nuestra autoría, en virtud de ello nos declaramos responsables del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, 28 de julio del 2017

Galo Geovanny Chacón Galarza
C.C.: 0503106692

Víctor Alfonso Tapia Tapia
C.C.: 0503285421

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

AUTORIZACIÓN

Nosotros, **GALO GEOVANNY CHACÓN GALARZA, VÍCTOR ALFONSO TAPIA TAPIA**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **"SISTEMA DE MONITOREO Y CONTROL DE LA TEMPERATURA DEL FLUJO DE AIRE MEDIANTE HARDWARE Y SOFTWARE LIBRE PARA SU USO DIDÁCTICO EN EL APRENDIZAJE DE CONTROL AUTOMÁTICO"** cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.

Latacunga, 28 de julio del 2017

Galo Geovanny Chacón Galarza
C.C.: 0503106692

Víctor Alfonso Tapia Tapia
C.C.: 0503285421

DEDICATORIA

Dedico este trabajo a mis padres Galo y Enma pilar fundamental en mi vida y son quienes me han inculcado valores de responsabilidad, respeto y honestidad además me han apoyado para seguir adelante y con ello llegar a cumplir una meta de ser profesional, a mis hermanos Oscar y Nancy que me han dado consejos para superar los problemas que se presentan en la vida diaria y en la vida estudiantil y a una angelita que tengo en el cielo mi hermana Mónica que cuida de mí y de toda mi familia.

Galo

Desde hoy en adelante todo lo que me proponga anhelos, deseos y sueños que quiero realizar así como este título que he conseguido se lo dedico a Dios, porque él sabe dónde quiero ir y estoy seguro que me pondrá en el camino que busco, una búsqueda de una realidad profundamente arraigada en la que pueda mirar a la vida con más objetividad y tener una visión crítica y analítica porque Dios me dio inteligencia para dudar de todo y así buscar medrar en un mundo tergiversado y así encontrar la felicidad en este viaje que Dios ya que si puedes soñarlo puedes cumplirlo.

Víctor

AGRADECIMIENTO

Agradezco a la Universidad de las Fuerzas Armadas Espe Extensión Latacunga por permitirme formarme profesionalmente en esta prestigiosa institución de educación superior, asi como también a los docentes que fueron parte fundamental en mi vida estudiantil brindándome sus conocimientos y experiencias contribuyendo para mi formación integral. También agradezco a mis compañeros de clase ya que con el compañerismo, amistad y apoyo que existió durante todos los niveles han aportado de una u otra manera para culminar y seguir adelante en la consecución de la Ingeniería.

Galo

Todas las palabras que utilice van a resultar incipientes para manifestar todas las formas de agradecimiento que tengo hacia mis padres, mi padre; Víctor R. Tapia y mi madre; Sonia P. Tapia. Espero que haya estado a la altura de las expectativas de ellos ya que hice el mejor de los esfuerzos y que a pesar de las adversidades y mi ganas de seguir algunas veces me abandonaron pero mi mente siguió en pie en especial por la satisfacción de mi madre al conseguir este objetivo en el que la verdad ni yo mismo creía que lo podía conseguir, ya que quise devolver todo lo que me han dado con esto que he logrado; ya que estoy seguro que estoy están más orgullosos que yo mismo. Sin duda todo esto se lo debo al más grande ya que en el peor momento más difícil me envió una luz que me acompañará toda mi vida, gracias Dios por tanto porque esto que he logrado no fue por mí sino por ti porque LA GLORIA ES DE CRISTO.

Víctor

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

El trabajo de titulación elaborado consiste en la implementación de un sistema de monitoreo y control de la temperatura de flujo de aire, compuesto por un variador de frecuencia, un motor trifásico, un ventilador centrífugo de aspas radiales, un ducto de aire, una resistencia calefactora, un relé de estado sólido y una HMI, que permite desarrollar prácticas a los estudiantes para obtener el conocimiento previo de los procesos industriales, además es una herramienta didáctica que permite mejorar el proceso de enseñanza aprendizaje en el área del control automático ya que se desarrolló un controlador básico como es el PID y un avanzado como el FUZZY LOGIC. Se usó la tarjeta de desarrollo Beaglebone Black, en la cual se implementaron los algoritmos de control, para la programación se empleó el lenguaje de programación Python en software libre en la distribución Linux Debian 7 para realizar los controladores que serán aplicados en el sistema implementado. Una vez desarrollados los controladores se verificó el funcionamiento de los mismos, siendo evaluados con controladores realizados en el software Labview para el análisis de resultados y con ello determinar las conclusiones y recomendaciones de este trabajo de investigación.

PALABRAS CLAVE:

- **MOTORES TRIFÁSICOS**
- **TARJETA BEAGLEBONE BLACK**
- **AIRE - CONTROL DE TEMPERATURA**

ABSTRACT

The work of degree consists in the implementation of an air flow temperature monitoring and control system, consisting of a frequency inverter, a three-phase motor, a centrifugal fan with radial blades, an air duct, a heating resistor, a Solid state relay and an HMI, which allows students to develop practices to obtain prior knowledge of industrial processes, also this is a didactic tool to improve teaching-learning process in the area of automatic control since a basic driver was developed such as the PID and an advanced such as the FUZZY LOGIC.

The Beaglebone Black development board was used in which the control algorithms were implemented. For programming, the free software Python programming language was used in Linux Debian 7 distribution to perform the drivers that will be applied to the implemented system. Once the controllers were developed, their operation was verified, being evaluated with controllers made in the Labview software for result analysis and hence to determine conclusions and recommendations of this research work.

KEYWORDS:

- **THREE-PHASE MOTORS**
- **BEAGLEBONE BACK CARD**
- **AIR - TEMPERATURE CONTROL**

CAPÍTULO I

PROBLEMA

1.1 Antecedentes

En la industria en cuanto se refiere al control de temperatura de flujo de aire se tiene en consideración una aplicación muy importante que son los intercambiadores de calor, como su nombre lo indica son dispositivos que permiten remover calor de un punto a otro, existen diferentes tipos en función del flujo y también depende de la base de su construcción como puede ser tubo y carcaza, y en base a las necesidades del sistema se tomará la decisión adecuada para el tipo de aplicación necesaria. (Jaramillo, 2007)

Continuando en el ámbito industrial relacionando a la temperatura del flujo de aire existen aplicaciones como la calefacción, refrigeración y aire acondicionado en sectores como centrales eléctricas, plantas químicas, plantas petroquímicas, petróleo y refinerías, procesamiento de gas natural y tratamiento de aguas residuales. (González, 2002)

El control de temperatura del flujo de aire es de vital importancia en el diseño de soluciones industriales; también se aplica en la industria petrolera, aquí se lo utiliza de manera frecuente en la refrigeración para controlar la presión de vapor de constituyentes volátiles existentes en estos procesos, otro ámbito es la industria química en donde también es favorable usar la refrigeración para el control de procesos de polimerización de la condensación para la elaboración de caucho y plásticos sintéticos. Para el control de estos sistemas se han implementado controles como el PID, control adaptativo, fuzzy logic entre otros a través de Microcontroladores, Controladores Lógicos Programables o tarjetas embebidas. (Villalba, 2011) (Handbook, 2012)

La implementación de algoritmos de control clásico y en los últimos años los controles avanzados, se han desarrollado en diferentes dispositivos o equipos, que permiten realizar los mismos controles, estos son PLC, Tarjetas Arduinos, Microcontroladores, entre los más utilizados. A través de estos controles se ha efectuado diferentes aplicaciones dentro del ámbito industrial, agricultura, inteligencia artificial etc. (Alvares, 1995)

Es de gran necesidad para un conocimiento integral en el aprendizaje de las carreras técnicas la existencia de sistemas didácticos, en cuanto a control automático se refiere ya que son una herramienta de mucha ayuda, siendo utilizado este recurso desde hace mucho tiempo atrás en todos los niveles de la educación, por lo que se considera que es de trascendental apoyo para los estudiantes. (Almache Coyago Luis Fernando, 2014)

1.2 Planteamiento del Problema

Las plantas de Sistemas de Control de Flujo de Aire del fabricante DEGEM existentes en los laboratorios del Departamento de Eléctrica y Electrónica de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga, tienen ya varios años desde que se las adquirió por lo cual se considera que han cumplido su tiempo de vida, dificultando el aprendizaje práctico ya que estos equipos no proporcionan valores reales dando así problemas en las mediciones perdiendo exactitud para la implementación de las aplicaciones de control que se encuentren en desarrollo.

La tecnología con la que viene implementada las plantas de dicho sistema es antigua basado en controladores analógicos es decir amplificadores operacionales mas no controladores avanzados. En la tecnología analógica es muy difícil almacenar, manipular, comparar, calcular y recuperar información con exactitud cuando esta ha sido guardada. Otro problema que se tiene es que

hay muchos estudiantes por módulo lo cual dificulta el proceso de enseñanza aprendizaje.

Dentro de los algoritmos de control más utilizados en la actualidad están el controlador clásico (PID) y controladores avanzados (FUZZY LOGIC, PREDICTIVO, REDES NEURONALES entre otros), usados en la industria en los que la electrónica está inmersa directa o indirectamente, un problema que existe al momento de aplicar dichos controles es el costo de los materiales o equipos que se utilizan como es el caso de los Controladores Lógicos Programables (PLC), que tienen software propietario, y por tal motivo está obligado a usar hardware y software propios del fabricante lo que incrementa el valor total de la implementación.

1.3 Justificación e Importancia

El sistema didáctico a desarrollarse en el proyecto de titulación está orientado a utilizar tarjetas de desarrollo y software libre para facilitar el monitoreo y control de temperatura de flujo de aire con pocos recursos económicos, ya que las soluciones que existen actualmente tienen un costo elevado por los materiales que se usan para el desarrollo de las mismas utilizadas en la industria como son los PLCs. Así como también el proyecto aportará al proceso de enseñanza aprendizaje de los alumnos de la Carrera de Ingeniería de Electrónica e Instrumentación, y afines a la misma, ya que con la implementación del mismo, y la aplicación de diferentes algoritmos de control como un control clásico y avanzado se complementará los conocimientos aprendidos teóricamente dentro de la formación académica brindando mayores herramientas a las conocidas y tradicionales; además los alumnos que utilicen posteriormente este sistema tendrán una visión de cómo se trabaja en la industria con este tipo de aplicaciones que son muy comunes en el ámbito laboral, en donde se desempeñan los Ingenieros Electrónicos e Instrumentistas.

1.4 Objetivos del Proyecto

1.4.1 General

- Implementar un sistema de monitoreo y control de la temperatura de flujo de aire mediante hardware y software libre para su uso didáctico en el aprendizaje de control automático.

1.4.2 Específicos

- Investigar la factibilidad de la implementación de controladores avanzados en la tarjeta de desarrollo así como conocer el procedimiento de conexión de las mismas.

- Implementar un sistema didáctico de temperatura de flujo de aire.

- Desarrollar los algoritmos de control clásico y avanzado en la tarjeta de desarrollo y la interfaz gráfica para el monitoreo y control de la temperatura de flujo de aire.

- Realizar las pruebas requeridas con el óptimo funcionamiento de los controladores y sus correspondientes evaluaciones.

1.5 Hipótesis

La implementación del sistema de monitoreo y control para la temperatura de flujo de aire, permitirá afianzar el conocimiento integral en el control automático de procesos a los estudiantes de las carreras técnicas.

1.6 Variables de la Investigación

- **Variable independiente.**
  Sistema de monitoreo y control.

- **Variable dependiente.**
  Aprendizaje de control automático.

CAPÍTULO II

GENERALIDADES

2.1 Motor Trifásico

Un motor trifásico es una máquina eléctrica, que tiene la capacidad de transformar la energía eléctrica trifásica abastecida, en energía mecánica. La energía eléctrica trifásica produce campos magnéticos giratorios en el bobinado del estator permitiendo el arranque de estos motores sin que sea necesario la utilización de un circuito complementario para ello, en relación al tamaño son más pequeños y de menor peso que un monofásico de inducción que tiene similar potencia, en consecuencia de estas características la fabricación del mismo tiene un menor precio. (Ecured, 2013)

Figura 1 Motor Eléctrico Trifásico
Fuente: (Ecured, 2013)

2.1.1 Principio de funcionamiento

El principio de funcionamiento de un motor trifásico consiste en que cuando la corriente pasa a través de los arrollamientos de las tres fases del motor, esto produce en el estator un campo magnético, el mismo que induce corriente en las barras del rotor; esta corriente da origen a un flujo que al reaccionar con el flujo del campo magnético del estator, originará un par motor que pondrá en movimiento al rotor, este movimiento es continuo, ya que las variaciones de la corriente alterna trifásica también son continuas.

Algo importante que se debe resaltar es que el rotor no puede ir a la misma velocidad que la del campo magnético giratorio, esto se debe a que a cada momento recibe impulsos del campo, pero al interrumpir en el empuje, el rotor se retrasa; conociéndolo a este fenómeno como deslizamiento, posterior a esto se producirá un nuevo empuje y por lo tanto un nuevo deslizamiento repitiendo el ciclo. Entendiendo de esta forma que el rotor nunca consigue alcanzar igual velocidad que la del campo magnético giratorio; y es por lo cual se los puede denominar sincrónico o asincrónico según sea el caso.

Si el rotor tiene la misma velocidad de giro que la del campo magnético rotativo, se puede decir que el motor es síncrono. En cambio si el rotor tiene una velocidad de giro mayor o menor que dicho campo magnético rotativo, el motor es asíncrono de inducción. (Ecured, 2013)

2.1.2 Aplicaciones

Debido a la diversidad en cuanto al tamaño y a la potencia de los motores, se los utiliza de manera frecuente en la industria más no en el medio doméstico y residencial, siendo la razón principal para ello que en este entorno no se tiene la corriente trifásica; dentro de la industria se los usa para activar, extractores,

ventiladores, bombas, montacargas, grúas eléctricas, máquinas, herramienta, elevadores, etc. (Ecured, 2013)

2.1.3 Motor Trifásico Sincrónico

Se denominan motores sincrónicos a los cuales el número de polos y la frecuencia de la corriente de alimentación determinan la velocidad de giro, y con ello queda de manera independiente de la carga que deba realizar el motor, dicha velocidad viene dada por la siguiente expresión:

$$N = 120\ f / p$$

donde f es la frecuencia de la red y p el número de polos de la máquina.

Las aplicaciones en la mayoría de casos en el país están ligadas a la producción de energía eléctrica; todas las centrales Hidroeléctricas y Termoeléctricas trabajan mediante generadores síncronos trifásicos. La utilización en motores se la hace cuando el valor de la potencia no sobrepasa el valor de 1 MV. (Ecured, 2013)

2.1.4 Motor Trifásico Asincrónico

Conocidos también como motores de inducción son las máquinas de impulsión eléctrica más usadas debido a que son de bajo costo, sencillas y seguras; en los motores asíncronos el tipo de rotor determina su tipo habiendo así: motores de rotor bobinado o de anillos rozantes y motores de rotor en jaula de ardilla. (Ecured, 2013)

2.1.5 Ventajas de Motores Trifásicos

Las ventajas de los motores trifásicos de entre las más importantes son las siguientes:

- El rendimiento de estos motores es muy alto (alrededor de un 75% incrementando la misma en relación con el aumento de la potencia de la máquina)
- Se los puede fabricar de cualquier dimensión.
- Según el tipo de motor, posee un par de giro alto es decir este es constante.
- En estos motores no se tiene la necesidad de tener una bobina de arranque y por consiguiente capacitores, y menos aún interruptores centrífugos usados frecuentemente en los motores monofásicos.
- Basta con cambiar dos de las tres líneas de la entrada para cambiar el sentido de rotación del motor.
- Para disminuir la corriente inicial admite diferentes tipos de conexiones en la configuración del sistema de arranque. (Ecured, 2013)

2.2 Variadores de Frecuencia

Denominados también como Convertidores de Frecuencia o Inversores de Frecuencia, aparecen con la necesidad de usar los motores a velocidades variables sin que esto implique una disminución considerable en la eficiencia de los mismos, y por ello ahora estos equipos conectados a los motores puedan ser empleados en diferentes aplicaciones especiales.

Los variadores de frecuencia están dentro de la familia llamada Drivers en AC (AC Drives), a la cual también pertenecen otros dispositivos que son utilizados para el manejo de motores de corriente alterna como lo son partidores suaves los mismos que son utilizados para el arranque y parada de motores mas no para poder alterar la velocidad del mismo de forma permanente. (Cobo, 2008)

2.2.1 Principio de Funcionamiento

El funcionamiento de estos equipos consiste en proporcionar voltaje y frecuencia variable según la exigencia del motor y la carga que esté conectada a él, para la consecución de este objetivo se toma de la red la alimentación eléctrica la misma que tiene frecuencia y voltaje fijo, transformándola a la misma en un voltaje continuo para posteriormente transformarla en un voltaje alterno trifásico que tenga las características de frecuencia y magnitud variable a través de un inversor; partiendo desde esa etapa se puede realizar la alimentación de estos motores por medio de un abastecimiento de corriente continua como por ejemplo baterías. Además es posible alimentar a un motor trifásico si se tiene un rectificador monofásico que puede ser conectado a la red, es decir a una alimentación monofásica. (Cobo, 2008)

En la salida de este dispositivo no debe ser de manera rigurosa una forma de onda de voltaje que sea sinusoide perfecta, ya que proporcionan una señal de pulso modulada que resulta a partir de una conmutación alta de la frecuencia; sin embargo en equipos de la actualidad en los cuales se puede encontrar la frecuencia de conmutación que están entre los 50 Khz en donde los armónicos son bastante bajos, debido a esto gracias a la implementación de filtros pasivos se cumplen con normativas y exigencias que son requeridas en muchos países

En estos dispositivos el usuario realiza la configuración de la relación frecuencia voltaje según la necesidad de la aplicación, una de las configuraciones más utilizadas es la relación lineal, ya que permite tener un torque constante en todo el rango de velocidad que se esté manejando, o si desea que el torque se reduzca a medida que la velocidad disminuye se usa una relación cuadrática. (Cobo, 2008)

2.2.2 Funciones Adicionales

En la actualidad para la fabricación de estos equipos se han tomado en cuenta aspectos muy importantes que son de mucha utilidad en el uso del mismo es decir incluir funciones adicionales a la principal para el que fue creado, funciones como: funciones de control que pueden ser usadas en diferentes aplicaciones, protecciones al motor, diferentes controles PID controles secuenciales y controles lógicos; para poder hacer uso de todas estas funciones se han incluido en estos dispositivos los terminales de control necesarios con lo cual se podrán conectar entradas y salidas tanto analógicas como digitales, así como también puertos para comunicación de datos y además lineamientos de configuración. En la figura 2 se muestran diferentes tipos de variadores. (Cobo, 2008)

Figura 2 Variadores de Frecuencia
Fuente: (Cobo, 2008)

2.3 Ventilador Centrífugo

Un ventilador centrífugo es aquel que emplea el principio de la fuerza centrífuga haciendo pasar a través de una rueda en rotación un volumen de aire o gas específico. Funcionando de la siguiente manera, el gas o aire ingresa al

rotor en forma axial y altera en su dirección justo a la entrada, el mismo que seguirá su camino en dirección radial sobre el aspa o álabe para salir expulsado en forma tangencial. Dentro de la denominada carcasa la misma que es una estructura de metal con acero estructural realiza su trabajo el rotor que en ocasiones se lo suele llamar rueda o rodete, esta carcasa cumple con dos funciones específicas:

- Guiar el flujo de aire hacia la boca de entrada del rotor de una manera controlada.
- Guiar el flujo de aire que es centrifugado hacia el exterior de la rueda hacia la descarga del ventilador y se lo realiza igual de una manera controlada.

Para el diseño de un ventilador centrífugo se lo puede hacer de dos maneras sea con una o dos entradas de aire, y para el accionamiento del mismo se lo pude realizar a través de un motor o si fuera el caso por medio de una transmisión de bandas y poleas. Un ejemplo de estos ventiladores se puede observar en la figura 3:

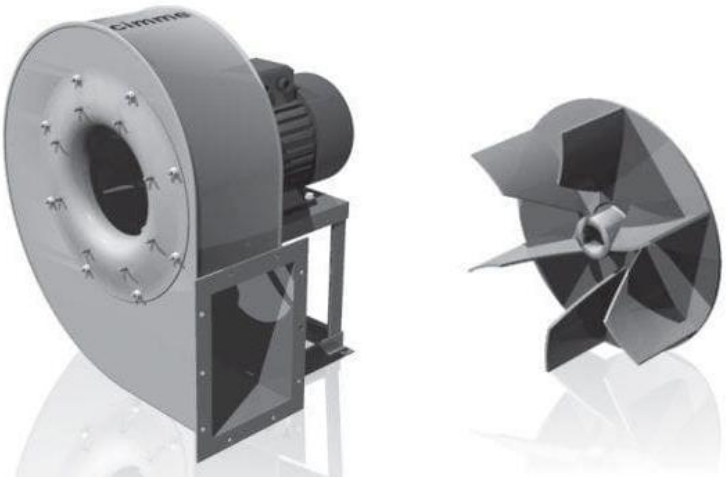

Figura 3 Ventilador Centrífugo de Álabes Radiales
Fuente: (Direct INDUSTRY, 2017)

En cuanto se refiere al tipo de aspas o álabes que se usan en los rotores, existen diferentes diseños los mismos que dependen de la cantidad de aire que se vaya a manejar, otro aspecto también es la presión en la aplicación o proceso, la temperatura, de entre otras consideraciones.

A continuación se describe los tipos de aspas o álabes que existen:

- Rotores de aspas airfoil considerados ventiladores de alta eficiencia
- Aspa curva inclinada hacia atrás usado en ventiladores que manejen aire puro es decir con una carga moderada de polvo.
- Aspas rectas inclinadas hacia atrás usado en ventiladores para el manejo de carga de polvo.
- Aspas radiales usado en ventiladores para el manejo de altas cargas de polvo). (Quiminet, 2010)

A continuación la figura 4 se tiene los tipos de álabes o rodetes que se tiene en los ventiladores centrífugos.

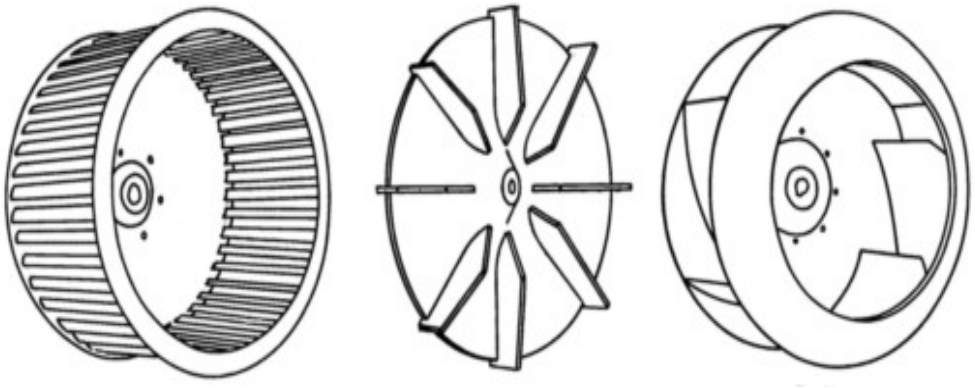

Figura 4 Ventiladores centrífugos de álabes curvados hacia delante, radiales y atrás.
Fuente: (Dani, 2008)

2.4 Conducto de Aire

2.4.1 Acero Galvanizado

El acero galvanizado es un una material que se obtiene mediante un proceso de galvanización es decir en este proceso lo que se realiza es proteger al acero recubriéndolo con una capa de zinc evitando así su oxidación. Ya que una de las características más importantes del zinc es proteger al acero a factores externos como la corrosión; este proceso consiste en sumergir las piezas de acero en zinc fundido y mediante una reacción metalúrgica entre estos dos materiales se forman aleaciones hierro – zinc permitiendo así una fuerte unión entre el acero y su recubrimiento. (Sinha, 2016)

2.4.2 Tubo de Acero

El acero galvanizado puede tomar diferentes formas pero de entre ellas la más usada comúnmente es en tuberías, ya que tiene muchas ventajas en cuanto a su uso como lo es el bajo nivel de corrosión, bajo costo con la relación a la vida útil del mismo, tiene un alta duración sin la necesidad de realizar un mantenimiento, entre otras características que hacen el uso frecuente de este material en el sector industrial. (Group, 2014)

2.4.3 Aplicaciones

De entre las aplicaciones de este material está la fabricación de estructuras, tuberías, depósitos industriales, en el sector de la construcción, también es muy utilizado en la agricultura y ganadería, etc. (Sinha, 2016)

2.5 Resistencia Calefactora

Una resistencia calefactora conocida comúnmente como niquelina es aquella que tiene la facultad de tener una determinada temperatura la misma que será usada en la calefacción de aire, según el grado de temperatura que se necesite tener.

La conductividad, longitud, la temperatura y la superficie transversal del objeto determinan la resistencia de un material conductor, la resistencia del material a una temperatura determinada es proporcional a la longitud del conductor e inversamente a la superficie transversal y a su conductividad; por lo general cuando la temperatura aumenta la resistencia del material incrementa. En la figura 5 se muestra diferentes tipos de resistencias calefactoras. (Moreno, 2002)

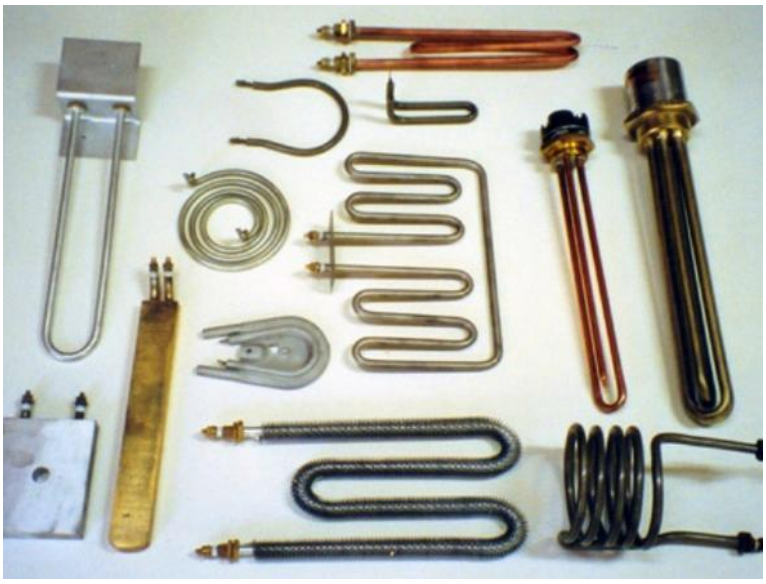

Figura 5 Tipos de Resistencias Calefactoras
Fuente: (IMEX, 2017)

2.6 Sensor

2.6.1 Sensores de Temperatura

La utilización de los sensores de temperatura en diferentes áreas es muy común e importante ya que con estos dispositivos se logra mantener en el rango de temperatura necesario los diferentes procesos lo que permite tener de una manera controlada y segura las diferentes aplicaciones como por ejemplo: climatización para control de ambientes, elaboración de alimentos, manejo y control de dispositivos en el sector automotriz, dispositivos médicos de entre otros usos.

Dentro de la clasificación de los sensores de temperatura se tiene dos grandes grupos que son: sensores sin contacto y con contacto; los sensores sin contacto son utilizados para entornos peligrosos es decir que la medición se la realiza a distancia, mientras que en los sensores de contacto están incluidos termistores y termopares que tienen contacto directo con lo que se vaya a medir. En la figura 6 se pueden observar diferentes tipos de sensores que existen en el marcado. (Mathas, 2011)

Figura 6 Sensores de Temperatura
Fuente: (MECALUX, 2016)

2.7 Software y Hardware Libre

2.7.1 Hardware libre

En los últimos años el denominado hardware libre, ha venido incursionando cada vez más, aunque por debajo del software libre, el objetivo del hardware libre es la creación de diferentes diseños de herramientas informáticos pero de manera libre, es decir que todos los usuarios tengan acceso a las mismas, o como mínimo a los planos de fabricación de los equipo o dispositivos; no siendo esto como algo novedoso; se remonta este tipo de tecnología a los años 70 ya que en aquella época las personas se puede decir aficionadas a los computadores se dedicaban a la fabricación de sus propios equipos dentro de sus hogares adquiriendo los materiales a diferentes fabricantes y con ello lograr sus propias implementaciones. (Digital, 2007)

"Hardware fuente abierta, esto se refiere al hardware para el cual toda la información del diseño se pone a disposición del público en general sin restricción alguna. Open Source hardware se puede basar en un Diseño de hardware libre"

2.7.2 Ventajas del hardware libre

- Favorece la calidad del hardware, a los estándares abiertos y que sean más de menor costo.
- Gracias al trabajo colaborativo sobre los diseños se admite la reutilización y la adaptación de los mismos para nuevas aplicaciones.
- Disminuye en cuanto al precio y tiempos en el diseño en sus trabajos.
- Se libera a los productores de los propietarios de alianzas globales. (EcuRed, 2016)

2.7.3 Software libre

En el software libre básicamente lo que se hace es respetar la libertad tanto de los usuarios como de la comunidad inmersa en este campo a copiar, ejecutar, estudiar, distribuir, modificar y mejorar un software determinado; es decir el software libre es un asunto de libertad mas no en el costo para la elaboración de proyectos, es decir libertad en la elección de materiales o herramientas. (GNU, 2016)

Comparando al software libre con el software propietario, el mismo que otorga algunos derechos a los usuarios se tiene algunas ventajas las mismas por las cuales este software es más usado por personas ajenas, por diferentes tipos de empresa e incluso por administraciones públicas; sin embargo el software libre ha sido parte de una serie mitos y mala información intentando cambiar los conceptos buenos del uso del mismo es decir haciendo dudar a los usuarios de su credibilidad. (Barahona Robles, 2003)

a. Ventajas del software libre

- **Costo**

  En su gran mayoría es muy atractivo el uso del software libre para usuarios individuales ya que el mismo por las libertades que presta garantiza no exceder en sobrecargos en el precio, mas no es el caso de la administración pública y empresas ya que en las mismas el costo de un software determinado es un punto importante al momento de la elección de nuevos sistemas informáticos en beneficio de su institución. (Más, 2003)

- **Innovación tecnológica**

  El modelo académico tradicional y científico es muy similar al modelo que utiliza el software libre en cuanto se refiere a la distribución de la información y el trabajo mancomunado, ayudando a estos modelos a tener muy buenos resultados ya que con la publicación de nuevos avances tecnológicos y científicos los mismos que servirán de inicio para nuevas investigaciones, siendo este el principal modelo en el que se ha basado la humanidad a lo largo de la historia y por medio se han tenido grandes logros para el desarrollo de la misma. (Más, 2003)

- **Requisitos de hardware y durabilidad de las soluciones**

  Aunque resulta imposible generalizar, sí que existen casos bien documentados donde las soluciones de software libre tienen unos requisitos de hardware menor, y por lo tanto son más baratas de implementar. Por ejemplo, los sistemas Linux que actúan de servidores pueden ser utilizados sin la interfaz gráfica con la consecuente reducción de requisitos de hardware necesarios. (Más, 2003)

- **Independencia del proveedor.**

  Dentro de la industria uno de los principales problemas que hay es la inevitable dependencia entre el cliente y el fabricante, esto hecho destaca con mayor trascendencia en el momento en el que el fabricante no proporciona el código fuente con lo cual el cliente queda ligado a él para nuevas actualizaciones o versiones del sistema entregado y en general para cualquier implementación de mejora que el cliente necesite. (Cusumano, 2004)

2.8 Beaglebone Black (BBB)

La Beaglebone Black es una tarjeta electrónica de bajo costo, que tiene un sistema operativo Linux Debian embebido de código abierto, que sirve para realizar aplicaciones con alto nivel de algoritmos de programa que brinda la libertad del software libre y un bajo nivel circuitos electrónicos ya que se utiliza las entradas y salidas de forma directa totalmente configurables. Para conocer un poco más de la misma, a continuación en la figura 7 se muestra la distribución de conexiones de la tarjeta de desarrollo Beaglebone Black y en la tabla 1 para conocer un poco más de la misma se detalla las características. (Richardson, 2013)

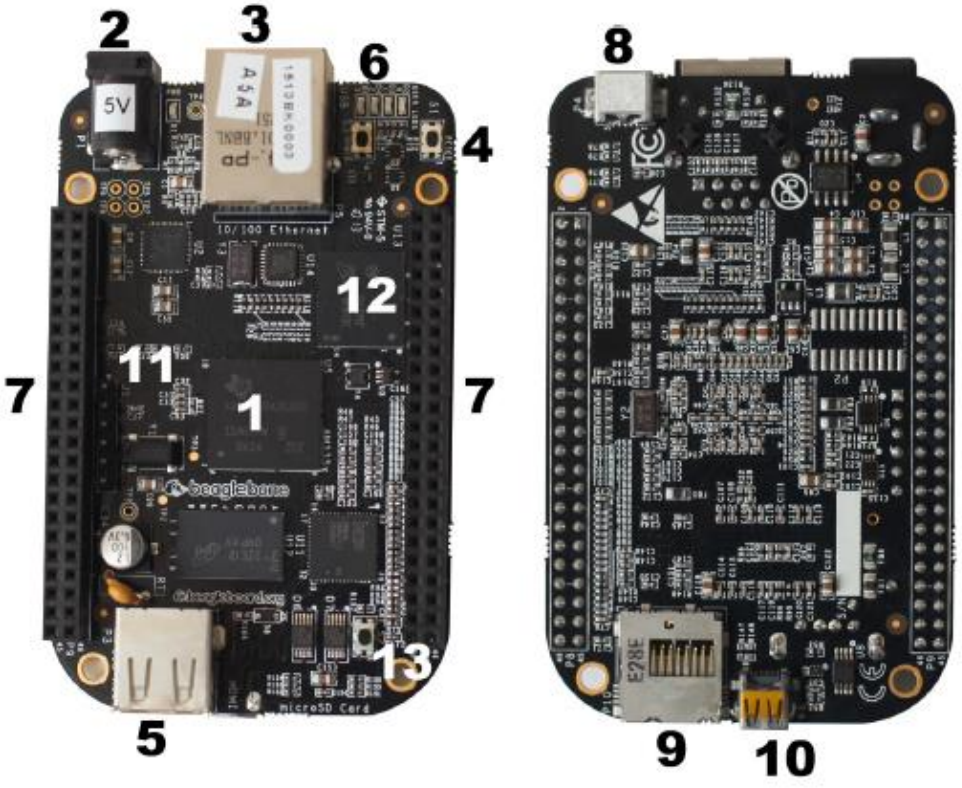

Figura 7 Tarjeta Beaglebone Black
Fuente: (Richardson, 2013)

En la tabla 1 se presenta las principales características de la tarjeta Beaglebone Black

Tabla 1

Características de la Tarjeta Beaglebone Black

	COMPONENTES	DETALLES
1	Procesador	1Ghz con 512MB de DDR3 RAM
2	Conector de Alimentación	La BBB necesita de 5V DC y 500mA
3	Puerto de Ethernet	Puerto estándar RJ45 para proyectos conectados a Internet, a un router.
4	Botón de Reinicio	Botón para reiniciar la placa al igual que un ordenador.
5	Puerto host USB	Al igual que un ordenador tiene un puerto USB para conectar hardware adicional como teclados, ratón, etc.
6	LEDs integrados	Tiene cuatro leds al lado del botón de reinicio. Led 0: Indica cuando el sistema está funcionando. Led 1: Parpadea cuando se está accediendo a la tarjeta SD. Led 2: Parpadea cuando la CPU está activa. Led 3: Parpadea cuando la memoria flash integrada se está accediendo.
7	Cabeceras de expansión	La placa tiene dos cabeceras de expansión

Continua

		denominados P8 y P9. La cual permite configurar una serie de funciones diferentes.
8	Mini puerto USB	Permite interactuar con la BBB como un dispositivo cuando se conecte a una computadora.
9	Ranura para la tarjeta MicroSD	La BBB no tiene un disco duro asi que utiliza una tarjeta MicroSD para almacenar el sistema operativo.
10	Puerto micro HDMI	Disponible solo en la BBB. Para conectar a un monitor o televisión.
11	Serial Header	Disponible solo en la BBB. Tiene salidas en serie para acceder al terminal.
12	Memoria Flash integrada	La BBB se puede arrancar sin una tarjeta MicroSD insertada, en los manuales se la conoce como eMMC.
13	Interruptor de arranque	Manteniendo presionado el interruptor de arranque se enciende la BBB desde la tarjeta MicroSD en lugar de la eMMC.

Fuente: (Richardson, 2013)

2.9 Linux

La distribución de Linux es una versión que se puede acceder de forma libre a los programas y herramientas de software a las diferentes versiones de Linux una de las de alto nivel que se puede instalar en Debian, OPEN SUSE entre otras. Al escoger una distribución de Linux en la tarjeta Beaglebone Black como sistema embebido se debería tomar en cuenta lo siguiente: (Molloy, 2015)

- Una distribución estable y bien soportada.
- Soporte de controladores para dispositivos para conectar con la tarjeta.

2.9.1 Distribución de Linux

Son algunas las distribuciones de Linux para sistemas embebidos que ofrecen una gran ventaja que es la programación en tiempo, cada distribución contiene herramientas y configuraciones tales que ofrece el software libre. Las principales distribuciones de Linux de la tarjeta Beaglebone Black son Debian, Angstrom y Ubuntu. (Molloy, 2015)

a. Debian

Es una distribución no comercial de Linux que se basa en el desarrollo de código abierto en libertad de software muy recomendada en tarjetas de sistemas embebidos por su estabilidad donde se distribuyen actualmente ya que ofrece un excelente soporte para el desarrollo de diferentes aplicaciones. (Molloy, 2015)

b. Angstrom

Es una distribución de Linux estable para dispositivos con sistema embebido como decodificadores, dispositivos móviles y de red. Es decir esta distribución se reduce a dispositivos con tan solo megabytes de almacenamiento. (Molloy, 2015)

c. Ubuntu

Tiene mucha relación con Debian porque la distribución antes mencionada es el soporte para el desarrollo de Ubuntu, siendo una de las distribuciones más populares debido a su enfoque al hacer más accesible para usuario nuevo en

software libre, fácil de instalar y soporte para controladores de escritorio. (Molloy, 2015)

2.10 Python

Python es un lenguaje de programación interpretado orientado a objetos, además de un código muy limpio que facilita la legitimidad de los algoritmos. Al ser un lenguaje interpretado es fundamental el uso de un intérprete en lugar de compilar el código, la gran ventaja es su velocidad además de su soporte y más flexibles. En Python el código fuente se traduce a pseudo código máquina. (Daniel & Vásconez, 2012)

2.10.1 Multiplataforma

Python está disponible en algunas plataformas como Unix, DOS, Windows, Linux, etc. Lo que es de mucha utilidad para que se pueda correr un programa en distintos sistemas sin la necesidad de hacer grandes cambios. (Daniel & Vásconez, 2012)

2.10.2 Orientado a Objetos

La programación orientada a objetos (POO) es una programación ligada a mensajes para hacer más legible el código para el uso de diferente tipo de programadores y así tener solución de problemas. A sido un desarrollo de la programación estructurada en la cual de definen funciones llamadas atributos. (Daniel & Vásconez, 2012)

2.10.3 Funciones de Python

Una función en Python y en la mayoría de los lenguajes de programación es una pequeña parte de código que por medio de un nombre que puede realizar varias series de tares previamente especificadas por el programador y que devuelven un valor o en otras no. Lo cual es de gran ayuda para depurar el algoritmo y la reutilización del código. (Daniel & Vásconez, 2012)

2.11 Controladores

2.11.1 Control PID

Los controladores son de mucha utilidad para mantener diferentes tipos de variables dentro de un punto de consigna y son los más utilizados en la industria, donde el 95% son lazos de control PID, los cuales han sobrevivido a los diferentes cambios tecnológicos por las prestaciones que ofrece y capacidad de resolver problemas que se encuentran en los procesos de control. (Sebastián & Julio, 2016)

Se da el nombre de control PID a un proceso controlado en lazo cerrado o realimentado, basado en acciones regulatorias como proporcional, integral y derivativa. Donde se logra la estabilidad del sistema a dichos cambios en sus variables. El regulador PID (proporcional, integral y derivativo) es apto para resolver los problemas de la industria. (Steeve, 2007)

A continuación se muestra un esquema de un control PID.

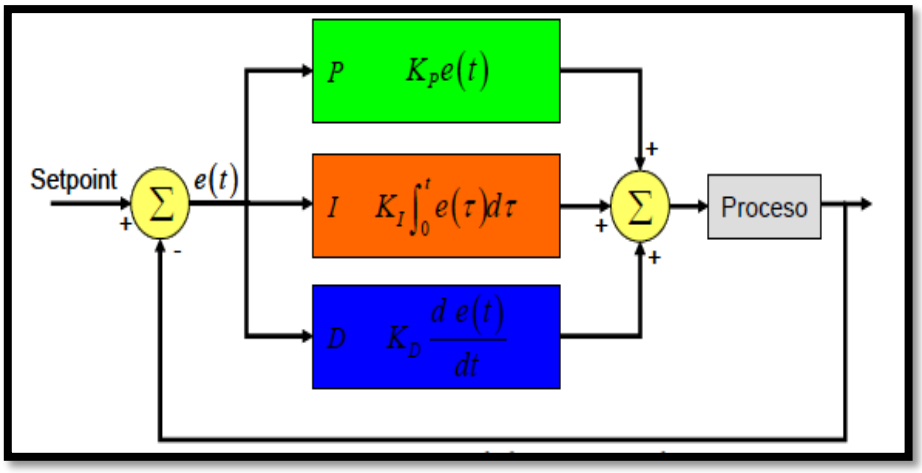

Figura 8: Funcionamiento Control PID
Fuente: (Steeve, 2007)

2.11.2 Control Digital PID

Un control digital o discreto en la salida muestrea cada intervalo de tiempo llamado periodo de muestreo y gracias a esto se obtiene una señal discretizada mediante un convertidor analógico digital (ADC), por lo tanto esta señal es con la que está trabajando en los algoritmos de control con lo que resulta mejor trabajar con ecuaciones en diferencia que con ecuaciones diferenciales. Y la señal de control es convertida nuevamente mediante un convertido digital analógico (DAC). (Dany)

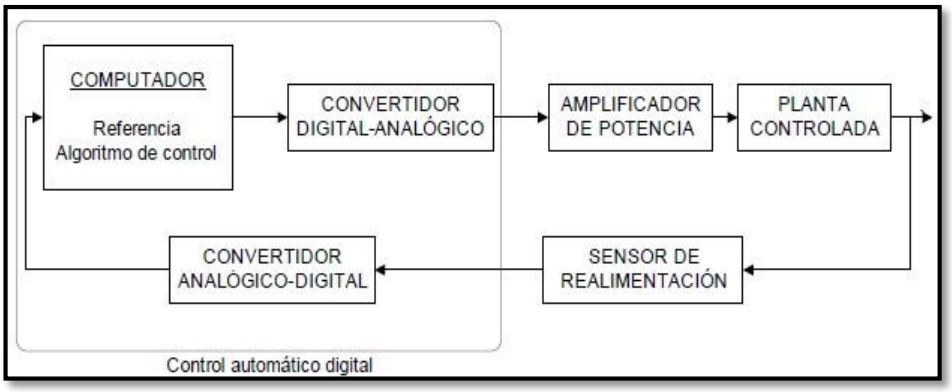

Figura 9 Estructura de un Control PID digital
Fuente: (Dany)

Características del control PID digital:

- El algoritmo puede ser implementado sin mucha dificultad.
- Los algoritmos de control presentan menos sensibilidad al ruido.
- La facilidad al realizar el ajuste en los controles digitales con tan solo modificando el programa.
- El costo es la principal característica para realizar un control digital. (Dany)

2.12 Control Difuso

El control difuso es una técnica de control que basa su funcionamiento en la lógica difusa, la misma que se parece a las decisiones que una persona toma conforme a lo experimentado u observado. El fin de la lógica difusa es solucionar situaciones dinámicas y complejas que son fácilmente explicados con palabras que realizarlos con modelos matemáticos.

En un controlador difuso se dispone de un algoritmo que transforma técnicas de control lingüística en una de control automático, en este controlador a través de una base de reglas se logra originar expresiones como la siguiente:

Si < Condiciones> Entonces <Acciones>

En la siguiente figura se muestra el funcionamiento de un controlador difuso (Cabezas & Capelo, 2012)

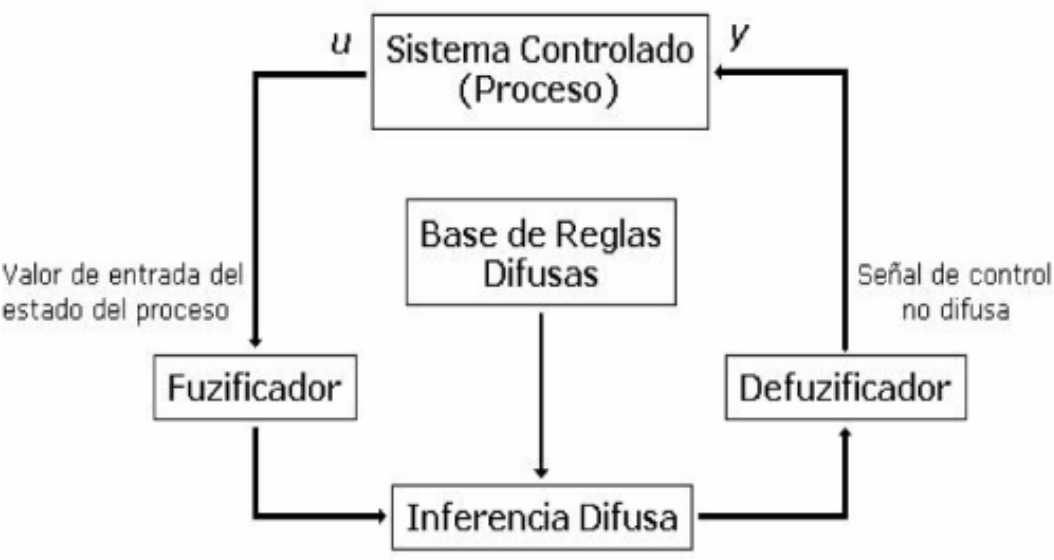

Figura 10 Funcionamiento Control Difuso

Fuente: (Cabezas & Capelo, 2012)

2.12.1 Fusificación

La tarea principal de la fusificación es transformar los valores que se miden del estado del proceso a valores lingüísticos, esto se lo hace a través de funciones de membrecía de las variables lingüísticas para obtener el grado de verdad o de pertenencia de cada uno de los conjuntos antes ya definidos.

2.12.2 Base de Reglas

En esta etapa se tiene todas las reglas lingüísticas que se usan para asociar los conceptos imprecisos con el comportamiento del sistema que se va a controlar, es decir en esta etapa esta la esencia del controlador ya que aquí se realiza la toma de decisiones que manejaran la forma de actuar del sistema.

2.12.3 Inferencia

En la inferencia toma lo que se realizó en la base de reglas para con ello originar las condiciones como la siguiente:

Si caso1 y caso3 cumplen entonces procede la accion1

2.12.4 Defuzificación

En esta última se hace el proceso de conversión de los valores difusos generados en los valores que serán utilizados para el proceso de control, esto se lo puede hacer a través de las funciones de membresía de la variable de salida. (Cabezas & Capelo, 2012)

CAPÍTULO III

DESARROLLO

En este capítulo se describe de forma detallada la realización de proyecto de investigación, como la selección de los materiales para la construcción del sistema de temperatura de flujo de aire, el mismo que posteriormente será objeto de pruebas, además la investigación para la implementación de los algoritmos de control y los entornos gráficos para la presentación de los resultados y con ello comprobar el funcionamiento total del sistema que será empleado.

3.1 Componentes usados en la fabricación del Sistema de Temperatura de Flujo de Aire

3.1.1 Variador de Frecuencia Siemens Sinamics G110

Se seleccionó este variador debido a las prestaciones que proporciona y las mismas se ajustan a las necesidades del proyecto como:

- Gracias a los ajustes realizados por defecto por parte del fabricante son ideales para el uso en aplicaciones sencillas para el control de motores.
- Tiene extensas funciones de seguridad lo que permite una extraordinaria protección tanto del variador como del motor.
- Puede ser usado en aplicaciones en las cuales esté integrado a los sistemas de automatización o aislado de los mismos.

Características Principales

- **Tensión de red:** 200 V a 240 V (±10%) 1AC
- **Frecuencia de Red:** 47 a 63 Hz

- **Potencia:** 120W a 3.0KW
- **Grado de Rendimiento:** 90% a 94% para equipos < 750 W y ≥ 95% para equipos ≥ 750 W
- **Entrada Digital:** 3
- **Entrada Analógica:** 1
- **Salida digital:** 1 salida de optoacoplador con separación galvánica
- **Métodos de Control:** Lineal V/f, cuadrática V/f, multipunto, V/f
- **Frenado:** Frenado por corriente continua.
- **Grado de Protección:** IP20
- **Funciones de Protección:** Subtensión, Sobretensión, Cortocircuito, Sobre temperatura del variador, Sobre temperatura del motor, Conexión a tierra.
- **Temperatura en servicio:** -10° C a + 40° C.

Figura 11 Variador Sinamics G110
Fuente: (Siemens, 2005)

Una vez conocidas las características más importantes del variador se debe conocer los parámetros que se deben configurar para ponerlo en servicio y no tener ningún inconveniente, lo que se detallará a continuación.

Para ponerlo en funcionamiento se lo hace a través de un BOP que quiere decir panel básico de operaciones por medio del cual se puede variar la velocidad o cambiar señales de control usando los botones pertinentes, este panel es el que se muestra en la figura 12:

Figura 12. BOP Sinamics G110
Fuente: (Siemens, 2005)

A continuación se detallan las funciones de los botones y estados que aparece en la pantalla de este dispositivo:

- En la pantalla cuando se tiene ***r*0000** indica un estado, es decir muestra los ajustes actuales del variador.

- Con este botón se pone en funcionamiento el variador, por defecto está bloqueado y para habilitarlo se debe ajustar el parámetro P0700 =1

-  Este botón es el de parada, tiene dos opciones de uso, la primera se llama OFF1 que se puede usar pulsando este botón con lo cual el motor se para con el tiempo de deceleración seleccionado por defecto está bloqueado y para habilitarlo se debe usar el parámetro P0700=1, y se tiene la opción dos que se llama OFF2 que funciona pulsando el botón dos veces o una vez de manera prolongada con ello el motor se detiene de forma natural es decir por inercia, esta función está habilitada de forma permanente.

-  Al pulsar este botón permite cambiar el sentido de giro del motor, el inverso se indica por medio de un signo negativo o también a través de un punto decimal intermitente, este botón se encuentra bloqueado por defecto y para habilitarlo se ajusta en el parámetro P0700=1.

- 

La función de este botón es que al pulsarlo el motor arranca y empieza a girar a la frecuencia Jog preseleccionada, y el motor se detiene cuando se deja de pulsar el botón, esta función no tiene ningún efecto cuando el motor ya está en marcha.

- 

Por medio de este botón se puede visualizar información adicional, pulsando de manera continua por dos segundos durante la marcha y desde cualquier parámetro muestra lo siguiente:

    - Frecuencia de salida en Hz
    - La tensión intermedia del circuito indicado mediante d. en unidades V.
    - La tensión de salida indicada mediante o. en unidades de V.
    - El valor seleccionado en el parámetro P0005.

Pulsando de manera rápida este botón es posible saltar desde cualquier parámetro a r0000 y estando allí si se vuelve a pulsar el botón FN irá de nuevo al punto inicial.

- Por medio de este botón se puede acceder a los parámetros de configuración.
- Con la pulsación de este botón aumenta el valor visualizado en la pantalla.
- Con la pulsación de este botón disminuye el valor visualizado en la pantalla.

También se puede realizar la variación de dígitos de manera individual en los valores de los diferentes parámetros lo cual se lo puede hacer de la siguiente manera:

1. Se debe estar seguro que se esté en el nivel de cambio de valor de parámetro.
2. Luego se pulsa el botón de funciones FN y con ello el digito derecho empieza a parpadear.
3. Para cambiar el valor de ese digito se los hace pulsando los botones de flecha hacia arriba o abajo.
4. Para cambiar el siguiente dígito se pulsa nuevamente la tecla de funciones FN y con ello parpadea el digito siguiente.
5. Y cuando se tenga el valor requerido visualizado en la pantalla se debe pulsar el botón P para salir del nivel de cambio.

Ahora para la selección de parámetros y realizar la modificación de los mismos utilizando el panel de operaciones básicas BOP se lo puede hacer de la siguiente manera:

Para acceder a un parámetro de configuración se aplasta el botón **P** y esto dará como resultado en la pantalla ***r*0000**, como ejemplo se accederá al nivel de acceso P0003, para lo cual se debe pulsar la tecla de flecha hacia ↑ arriba hasta que aparezca este parámetro, posterior a esto se debe pulsar

nuevamente la tecla **P** hasta acceder al valor del parámetro y con las teclas de flecha hacia arriba ↑ o hacia abajo ↓ se escoge el nivel como por ejemplo el **3,** por último nuevamente se debe pulsar la tecla **P** para con ello confirmar y guardar el valor escogido, con el nivel de acceso 3 se pueden modificar o seleccionar todos los parámetros del nivel 1 al 3.

En ocasiones al intentar modificar algún parámetro en la pantalla del BOP puede mostrar la palabra **buSY**, lo que quiere que el variador esté ocupado realizando tareas que sean de mayor prioridad.

Una vez conocido como varias los parámetros, para poner en funcionamiento el variador se escogió la puesta en servicio rápida lo cual se lo realiza de la siguiente manera:

Se accede al parámetro de configuración P0010 el mismo que se usa para la puesta en servicio rápida y también es una forma óptima para adaptar el variador a un motor determinado, el mismo que se lo debe poner en **1** ya que este nivel es el de guía básica para la configuración, dentro de este existen niveles como el **2** que se lo usa solo para tareas de revisión, de entre otros niveles que se pueden configurar.

Ahora bien una vez escogida el nivel P0010, se ingresa al mismo, dentro del cual primero se elige la región de trabajo en la que se vaya a utilizar el dispositivo en este caso se ingresa a **P0100** y en el cual se escoge como nivel de acceso en 1 que corresponde Norte América con una frecuencia de trabajo de 60Hz y se ubica como constante la C, que es lo que recomienda el manual para esta configuración.

Luego de ello otro parámetro importante es el **P0304** que se refiere a la tensión nominal del motor que se va a usar, el mismo que se puede observar

en la placa del motor la misma que lleva las características importantes del mismo, para este caso se escoge como nivel de acceso 1 y nuevamente como constante la C, importante que este parámetro solo se lo puede configurar cuando **P0010** este en el nivel 1.

El siguiente parámetro a modificar es el **P0305** el que se refiere a la corriente nominal del motor que de igual manera que la tensión nominal esto se lo puede encontrar en la placa de características del motor, aquí se escoge el nivel de acceso 1 y la constante C.

Otro parámetro a configurar es el **P0307** correspondiente a la potencia nominal del motor que de igual manera se debe poner de acuerdo a la placa del motor , lo que se debe tener en cuenta que si P0100 está en el nivel 1, se debe tomar el valor de la placa del motor en hp. En la figura siguiente se muestra una placa de un motor en las que se puede ver las características antes mencionadas para la configuración de la puesta en servicio rápida de un variador Sinamics G110.

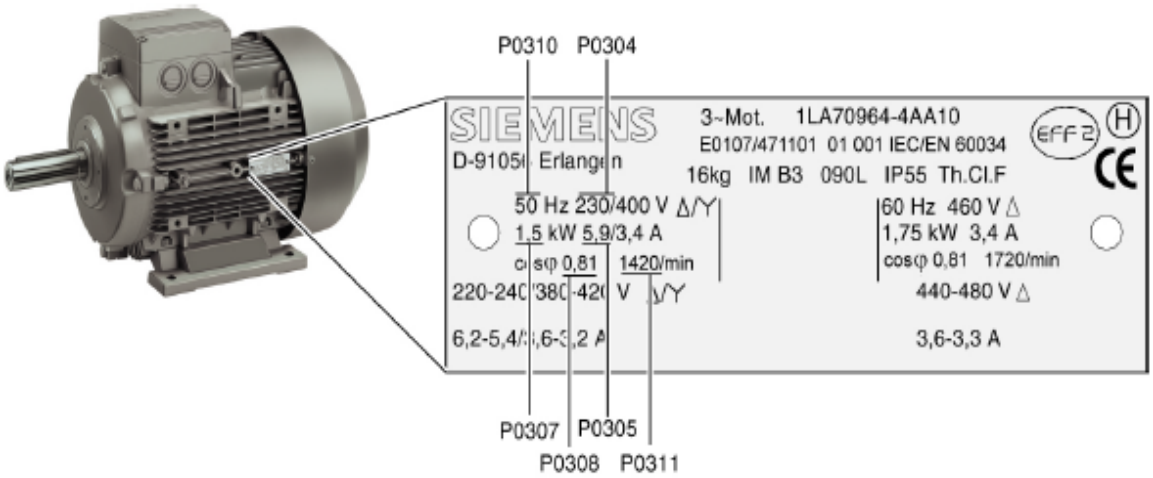

Figura 13 Ejemplo de una placa de características de un motor
Fuente: (Siemens, 2005)

Con la configuración de los parámetros antes descritos su puede poner en funcionamiento el variador Sinamics G110, sin tener ningún inconveniente y usarlo en cualquier uso que se lo vaya a dar.

3.1.2 Motor Trifásico ABB

El proyecto se lo realizará con un motor trifásico ya que con lo que se está realizando se quiere tener un sistema que se asemeje a lo que se tiene en la industria, ya que con motores trifásicos se ayuda a la disminución del consumo de energía, también se tiene mayor potencia que un motor monofásico aproximadamente de un 150% más y por lo tanto para el sector industrial representa un ahorro.

Ahora el motor que se ha seleccionado es de Marca ABB las características del mismo se describen a continuación.

Características principales

- **Marca:** ABB Motors
- **IEC:** 600I34
- **Voltaje** Δ**:** 208 - 230
- **Voltaje** **:** 440 - 460
- **Potencia:** 2.2 KW
- **Cos φ :** 0.74
- **Frecuencia:** 60Hz

En la figura 14 se observa el motor que se usó para el armado de la planta.

Figura 14 Motor Trifásico Marca ABB
Fuente: (Ecured, 2013)

3.1.3 Ventilador Centrífugo Palas Radial

De entre los diferentes tipos de ventiladores que existen, para la aplicación que se desarrolló, se consideró usar un ventilador centrífugo de palas o álabes radiales ya que es el de configuración más simple y además no se tiene la necesidad que el mismo tenga un alto rendimiento, es la mejor opción para el desarrollo del sistema de temperatura de flujo de aire. En la figura 15 se observa en ventilador adaptado al motor para la generación de flujo de aire.

Figura 15 Ventilador centrífugo de alabes radiales

3.1.4 Resistencia Calefactora

Para la resistencia calefactora se tomó en cuenta las necesidades del sistema como el voltaje de trabajo, el rango de temperatura en el que se va a controlar entre otros parámetros, con lo cual la resistencia fabricada se puede observar en la figura 16 y tiene las siguientes características:

Voltaje: 220V

Corriente: 11,3 A

Potencia: 2500 W

Calibre: 3/16

Longitud: 180 cm

Temperatura Máxima: 100°C

Figura 16 Resistencia Calefactora Espiral
Fuente: (Industrial, 2017)

3.1.5 Ducto de Aire

Para el ducto de aire del sistema didáctico de entre las diferentes opciones, se escogió el acero galvanizado ya que este material es uno de los más

utilizados dentro del sector industrial debido a sus prestaciones como es el bajo nivel de corrosión que tiene, el bajo costo, y su alta duración sin la necesidad de realizar un mantenimiento; el mismo que tiene de diámetro 4 pulgadas y de largo 100 cm. En la figura siguiente se observa el ducto fabricado para el sistema.

Figura 17 Ducto de flujo aire

3.1.6 Sensor

Luego de comparar entre los diversos sensores de temperatura, y según las necesidades del proyecto se usó el sensor digital DS18B20 encapsulado ya que cuenta con salida lineal a los grados centígrados, el mismo tiene como característica principal y de gran ayuda que utiliza la comunicación OneWire que es un protocolo que permite enviar y recibir datos por un solo cable a diferencia de la mayoría de protocolos que necesitan de dos cables como se lo observa en la figura 18, además presenta las siguientes características importantes:

- Sensor Digital
- Resolución de 9 y 12 bits
- Rango de operación de -50 a 125 grados Centígrados
- Precisión +- 0.5 grados

- Protocolo OneWire
- De bajo costo debido a su estructura.
- No requiere componentes externos

Figura 18 Sensor DS18B20
Fuente: (Caldas, 2017)

A continuación en la figura 19 se muestra la conexión los cables según el color para que no exista ningun problema en el funcionamiento del mismo:

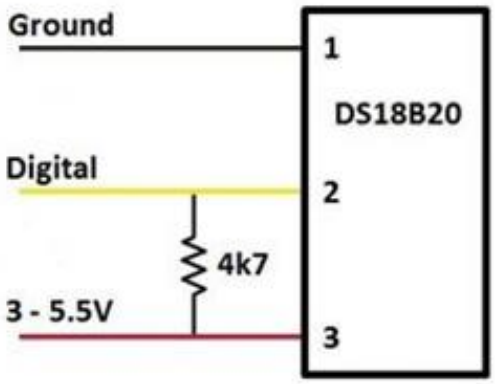

Figura 19 Nombres cables sensor DS18B20
Fuente: (Caldas, 2017)

3.1.7 Relé de Estado Sólido HFS15

Para actuador de este sistema se usó el relé de estado sólido HFS15 como el de la figura 20, debido a las características que brinda las cuales se presentan a continuación:

- Salida Triac AC
- Control DC o AC
- LED indicador de funcionamiento
- Rango de Control de 1.5 – 32 VDC
- Rango de Voltaje hasta 240 VAC
- Carga 15 A
- Carga Resistiva
- Montaje en Panel
- Rango de temperatura de funcionamiento de -30°C a 80°C

Figura 20 Relé de Estado Sólido HFS15
Fuente: (Electrónica, 2017)

A continuación en la figura 21 se presenta el esquema eléctrico del sistema de temperatura de flujo de aire, el mismo que comprende de la conexión de la línea de tensión trifásica al variador y luego al motor, L1 y L2 son las líneas de tensión de la red, PE se refiere a la protección a tierra, mientras que las letras U, V y W son los cables de conexión entre el variador y el motor. Mientras que R2 es un relé de protección que está usado de puente para que arranque el motor.

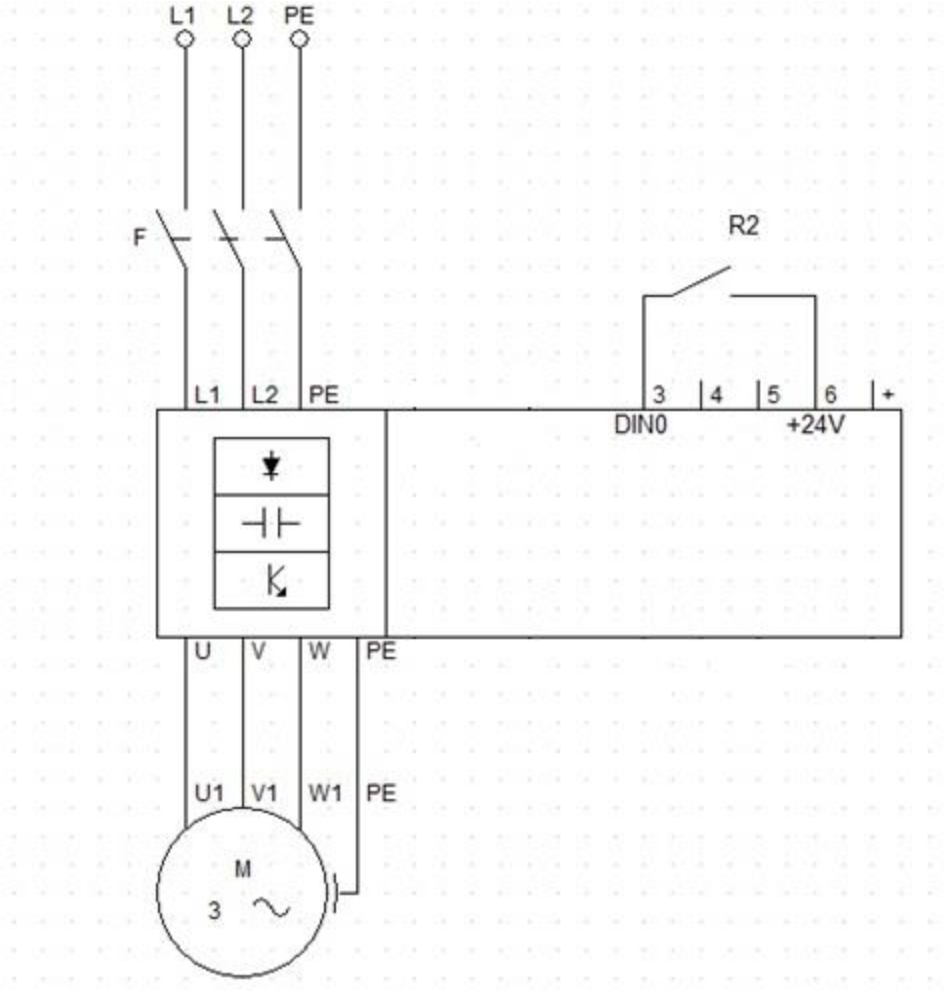

Figura 21 Circuito Eléctrico de la Planta de Flujo de Aire

En la figura 22 se muestra el circuito de control del sistema de tempearatura de flujo de aire, en el mismo se encuentra F1 y F2 que son fusibles de protección, R1 es un relé electrónico que permite enviar la señal de control, hacia la resistencia calefactora, R2 es un relé de protección y además esta usado como puente para que arranque el variador y por lo tanto el motor, RC es la resistencia calefactora, LV indica que el sistema esta encendido y por úlitmo LR indica que el sistema empezó a funcionar.

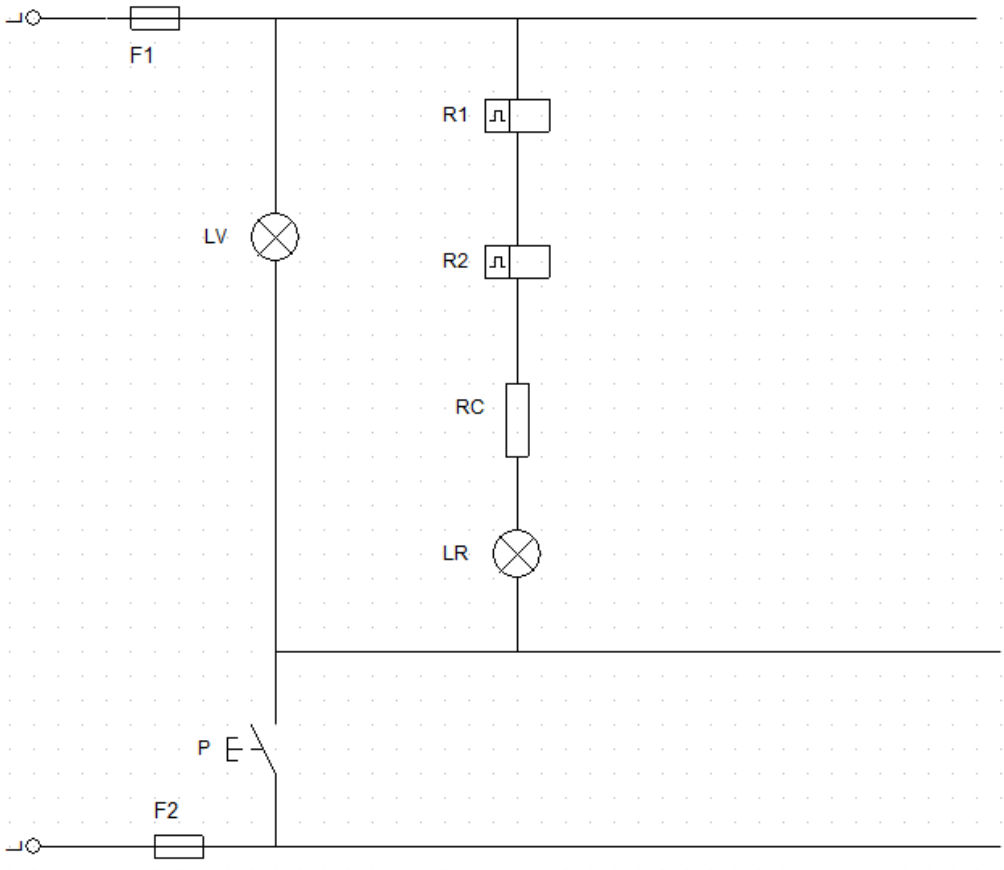

Figura 22 Circuito de Control de la Planta de Flujo de Aire

Una vez descritos todos los elementos que componen el sistema a continuación se presenta la implementación de la planta de temperatura de flujo de aire, como se observa en la figura 23.

Figura 23 Planta de temperatura de flujo de aire

3.2 Instalación del sistema operativo en la tarjeta Beaglebone Black Rev C

Para el proceso de instalación del sistema operativo de la tarjeta Beaglebone Black Rev C, se debe ingresar a la página recomendada por el fabricante la misma que tiene de nombre beagleboard.org se puede acceder a ella en el siguiente link: https://beagleboard.org/.

Una vez en la página, lo primero que se debe hacer es descargar la imagen, para el caso del proyecto se decidió utilizar el sistema operativo Linux Debian 7.9, ya que este sistema es el que mejor se ajusta a las características deseadas como el rendimiento y la compatibilidad necesaria que debe existir

entre hardware y software. Para la descarga de la imagen que se va a utilizar se ingresa a la siguiente dirección: https://beagleboard.org/latest-images.

Una recomendación que se debe tomar en cuenta es que la instalación del sistema operativo se lo debe hacer en una tarjeta microSD que tenga la capacidad mayor a 4GB, para este caso se usó una de 16 GB, además la tarjeta debe ser de clase 10 lo que permite la escritura a 10Mb/s.

Una vez hecho la descarga del archivo que tiene una extensión **.img.xz**, se debe hacer uso del programa 7zip para poder extraer el contenido de la imagen y poder avanzar sin ningún inconveniente en la instalación del sistema como lo muestra la figura 24.

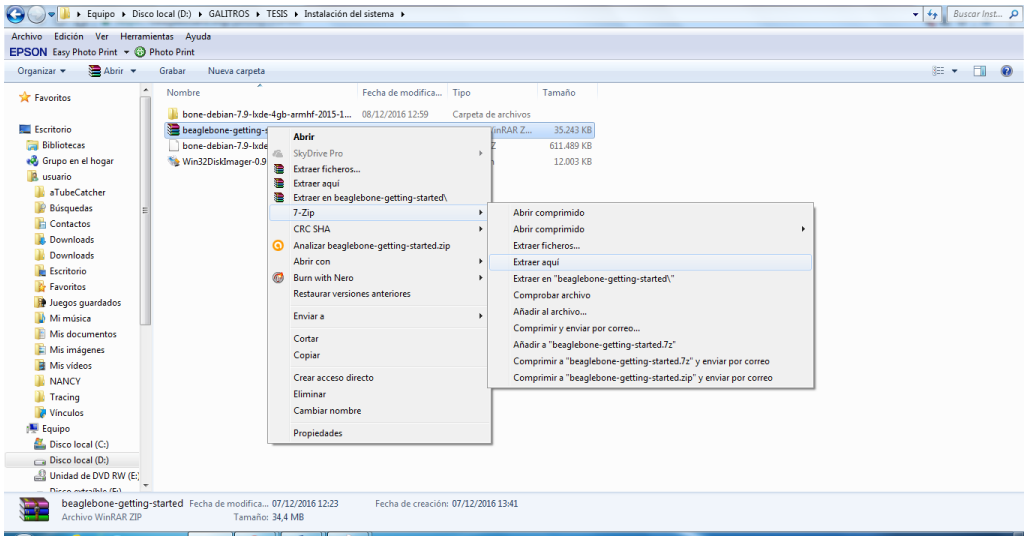

Figura 24 Extracción del sistema Operativo Linux Debian 7.9 en 7-Zip

Una vez descomprimida la imagen, con la ayuda del programa Win32 Disk Imager se debe escribir la misma en la tarjeta MicroSD como lo muestra la figura 25, dicho programa se puede descargar de la siguiente dirección: https://sourceforge.net/projects/win32diskimager/files/latest/download,

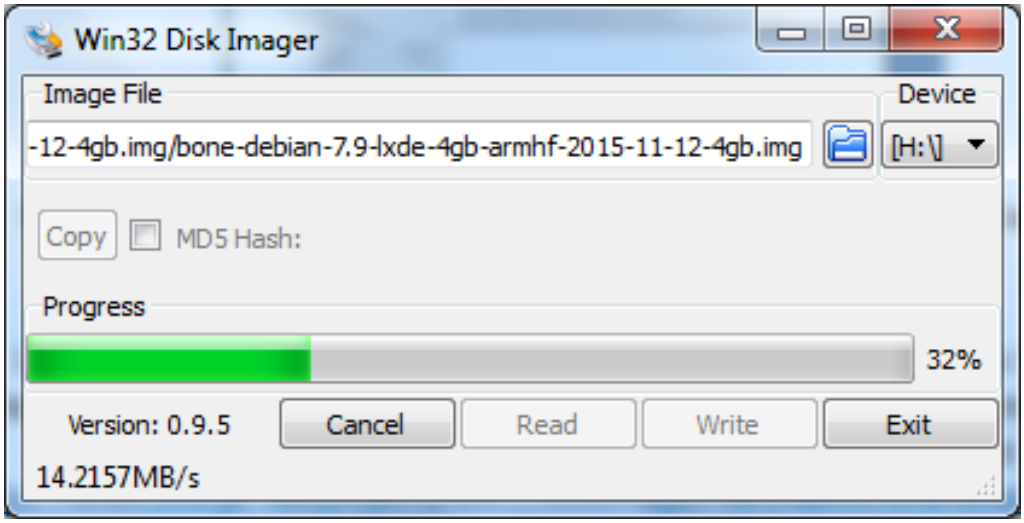

Figura 25 Escritura de la imagen en la tarjeta MicroSD

Una vez instalado el sistema operativo Linux Debian en la MicroSD se inserta en el puerto respectivo de la Beaglebone Black, posterior a esto se enciende la tarjeta y para empezar a usarla de inicio aparecerá una ventana en la cual se debe ingresar un usuario y una contraseña, para este caso se ingresó como usuario **root** el mismo que no requiere de contraseña luego de esto aparecerá el entorno de escritorio como se puede observar en la figura 26:

Figura 26 Escritorio Linux Debian

Ahora para la programación como tal en Linux Debian se usa la consola LXTerminal a la cual se accede haciendo clic en el icono que está en la esquina inferior izquierda, luego escoger la opción de accesorios y posteriormente dar clic en LXTerminal y aparecerá una ventana como se observa en la figura 27:

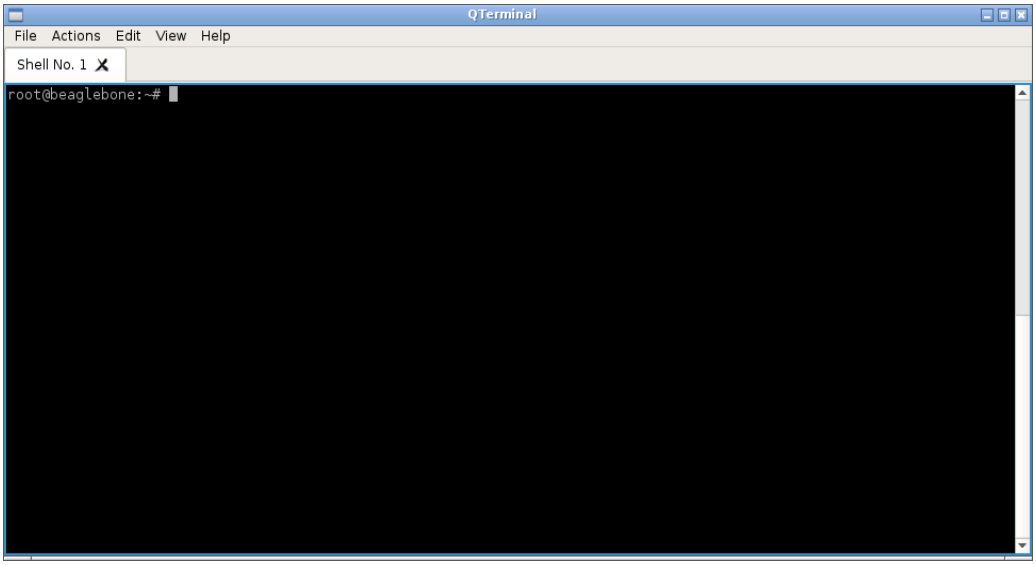

Figura 27 Ventana de Consola LXTerminal

A través de esta consola LXTerminal se podrá ejecutar instrucciones y comandos con los cuales se desarrollará con más facilidad las diferentes aplicaciones. En esta ventana aparece de inicio root@beaglebone:~#.

Los primeros comandos que pueden ser usados para el inicio dentro de Linux son:

- **more /etc/issue:** permite retornar la distribución de Linux que se está usando.
- **ps –p $$:** devuelve el shell que se está utilizando.
- **uptime:** permite conocer cuánto tiempo ha estado funcionando el sistema.
- **top:** enumera todos los procesos que se están ejecutando actualmente.

3.2.1 Comandos Básicos en Archivos de Linux

A continuación se describen los comandos básicos que se necesitará para moverse y manejar sin ningún inconveniente los archivos de sistema de Linux.

Una recomendación que se debe tomar en cuenta es que cuando se esté utilizando cuentas de usuario de Debian y Ubuntu, con frecuencia se debe prefijar la palabra “sudo” al inicio de algunos comandos, esto es debido a que “sudo” permite a los usuarios ejecutar programas con ciertos privilegios de seguridad de super usuario.

Los comandos básicos se describen en la siguiente tabla:

Tabla 2

Comandos básicos usados en archivos de sistema de Linux

Nombre	Comando	Opciones e Información	Ejemplo
Listar archivos	ls	- a muestra todo incluso archivos ocultos - S clasifica por tamaño de archivo - h proporciona tamaño de archivos legibles por el usuario - t ordena las últimas modificaciones	ls -al
Directorio actual	pwd	Imprime el directorio de trabajo - P imprime la dirección física	pwd -p
Cambiar	cd	Cambio de directorio	cd /home/root

Continua

directorio		cd luego enter o cd ~/ le lleva al inicio de directorio cd/ le lleva al sistema de archivos de raíz cd **..** le lleva hasta un nivel	cd /
Hacer un directorio	mkdir	Hace un directorio	mkdir test
Borrar un archivo o directorio	rm	Borra un archivo - r borrado recursivo usado para directorios -d borrado de directorios vacíos	rm bad.txt rm –r test
Copiar un archivo o directorio	cp	-r copia recursiva -u copiar solo si el origen es más reciente que el destino o el destino está perdido -v copia detallada	cp a.txt b.txt cp –r test testa
Mover un archivo o directorio	mv	-i menajes antes de sobrescribir, no usar –r para directorios, para cambiar al mismo directorio se realiza un cambio de nombre	mv a.txt c.txt mv test testb
Crear un archivo vacío	touch	Crea un archivo vacío o actualiza la fecha de modificación de un archivo	touch d.txt

Continua

Ver el contenido de un archivo	more	Ver el contenido de un archivo. Use la tecla de espacio para la siguiente pagina	more d.txt
Obtener el calendario	cal	Muestra un calendario basado en texto	cal 01 2015

Fuente: (Richardson, 2013)

A continuación se detallan algunos comandos que facilitarán el trabajo en la mayoría de shell de Linux:

- **Flecha hacia arriba:** proporciona el último comando escrito, y con más pulsaciones los comandos anteriormente usados.
- **Tecla de tabulación:** permite completar automáticamente el nombre del archivo, el nombre del directorio e inclusive el nombre del comando ejecutable. Ejemplo, para cambiar al directorio Linux/tmp, puede escribir cd/t y luego pulsar Tab, que completara automáticamente el comando cd/tmp/. Si existe muchas opciones, presione la tecla Tab nuevamente para ver todas las opciones como una lista.
- **Ctrl + A:** permite ir al inicio de la línea que se esté escribiendo
- **Ctrl + E:** permite ir al final de la línea que se está escribiendo.
- **Ctrl + U:** permite borrar hasta el inicio de la línea. Ctrl+E y luego control Ctrl+U borra la línea.
- **Ctrl + L:** permite borrar la pantalla.
- **Ctrl + C:** permite eliminar cualquier proceso en curso.
- **Ctrl + Z:** pone el proceso actual en segundo plano, se escribe bg a continuación y se ejecuta en el fondo, y presionando fg lo trae nuevamente al primer plano.

Para tener una idea claro de algunos comando a continuación se muestra un ejemplo se creara un directorio de nombre test en el que se crea un archivo de

texto vacío hello.txt. Todo el directorio de prueba se copia en el directorio /tmp, que está fuera del directorio de la raíz.

root@beaglebone:~# **cd ~/**

root@beaglebone:~# **pwd**

/root

root@beaglebone:~# **mkdir test**

root@beaglebone:~# **cd test**

root@beaglebone:~/test# **touch hello.txt**

root@beaglebone:~/test# **ls**

hello.txt

root@beaglebone:~/test# **cd . .**

root@beaglebone:~# **cp –r test /tmp**

root@beaglebone:~# cd /tmp/test/

root@beaglebone: /tmp/test# **ls**

hello.txt

3.2.2 Edición Básica de Archivos

Existen una gran variedad de editores disponibles en el mercado, sin embargo uno de los más fáciles de usar cuando se está iniciando en el software libre y por lo tanto en Python y que además es uno de los más poderosos es el editor GNU nano. Para empezar el uso del editor se escribe en la consola nano seguido del nombre de un archivo existente o nuevo si es la primera vez que lo va a usar según el caso; por ejemplo nano hello.txt y se mostrara la ventana del editor como la figura 28.

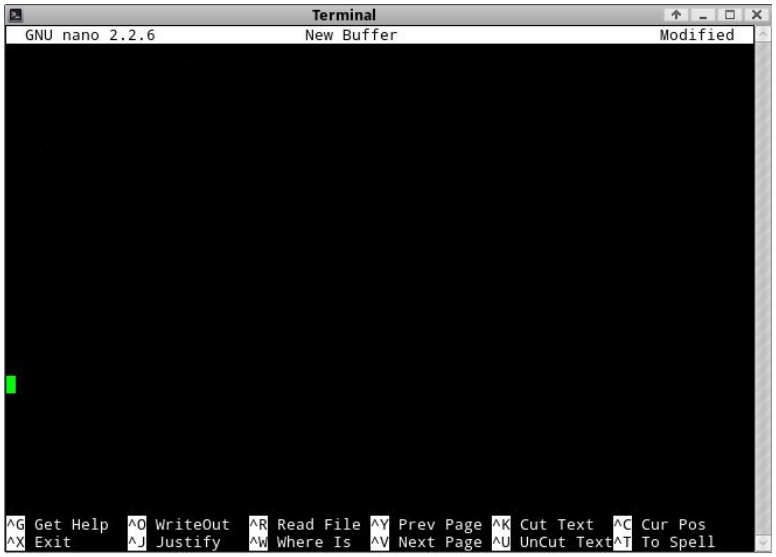

Figura 28 Editor GNU nano

Para moverse de mejor manera y con más facilidad alrededor del editor se usa las teclas de las flechas y escribir los códigos en la ubicación del cursor, en la parte inferior de la ventana se muestran teclas de atajo, a continuación se describen la funcionalidad de todas ellas en la siguiente tabla:

Tabla 3

Comandos básicos usados en archivos en el Editor GNU Nano

Comando	Función
Ctrl + G	Ayuda
Ctrl + C	Cancelar
Ctrl + X	Salir (los mensajes quedan guardados)
Ctrl + L	Habilita el envolvimiento de líneas largas
Ctrl + O	Guardar
Flechas	Moverse alrededor
Ctrl + A	Ir al inicio de la línea

Continua

Ctrl + E	Ir al final de la línea
Ctrl + Espacio	Siguiente palabra
Alt + Espacio	Palabra previa
Ctrl + V	Siguiente pagina
Ctrl + Y	Página previa
Ctrl+_ o Ctrl+/	Ir a un número de línea
Alt + /	Ir al final del archivo
Ctrl + 6	Marca texto (con las flechas se resalta)
Ctrl+K o Alt+6	Corta el texto marcado
Ctrl + U	Pega el texto
Ctrl + R	Inserta el contenido de otro archivo
Ctrl + W	Buscar una cadena
Ctrl + D	Eliminar el carácter bajo el cursor
Ctrl + K	Eliminar toda la línea

Fuente: (Richardson, 2013)

Para el desarrollo como tal de los algoritmos de control que se va a implementar, siendo estos el control PID y el control Fuzzy Logic se lo hará en el editor GNU nano, se utilizará éste ya que es el que viene instalado en el sistema operativo de Linux Debian para el lenguaje de programación Python, este es un lenguaje que presenta muchas facilidades para el desarrollo de diferentes aplicaciones en software libre por lo cual se lo ha escogido, este es

un lenguaje que permite la optimización para el programador en cuanto a la productividad se refiere, la legibilidad del código y la calidad de software.

Está enfocado principalmente a la programación orientada a objetos ya que desde sus inicios los modelos soportan múltiple herencia, la sobrecarga del operador y los modelos de clase como son los polimorfismos; también resulta más fácil la programación orientada a objetos en Python que en otros lenguajes.

Python también en cuanto a la extensión del código, es mucho menor en cuanto a otros lenguajes como Java o C++, ahora con esto como se debe digitar menos, hay menos que revisar, corregir o mejorar, así como también se puede correr un programa sin la necesidad de que exista la compilación de otras herramientas de esta manera mejorando la velocidad del programa.

Con la uniformidad que permite tener un código de Python lo hace más fácil de comprender y además de que tiene un gran soporte para el uso de software más avanzado, como la programación funcional mejorando la productividad del desarrollador en comparación con lenguajes como Java o C++.

Este lenguaje es combinable lo que quiere decir que programas de Python pueden ser unidos fácilmente a otros componentes escritos en distintos idiomas en una variedad de formas.

3.3 Configuración Sensor DS18B20

Algo muy importante para la implementación de los algoritmos de control es la configuración del sensor que será usado en el sistema de monitoreo y control para lo cual se debe hacer lo siguiente:

Se debe crear un archivo en el editor de texto GNU con el siguiente nombre y extensión DS1820 – 00A00.dts como se lo observa en la figura 29:

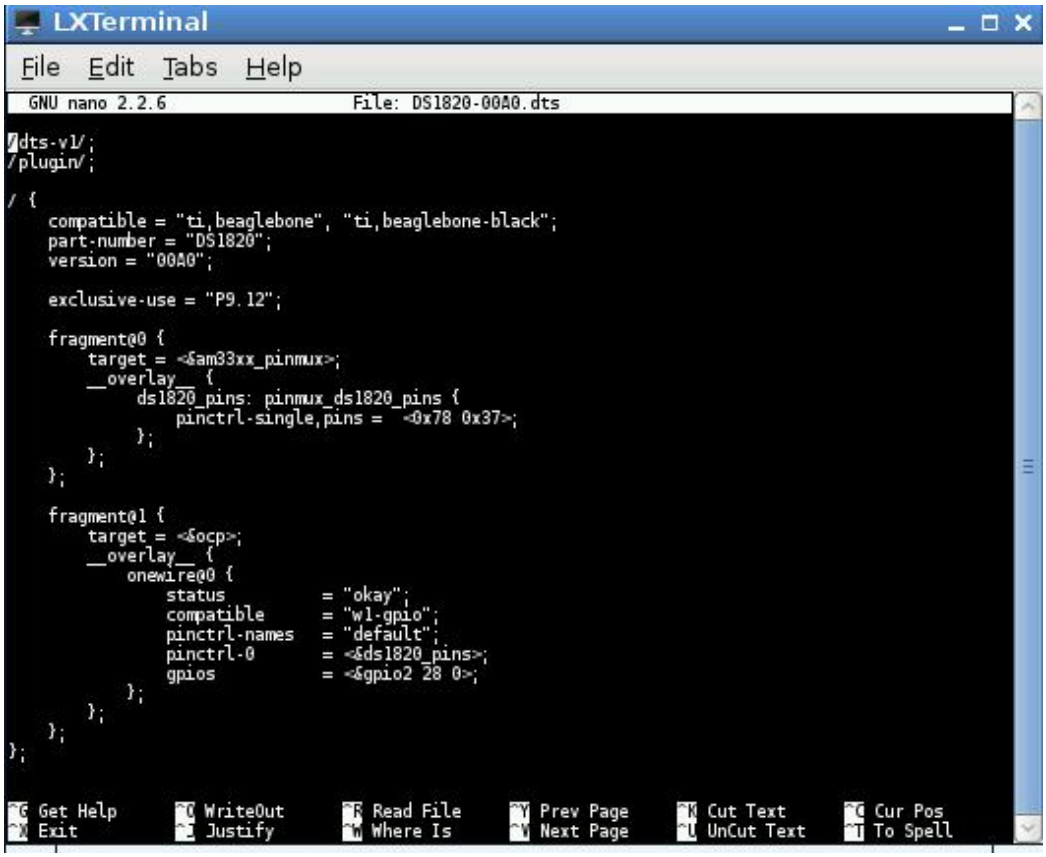

Figura 29 Archivo de Configuración del Sensor

Como se ve en la figura 29, al inicio del código se encuentra la línea compatible = “ti.beaglebone”, “ti,beaglebone-black” que permite a la tarjeta reconocer al sensor sin ningún inconveniente, también se encuentra la línea de código exclusive.use P9.12 que indica que dentro de la cabecera 9 se usará el pin 12 para el envió de los datos que proporcione el sensor.

Una vez activado el sensor para su reconocimiento ahora se debe verificar que esto suceda, para lo cual se debe ingresar en consola el código que se observa en la figura 30.

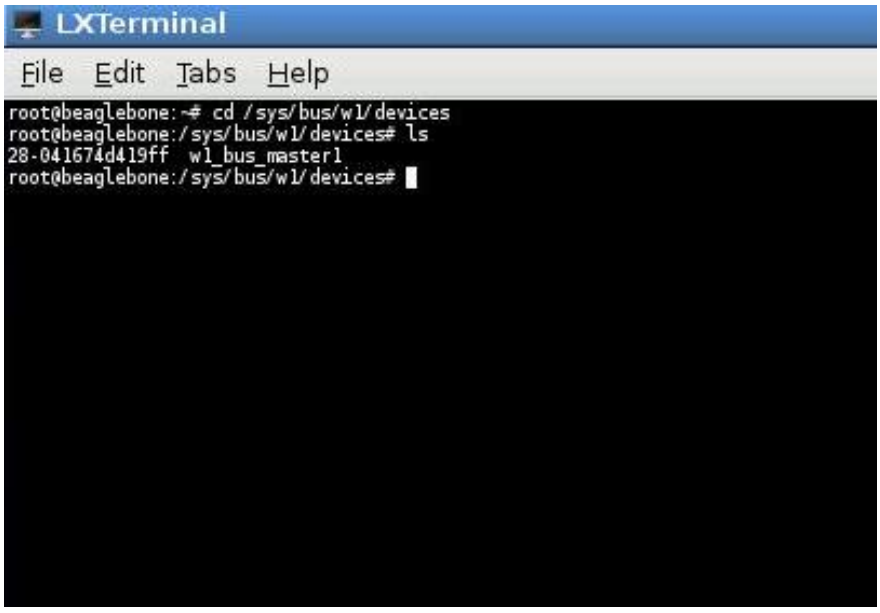

Figura 30 Verificación y Reconocimiento del sensor

Como se puede ver en la figura 30 con el código **cd/sys/bus/wl/devices** se cambia de directorio para realizar la búsqueda del sensor, luego se ingresa el comando **ls,** el mismo permite devolver la información que está dentro del directorio de la tarjeta, si las conexiones físicas del sensor a la tarjeta estuvieron bien hechas en el terminal o consola aparecerá una numeración como la siguiente **28 – 041674d419ff.**

Una vez verificado que el sensor fue reconocido por la tarjeta en el terminal se debe ingresar el código **dtc -0 dtb –o /lib/firmware/DS18B20-00A0.dtbo –b 0 - @ DS1820-00A0.dts,** este permite compilar la especificación del archivo que fue creado al inicio y luego de ello se ingresa el código **echo DS1820 > /sys/devices/bone_capemgr.*/slots**, para con ello finalmente concluir con la configuración del sensor.

3.4 Desarrollo de Algoritmo de Control PID

Dentro de la variedad de controles que existen, los controladores PID con mucha ventaja son los más utilizados en la industria de control de procesos, ya que por su sencillez, versatilidad y sobre todo por la capacidad de resolver problemas de complejidad básica dentro de los procesos, estos han sufrido cambios tecnológicos es decir desde los primeros controladores que fueron hechos a base de elementos neumáticos, se han venido dando cambios significativos pero sin perder la esencia del controlador.

Para el desarrollo de este algoritmo se tiene la necesidad de hacer uso de algunas librerías que harán la tarea más fácil para el programador, las mismas que se detallarán según la necesidad que tenga de usarlas.

Con lo primero que se debe comenzar es con la instalación de la librería Adafruit, que es la que permitirá la activación de los pines de funcionamiento de la señal de salida PWM de la tarjeta BeagleBone Black, importante para la excitación del relé de estado sólido el mismo que permite el control de la resistencia calefactora o niquelina.

3.4.1 Instalación de Librería Adafruit

Para la instalación de esta librería primero se debe conectar la tarjeta y posterior a ello se debe actualizar la fecha y la hora exacta, para lo que se ingresa en el terminal la siguiente línea de código:

sudo ntpdate pool.npt.org

Luego de esto se debe instalar las dependencias para lo cual se ingresa las siguientes líneas de código:

sudo apt-get update

sudo apt-get install build-essential python-dev-setuptools python-smbus-y

Ahora dependiendo la versión que se tenga instalado de Debian o Ubuntu es posible la necesidad de tener una versión de dtc. La versión parcheada de dtc que significa compilador de árbol de dispositivos, que incluye la capacidad de compilar superposiciones, ya que la biblioteca Adafruit_BBIO compila un conjunto de superposiciones para SPI y UART, ahora una vez se determine si se tiene la necesidad de instalar o no la versión parcheada de dtc, se ejecuta el comando para instalar la librería BBIO ingresando la siguiente línea de código:

sudo pip install Adafruit_BBIO

Ahora una función importante para el desarrollo de los controladores es la de PWM que se describe a continuación.

La función de PWM de la librería de Adafruit no resulta muy complicada de usarlo, a continuación para la configuración de un pin se debe escribir la siguiente línea de código:

import Adafruit_BBIO.PWM as PWM
PWM.start("P9_14", 50)

Los valores que están permitidos para el uso de PWM son de 0.0 a 100.0. El método de inicio activa a PWM en el canal referenciado. Por lo que no es necesario configurar los canales con Adafruit_BBIO.PWM. También se puede variar el ciclo de trabajo o la frecuencia según los requerimientos. Un ejemplo de esto son las dos líneas de código siguiente:

PWM.set_duty_cycle("P9_14", 25.5)
PWM.set_frequency("P9_14", 10)

Además también se puede desactivar un canal específico o limpiarlos todos cuando no se tenga la necesidad de usarlos.

3.4.2 Instalación de Librería Sleep

Esta es una función importante dentro del módulo de tiempo existente para Python que como su nombre lo indica, lo que hace es detener el programa por unos segundos, el tiempo de suspensión puede ser inferior al que se determine ya que cualquier señal que se detecte dará por terminado la ejecución de la rutina de la captura de la señal. Para la utilización de la misma, la sintaxis que se debe usar es la siguiente:

time.sleep(t)

Dentro del parámetro t se ingresa el número de segundos que se desea detener el programa, esta función no devuelve ningún valor. Un ejemplo de cómo usar es como las siguientes líneas de código:

```
import time
time.sleep(5)
```

Ahora se describirá la instalación de librerías necesarias para realizar los controladores y la interfaz gráfica.

3.4.3 Instalación Librería Tkinter

Tkinter pertenece la biblioteca gráfica Tcl/Tk utilizado en el lenguaje de programación Phyton y en otros lenguajes, con esta librería se conseguirán buenos resultados ya que se podrá realizar interfaces gráficas que interactúen con el usuario ayudando al fácil manejo de las diferentes aplicaciones que se hagan. Para el uso de Tkinter se debe realizar los siguientes pasos:

Se debe verificar la versión de Python que se esté usando, para lo cual se escribe en la consola el comando siguiente, como se observa en la figura 31:

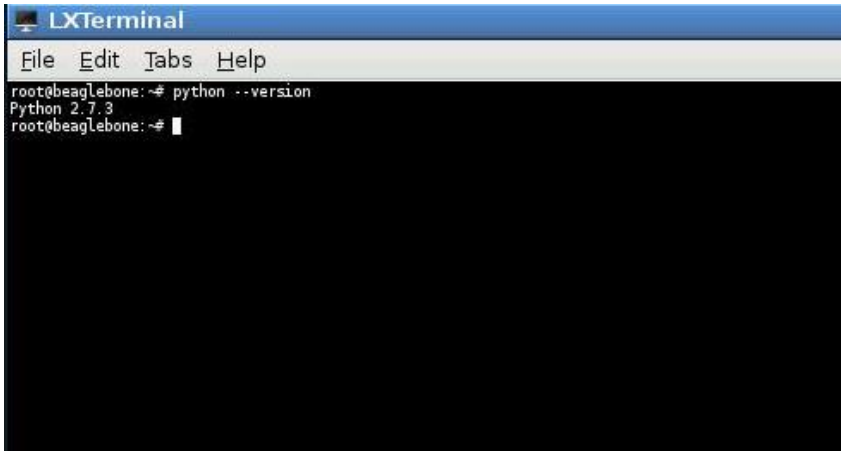

Figura 31 Verificación versión de Python

Esto devolverá la versión que esté instalada, si por algún motivo no estuvo instalado Python se lo puede hacer con el siguiente comando:

sudo apt-get install python

Ahora para la instalación de la librería se escribe la línea siguiente en el terminal:

sudo apt-get install python – tk

Una vez instalada la librería, lo que se debe hacer es importar el módulo, para lo cual se tiene dos alternativas.

La primera forma es con el código:

from Tkinter import*

La segunda forma es a través del código:

import Tkinter

Ahora la diferencia que existe en usar cualquiera de las opciones, es que si se usa "**import Tkinter**" cada momento que se use una función del módulo se debe anteponer la palabra 'Tkinter', y en la otra forma "**from Tkinter import*"** solo se debe escribir directo el nombre de la función sin necesidad de poner el nombre del módulo y además en esta forma se importa todas las funcionalidades del mismo, para quede claro se usa el ejemplo siguiente.

Primera forma:

```
import time
time.sllep(10)
```

Segunda forma:

```
from time import sleep
sleep(10)
```

Después de haber realizado todos los pasos descritos anteriormente, ahora se detallará el uso de las funciones de los diferentes elementos necesarios inmersos para la realización de una interfaz gráfica para el fácil manejo del sistema por parte de los usuarios.

Para la creación de una ventana se usa la función llamada **"ventana=Tkinter.Tk()",** con ello a la variable ventana se le atribuye las funciones de Tkinter, luego de esto para inicializar la ventana se usa "**ventana.mainloop()"** para con esto mostrar la misma en código queda de la siguiente manera:

```
from Tkinter import*
ventana=Tkinter.Tk()
ventana.mainloop()
```

Después de haber creado la ventana se le puede cambiar el título de la misma con la función **title(),** quedando en el código de la siguiente manera y el resultado como se lo observa en la figura 33:

```
from Tkinter import*
ventana=Tkinter.Tk()
ventana.title('Ventana Principal')
ventana.mainloop()
```

3.4.3.1 Etiquetas

Para el uso de etiquetas a través de Tkinter se lo hace con la función **Label (ventana,text)**, en esta función el primer parámetro es la instancia de la ventana lo que quiere decir el resultado de ventana=Tkinter.tk(), y a la variable se le debe asignar una valor tipo cadena entre comillas; luego para posicionar la etiqueta se usa la función place(x='', y="") que está delimitada según las dimensiones de la ventana definiendo su posición en los ejes horizontal y vertical. En código quedaría de la siguiente manera:

```
from Tkinter import*
ventana=Tkinter.Tk()
ventana.title('VENTANA PRINCIPAL')
etiqueta1=Label(ventana,text="etiqueta").place(x=0 , y=50)
ventana.mainloop()
```

3.4.3.2 Botones

Para el uso de botones se usa la función **Button(ventana,text="Nombre del botón")** y se escriben los parámetros usados anteriormente para agregar una etiqueta, es decir tanto la instancia del botón como el valor del texto del

mismo, pero en un botón también se puede cambiar otros aspectos para cambiar su apariencia, para ello se tiene el código siguiente:

```
from Tkinter import*
ventana=Tkinter.Tk()
def boton():
    print “Boton”
ventana.title(‘VENTANA PRINCIPAL’)
ventana.geomtery(“300x300”)
etiqueta1=Label(ventana,text=”etiqueta”).place(x=0 , y=50)
botón= Button(ventana,text=”Boton”,command=botón).place(x=80, y=50)
ventana.mainloop()
```

3.4.4 Instalación Librería Numpy

Numpy es una herramienta esencial dentro del lenguaje de programación Python, ya que con ésta se tiene mayor versatilidad para el manejo de vectores y matrices convirtiéndose en una biblioteca fundamental en el tratamiento de funciones matemáticas de alto nivel.

Para la instalación de esta librería se ingresa el código siguiente en consola:

sudo pip install numpy

3.4.5 Instalación Librería Matplotlib

Matplotlib es una biblioteca de funciones muy utilizada para la realizar trabajos en 2D y 3D dentro del lenguaje de programación python, se pueden realizar gráficas, gráficos de barras, histogramas, diagramas de dispersión entre otras más, es decir se la usa para la publicación de datos con figuras de calidad en diferentes formatos.

Esta librería está compuesta por tres elementos que son:

- PYLAB el mismo que está compuesto por un conjunto de funciones que son las que permiten realizar gráficas es decir plot's semejantes a los de Matlab.

- MATPLOTLIB que contiene una interfaz con clases con las cuales se puede crear y gestionar gráficos, imágenes, textos y plot's.

- BACKEND con esta se podrá realizar funciones extra como archivos PNG, PDF, etc.

Para la instalación de esta librería se ingresa en consola el código que se muestra a continuación:

sudo pip install matplotlib

3.4.6 Sintonización Controlador PID

Para la sintonización de este controlador se utilizó el método de Ziegler - Nichols el mismo que consiste en:

Fijar un $T_i = \infty$ y $T_d = 0$ usando únicamente la acción proporcional como indica la figura 32, se incrementa K_p desde cero hasta un valor critico k_{cr}, para que se genere en la salida oscilaciones sostenidas, algo que se debe tomar en cuenta es que si para cualquier valor de K_p en la salida no se presenta las oscilaciones sostenidas este método no se lo puede aplicar, luego de tener las oscilaciones. De esta manera la ganancia crítica K_{cr} y el período P_{cr} correspondiente se determinan de forma experimental como se puede observar en la figura 33.

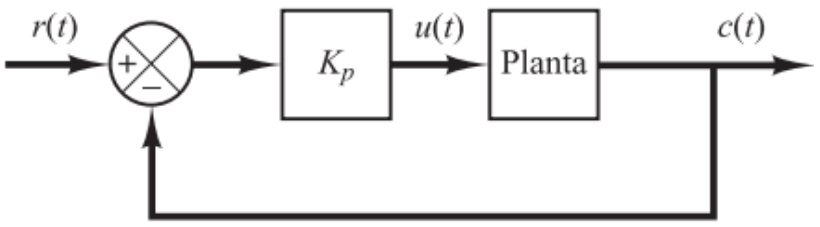

Figura 32 Sistema de Lazo Cerrado con Ganancia proporcional

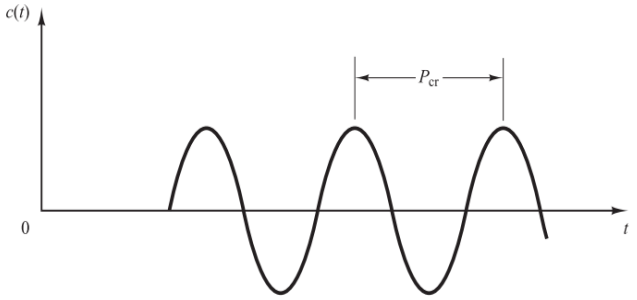

Figura 33 Oscilaciones Sostenidas con periodo P_{cr} en segundos

Ahora para continuar con este método, Ziegler – Nichols determinaron que se establecieran los valores de los parámetros K_p , T_i y T_d de acuerdo a la fórmula que se muestra en la siguiente tabla:

Tabla 4

Regla de Sintonía de Ziegler – Nichols basada en Ganancia Critica K_cr y Período Critico P_cr

Tipo de Controlador	K_P	T_i	T_d
P	0.5 K_{cr}	∞	0
PI	0.45 K_{cr}	$\frac{1}{1.2} P_{cr}$	0
PID	0.6 K_{cr}	0.5 P_{cr}	0.125 P_{cr}

Luego de haber explicado cómo se realizó la sintonización del controlador se estableció como constantes los valores presentados en la Tabla 3.

Tabla 5

Constantes Controlador PID

Constante	Valor	Dimensión
KP	1.5	[adim]
Ti	0.09	[min]
Td	0.03	[min]

3.4.7 Interfaz Gráfica PID

Para poder observar los resultados se diseñó la interfaz gráfica que se muestra a continuación en la figura 34 y 35 y el sistema funcionando en la figura 36.

Figura 34 Interfaz Gráfica Control PID

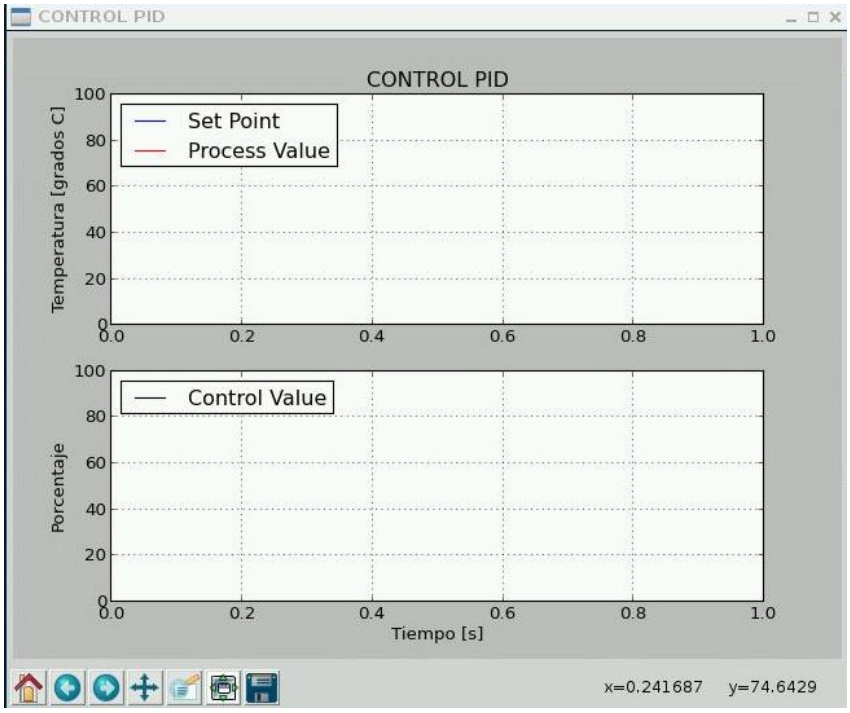

Figura 35 Interfaz Tendencias del PID

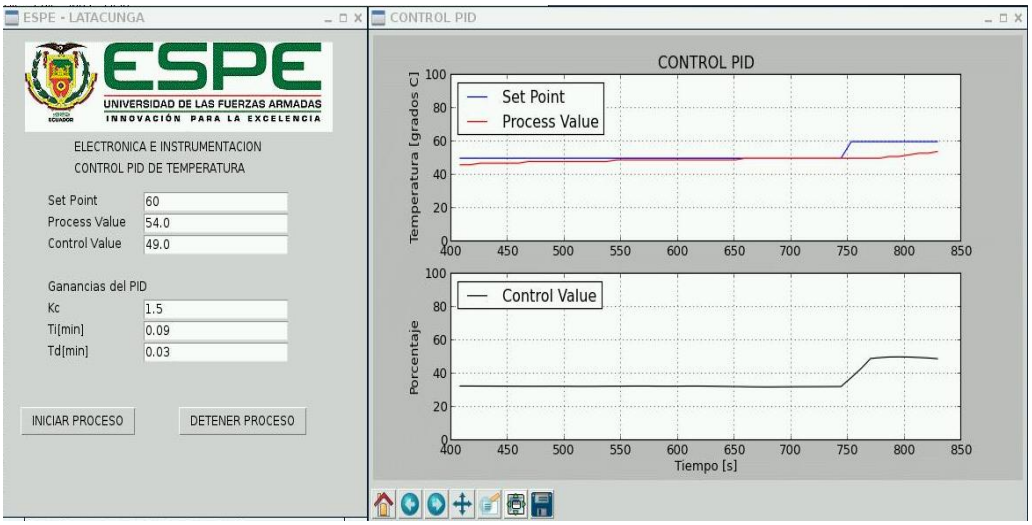

Figura 36 HMI Controlador PID

A continuación en la figura 37 se presenta un diagrama de flujo del desarrollo del algoritmo de control PID:

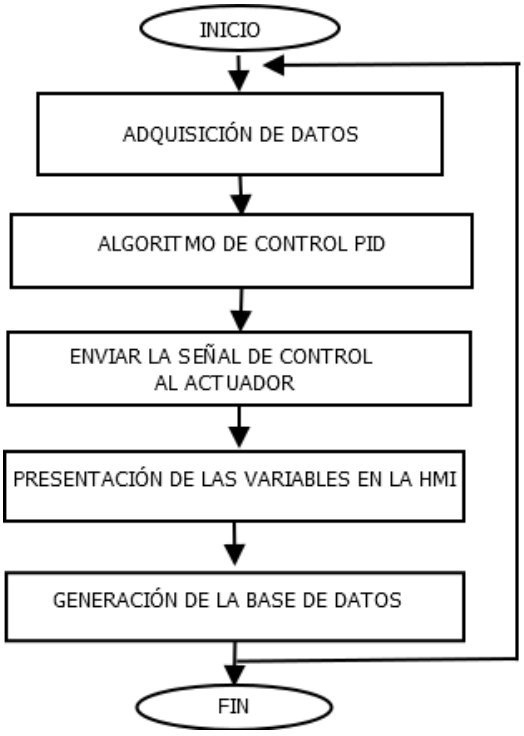

Figura 37 Diagrama de Flujo Algoritmo PID

3.5 Desarrollo Algoritmo de Control Fuzzy Logic

Para el desarrollo del algoritmo de control Fuzzy Logic se tiene la necesidad de instalar algunas librerías que permitirán implementar sin inconvenientes este control las mismas que son:

- Scikit – Fuzzy
- Matplotlib
- Tkinter
- Adafruit
- Sleep

Ahora de la lista anterior queda explicar cómo se instala la librería Scikit – Fuzzy, ya que el resto se explicó en el desarrollo del algoritmo de control PID en

el que él también fue necesario utilizar funciones, entonces a continuación se detalla el proceso de instalación de la librería Scikit – Fuzzy.

3.5.1 Librería Scikit – Fuzzy

Scikit Fuzzy o conocido como skfuzzy es un importante conjunto de algoritmos que tienen un papel fundamental dentro de la lógica difusa, esta funcionalidad es realizada por la comunidad SciPy, que direccionan su investigación para el lenguaje de programación Pyhton, es por esta razón que se va a usar este paquete para el desarrollo del algoritmo de control.

Para la instalación de Scikit-Fuzzy, primero se debe conocer acerca de **PyPi** que significa índice para paquetes de python, es decir es el repositorio oficial de paquetes en código abierto para el desarrollo de aplicaciones en el lenguaje de programación Python; para acceder a esta página lo puede hacer a través del siguiente link:

https://pypi.python.org/pypi

Ahora bien dentro PyPi se encuentra el paquete de scikit - fuzzy con extensión .tar.gz, y además en las especificaciones del mismo indica que se debe tener como requerimiento instalado antes la librería Numpy con versión mayor o igual a 1.6.0, además, scipy con versión mayor o igual a 0.9.0 y finalmente NetworkX con versión mayor o igual a 1.9.0 para no tener ningún inconveniente en la instalación. Para la descarga de scikit – fuzzy se lo puede hacer desde el siguiente link:

https://pypi.python.org/pypi/scikit-fuzzy

Para la instalación de SciPy se lo puede hacer a través de **pip** que es la herramienta recomendada por PyPi para la instalación de los paquetes de

pyhton. Para poder hacer uso de esta función se la debe instalar y se lo hace a través de la consola con la línea de código siguiente:

sudo apt – get install python - pip

Ahora para instalar SciPy que es una biblioteca de funciones de fuente abierta que específica su trabajo en procesamiento de señales y de imagen, interpolación, optimización, algebra lineal de entre otras aplicaciones, de ahí su uso y su importancia para el desarrollo de la investigación dentro del software libre. Para la instalación de SciPy se usa la siguiente línea de código:

sudo pip install scipy

Luego de esto se procede a instalar la librería Numpy que es uno de los tres requisitos que se necesarios para la instalación de skfuzzy para lo cual se usa el código como la figura 53:

sudo pip install numpy

Y por último se instala el paquete NetworkX para lo cual se usa la siguiente line de código como la figura 54:

sudo pip install NetworkX

Una vez instalado los tres requisitos no se deberá tener ningún inconveniente para instalar y hacer uso de skfuzzy y desarrollar cualquier aplicación según las necesidades.

Una vez instalada la librería ahora se detallará el desarrollo del algoritmo, en el inicio del mismo se debe importar las librerías que se van a utilizar.

Una forma de importar la librería es como en la línea de código siguiente:

import numpy as np

En esta línea se define que se va a importar la librería con la palabra **“import”** mientras que **“as”** se utiliza para dar un nombre corto dentro del programa para facilidad del programador en este caso se usó **np.**

Otra forma de importar una librería es como se muestra a continuación:

from skfuzzy import control as ctrl

En esta forma está definido que con la palabra **“from”** especifica que desde la librería skfuzzy se importará la función **control** y de la misma manera que antes se le define con un nombre corto o alias por medio de la palabra **“as”**, en este caso **ctrl,** esta función será usada ya que dentro de sistemas de control difusos tienen la tarea de mantener una variable cerca de un valor específico, por lo que para lograr este objetivo se debe tener en cuenta dos puntos importantes para la consecución del control como lo es el error y la derivada del error, y también la forma en que el error está cambiando a la cual se la llamará la primera derivada matemática dentro del proceso y a la cual se la denominará delta del error.

Ahora luego de definir todas las librerías que se van a usar, lo siguiente para el control es definir el dominio de trabajo que dentro del lenguaje python se lo denomina universo, para ello se lo hace de la siguiente manera:

#Universe

x_error = np.arange(0, 11, 1)

x_delta = np.arange(0, 11, 1)

x_salida= np.arange (0, 26, 1)

En estas líneas se utiliza **np.arange** el mismo que permite construir un vector dentro del cual esta una progresión aritmética, el primer parámetro del argumento indica el primer valor la secuencia, el segundo parámetro es el de limitación es decir el ultimo valor de la secuencia nunca será mayor o igual a este valor, y el ultimo parámetro del argumento indica la diferencia común entre los valores sucesivos del vector, el valor por defecto es 1, y de ahí se parte para la creación de los conjuntos difusos del controlador.

Posterior a esto se debe crear las variables de las entradas y las salidas como por ejemplo las líneas siguientes:

```
error  = ctrl.Antecedent(x_error, 'error')
delta  = ctrl.Antecedent(x_delta,'delta')
salida = ctrl.Consequent(x_salida, 'salida')
```

En estas líneas se usa **ctrl.Antecedent** que es usado pasa definir las variables fuzzy en este caso las entradas, y también se usa **ctrl.Consequent** que es usado para definir las salidas dentro del controlador.

Una vez definido tanto las entradas como las salidas se debe definir los conjuntos del controlador, primero para el error, para lo cual se utilizara líneas de código como las siguientes:

```
error_pobre = fuzz.trimf(x_error, [0, 0, 5])
error_decen = fuzz.trimf(x_error, [0, 5, 10])
error_great = fuzz.trimf(x_error, [5, 10, 10])
```

Para realizar los conjuntos se usó **fuzz.trimx,** el mismo que permite realizar un conjunto difuso en forma de triángulo y dentro del argumento de esta función se limita los valores o puntos de inicio y fin para este caso de los triángulos. Posterior a esto se realizó una reasignación de variables que serán usadas a futuro para hacer las gráficas y las reglas del controlador, como se lo puede ver en las líneas siguientes:

```
error['pobre'] = error_pobre
error['decen'] = error_decen
```

error['great'] = error_great

Luego de esto se debe nuevamente definir los conjuntos, pero ahora para la derivada de error de forma similar como las del error como se ve en las líneas siguientes:

delta_mala = fuzz.trimf(x_delta, [0, 0, 5])

delta_acep = fuzz.trimf(x_delta, [0, 5, 10])

delta_amaz = fuzz.trimf(x_delta, [5, 10, 10])

Y luego se debe otra vez realizar la reasignación de variables para las gráficas y las reglas del controlador como se lo hizo con el error, quedando de forma similar como las líneas siguientes:

delta['mala'] = delta_mala

delta['acep'] = delta_acep

delta['amaz'] = delta_amaz

Por último se realiza tanto la definición de los conjuntos como la reasignación de variables para la salida de forma similar al error y la derivada, como se observa en las líneas de código siguientes:

salida_baja = fuzz.trimf(x_salida, [0, 0, 13])

salida_medi = fuzz.trimf(x_salida, [0, 13, 25])

salida_alta = fuzz.trimf(x_salida, [13, 25, 25])

salida['baja'] = salida_baja

salida['medi'] = salida_medi

salida['alta'] = salida_alta

Ahora bien una vez definido los conjuntos y la reasignación de variables, se realizó la gráfica de los conjuntos por medio de las siguientes líneas:

```
fig, (ax0, ax1, ax2) = plt.subplots(nrows=3)

ax0.plot(x_error, error_pobre, 'b', linewidth=1.5, label='pobre')

ax0.plot(x_error, error_decen, 'g', linewidth=1.5, label='decen')

ax0.plot(x_error, error_great, 'r', linewidth=1.5, label='Great')

ax0.set_title('error')

ax0.legend()
```

En la primera línea de define el número de gráficas que se va hacer con **fig, (ax0, ax1, ax2) = plt.subplots(nrows=3)** en la que se indica que se tendrán tres graficos.

Luego en la línea **ax0.plot(x_error, error_pobre, 'b', linewidth=1.5, label='pobre')** se define como primer parámetro que se graficará del error, luego como antes se hizo una reasignación de variables que se dijo que utilizará posteriormente se lo usa aquí como un parámetro para la gráfica, el tercer parámetro la letra **“b”** indica el color de la línea en esta casi azul, el siguiente parámetro linewidth define el grosor de la línea y por último, esta la etiqueta; y en las líneas siguientes se hace lo mismo pero con las otras subdivisiones del error.

En línea siguiente se usa **ax0.set_title('error')** que sirve para dar nombre al conjunto, y se desea que aparezca la misma se usa la última línea **ax0.legend()**

Luego de esto se debe hacer lo más importante de los controladores fuzzy logic como lo son la base de reglas, la misma que comprende de entradas o conocidos como antecedentes como se los había denominado al inicio del

algoritmo con estas se realiza el cálculo necesario para poder determinar una salida o conocido como consecuente en este controlador.

Dentro de la base de reglas es importante que se tengan algunas propiedades como la integridad ya que con cualquier combinación en la entrada se tiene un valor apropiado en la salida, otro punto importante es la consistencia ya que un conjunto de reglas no debe tener contradicciones entre las entradas o consecuentes para no tener problemas para que el controlador discierna con las salidas. Entonces dentro del algoritmo queda como las siguientes líneas:

```
rule1 = ctrl.Rule(error['pobre'] | delta['mala'], salida['baja'])
rule2 = ctrl.Rule(error['decen'] | delta['acep'], salida['medi'])
rule3 = ctrl.Rule(error['great'] | delta['amaz'], salida['alta'])
```

En estas líneas se usa **ctrl.Rule** que es una función que permite definir las reglas del controlador, es decir la relación difusa, como en las reglas anteriores se tiene dos entradas o antecedentes para con ellas definir la salida o consecuentes.

Ahora bien una vez creadas las reglas definidas, el siguiente paso es crear un sistema de control el mismo que se lo hace a través de:

```
control = ctrl.ControlSystem([rule1, rule2, rule3])
```

Dentro del argumento de control se debe incluir todas las reglas que se hicieron previamente. Ahora para la simulación de este sistema de control, se crea ControlSystemSimulation y en línea de código queda de la siguiente manera:

controlar = ctrl.ControlSystemSimulation(control)

Ahora se puede simular el sistema de control, lo que se debe hacer es simplemente especificar las entradas y llamar al método de cálculo. Un ejemplo de ello es como las líneas siguientes de código:

controlar.input['error'] = 6.5

controlar.input['delta'] = 9.8

Posterior a esto se realiza el cáculo del valor de salida a través de la función compute, quedado en línea de código asi:

controlar.compute ()

Y finalmente si se quiere imprimir el valor del cálculo es decir el valor de salida se usa la siguiente línea de código:

print controlar.output['salida']

3.5.2 Definición Variables Lingüísticas del Controlador Fuzzy

Luego de haber explicado el diseño del controlador fuzzy se indica lo utilizado en el sistema de monitoreo y control de la temperatura de flujo de aire se definieron las siguientes variables lingüísticas:

- **Error**
- **Delta Error**
- **Salida**

Para la variable lingüística del error se definieron los siguientes nombres para los conjuntos:

- **Error Positivo**
- **Error Cero**
- **Error Negativo**

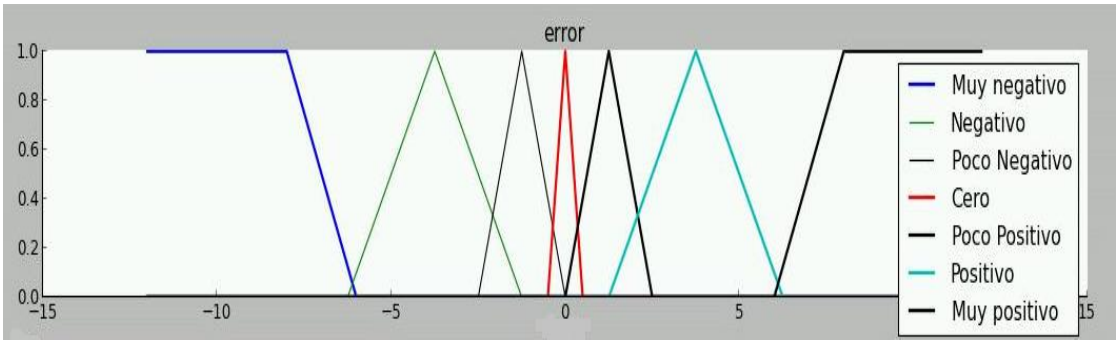

Figura 38 Conjuntos Difusos Error

Para la variable lingüística delta error se definieron los siguientes nombres para los conjuntos:

- **Delta Muy Positivo**
- **Delta Positivo**
- **Delta Cero**
- **Delta Negativo**
- **Delta Muy Negativo**

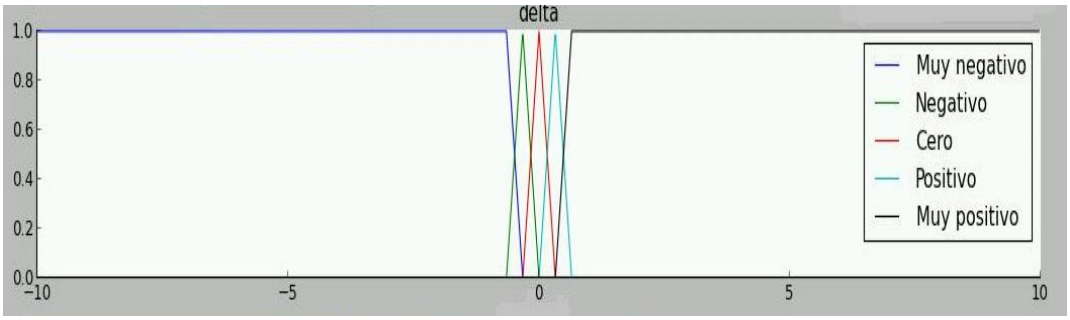

Figura 39 Conjuntos Difusos Delta Error

Para la variable lingüística delta error se definieron los siguientes nombres para los conjuntos:

- **Muy Positiva**
- **Bastante Positiva**
- **Algo Positiva**
- **Poco Positiva**
- **Normal**
- **Poco Negativa**
- **Algo Negativa**
- **Bastante Negativa**
- **Muy Negativa**

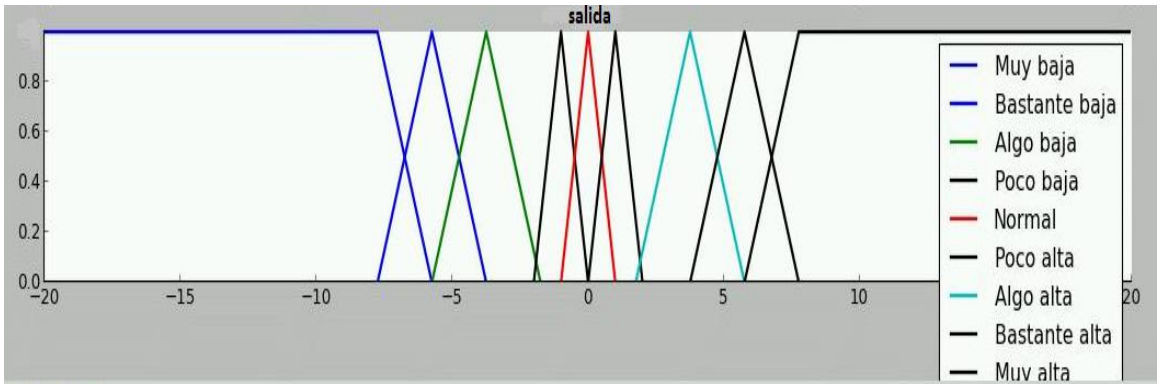

Figura 40 Conjuntos Difusos Salida

3.5.3 Definición de Reglas Controlador Fuzzy

En la siguiente tabla se muestra las reglas implementadas para este controlador:

Tabla 6

Reglas del Controlador Fuzzy

		Derivada del Error				
		Muy Negativo	**Negativo**	**Cero**	**Positivo**	**Muy Positivo**
Error	**Muy Negativo**	Muy Baja	Bastante Baja	Poco Baja	Normal	Normal
	Negativo	Bastante Baja	Algo Baja	Poco Baja	Normal	Normal
	Poco Negativo	Algo Baja	Poco Baja	Normal	Poco Alta	Poco Alta
	Cero	Normal	Normal	Normal	Normal	Normal
	Poco Positivo	Poco Baja	Poco Baja	Normal	Poco Alta	Algo Alta
	Positivo	Normal	Normal	Poco Alta	Algo Alta	Bastante Alta
	Muy Positivo	Normal	Normal	Poco Alta	Bastante Alta	Muy Alta

3.5.3 Interfaz Gráfica Controlador Fuzzy

Para observar los resultados se diseñó la interfaz gráfica que se muestra a continuación en la figura 41 y 42, en la figura 43 se muestran los conjuntos difusos y en la figura 44 se muestra el sistema funcionando.

Figura 41 Interfaz Gráfica Controlador Fuzzy

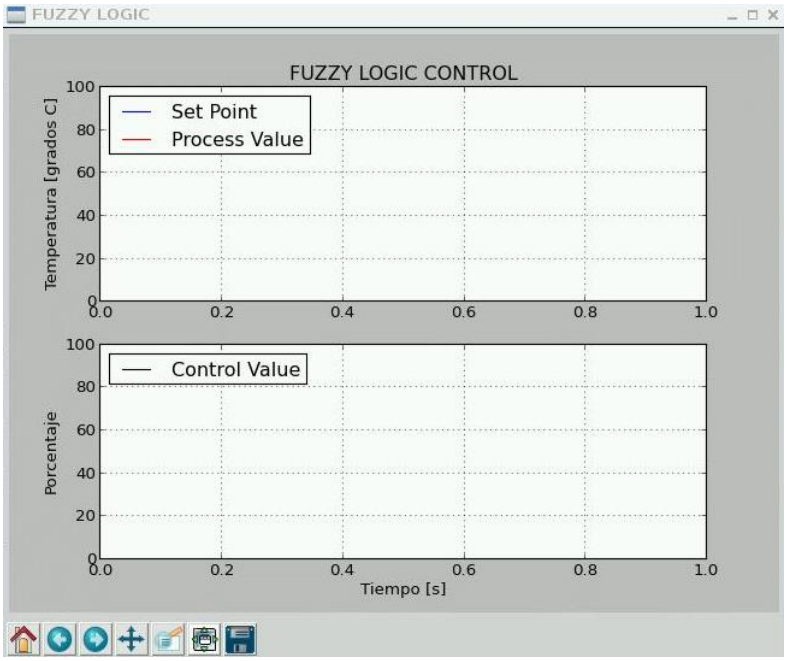

Figura 42 Interfaz Tendencias de las Variables Control Fuzzy

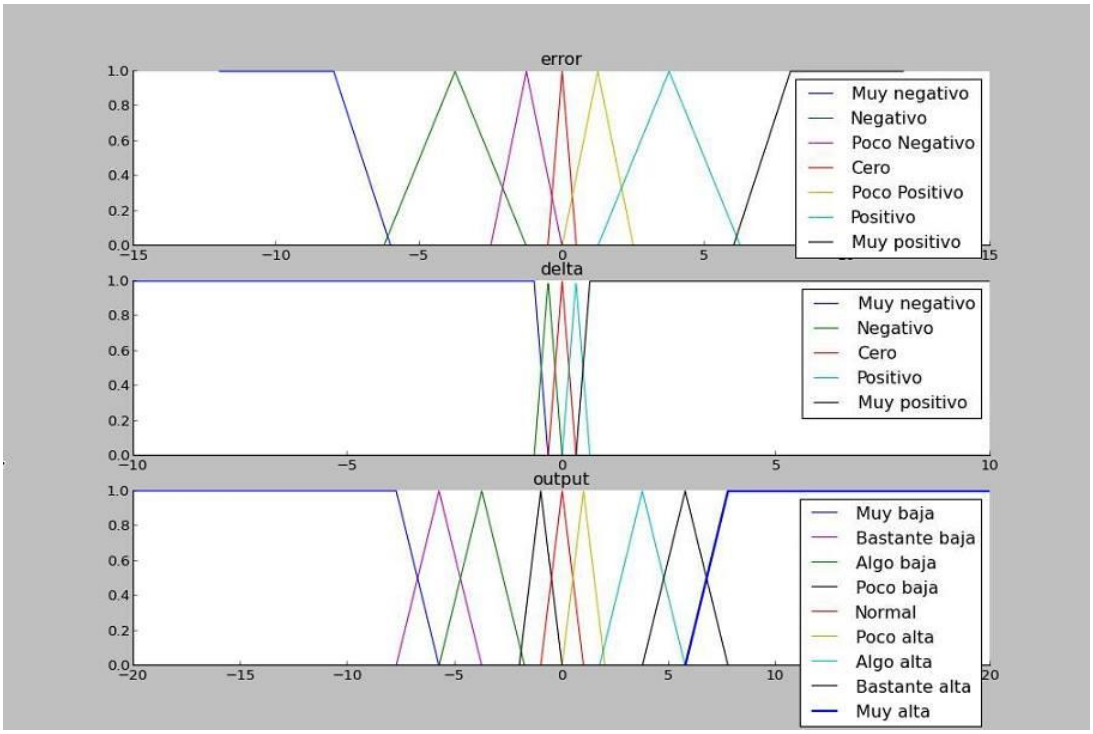

Figura 43 Conjuntos Difusos del Controlador Difuso

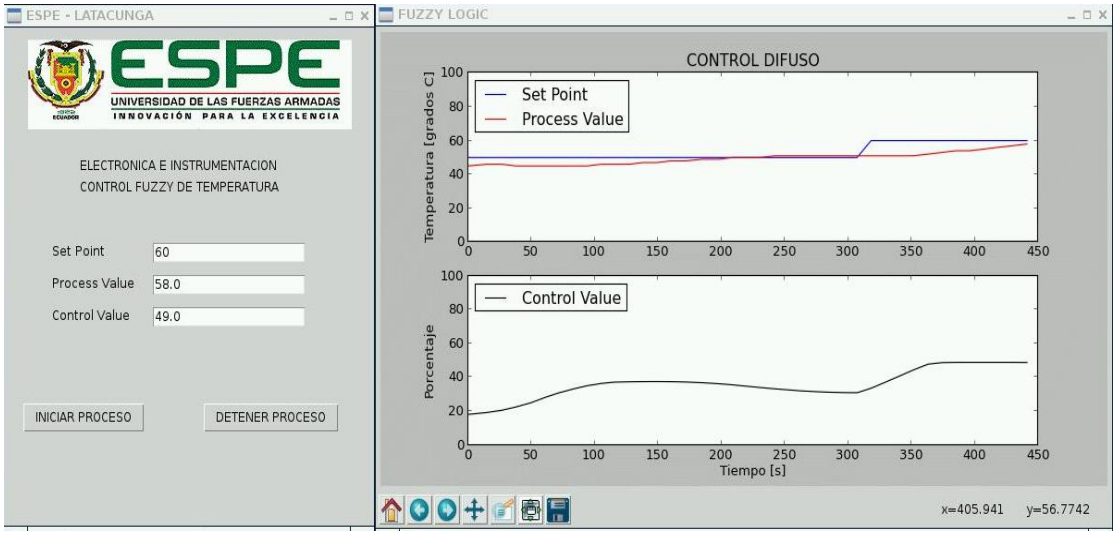

Figura 44 HMI Controlador Difuso

A continuación en la figura 45 se presenta un diagrama de flujo del desarrollo del algoritmo de control Difuso:

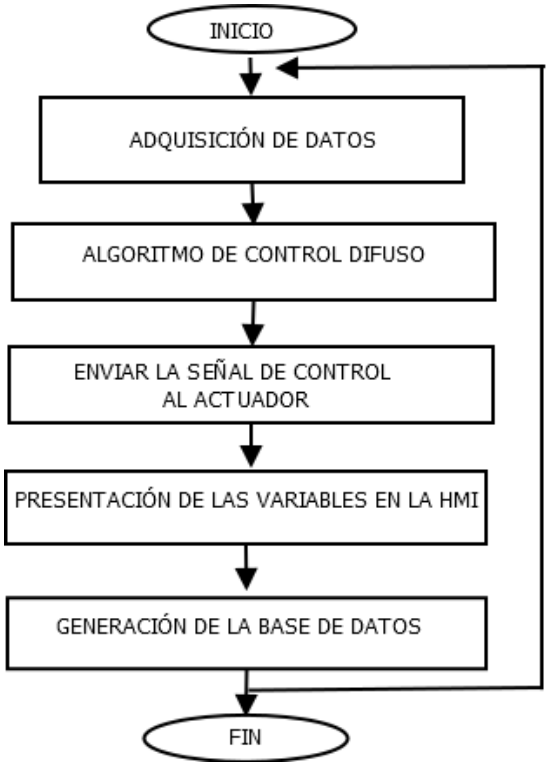

Figura 45 Diagrama de Flujo Control Difuso

CAPÍTULO IV

RESULTADOS DE LA INVESTIGACIÓN

4.1 Generalidades

En este capítulo se describen las pruebas realizadas para verificar el correcto funcionamiento de los algoritmos de control realizados en la programación PID y Fuzzy para el sistema de monitoreo y control de la temperatura de flujo de aire realizados en la tarjeta Beaglebone Black con el análisis de los resultados y posterior a ello se contrasta los resultados obtenidos con los controladores desarrollados en Labview.

4.2 Pruebas realizadas al sistema de monitoreo y control.

Controlador PID

En las figuras que se muestran a continuación se presentan los resultados de los controladores implementados en el sistema de monitoreo de temperatura de flujo de aire.

En la figura 46 se muestra las curvas de respuesta del Process Value del controlador PID con diferentes Set Point.

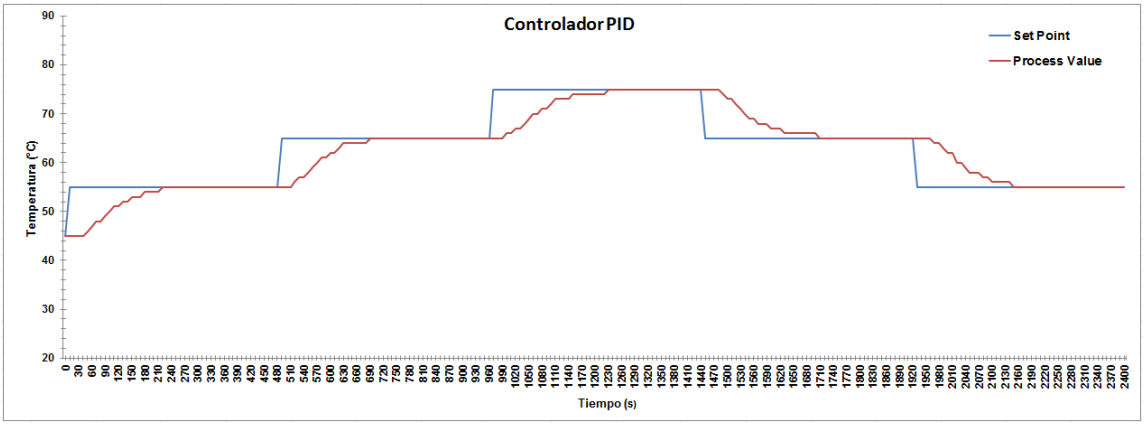

Figura 46 Curvas de Respuesta Process Value Control PID

En la figura 47 se muestra las curvas de respuesta del Control Value del controlador PID con diferentes Set Point.

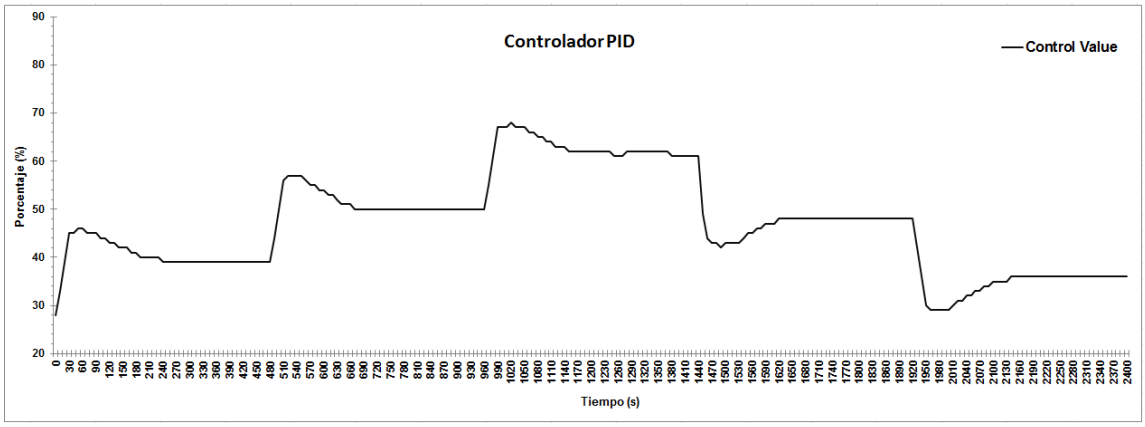

Figura 47 Curvas de Respuesta Control Value Control PID

Controlador Difuso

En la figura 48 se muestra las curvas de respuesta del process value del controlador Difuso con diferentes Set Point.

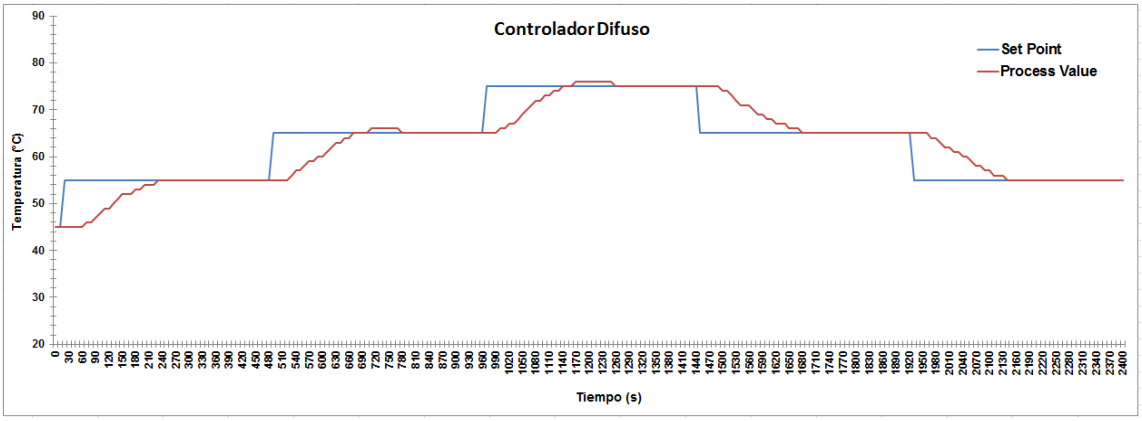

Figura 48 Curvas de Respuesta Process Value Control Difuso

En la figura 49 se muestra las curvas de respuesta del Control Value del controlador Difuso con diferentes Set Point.

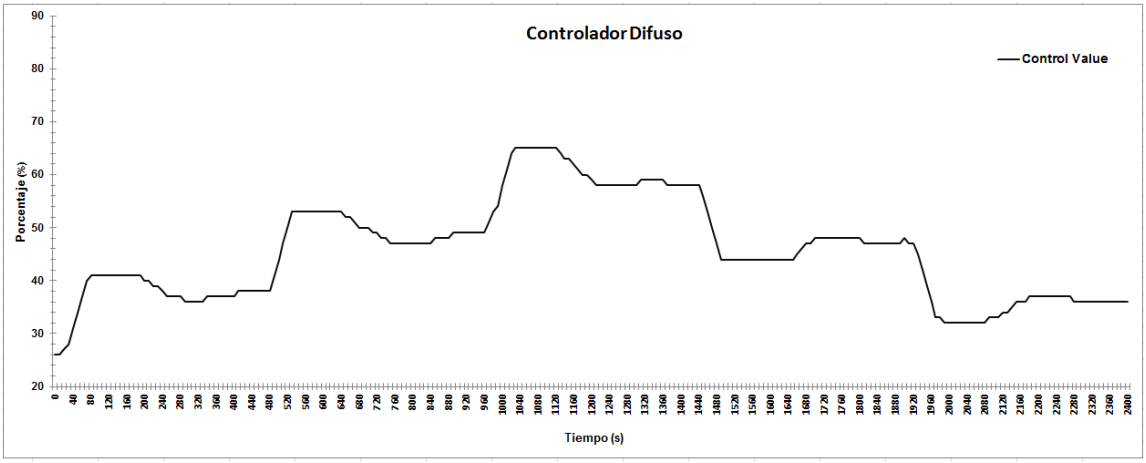

Figura 49 Curvas de Respuesta Control Value Control Difuso

Comparaciones Controlador PID vs Difuso

A continuación en la figura 50 se presenta la comparación del comportamiento de los Process Value del controlador PID y el controlador Difuso

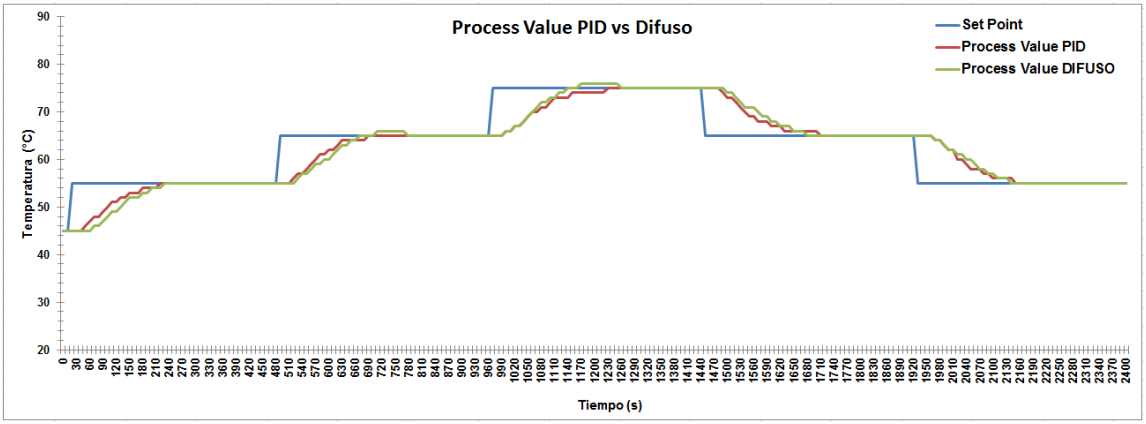

Figura 50 Comparación Process Value PID vs Difuso

A continuación en la figura 51 se presenta la comparación del comportamiento de los Control Value del controlador PID y el controlador Difuso

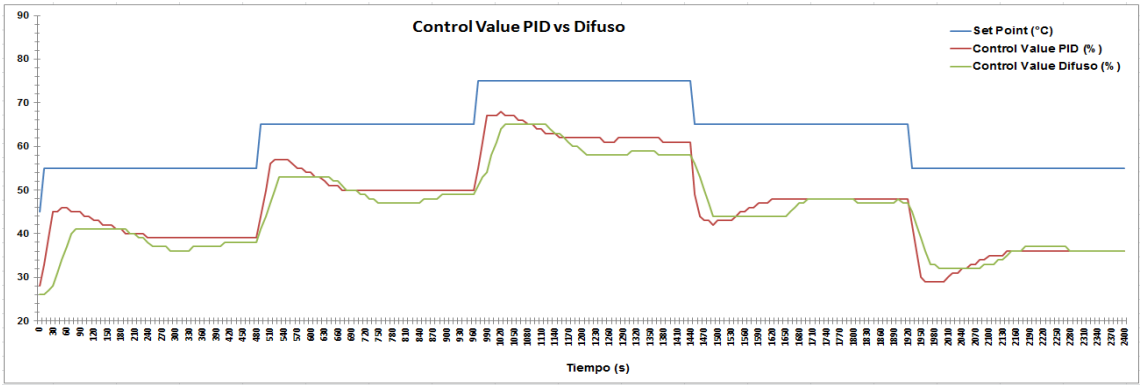

Figura 51 Comparación Control Value PID vs Difuso

En la figura 52 se muestra una comparación del comportamiento de un escalón de los Process Value del Controlador PID vs el Difuso

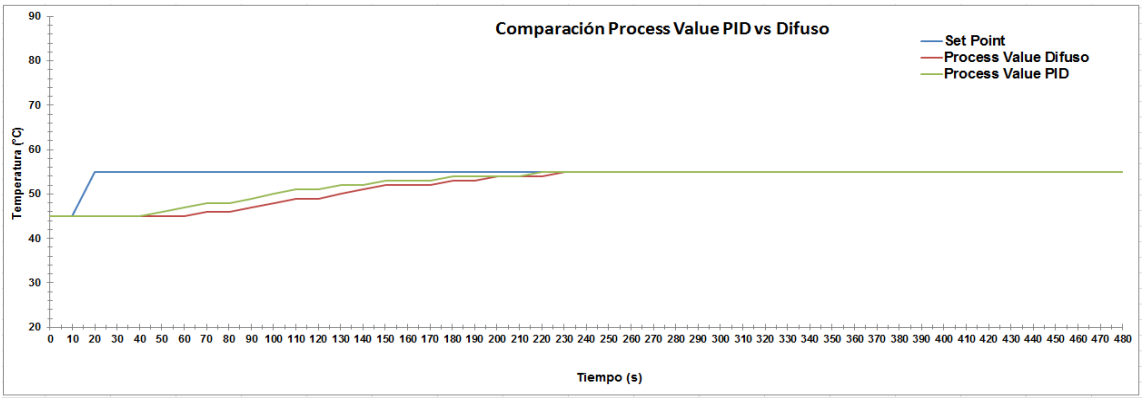

Figura 52 Comparación Escalón Process Value PID vs Difuso

Tabla 7

Comparativa de parámetros Control PID vs Difuso

PARAMETROS	PID	FUZZY
Sobre impulso	0%	0%
Tiempo de Retardo	30 seg	50 seg
Tiempo de Subida	180 seg	170 seg
Tiempo de Establecimiento	210 seg	230 seg
Error	0%	0%

De acuerdo a los datos presentados en la tabla 5 el controlador PID es más eficiente que el Difuso, debido a la poca capacidad de procesamiento de la tarjeta, ya que si se aumentan más reglas al algoritmo de control difuso, se demora más tiempo e incluso se pierde el control, lo que se comprobó en las diferentes pruebas que se realizó para el desarrollo de esta investigación.

4.3 Evaluación de los Controladores

Luego de haber realizado los controladores en la tarjeta de desarrollo de la investigación para evaluar los mismos se realizaron los controladores en el software Labview a continuación se muestran los resultados.

En la figura 53 se muestra las curvas de respuesta del Process Value del controlador PID con diferentes Set Point realizado en Labview.

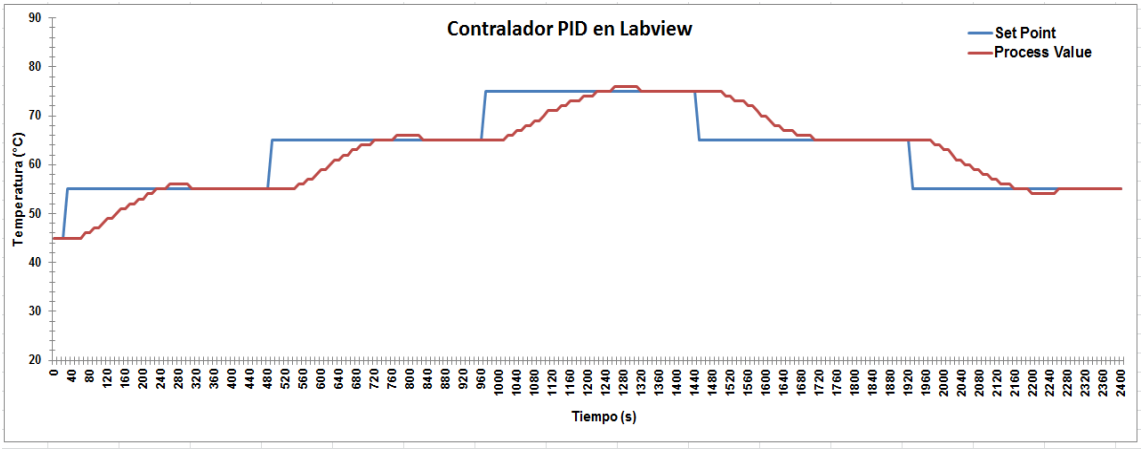

Figura 53 Curvas de Respuesta Process Value Control PID Labview

En la figura 54 se muestra las curvas de respuesta del Control Value del controlador PID con diferentes Set Point realizado en Labview.

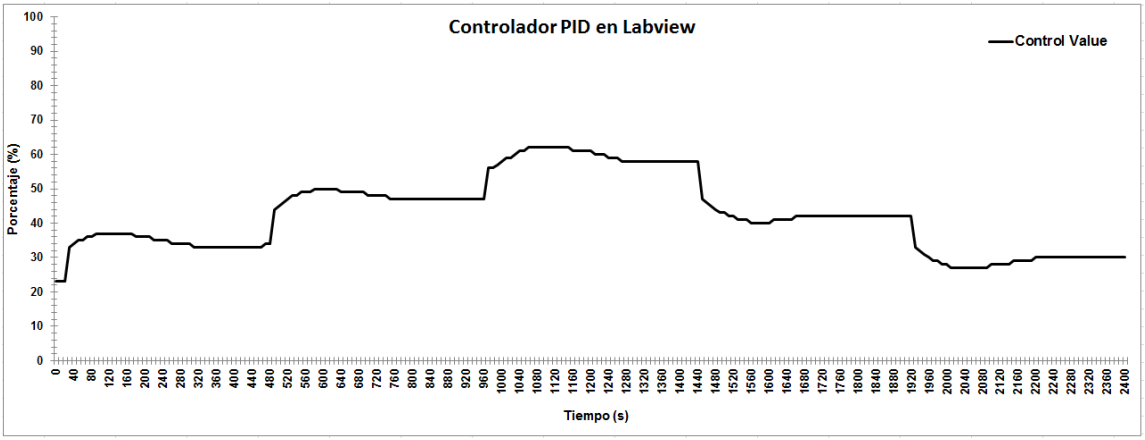

Figura 54 Curvas de Respuesta Control Value Control PID Labview

En la figura 55 se muestra las curvas de respuesta del Process Value del controlador Difuso con diferentes Set Point realizado en Labview.

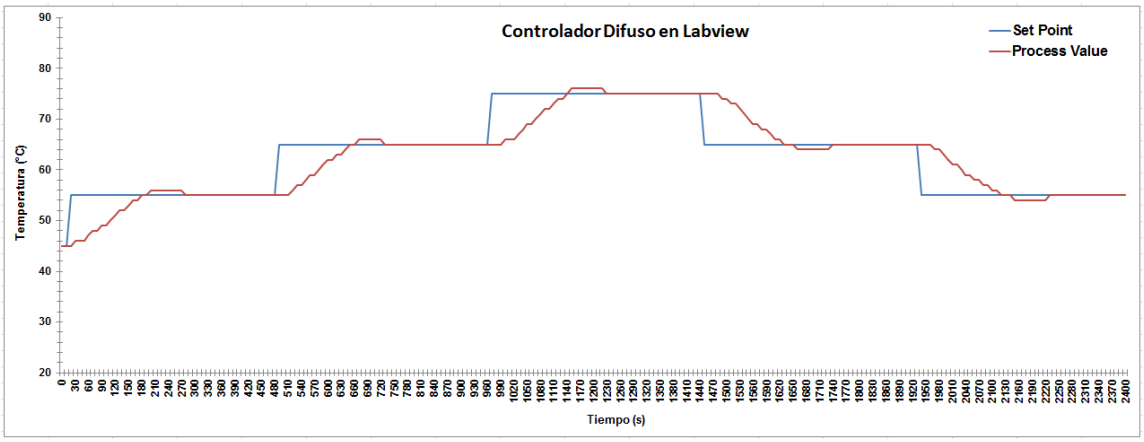

Figura 55 Curvas de Respuesta Process Value Control Difuso Labview

En la figura 56 se muestra las curvas de respuesta del Control Value del controlador Difuso con diferentes Set Point realizado en Labview.

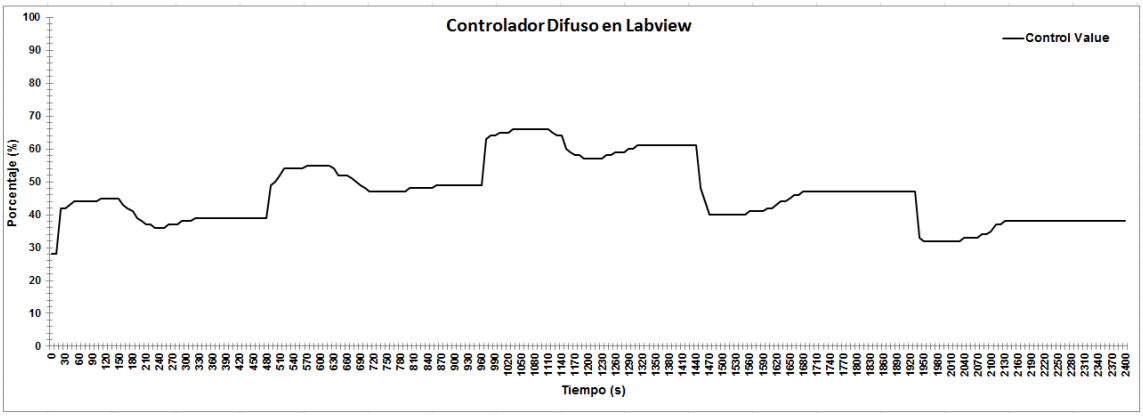

Figura 56 Curvas de Respuesta Control Value Control Difuso Labview

En la figura 57 se muestra las curvas de respuesta del Process Value del controlador PID realizado en la tarjeta Beaglebone Black vs Labview.

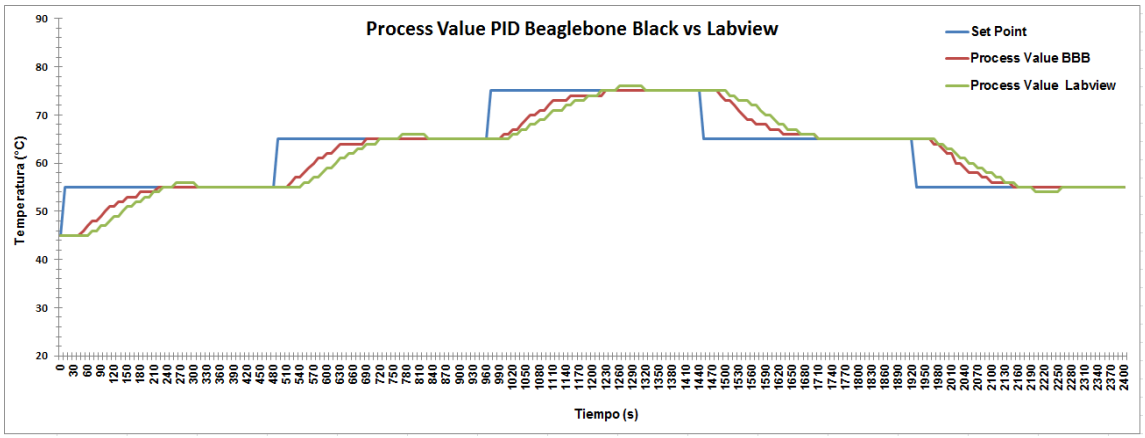

Figura 57 Comparación Process Value Control PID Beaglebone Black vs Labview

En la figura 58 se muestra las curvas de respuesta del control value del controlador PID realizado en la tarjeta Beaglebone Black vs el realizado en Labview.

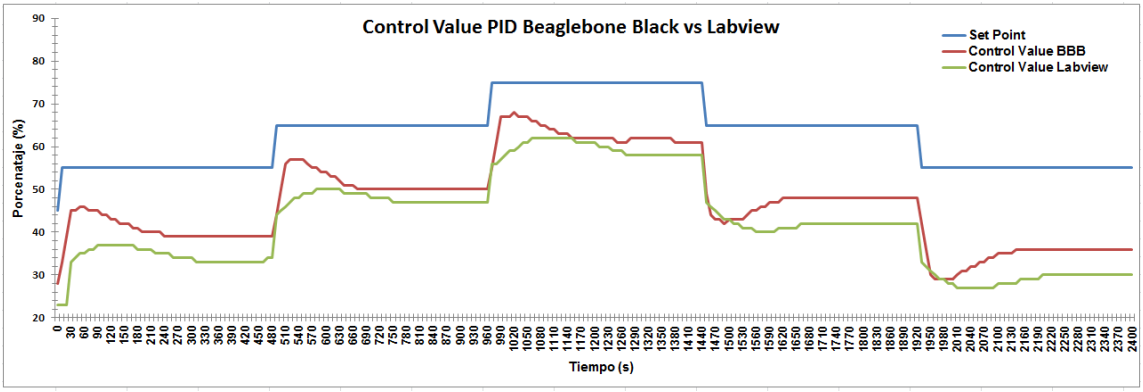

Figura 58 Comparación Control Value Control PID Beaglebone Black vs Labview

En la figura 59 se muestra las curvas de respuesta del process value del controlador Difuso realizado en la tarjeta Beaglebone Black vs el realizado en Labview.

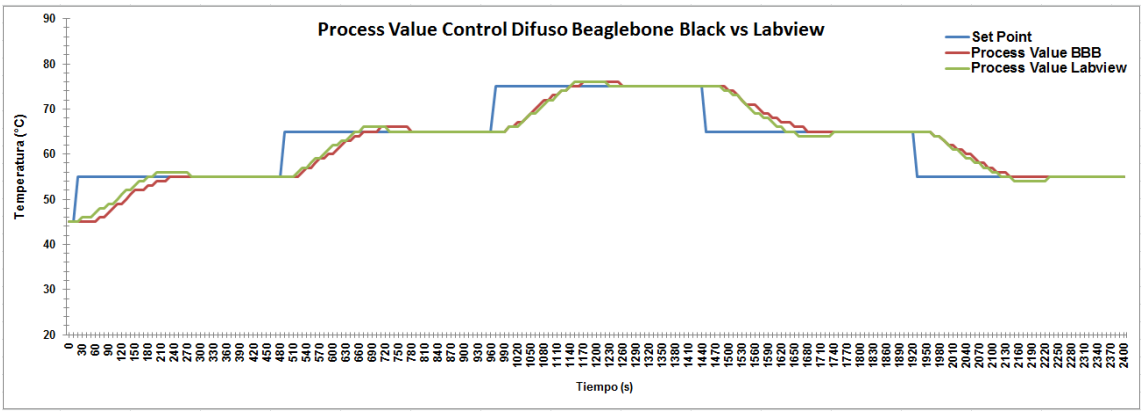

Figura 59 Comparación Process Value Control Difuso Beaglebone Black vs Labview

En la figura 60 se muestra las curvas de respuesta del control value del controlador Difuso realizado en la tarjeta Beaglebone Black vs el realizado en Labview.

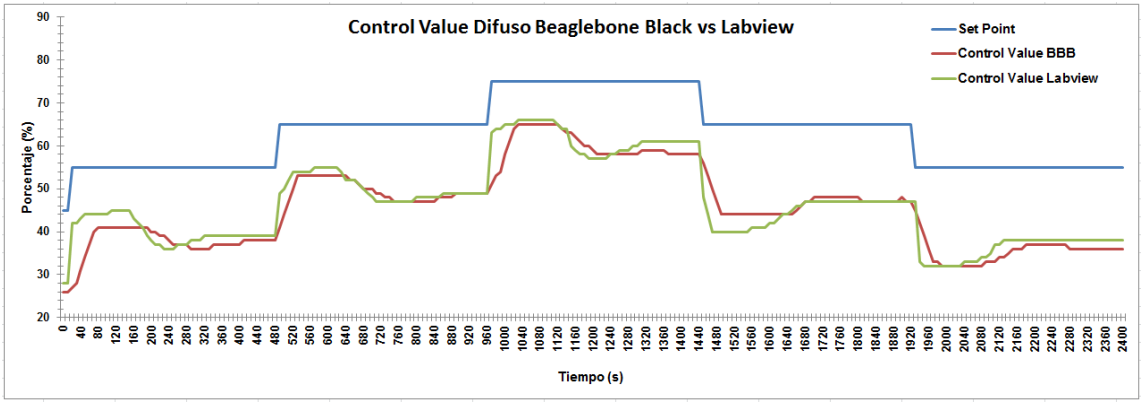

Figura 60 Comparación Control Value Control Difuso Beaglebone Black vs Labview

A continuación en la figura 61 se presenta la comparación del comportamiento de un escalón de los Process Value del controlador PID realizados en la tarjeta Beaglebone Black y en Labview

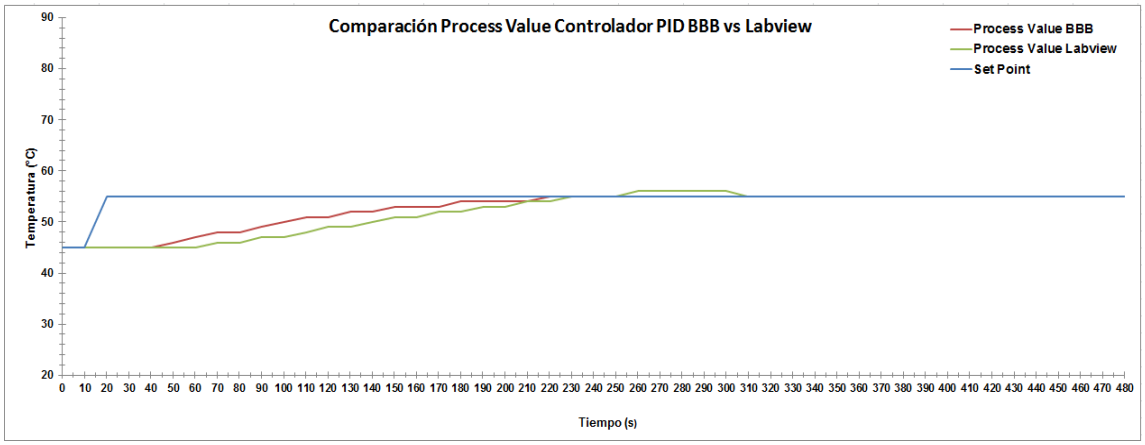

Figura 61 Comparación Escalón Process Value PID BBB vs Labview

A continuación en la figura 62 se presenta la comparación del comportamiento de un escalón de los Process Value del controlador Difuso realizados en la tarjeta Beaglebone Black y en Labview

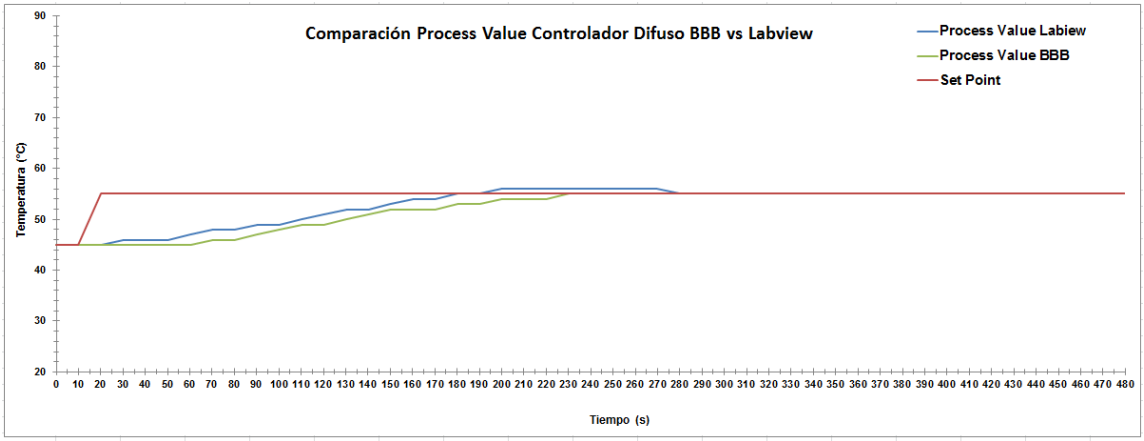

Figura 62 Comparación Escalón Process Value Difuso BBB vs Labview

Tabla 8

Comparativa de parámetros Beaglebone Black vs Labview

	PID		DIFUSO	
PARAMETROS	**Beaglebone Blacl**	**Labview**	**Beaglebone Black**	**Labview**
Sobre impulso	0%	10%	0%	10%
Tiempo de Retardc	30 seg	50 seg	50 seg	15 seg
Tiempo de Subida	180 seg	170 seg	170 seg	155 seg
Tiempo de Establecimiento	210 seg	300 seg	230 seg	270 seg
Error	0%	0%	0%	0%

Los controladores desarrollados en Labview presentan resultados similares a los obtenidos en la tarjeta Beaglebone Black, con lo que se puede determinar que los algoritmos implementados en la tarjeta son de igual eficiencia a los que provee un software propietario, en el caso de la variable temperatura motivo de esta investigación.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La implementación del sistema de monitoreo y control de la temperatura de flujo de aire, ayuda a los estudiantes a familiarizarse con los procesos industriales específicamente con la variable muy utilizada como es la temperatura, aportando a la consolidación del estudio del control automático, temas existentes dentro de la formación académica de los estudiantes de la Carrera de Ingeniería en Electrónica e Instrumentación.

- El sistema sirve como herramienta de aprendizaje en el estudio y desarrollo de sistemas de control clásicos como el PID y el diseño de controles avanzados como la Lógica Difusa.

- En la tarjeta Beaglebone Black se puede realizar el diseño e implementación de controladores, los mismos que pueden resultar no tan efectivos debido a la poca capacidad de procesamiento de la tarjeta.

- El tiempo de procesamiento de la tarjeta Beaglebone Black es lento, debido a que para ejecutar instrucciones de control, interfaz gráfica y la salida de los resultados produce un retardo en este proceso.

- El uso del hardware libre permite la integración de varios dispositivos para una solución integral sin que esto afecte al producto final de la aplicación en desarrollo.

- Se evidenció que en la ejecución de los algoritmos de control tanto para el PID como para el Difuso, el lenguaje de programación más adecuado para la implementación dentro del sistema embebido Beaglebone Black

es Python, ya que presenta facilidad en la escritura, tiene flexibilidad respecto al uso de librerías, declaración de variables y además por la baja demanda de recursos computacionales.

5.2 Recomendaciones

- Utilizar la versión Python 2 ya que en ella existen una mayor variedad de recursos que permiten mayor facilidad para la programación.

- Las condiciones de diseño de la resistencia calefactora se las debe hacer de acuerdo a las necesidades de la aplicación en desarrollo.

- Realizar una interfaz gráfica de fácil acceso para que de esta manera el usuario no tenga inconvenientes para interactuar con la misma.

- Para el uso del sistema se recomienda usar protecciones tanto para el motor como el variador de frecuencia para posibles sobrecargas externas, así como el uso de un paro general para protección del sistema.

- Hacer uso del sensor adecuado, que tenga la precisión y la exactitud necesaria para que no exista inconvenientes con la toma de datos y por lo tanto con los controles desarrollados.

REFERENCIAS BIBLIOGRÁFICAS

Almache Coyago Luis Fernando, T. S. (14 de 05 de 2014). *Diseño e implementación de módulo didáctico para el monitoreo y control automático de los sistemas de presión y temperatura para el laboratorio de redes industriales y control de procesos de la Universidad de las Fuerzas Armadas espe extensión latacunga .* Recuperado el 18 de 04 de 2016, Obtenido de http://repositorio.espe.edu.ec/handle/21000/8444

Alvares, F. J. (1995). *Diseño sistemático de controladores difusos usando razonamiento inductivo*. (En la Universitat Politècnica de Catalunya (España)) Recuperado el 30 de Marzo de 2016, Obtenido de https://dialnet.unirioja.es/servlet/tesis?codigo=10254

Barahona Robles, J. &. (2003). *Introducción al software libre.* Catalunia: Universidad de Catalunia.

Cabezas, A., & Capelo, A. (2012). *Repositorio Digital ESPE.* Obtenido de Repositorio Digital ESPE: http://repositorio.espe.edu.ec/bitstream/21000/5266/1/T-ESPE-033208.pdf

Cobo, R. (05 de Mayo de 2008). *EL ABC de la Automatización*. Obtenido de http://www.aie.cl/files/file/comites/ca/abc/DRIVES-variadores%20de%20velocidad%20.pdf

Cusumano, M. (2004). *The business of Software.* Free press.

Dani, R. P. (07 de 10 de 2008). *Coriolys Ingenieria y Tecnología*. Obtenido de https://coriolisblog.wordpress.com/2008/11/07/ventiladores-centrifugos/

Daniel, G., & Vásconez, D. (2012). *Maskay Electrónica.* Recuperado el 5 de Diciembre de 2015, Obtenido de Maskay Electrónica: http://journal.espe.edu.ec/index.php/maskay/article/view/117

Dany, O. (s.f.). *Galileo.* Recuperado el 2016 de Diciembre de 2016, Obtenido de https://www.ucuenca.edu.ec/ojs/index.php/galileo/article/view/166

Digital, P. (2007). *Página Digital*. Recuperado el 16 de Marzo de 2016, Obtenido de http://www.paginadigital.com.ar/articulos/2007/2007prim/tecnologia41/hardware-mi-211107.asp

Direct INDUSTRY. (18 de 01 de 2017). *Direct INDUSTRY*. Obtenido de http://www.directindustry.es/prod/cimme/product-15998-718031.html

Ecured. (15 de Mayo de 2013). *Ecured*. Obtenido de Ecured: https://www.ecured.cu/Motor_el%C3%A9ctrico_trif%C3%A1sico

EcuRed. (2016). *EcuRed*. Recuperado el 16 de Marzo de 2016, Obtenido de http://www.ecured.cu/index.php/Hardware_libre

GNU. (2 de Enero de 2016). *El sistema operativo GNU*. (GNU) Recuperado el 16 de Marzo de 2016, Obtenido de https://www.gnu.org/philosophy/free-sw.es.html

González, D. (Marzo de 2002). *Intercambiadores de calor - Tipos Generales y Aplicaciones*. (Departamento de Termodinámica y Fenómenos de Transferencia - Universidad Simón Bolívar) Recuperado el 30 de Marzo de 2016, Obtenido de https://operaciones1.files.wordpress.com/2009/07/intercambiadores-de-calor-tipos-generales-y-aplicaciones.pdf

Group, I. D. (15 de Mayo de 2014). *IPAC Duferco Group*. Obtenido de http://www.ipac-acero.com/somos.php

Handbook, M. E. (25 de Marzo de 2012). *Refrigeración en la industria química*. (Refrigeration, Cryogenics) Recuperado el 4 de Abril de 2016, Obtenido de http://todoproductividad.blogspot.com/2012/03/refrigeracion-en-la-industria-quimica.html

IMEX, R. . (12 de 01 de 2017). *Resistencias Electri.Indistriales*. Obtenido de http://tlahuac.aki.com.mx/empresa/rei-imex-resistencias-electri-industriales-.html

Jaramillo, O. A. (20 de Noviembre de 2007). *Intercambiadores de Calor*. (Centro de Investigación en Energía - Universidad Nacional Autónoma de México) Recuperado el 30 de Marzo de 2016, Obtenido de http://www.cie.unam.mx/~ojs/pub/HeatExchanger/Intercambiadores.pdf

Más, J. (Octubre de 2003). *UOC*. (UOC) Recuperado el 16 de Marzo de 2016, Obtenido de http://www.uoc.edu/dt/20327/index.html

Mathas, C. (27 de 10 de 2011). *Digi - Key Electronics*. Obtenido de Biblioteca de Artículos: Obtenido de http://www.digikey.com/es/articles/techzone/2011/oct/temperature-sensors-the-basics

MECALUX. (18 de 01 de 2016). *MECALUX Logismarket*. Obtenido de https://www.logismarket.cl/comind/sensores-temperatura/5269148302-4245933009-p.html

Molloy, D. (2015). *Exploring BeagleBone Tools And Techniques For Building With Embedded Linux.* Indianapolis: John Willey & Sons, Inc.

Moreno, J. E. (10 de Octubre de 2002). *Repositorio Digital de la Universidad de las Fuerza Armadas*. Obtenido de http://repositorio.espe.edu.ec/bitstream/21000/3575/1/T-ESPEL-0067.pdf

Quiminet. (25 de Mayo de 2010). *Quiminet*. Obtenido de http://www.quiminet.com/articulos/características-de-un-ventilador-centrifugo-42822.htm

Richardson, M. (2013). *Getting Started with BeagleBone.* USA: Maker Media.

Sebastián, H., & Julio, H. (Febrero de 2016). *Reposiorio UTP.* Recuperado el 05 de Diciembre de 2016, Obtenido de http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/6166/6213822H565.pdf?sequence=1

Sinha, S. (20 de 12 de 2016). *Arquigráfico*. Obtenido de http://www.arkigrafico.com/el-acero-galvanizado-en-la-construccion/

Steeve, T. (Octubre de 2007). *Diseño de Controladores PID en Tiempo Discreto, y Análisis de Respuesta Utilizando Herramientas Computacionales.* Recuperado el 5 de Diciembre de 2016, Obtenido de https://www.academia.edu/6625598/Dise%c3%91o_de_controladores_pid_en_tiempo_discreto_y_an%c3%81lisis_de_respuesta_utilizando_herramientas_computacionales

Ultraplast. (17 de Marzo de 2016). *Acrilico y Policarbonato.com*. Obtenido de http://www.acrilico-y-policarbonato.com/acrilico-tubo.html

Villalba, M. (Octubre de 2011). *Sistema de control de temperatura para la climatización del quirófano y la sala de reuperación del hospital Nuestra Señora de la Merced de la ciudad de Ambato*. (Carrera de Ingeniería en Electrónica y Comunicaiones) Recuperado el 4 de Abril de 2016, Obtenido de http://repo.uta.edu.ec/bitstream/123456789/680/1/Tesis_t650ec.pdf

ANEXOS

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por los señores:

GALO GEOVANNY CHACÓN GALARZA, VÍCTOR ALFONSO TAPIA TAPIA,

En la ciudad de Latacunga, a los 28 días del mes de julio del 2017.

Ing. Marco Pilatásig

DIRECTOR DEL PROYECTO

Aprobado por:

Ing. Franklin Silva

DIRECTOR DE CARRERA

Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO