



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

**DISEÑO DE UN SISTEMA DE CONTROL MODULAR Y AUTOMATIZADO
PARA REGULACIÓN TÉRMICA Y DE HUMEDAD EN HÁBITATS DE
ORGANISMOS ECTOTERMOS**

Lionel Antonio Mazariegos Hernández

Asesorado por la Inga. Ingrid Salomé Rodríguez de Loukota

Guatemala, julio de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE UN SISTEMA DE CONTROL MODULAR Y AUTOMATIZADO
PARA REGULACION TÉRMICA Y DE HUMEDAD EN HÁBITATS DE
ORGANISMOS ECTOTERMOS**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

LIONEL ANTONIO MAZARIEGOS HERNÁNDEZ

ASESORADO POR LA INGA. INGRID SALOMÉ RODRÍGUEZ DE LOUKOTA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, JULIO DE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Jurgen Andoni Ramírez Ramírez
VOCAL V	Br. Oscar Humberto Galicia Nuñez
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. Marvin Marino Hernández Fernández
EXAMINADORA	Inga. María Magdalena Puente
EXAMINADOR	Ing. José Antonio de León Escobar
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

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

DISEÑO DE UN SISTEMA DE CONTROL MODULAR Y AUTOMATIZADO PARA REGULACION TÉRMICA Y DE HUMEDAD EN HÁBITATS DE ORGANISMOS ECTOTERMOS

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Eléctrica, con fecha 17 de julio de 2014.

Lionel Antonio Mazariegos Hernández

Guatemala 21 de octubre de 2016

Ingeniero
Carlos Eduardo Guzmán Salazar
Coordinador del Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

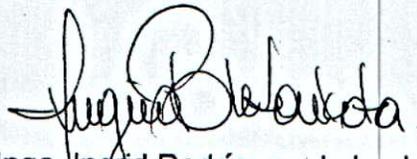
Estimado Ingeniero Guzmán.

Me permito dar aprobación al trabajo de graduación titular: "**Diseño de un sistema de control modular y automatizado para regulación térmica y de humedad de hábitats de organismos ectotermos**", del señor Lionel Antonio Mazariegos Hernández, por considerar que cumple con los requisitos establecidos.

Por tanto, el autor de este trabajo de graduación y, yo, como su asesora, nos hacemos responsables por el contenido y conclusiones del mismo.

Sin otro particular, me es grato saludarle.

Atentamente,



Inga. Ingrid Rodríguez de Loukota
Colegiada 5,356
Asesora

Ingrid Rodríguez de Loukota
Ingeniera en Electrónica
colegiada 5356



Ref. EIME 70.2016.
Guatemala, 25 de OCTUBRE 2016.

Señor Director
Ing. Francisco Javier González López
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

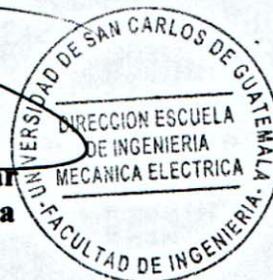
Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado:
**DISEÑO DE UN SISTEMA DE CONTROL MODULAR Y
AUTOMATIZADO PARA REGULACIÓN TÉRMICA Y DE
HUMEDAD DE HÁBITATS DE ORGANISMOS ECTOTERMOS,**
del estudiante **Lionel Antonio Mazariegos Hernández,** que cumple con
los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
DID Y ENSEÑAD A TODOS

Ing. **Carlos Eduardo Guzmán Salazar**
Coordinador Área Electrónica



SRO

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA

REF. EIME 70. 2016.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; **LIONEL ANTONIO MAZARIEGOS HERNÁNDEZ** titulado: **DISEÑO DE UN SISTEMA DE CONTROL MODULAR Y AUTOMATIZADO PARA REGULACIÓN TÉRMICA Y DE HUMEDAD DE HÁBITATS DE ORGANISMOS ECTOTERMOS**, procede a la autorización del mismo.

Ing. Francisco Javier González López



GUATEMALA, 9 DE NOVIEMBRE 2016.

Universidad de San Carlos
de Guatemala

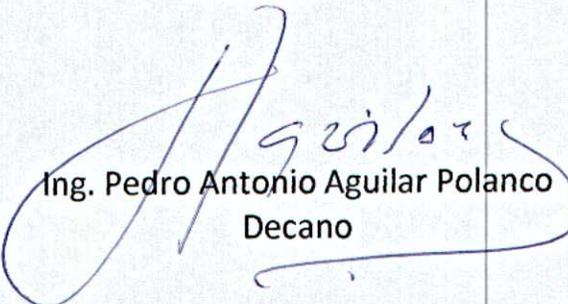


Facultad de Ingeniería
Decanato

DTG. 291.2017

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO DE UN SISTEMA DE CONTROL MODULAR Y AUTOMATIZADO PARA REGULACIÓN TÉRMICA Y DE HUMEDAD DE HÁBITATS DE ORGANISMOS ECTOTERMOS**, presentado por el estudiante universitario: **Lionel Antonio Mazariegos Hernández**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:


Ing. Pedro Antonio Aguilar Polanco
Decano

Guatemala, julio de 2017

/gdech



ACTO QUE DEDICO A:

Dios	Por darme la paciencia y las fuerzas para realizar esta tarea, por ser la motivación para seguir y por mantenerme firme en el propósito de vida.
Mis padres	Por apoyarme durante la carrera y por sus enseñanzas que me ayudaron a seguir durante todo este trayecto.
Mi hermano	Pablo Mazariegos, por el apoyo, la honestidad y por compartir mi visión hacia el futuro.
Mis amigos y compañeros	Por vivir esta experiencia conmigo. Y seguir hasta aquí.

AGRADECIMIENTOS A:

**Universidad de San Carlos
de Guatemala**

Por acogerme durante todos estos años y darme la oportunidad de ser parte de tan gloriosa universidad.

Facultad de Ingeniería

Por brindarme y transmitirme el conocimiento que me permite estar hoy aquí.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS	VII
GLOSARIO	IX
RESUMEN.....	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN	XVII
1. CONTEXTO PARA EL DESARROLLO DEL SISTEMA	1
1.1. Biología de un organismo ectotermo	1
1.1.1. Concepto	1
1.1.2. Diferencias entre organismos endotermos y ectotermos.....	2
1.1.3. Organismos poiquilotermos y homeotermos.....	2
1.1.4. Métodos para la obtención de calor	3
1.1.5. Organismos que están comprendidos dentro del problema.....	4
1.2. Reptiles.....	4
1.2.1. Biología del reptil	4
1.2.2. Condiciones óptimas de vida para un reptil	6
1.2.3. Condiciones de vida de un reptil en cautiverio.....	7
1.3. Definición del problema	8
1.3.1. El hábitat como un sistema de control	8
1.3.1.1. Definición de un sistema de control	8
1.3.1.2. El hábitat como un lazo abierto.....	9
1.4. Planteamiento de la solución.....	10

1.4.1.	El hábitat como un lazo cerrado	10
2.	TECNOLOGÍA DE DISEÑO PARA EL SISTEMA DE CONTROL.....	13
2.1.	Fuentes de calor	13
2.1.1.	Lámparas incandescentes.....	13
2.1.2.	Fuentes de calor infrarrojas.....	15
2.2.	Retroalimentación térmica.....	15
2.2.1.	Diseño del circuito medidor de temperatura	17
2.3.	Diseño de un módulo de fuente de calor regulable mediante tiristores (TRIAC).....	19
2.3.1.	El tiristor	19
2.3.1.1.	Estructura de un tiristor	20
2.3.1.2.	Funcionamiento de un tiristor	20
2.3.2.	El tiristor para corriente alterna	21
2.3.3.	Diferencias con un tiristor	21
2.3.4.	Diseño del regulador de potencia para la fuente de calor.....	22
2.3.4.1.	Componentes necesarios para el circuito regulador.....	22
2.3.4.2.	El opto acoplador	25
2.4.	Diseño del circuito regulador de potencia para la lámpara de calor	27
2.5.	Sensor de humedad.....	29
2.6.	Humidificador común.....	32
2.6.1.	Conexión del humidificador al sistema de control ...	33
2.6.2.	Diseño del circuito del humidificador	34
3.	MÓDULO DE CONTROL.....	37
3.1.	La Raspberry Pi como controlador	37

3.1.1.	Especificaciones del ordenador	37
3.1.2.	GPIO.....	39
3.2.	Software y conceptos de control para la Raspberry Pi b+	41
3.2.1.	Control proporcional integral diferencial	41
3.2.2.	Creación de las rutinas de control y entrada/salida para el programa principal.....	43
3.2.2.1.	PWM.....	46
3.3.	Creación de la interfaz de control como página web	48
3.3.1.	Creación de un servidor LAMP	48
3.3.1.1.	Linux	49
3.3.1.2.	Apache	50
3.3.1.3.	Mysql	50
3.3.1.3.1.	Creación de la base de datos para temperatura y humedad	51
3.3.1.4.	PHP	52
3.3.1.5.	HTML.....	52
3.3.2.	Diseño de la página web	53
3.3.3.	Entradas	55
3.3.4.	Salidas	56
4.	ESCALABILIDAD DEL DISEÑO.....	59
4.1.	El protocolo IP/Ethernet (IP)	59
4.1.1.	El Modelo OSI.....	59
4.2.	Conectividad física	61
	CONCLUSIONES	63
	RECOMENDACIONES	65

BIBLIOGRAFÍA.....67
APÉNDICES.....69

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Sensor DS18B20	16
2.	Diagrama de conexiones.....	18
3.	Diagrama del circuito electrónico	18
4.	Modelo en 3D	19
5.	Diagrama y construcción de un tiristor	20
6.	Diagrama y construcción de un Triac	22
7.	Resistor común de carbón	24
8.	Capacitor.....	25
9.	Opto acoplador.....	26
10.	Diagrama de conexiones.....	27
11.	Diagrama del circuito electrónico	28
12.	Modelo en 3D.....	28
13.	Conexión del sensor dht 11	30
14.	Diseño de la placa.....	31
15.	Modelo en 3D	32
16.	Humidificador ultrasónico	33
17.	Diagrama del circuito eléctrico	34
18.	Diagrama de conexiones.....	34
19.	Modelo en 3D	35
20.	Raspberry Pi B+	39
21.	Diagrama de pines GPIO	40
22.	Estructura control PID	42
23.	Rutinas de control y entrada/salida para el programa principal.....	44

24.	Controlador de temperatura	46
25.	Salida del controlador PID	47
26.	Raspberry Pi	54
27.	Cliente remoto.....	54
28.	Medición de temperatura DS18B20	55
29.	Sensor de humedad DHT11	56
30.	Conexión de salida	56
31.	Humificador.....	57
32.	Topología de red.....	61

TABLAS

I.	Condiciones óptimas de vida para un reptil	7
II.	Comparación de temperatura y potencia	15
III.	Relación temperatura/valores binarios.....	17
IV.	Características del Triac	23
V.	Especificaciones del opto acoplador.....	26
VI.	Especificaciones del sensor.....	31
VII.	Especificaciones del ordenador Raspberry Pi B+	38

LISTA DE SÍMBOLOS

Símbolo	Significado
A	Amperio
HP	Caballos de fuerza
AC	Corriente alterna
DC	Corriente directa
\$	Dólar estadounidense
f	Frecuencia
Hz	Hercio
kWh	Kilovatio por hora
®	Marca registrada
Ω	Ohm
p	Polo
ω	Velocidad angular
V	Voltio
W	Vatio
J	<i>Joule</i> (julio)

GLOSARIO

Artrópodo	El filo más diverso y numeroso del reino animal lo constituyen invertebrados con externo y apéndices articulados.
Calor metabólico	Reacción química en el metabolismo que produce calor con base en energía de hidratos de carbono.
Clase	Subdivisión de un reino de seres vivos. Tercer grado de clasificación taxonómica.
Conducción	Modo de transferencia de calor que se produce a través de un medio estacionario.
Convección	Modo de transferencia de calor en el que un fluido se encarga de llevar la temperatura a distintas áreas con diferente temperatura.
Ectotermia	Proceso por el cual ciertos organismos regulan su temperatura a partir de la temperatura, ambiental.
Endotermia	Proceso por el cual ciertos organismos regulan su temperatura interna mediante procesos químicos, manteniéndola constante sin depender del ambiente.

Filo	Segunda división de un reino, una clasificación taxonómica un grado más específico que el reino.
Homeotermia	Proceso por el cual un organismo mantiene una temperatura constante todo el tiempo.
Organismo	Parte material y física de un ser vivo, en la que interactúan sistemas de comunicación molecular, y participan en un intercambio de materia y energía de forma ordenada.
Poiquiloterma	Proceso por el cual un organismo varía su temperatura fácilmente.
Radiación	Modo de transferencia de calor en el que la temperatura o calor es transferido por medio de ondas electromagnéticas.
Reino	Representa cada una de las divisiones mayores en las que se dividen los seres vivos.
Reptil	Grupo de vertebrados que poseen escamas. Estos son ovíparos e incapaces de regular su temperatura corporal por medios internos.
Tasa metabólica	Conjunto de reacciones químicas y biológicas que permite diversas actividades celulares tales como crecer reproducirse y responder a estímulos.

**Termorregulación
conductual**

Proceso mediante el cual un ser vivo adopta conductas periódicas que le permiten regular su temperatura corporal.

RESUMEN

La biodiversidad es muy importante en el ciclo de la vida, en la actualidad este tema ha tomado gran importancia por lo que surge la necesidad de protegerla, sobre todo, porque Guatemala es un país rico en vida y en fauna y existen ciertos tipos de animales que necesitan un cuidado especial para vivir.

En el primer capítulo de este trabajo de graduación se presentan las especies en las que se enfocará el estudio, su modo de vida, sus hábitos, sus condiciones óptimas de supervivencia, y el comportamiento que amerita un sistema de control automatizado. Se propone también una solución tecnológica para mejorar la calidad y esperanza de vida de los organismos.

La calidad de vida se puede mejorar mediante tecnología; es por eso que en el segundo capítulo se presenta la tecnología necesaria y adecuada para automatizar el clima. Se describe de una manera concisa el planeamiento y el diseño de cada circuito electrónico que se va a utilizar.

Toda la electrónica presentada en el segundo capítulo no puede funcionar sin una unidad de control. Por ello, en el tercer capítulo se detalla el funcionamiento y la programación de la unidad de control utilizando una *Single board computer* (SBC). Esto reduce costos y hace más asequible el proyecto de automatizar un hábitat.

En el cuarto capítulo, luego de describir la tecnología y el control, se propone una estrategia de integración tanto para el hábitat como para la electrónica. Con ello, es necesario que todo funcione de manera simbiótica para

que los organismos que se vayan a beneficiar de este proyecto no logren notar la diferencia entre su hábitat natural y un clima automatizado.

Por último, al ser una solución tecnológica escalable, el último y quinto capítulo presenta la manera de hacer la solución modular. Esto es, utilizado en varios de ellos, sistemas para controlar distintos hábitats de una manera mucho más fácil y ordenada.

OBJETIVOS

General

Diseñar un sistema de control retroalimentado que regule las condiciones climatológicas de un hábitat de organismos ectotermos, para aumentar su calidad de vida.

Específicos

1. Diseñar los módulos de operación utilizados por el sistema de control.
2. Diseñar un sistema de fácil adaptabilidad para distintos hábitats.
3. Crear un sistema de control capaz de crecer de acuerdo con las necesidades del usuario.
4. Centralizar el control mediante páginas web dirigidas a un solo servidor mediante una red Ethernet.

INTRODUCCIÓN

Un organismo ectotermo es aquel que no es capaz de utilizar su energía para regular su metabolismo, por lo que necesita de condiciones exteriores para hacerlo. Dentro de este grupo se encuentran los reptiles y los artrópodos (insectos, arácnidos, moluscos). Cada uno de estos organismos tienen diferentes necesidades térmicas y climatológicas. Actualmente, en el país existe poco control sobre estos hábitats deteriorando la calidad de vida de los organismos.

Al automatizar y retroalimentar el control se puede obtener una mayor esperanza de vida y condiciones favorables de reproducción para los organismos dentro del hábitat, haciendo necesario diseñar una red de retroalimentación y control para desarrollar condiciones favorables, para la investigación o el cuidado de dichos organismos.

El diseño de este sistema de control ampliará la calidad de vida de los organismos que se encuentren viviendo en el hábitat. También facilitará el desarrollo de condiciones favorables para su reproducción.

Criaderos de insectos, herpetarios o entidades públicas al cuidado de estos organismos podrán beneficiarse utilizando este diseño al no tener un costo tan elevado, y con el que se obtienen resultados de condiciones de vida óptimos para el ambiente.

1. CONTEXTO PARA EL DESARROLLO DEL SISTEMA

1.1. Biología de un organismo ectotermo

Se define como ectotermia al proceso de regulación de temperatura en el que determinados organismos vivos se valen de la temperatura ambiental, para establecer su propia temperatura corporal.

1.1.1. Concepto

Los organismos que utilizan el proceso de ectotermia son denominados ectotermos. Estos organismos aprovechan el calor del ambiente mediante comportamientos específicos para aprovechar completamente el calor que se transfiere de su ambiente a su sistema.

Al utilizar condiciones externas para regular la temperatura corporal, el proceso de ectotermia produce condiciones muy estables de calor dentro del organismo. Esto hace que el metabolismo del animal pueda ser regulado de una manera estable.

Los filos del reino animal, que son típicamente descritos por esta característica de termorregulación, son los reptiles y los artrópodos. Estos presentan conductas prominentemente ectotérmicas. Si bien estos son los filos que más se caracterizan por este comportamiento, algunos peces y aves también presentan conductas de aprovechamiento del calor ambiental. Esto para favorecer a su metabolismo o regular su temperatura interna.

El proceso inverso a la ectotermia es el termorregulador que no depende de condiciones externas, sino de procesos químicos. Esto, para producir energía calórica mediante la síntesis de proteínas, se denomina endotermia.

1.1.2. Diferencias entre organismos endotermos y ectotermos

La mayor diferencia entre estos organismos es su relación de dependencia con su ambiente. Un organismo endotermo depende de la comida que puede conseguir para transformarla en energía calórica, mientras que un ectotermo puede pasar largos períodos sin comida. Algunas porque son capaces de regular su metabolismo externamente.

Un organismo ectotermo no depende de su alimentación, pero a menudo vive en condiciones de temperatura extrema, y son extremadamente susceptibles a los cambios drásticos de temperatura en el ambiente. Es por eso que es necesario tener un hábitat controlado, para que estos organismos no sufran los cambios bruscos de temperatura afectando directamente su metabolismo.

1.1.3. Organismos poiquilothermos y homeotermos

Un organismo poiquilothermo es aquel que posee una temperatura interna variable. Por lo contrario, un organismo homeotermo posee una temperatura interna relativamente estable, por arriba o por debajo de la temperatura externa.

Se concluye, entonces, que un organismo ectotermo es poiquilothermo por definición. Pero pueden existir también organismos ectotermos homeotermos que utilizan la termorregulación conductual para tener menos varianza en su temperatura interna.

1.1.4. Métodos para la obtención de calor

El metabolismo de un organismo ectotermo está directamente ligado a su temperatura, por medio de su tasa metabólica.

La tasa metabólica es la velocidad a la que un organismo utiliza la energía disponible. La tasa metabólica se incrementa cuando aumenta la temperatura y disminuye a medida que la temperatura desciende.¹

Debido a que la tasa metabólica de un organismo ectotermo depende de un factor externo (el ambiente), este no consume energía. Lo anterior es para producir calor haciéndolo un organismo bastante adaptivo a condiciones extremas. Adaptivo se refiere a que los seres vivos pueden encontrar sitios donde regular su temperatura o bien pueden tomar ciertas conductas, para hacerlo. Esto si no encuentran sitios donde las condiciones sean favorables para que ellos puedan obtener en su cuerpo la temperatura deseada.

Asimismo, son capaces de entrar en estado de hibernación o soportar condiciones de congelación en el caso de anfibios y reptiles si las condiciones no son favorables para gastar la energía que han logrado sintetizar. Pueden tomar también distintos comportamientos alimenticios durante determinados períodos para contribuir a la termorregulación.

Los insectos voladores son capaces de generar calor o regular su temperatura mediante conductas grupales. Estas pueden ser batir sus alas hasta generar una diferencia de temperatura o agruparse en el caso de las abejas.

¹ Curtis. *Curtis biología*. p. 128.

1.1.5. Organismos que están comprendidos dentro del problema

Son aquellos que están en cautiverio. Estos, aun teniendo un hábitat, no son capaces de desarrollarse plenamente puesto que no poseen los medios para regular su temperatura correctamente o el ambiente de cautiverio no ha sido bien diseñado. Los más comunes son los reptiles y los insectos.

1.2. Reptiles

A continuación, se define la biología de los reptiles.

1.2.1. Biología del reptil

Los reptiles regulan su temperatura conductualmente. Su metabolismo se ve directamente afectado por el intercambio de calor con el medio. Al tener períodos de actividad, pueden mantener una temperatura y un metabolismo constante de acuerdo con el medio en el que están realizando sus actividades. Si este medio llegara a cambiar, sus funciones metabólicas se verían afectadas por el cambio de temperatura.

Al descender la temperatura, las funciones metabólicas del reptil comienzan a disminuir. Esto lleva a los reptiles a un estado de reposo prolongado en el que sus funciones de nutrición cesan por completo. Esto obliga al reptil a disminuir su consumo de oxígeno a un mínimo y a agotar las reservas de energía que acumuló durante su período de actividad.

Los reptiles presentan diferentes tipos de mecanismos para regular su temperatura:

- Absorción de radiación infrarroja (luz solar): son capaces de controlar la radiación que absorben al movilizarse de la sombra al sol o solo. Ellos van exponiendo ciertas partes de su cuerpo a la radiación solar haciéndolo que varíe en su temperatura.
- Producción metabólica de calor: durante períodos de reposo prolongado, los reptiles pueden producir calor agotando sus reservas de energía acumulada. Este es un proceso químico en el que ciertos nutrientes se convierten en energía calórica. A esto se le llama calor metabólico.
- Convección: la regulación de temperatura se produce mediante el intercambio de calor entre el aire y el organismo. En organismos pequeños la temperatura al ser regulada por este medio es la misma que la del aire.
- Evaporación y condensación: los reptiles son capaces de evaporar el agua que contiene su cuerpo mediante conductas de jadeo o agitación muscular. Al evaporarse el agua de su piel se produce calor, lo que acondiciona al reptil a la temperatura deseada.
- Conducción: este método es un intercambio de calor directo entre el reptil y la superficie en donde este se encuentra. La cantidad de calor intercambiado es proporcional al área corporal del reptil que está en contacto con esta superficie.

1.2.2. Condiciones óptimas de vida para un reptil

Debido a que un reptil, depende mucho de su hábitat, es necesario que se definan las condiciones óptimas para su supervivencia.

Los reptiles dedican poca energía a procesos químicos como la digestión y la producción de calor metabólico. Al dedicar poca energía a estos procesos pueden utilizar el sobrante para su reproducción y crecimiento. Por lo tanto, el crecimiento del reptil está directamente ligado a qué tan buenas sean sus condiciones ambientales.

La regulación de temperatura y las condiciones óptimas de vida dependen también de la especie de reptil con la que se esté tratando. Si la temperatura del ambiente es mayor o menor a la óptima el reptil puede permanecer en estado de reposo, por tiempos muy prolongados o morir por falta de obtención de energía calórica.

La transferencia de calor depende también de la humedad del ambiente. Esta afecta los métodos de transferencia de calor previamente vistos y puede afectar al desarrollo del organismo. Es necesario también definir las condiciones de humedad para lograr una condición de vida óptima en el reptil.

Existen cuatro tipos de ambientes básicos para la supervivencia de un reptil, generalizando por hábitats y especies. Las condiciones promedio en las que un reptil puede sobrevivir de manera óptima son:

Tabla I. **Condiciones óptimas de vida para un reptil**

Ambiente	Temperatura	Temperatura De termorregulación	Humedad
Árido	18-29 °C	32-43 °C	10-30 %
Templado	18-29 °C	32-38 °C	30-50 %
Tropical	21-29 °C	29-35 °C	50-80 %
Húmedo	16-24 °C	27-35 °C	50-80 %

Fuente: *Condiciones óptimas de vida para un reptil*. <http://www.petco.com.mx/sitefiles/aprendiendo/reptario-okpdfpdf.pdf>. Consulta: julio de 2015.

1.2.3. **Condiciones de vida de un reptil en cautiverio**

Un reptil puede estar en cautiverio por diversos motivos, ya sea que esté en exhibición o esté siendo utilizado como mascota. De cualquier manera, sus condiciones de vida están lejos de ser las anteriores.

En el Museo de Historia Natural de la ciudad de Guatemala los reptiles vivos en exhibición carecen de fuentes de calor externas para la termorregulación. No existe control alguno sobre la humedad del ambiente, las condiciones de vida para el reptil son deplorables, lo cual disminuye su esperanza de vida y su desarrollo natural.

En el zoológico La Aurora existen hábitats con fuentes de calor externas, pero carecen de control alguno sobre la humedad. Existen hábitats con pequeños estanques de agua que son usados para regular la humedad, pero si la especie de reptil no está adaptada a un estanque, aun con las condiciones de humedad adecuadas, su hábitat no sería el adecuado y también se vería coartado su desarrollo natural.

1.3. Definición del problema

Debido a que los reptiles en promedio necesitan ciclos de doce horas de calor, temperatura templada y rangos de temperatura controlados para desarrollar sus funciones normales, se complica el diseño de un hábitat.

1.3.1. El hábitat como un sistema de control

Debe cumplir con las condiciones de vida de un reptil normal sin tener retroalimentación alguna del ambiente y la humedad. En la mayoría de los casos el elemento de retroalimentación es el ser humano a cargo del organismo y este no puede estar presente regulando la humedad y temperatura, en la mayoría de casos.

En todo caso, cabe anotar que sí es posible modelar un hábitat con estas condiciones de operación, como un sistema de control.

1.3.1.1. Definición de un sistema de control

Es un conjunto de componentes capaces de regular su operación o la operación de otros elementos, para formar condiciones óptimas de funcionamiento y evitar fallos.

El principal objetivo de un sistema de control es promover la estabilidad dentro de las variables controladas. Es decir, el sistema debe ser capaz de mantener sus condiciones previamente establecidas aun existiendo perturbaciones en el proceso de control.

Los sistemas de control se utilizan hoy día para sustituir trabajadores pasivos. Estos pueden ser humanos o mecánicos, si un trabajador pasivo controla un proceso existe una probabilidad de error muy grande y el sistema puede volverse ineficiente debido a que no existe una retroalimentación propia y objetiva.

Existen dos tipos de sistemas de control:

- Lazo abierto: estos sistemas realizan su trabajo sin tener ninguna información sobre si se realizó correctamente o no. Estos controlan las variables de entrada solamente, la salida no es controlada, tampoco hay comparación entre variables, lo cual produce un sistema altamente inestable.
- Lazo cerrado: esto presenta una comparación del trabajo realizado a la entrada contra lo que se obtiene a la salida, es decir, existe una retroalimentación y un control más preciso sobre cualquier proceso realizado. Este sistema puede ser un poco más complejo, pero presenta mayor estabilidad.

1.3.1.2. El hábitat como un lazo abierto

Definiendo entonces qué es un sistema de control y los tipos de sistemas de control se cataloga el hábitat como un sistema de lazo abierto. Sobre él no actúa ninguna retroalimentación ni trabajadores pasivos, esto representa un problema para la esperanza de vida del organismo.

Si estos organismos son importantes, ya sea para investigación, para tenencia doméstica o para exhibición, es necesario proveer las condiciones

óptimas de vida para dichos organismos. Un sistema de control de lazo abierto no es capaz de proveer estas condiciones óptimas.

1.4. Planteamiento de la solución

A continuación se presenta el planteamiento de la solución.

1.4.1. El hábitat como un lazo cerrado

En un lazo cerrado existen varios elementos que proveen estabilidad a las variables controladas, es posible comparar una variable contra otra variable de referencia. Con ello es posible retroalimentar, mediante sensores, que determinen si las condiciones óptimas se cumplen o no.

Si se diseña el hábitat desde el principio, como un lazo cerrado, la esperanza de vida del animal crecerá por las siguientes razones:

- Existe control sobre la variable temperatura
- Existe control sobre la variable humedad
- Existe un elemento de retroalimentación
- Se elimina el elemento humano pasivo
- Se puede estar monitoreando el clima durante largos períodos de tiempo

El tener un clima controlado y la habilidad de observar las variables y cambiarlas a conveniencia promueve un clima estable. Por lo tanto, un desarrollo eficiente del organismo ectotermo en todas sus funciones.

El cuidado se facilita puesto que el elemento de control ya no depende de la habilidad del elemento humano pasivo, sino que de parámetros definidos por

experiencia o por el tipo de reptil. La facilidad de control y la estabilidad se incrementan a medida que más elementos de sonorización se agregan al sistema, por lo que se puede tener una medición y un monitoreo bastante acertado sobre lo que está sucediendo.

2. TECNOLOGÍA DE DISEÑO PARA EL SISTEMA DE CONTROL

2.1. Fuentes de calor

Dentro de la tecnología de diseño es necesario tener una fuente de calor regulable con la potencia suficiente. Esto para llegar a las condiciones óptimas de temperatura del hábitat. Actualmente, existen dos dispositivos capaces de cumplir con los requisitos para ser controlados por un sistema de control y producir la energía calórica necesaria para que los organismos se desarrollen bien:

- Lámparas incandescentes
- Calentadores cerámicos

Ambas fuentes de calor son alimentadas por corriente alterna 120 voltios. Esto presenta una ventaja puesto que pueden ser elementos controlados, la corriente alterna se puede regular mediante un circuito electrónico.

2.1.1. Lámparas incandescentes

Estas se utilizan como fuente de calor puesto que el 85 % de energía que consumen la transforman en calor. Esto la hace ineficiente para la iluminación, pero eficiente para la transferencia de calor.

Las lámparas incandescentes funcionan por el efecto *joule*, este dicta en principio que una corriente eléctrica que fluye por un conductor disipa calor.

La ecuación del efecto *joule* para corriente alterna está definida de la siguiente manera:

$$P(t) = V(t)I(t)$$

Esta potencia en vatios, ya viene definida en la lámpara incandescente. Para traducir la potencia en vatios a temperatura se realiza lo siguiente:

- Se multiplica la potencia por la cantidad de tiempo que se necesita radiar calor. Esto da como resultado la ganancia de energía en *joules* (julios).
- Al obtener la ganancia de energía en *joule* (julios), se divide por la masa del objeto que va a recibir calor, en este caso, es el aire del ambiente.
- Ya con la relación energía/masa se multiplica por el calor específico del elemento que se desea calentar obteniendo la ganancia de temperatura.

La fuente de calor va a ser utilizada para calentar aire por lo que realizarán los siguientes cálculos con el calor específico del aire:

$$T = \frac{\frac{\text{Potencia} * (\text{tiempo de radiación})}{\text{masa}}}{c_e(\text{aire})}$$

$$c_e(\text{aire}) = 1\,0006 \frac{\text{J}}{\text{g} \cdot ^\circ\text{C}}$$

$$\text{masa (aire)} = 1 \text{ kg}$$

$$\text{Potencia (watts)} = 50$$

$$\text{tiempo de radiación (t)} = 10 \text{ min}$$

$$T = \frac{(50 * (10 * 60))/1\ 000}{1\ 0006} = 28,3\ ^\circ\text{C}$$

Al realizar una medición para diferentes potencias encontradas en las lámparas incandescentes, se desarrolla una tabla para regular el calor con el sistema de control.

Tabla II. **Comparación de temperatura y potencia**

Potencia	Ganancia de temperatura
50 vatios	24-32 °C
75-100 vatios	32-38 °C
100-150 vatios	C

Fuente: elaboración propia.

2.1.2. Fuentes de calor infrarrojas

Los calentadores cerámicos constan de una resistencia eléctrica cubierta de un material de cerámica que reparte el calor uniformemente y aísla la fuente de calor aumentando su vida útil. Estos son los más eficientes puesto que transforman el 85 % de energía eléctrica en calor. Ambas fuentes de calor incandescentes e infrarrojas funcionan con corriente alterna, por lo que su control dentro del sistema puede ser accionado mediante tiristores.

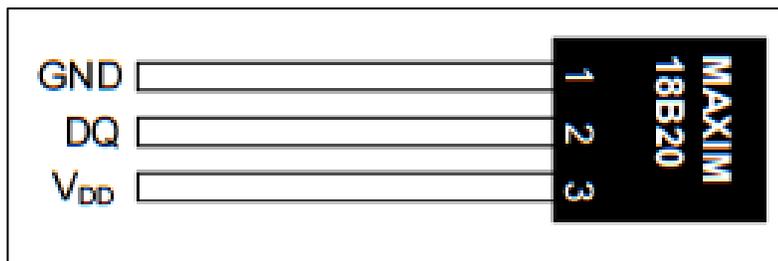
2.2. Retroalimentación térmica

Un sensor electrónico capta las magnitudes físicas y las convierte a magnitudes eléctricas que pueden ser cuantizadas y manipuladas. Al tener magnitudes cuantizadas el trabajo puede hacerse más eficiente y más rápido.

Para la retroalimentación térmica del hábitat, para organismos ectotermos, se utilizarán sensores electrónicos de temperatura. Estos son ubicados alrededor de todo el hábitat para obtener una temperatura promedio de todo el ambiente.

El sensor por utilizar es un DS18B20.

Figura 1. **Sensor DS18B20**



Fuente: *Sensor DS18B20*. <https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/temperature>.

Consulta: julio de 2015.

Este sensor posee un rango de medición de -55 a 125 °C haciéndolo adecuado para el rango de medición que se necesita viene en un encapsulado TO-92 y utiliza una sola fuente de alimentación de +3,3 V.

Este sensor es lineal en su relación temperatura/voltaje. Esto hace más fácil el trabajo de programación y acertada la medición de la sensación térmica.

Este sensor despliega un hilo binario de datos. Para cada temperatura descrita existe un método ya establecido, dentro del programa Python para leer este sensor.

Tabla III. **Relación temperatura/valores binarios**

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

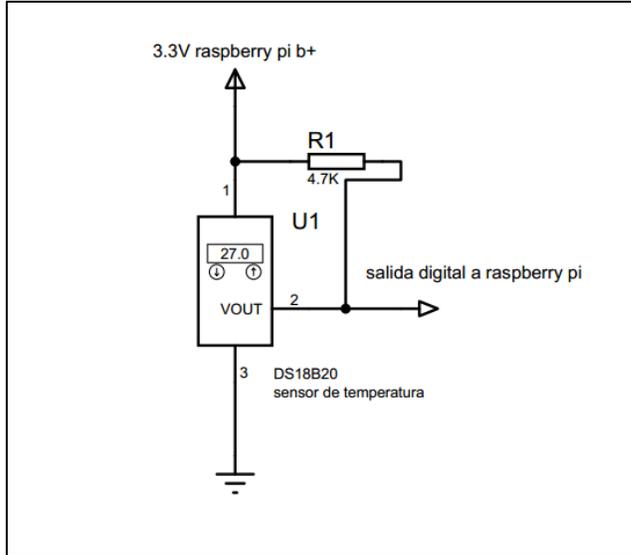
Fuente: *Relación temperatura valores binarios*. <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>. Consulta: agosto de 2015.

Este sensor tiene una incerteza de ± 2 °C y utiliza una junta bimetálica para obtener una relación de voltaje-temperatura, que es muy parecido en funcionamiento a una termo copla.

2.2.1. Diseño del circuito medidor de temperatura

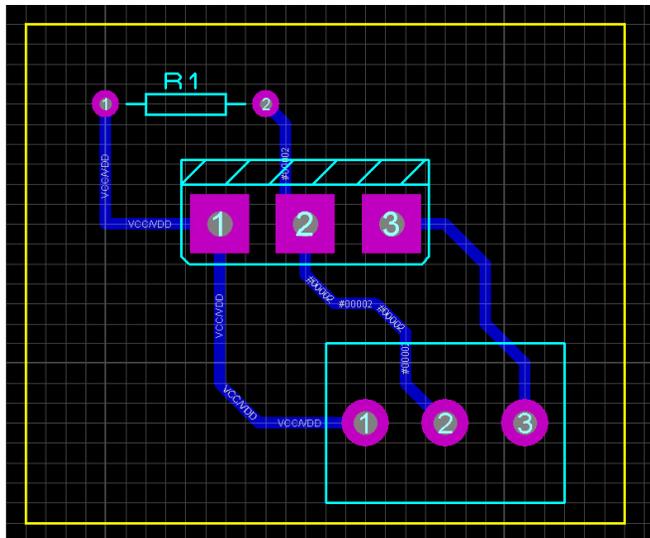
Se presenta el diseño de la circuitería realizado por computadora mediante el *software* Proteus 7 y una visualización 3D de los componentes.

Figura 2. Diagrama de conexiones



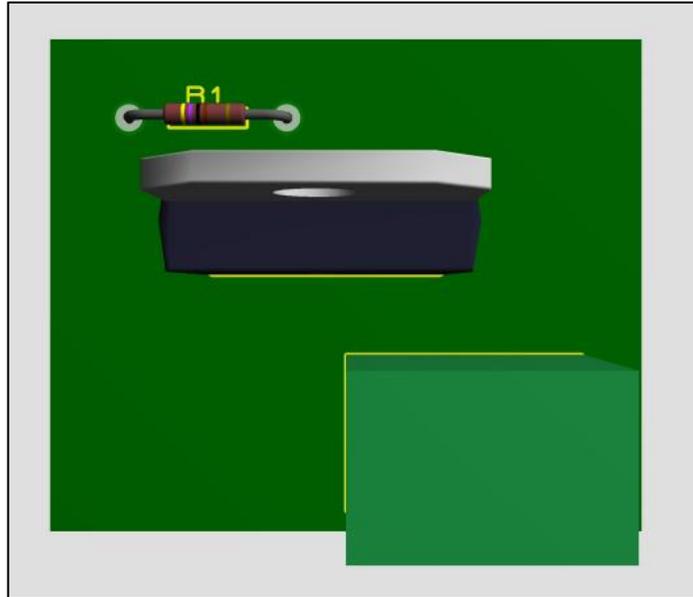
Fuente: elaboración propia, empleando Proteus 7.

Figura 3. Diagrama del circuito electrónico



Fuente: elaboración propia, empleando Proteus 7.

Figura 4. **Modelo en 3D**



Fuente: elaboración propia, empleando Proteus 7.

2.3. Diseño de un módulo de fuente de calor regulable mediante tiristores (TRIAC)

Se define como tiristor todo aquel componente semiconductor que posee dos estados estables, y funciona a base de retroalimentación regenerativa.

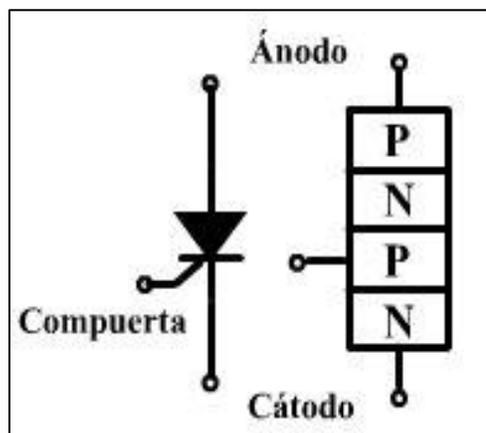
2.3.1. El tiristor

Es un dispositivo que conduce la corriente en un solo sentido. Es el equivalente electrónico a un conmutador mecánico. Su función es dejar pasar la corriente eléctrica o bloquear su paso completamente.

2.3.1.1. Estructura de un tiristor

El tiristor consta de dos juntas PN, es decir, dos uniones de semiconductor dopado positivo y semiconductor dopado negativo. Al unir las dos junturas se obtiene una estructura PNPN, como lo ilustra la siguiente figura.

Figura 5. Diagrama y construcción de un tiristor



Fuente: *Diagrama y construcción de un tiristor*. http://www.dte.uvigo.es/recursos/potencia/dc-ac/scr_pnpn.jpg. Consulta: julio de 2015.

Este componente consta de dos terminales primarias que son el ánodo y el cátodo y una terminal auxiliar para disparo también llamada compuerta.

2.3.1.2. Funcionamiento de un tiristor

Como ya se describió el tiristor es el equivalente a un interruptor mecánico y conduce corriente desde el ánodo hacia el cátodo. Para que se logre un flujo de corriente se necesita tener una corriente positiva para que fluya desde el ánodo hacia el cátodo y una pequeña excitación en la compuerta, para que este empiece a conducir.

La única manera de detener el tiristor es cortocircuitando la compuerta con el cátodo o haciendo fluir una corriente inversa. Es decir, positiva desde cátodo hacia el ánodo, para que este se bloquee y pare el flujo eléctrico por completo.

Al aplicar cierta cantidad de voltaje en la compuerta se puede ir regulando la capacidad de conmutación del tiristor, esto es que se regula la cantidad de corriente que va a pasar a través del conmutador electrónico.

Se puede decir que se comporta como un diodo rectificador con conducción controlada por compuerta.

2.3.2. El tiristor para corriente alterna

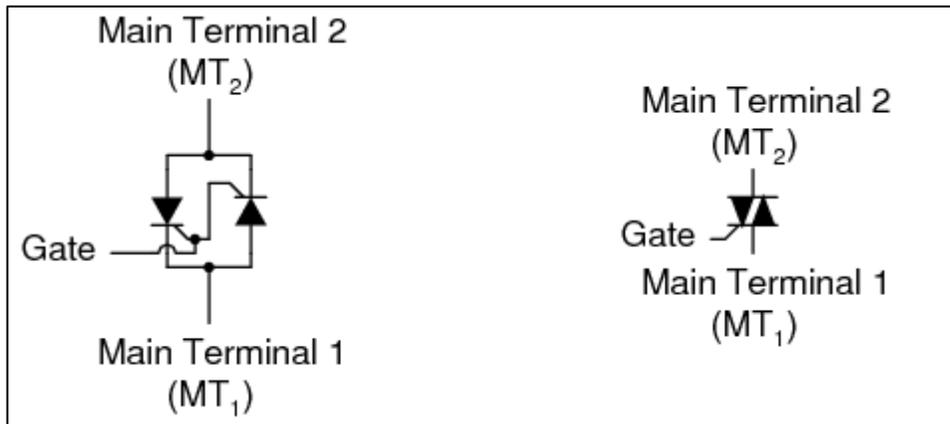
Al ser el tiristor un dispositivo unidireccional, es decir que solo conduce corriente en un solo sentido, es incapaz de conmutar corriente alterna.

2.3.3. Diferencias con un tiristor

El tiristor de corriente alterna, es capaz de bloquear la tensión y conducir corriente en ambos sentidos (bidireccional) entre sus terminales T1 y T2.

Este es el equivalente a dos tiristores comunes unidos en paralelo activados con la misma compuerta, el Triac no funciona igual a un tiristor común. Este conducirá solo cuando la compuerta esté activada.

Figura 6. **Diagrama y construcción de un Triac**



Fuente: *Diagrama y construcción de un Triac*. <http://sub.allaboutcircuits.com/images/03223.png>.

Consulta: julio de 2015.

Debido a que la fuente de calor debe ser regulable y utiliza corriente alterna, es necesario diseñar un circuito regulador. Debe recortar o ampliar la cantidad de potencia entregada a la fuente de calor.

2.3.4. Diseño del regulador de potencia para la fuente de calor

A continuación se presentan los componentes necesarios para el circuito regulador.

2.3.4.1. Componentes necesarios para el circuito regulador

Para crear un circuito regulador son necesarios componentes de aislamiento y parametrización. Estos permiten separar las fuentes de poder y

evitar cortocircuitos innecesarios. Además de establecer los parámetros de funcionamiento correcto de la red de corriente alterna.

Los componentes que se requieren para el regulador de potencia de la fuente de calor son:

- Triac: se utilizará un tiristor de corriente alterna MAC 12N. A continuación se presenta la tabla de especificaciones del mismo:

Tabla IV. **Características del Triac**

Rating	Symbol	Value	Unit
Peak Repetitive Off-State Voltage (Note 1) ($T_J = -40$ to 125°C , Sine Wave, 50 to 60 Hz, Gate Open)	V_{DRM} , V_{RRM}	400 600 800	V
On-State RMS Current (All Conduction Angles; $T_C = 70^\circ\text{C}$)	$I_{T(RMS)}$	12	A
Peak Non-Repetitive Surge Current (One Full Cycle, 60 Hz, $T_J = 125^\circ\text{C}$)	I_{TSM}	100	A
Circuit Fusing Consideration ($t = 8.33$ ms)	I^2t	41	A^2sec
Peak Gate Power (Pulse Width ≤ 1.0 μs , $T_C = 80^\circ\text{C}$)	P_{GM}	16	W
Average Gate Power ($t = 8.3$ ms, $T_C = 80^\circ\text{C}$)	$P_{G(AV)}$	0.35	W
Operating Junction Temperature Range	T_J	-40 to +125	$^\circ\text{C}$
Storage Temperature Range	T_{stg}	-40 to +150	$^\circ\text{C}$

Fuente: *Características del Triac*. http://www.onsemi.com/pub_link/Collateral/MAC12-D.PDF.

Consulta: julio 2015.

Se estará utilizando el Triac Mac 12N con capacidad de soportar 800 V y una corriente máxima de 12 A. Haciendo los cálculos de potencia disipada para una lámpara de 200W se obtendrá el rango de operación del Triac comprobando que cumple con los requisitos de potencia.

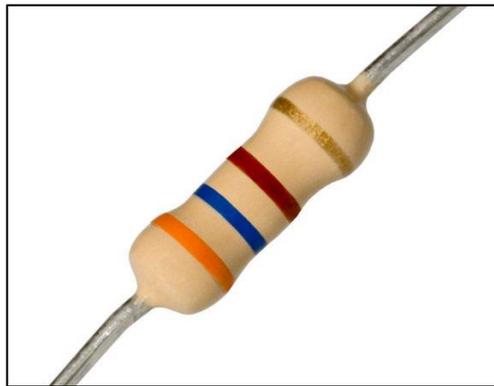
$$P = V * I$$

$$P = 800 * 12 = 9\ 600\ W$$

9 600 W son suficientes para regular el circuito de poder.

- Resistor: el resistor se opone al paso de corriente. Hecho de material de carbón provee las diferencias de potencial, mediante la Ley de Ohm. Estas son necesarias para hacer que el Triac funcione a los valores de potencia que necesitará el circuito regulador.

Figura 7. **Resistor común de carbón**



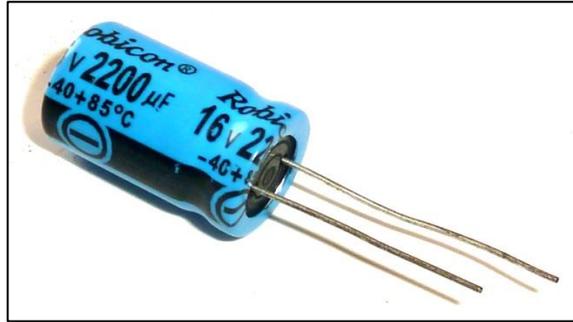
Fuente: *Resistor común de carbón*. <http://develissimo.com/media/images/productimage-picture-resistor-set-carbon-film-84.jpg> Consulta: Julio 2015

Las resistencias por utilizar dentro de este circuito son de 190Ω y de $2,5k\Omega$

- Capacitor: es un dispositivo pasivo que se utiliza para suprimir las elevaciones de tensión repentinas. Este podría provocar la conmutación de corriente alterna hecha por el Triac.

El capacitor por utilizar en este circuito es de $0,1 \mu\text{F}$

Figura 8. **Capacitor**



Fuente: *Capacitor*. <http://www.build-electronic-circuits.com/wp-content/uploads/2013/05/electrolytic-capacitor.jpg>. Consulta: julio de 2015.

2.3.4.2. El opto acoplador

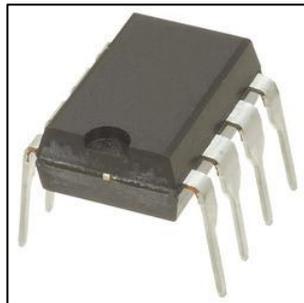
Es un dispositivo electrónico que se utiliza cuando se necesita enviar una señal de un circuito a otro, sin alimentar con la misma fuente de poder ambos circuitos. Las ventajas primordiales de este dispositivo son que puede aislar o proteger la electrónica de control.

Funciona mediante *leds* infrarrojos que envían una señal del circuito de control al receptor óptico del otro lado del opto acoplador. Este la recibe y activa un interruptor dejando pasar corriente eléctrica sin dañar algún otro equipo que esté conectado del otro lado de la red o circuito.

Cabe mencionar que, si hubiera alguna falla, dentro del circuito regulador de potencia, este dispositivo sería el que reciba el golpe de tensión ocasionado por la corriente alterna. Debido a que este sistema es modular, es necesario hacer este dispositivo fácilmente intercambiable. Esto mediante un adaptador de circuito integrado.

El opto acoplador por utilizar es el MOC3011 que está específicamente diseñado para controlar Triacs. Esto lo hace idóneo para el circuito de regulación de temperatura que se diseñará a continuación.

Figura 9. **Opto acoplador**



Fuente: *Opto acoplador*. <http://3.imimg.com/data3/PH/VX/MY-2644000/transistor-optocoupler-250x250.jpg>. Consulta: julio de 2015.

Tabla V. **Especificaciones del opto acoplador**

INFRARED EMITTING DIODE			
Reverse Voltage	V_R	3	Volts
Forward Current — Continuous	I_F	60	mA
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Negligible Power in Transistor Derate above 25°C	P_D	100 1.33	mW mW/ $^\circ\text{C}$
OUTPUT DRIVER			
Off-State Output Terminal Voltage	V_{DRM}	250	Volts
Peak Repetitive Surge Current (PW = 1 ms, 120 pps)	I_{TSM}	1	A
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	300 4	mW mW/ $^\circ\text{C}$

Fuente: *Especificaciones del opto acoplador*. <http://www.danomsk.ru/pdf/145319.pdf>.
Consulta: julio de 2015.

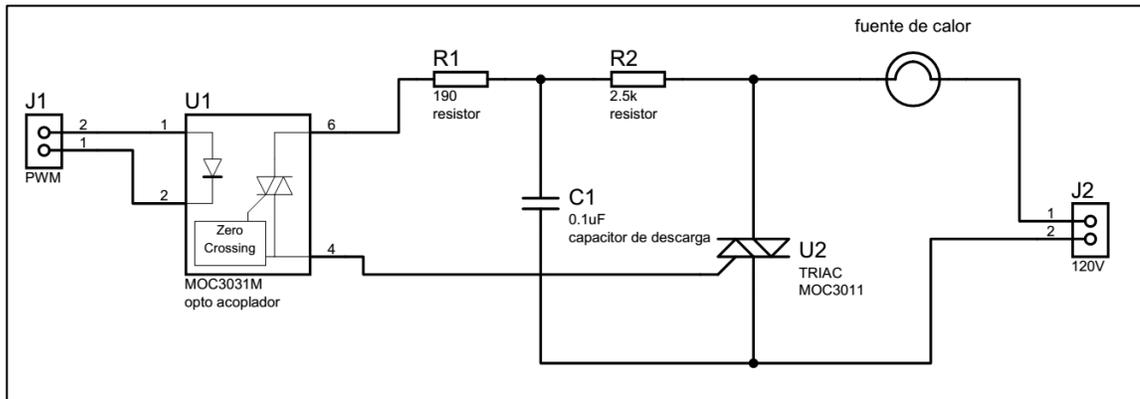
Este opto acoplador cumple con los requisitos mínimos de potencia del circuito regulador de calor. Esto representa una medida de seguridad en el momento de operación del mismo.

Al tener los requisitos mínimos de 250W y 1 A evita dañar otros equipos o al habitante del ecosistema con termorregulación. Esta es la primera barrera de protección de todo el sistema de control. El Triac podrá seguir operando hasta 12 A y 800 V, pero si el opto acoplador no envía la señal de conmutación todo el circuito se desactiva.

2.4. Diseño del circuito regulador de potencia para la lámpara de calor

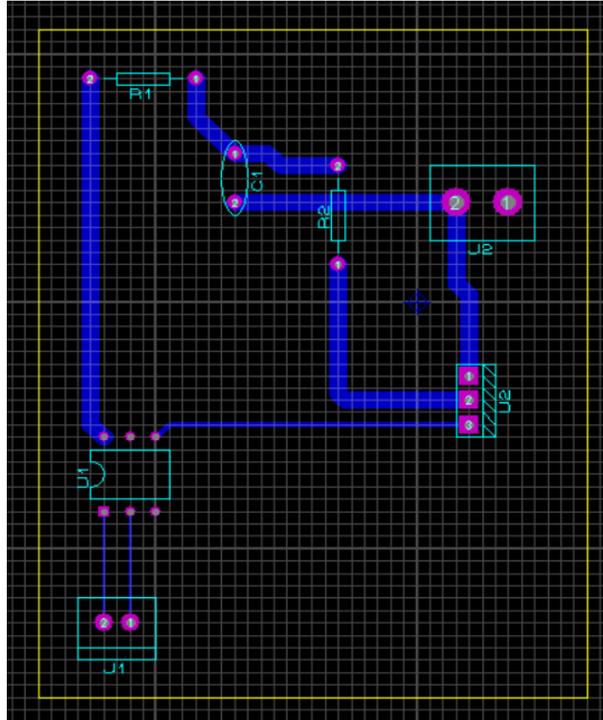
A continuación se presenta el diseño realizado en Proteus 8 Professional.

Figura 10. Diagrama de conexiones



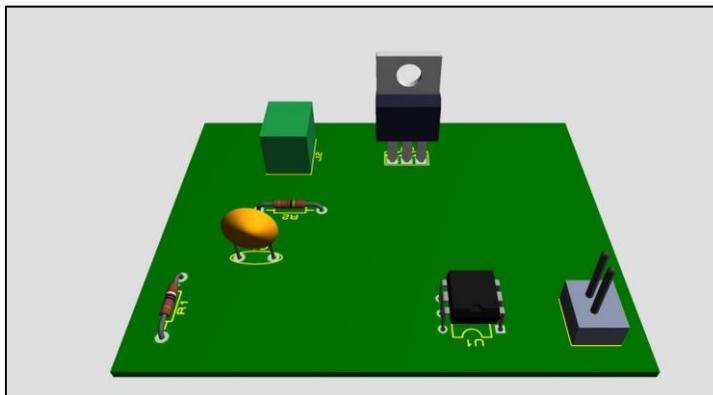
Fuente: elaboración propia, empleando Proteus 7.

Figura 11. Diagrama del circuito electrónico



Fuente: elaboración propia, empleando Proteus 7.

Figura 12. Modelo en 3D



Fuente: elaboración propia, empleando Proteus 7.

2.5. Sensor de humedad

Debido a que el ambiente en el que desarrollan los organismos ectotermos, necesitan una humedad relativa controlada, es necesario medirla. Existen varios tipos de tecnología aplicables a la medición de humedad. El principio básico es que las propiedades de cierto material o elemento cambian cuando entra en contacto con la humedad relativa del ambiente.

Hay dos tipos comunes de sensores de humedad:

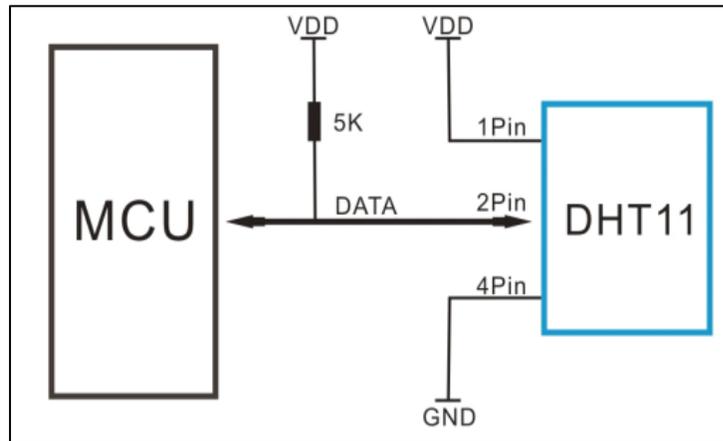
- Resistivos
- Capacitivos

El que se va a utilizar para controlar la humedad relativa del hábitat para organismos ectotermos es un dht11. Este posee un elemento resistivo de sensado de humedad.

Este sensor es digital y provee una salida con resolución de 8 bits. Esto traduce en una incerteza de humedad relativa de 5 %, su rango de medición es de 20-90 % de humedad relativa.

Debido a que la comunicación es digital, es necesario establecer un protocolo de comunicación entre el sensor y la unidad de control. Según las especificaciones del sensor este funciona con tecnología de transmisión de 1 hilo, la cual se presenta a continuación:

Figura 13. **Conexión del sensor dht 11**



Fuente: elaboración propia, empleando Photoshop.

Este sensor posee cuatro pines de conexión:

- Pin 1: conexión a la fuente de poder
- Pin 2: conexión de datos de 8 bits hacia el sistema de control
- Pin 3: no utilizado
- Pin 4: referencia a tierra

El resistor conectado al pin 2 es utilizado para evitar que el ruido eléctrico o el causado por otras fuentes interrumpa la transmisión de datos. Para distancias de transmisión menores a 20 m se utiliza un resistor de 5 k Ω . Para propósitos del diseño del hábitat este resistor está dentro del rango.

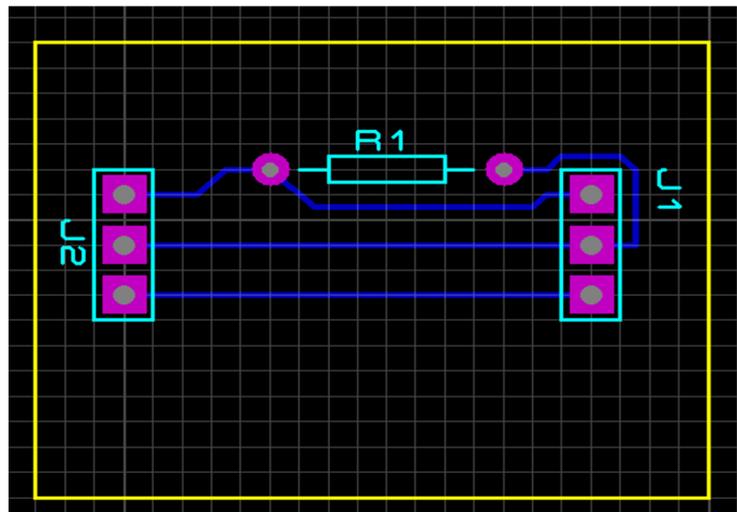
Tabla VI. Especificaciones del sensor

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH 8 Bit	1%RH
Repeatability			± 1%RH	
Accuracy	25°C		± 4%RH	
	0-50°C			± 5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25°C, 1m/s Air	6 S	10 S	15 S
Hysteresis			± 1%RH	
Long-Term Stability	Typical		± 1%RH/year	
Temperature				
Resolution		1°C	1°C	1°C
		8 Bit	8 Bit	8 Bit
Repeatability			± 1°C	
Accuracy		± 1°C		± 2°C
Measurement Range		0°C		50°C
Response Time (Seconds)	1/e(63%)	6 S		30 S

Fuente: *Especificaciones del sensor*. <http://akizukidenshi.com/download/ds/aosong/DHT11.pdf>

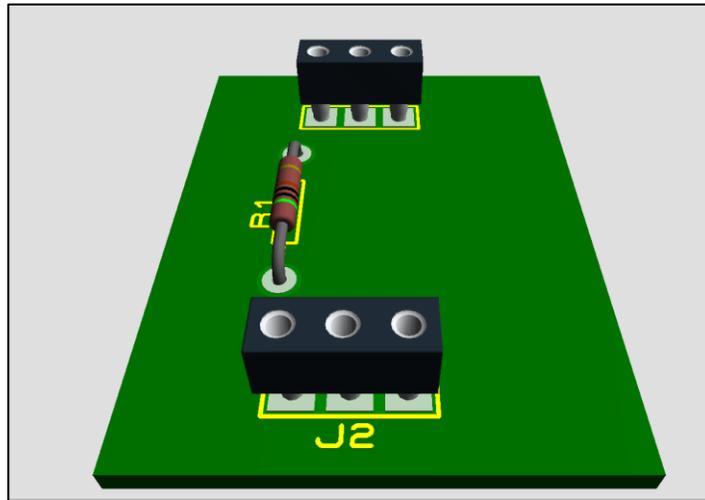
Consulta: julio de 2015.

Figura 14. Diseño de la placa



Fuente: elaboración propia, empleando Proteus 7.

Figura 15. **Modelo en 3D**



Fuente: elaboración propia, empleando Proteus 7.

2.6. **Humidificador común**

Es un aparato que sirve para cambiar la humedad relativa del ambiente en donde se utiliza, en este caso, el hábitat de condiciones reguladas.

El humidificador que se estará empleando en este diseño, es un humidificador comercial frío. Estos producen la nebulización del agua mediante vibraciones de alta frecuencia, lo cual los hace muy silenciosos e ideales para la interacción con seres vivos. Consumen muy poca potencia (20-35W) y son regulables.

Al ser comerciales, ya vienen listos para conectarse a la red y ser utilizados es un aparato completamente funcional y no se requiere de ninguna configuración. Como es un diseño modular, si el humidificador se llegara a estropear o presentara algún fallo es completamente reemplazable por otro, de

igual o diferente fabricante, no es necesario un humidificador específico. Esto facilita la construcción y reparación del hábitat.

Figura 16. **Humidificador ultrasónico**



Fuente: *Humificador ultrasónico*. <http://previo.vppharma.com/wp-content/uploads/humificador-ultrasonico-vp-natura.jpg>. Consulta: julio de 2015.

2.6.1. Conexión del humidificador al sistema de control

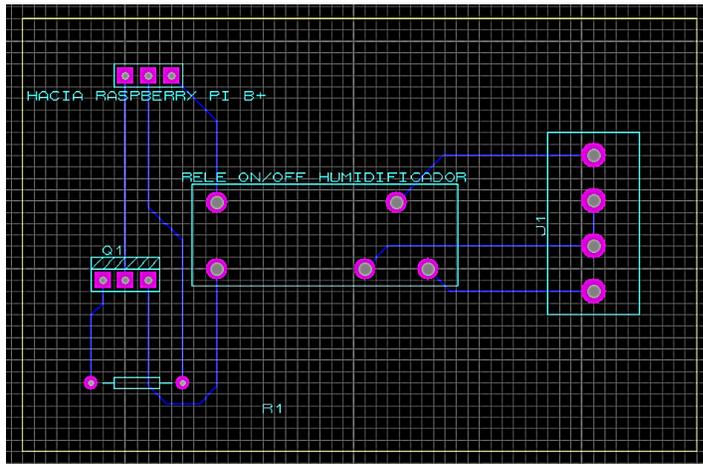
Dadas las condiciones de operación del humidificador y debido a que este es comercial, el sistema de control se vale de una placa. Esta contiene un *relay* de corriente alterna para apagar y encender por intervalos el humidificador, hasta lograr las condiciones de humedad relativa deseadas.

Esto presenta también una barrera de protección; para la electrónica de control el *relay* es activado por un transistor; el humidificador, por el *relay*. De esta manera se aísla completamente un circuito de otro.

2.6.2. Diseño del circuito del humidificador

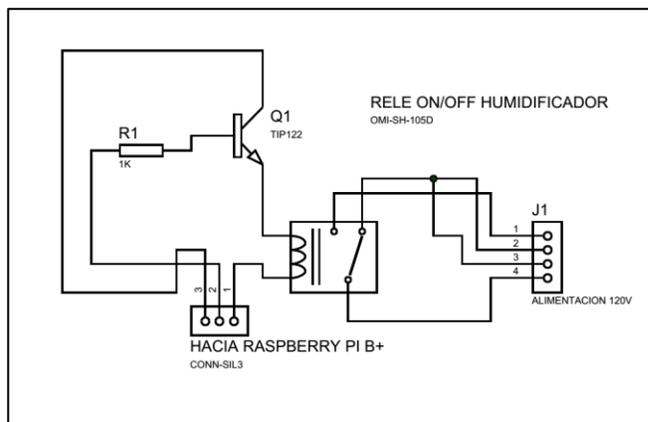
A continuación se presenta el diagrama del circuito del humidificador.

Figura 17. Diagrama del circuito eléctrico



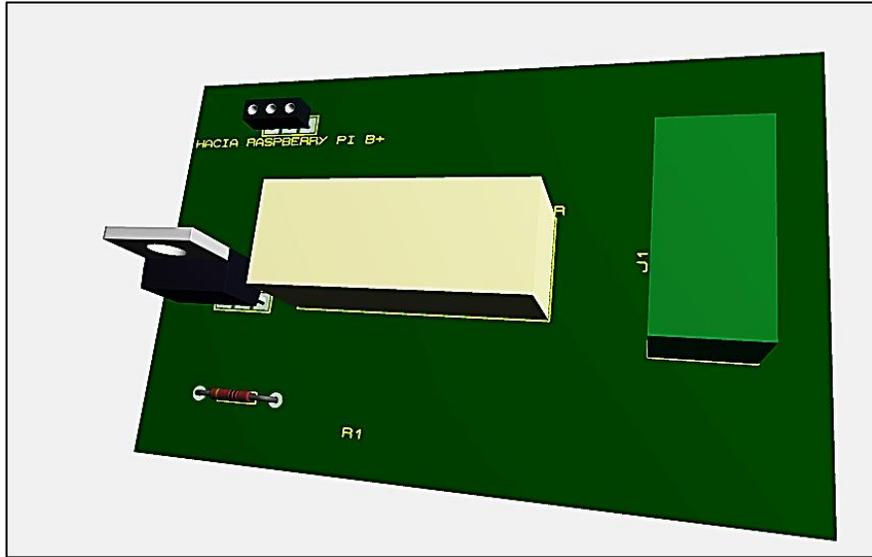
Fuente: elaboración propia, Proteus 7.

Figura 18. Diagrama de conexiones



Fuente: elaboración propia, Proteus 7.

Figura 19. **Modelo en 3D**



Fuente: elaboración propia, empleando Proteus 7.

3. MÓDULO DE CONTROL

Toda la tecnología presentada en el capítulo anterior no funciona por sí sola, es necesario tener un módulo o unidad de control que logre hacer que todas las mediciones y circuitos de operación converjan en un solo sistema. Con esto se busca una solución costeable y fácil de utilizar para que no solo el sistema opere de una manera sencilla, sino que también el individuo que vaya a hacer uso de este sistema tenga una experiencia amigable e intuitiva.

La solución provista para el módulo de control es un ordenador de placa única de bajo costo llamado Raspberry Pi. Básicamente, es una computadora de una sola placa electrónica con todas las funciones integradas, capaz de realizar cálculos, controlar salidas y entradas (analógicas y digitales) y desplegar datos en pantalla.

3.1. La Raspberry Pi como controlador

Ya presentado este ordenador como la solución al problema se procede a examinar sus funciones y los requerimientos básicos de control para un hábitat de organismos ectotermos.

3.1.1. Especificaciones del ordenador

Este sistema en un solo chip consta de diferentes tipos de interfases de comunicación necesarias. Para este proyecto, estos son: la comunicación digital serial, los módulos digitales de entrada/salida (I/O Pins), los de señales analógicas y los de control de ancho de pulsos.

El ordenador es el Raspberry Pi B+ que cuenta con 512Mb de memoria RAM y un procesador Broadcom de arquitectura ARM funcionando a 700 MHz.

Estas son las especificaciones completas:

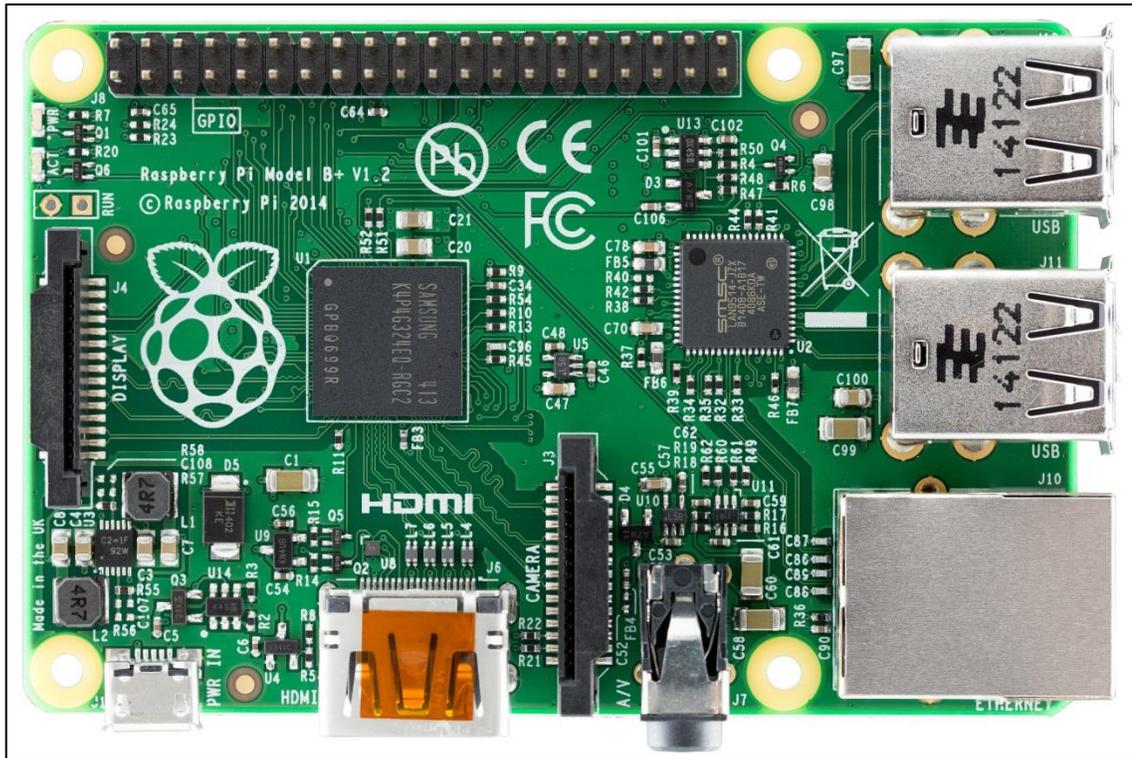
Tabla VII. **Especificaciones del ordenador Raspberry Pi B+**

Chip	Broadcom BCM2835 SoC
Arquitectura	ARM11
CPU	Núcleo único ARM1176JZFS 700MHz
GPU	Dual Core VideoCore IV
Memoria	512 Mb SDRAM
Sistema operativo	Raspbian (imagen de LINUX optimizada para el ordenador)
Alimentación de poder	Via MicroUSB 5V, 2ª
Ethernet	Socket de ethernet . 10/100 BaseT
Video	HDMI (revisión 1.3 y 1.4)
Audio	HDMI y Jack de 3,5"
USB	4 puertos usb 2,0 con capacidad de 1,2ª
Interfaz I/O	40 pines que suplen 27 pines de entrada/salida y alimentación partida de 3,3V y 5V
Memoria	Micro SD 32 Gb

Fuente: *Especificaciones del ordenador Raspberry Pi B+*. <http://www.geeetech.com/wiki/images>.

Consulta: julio de 2015.

Figura 20. Raspberry Pi B+



Fuente: *File:Raspberry Pi B+ to.* https://upload.wikimedia.org/wikipedia/commons/6/6f/Raspberry_Pi_B%2B_top.jpg. Consulta: julio 2015

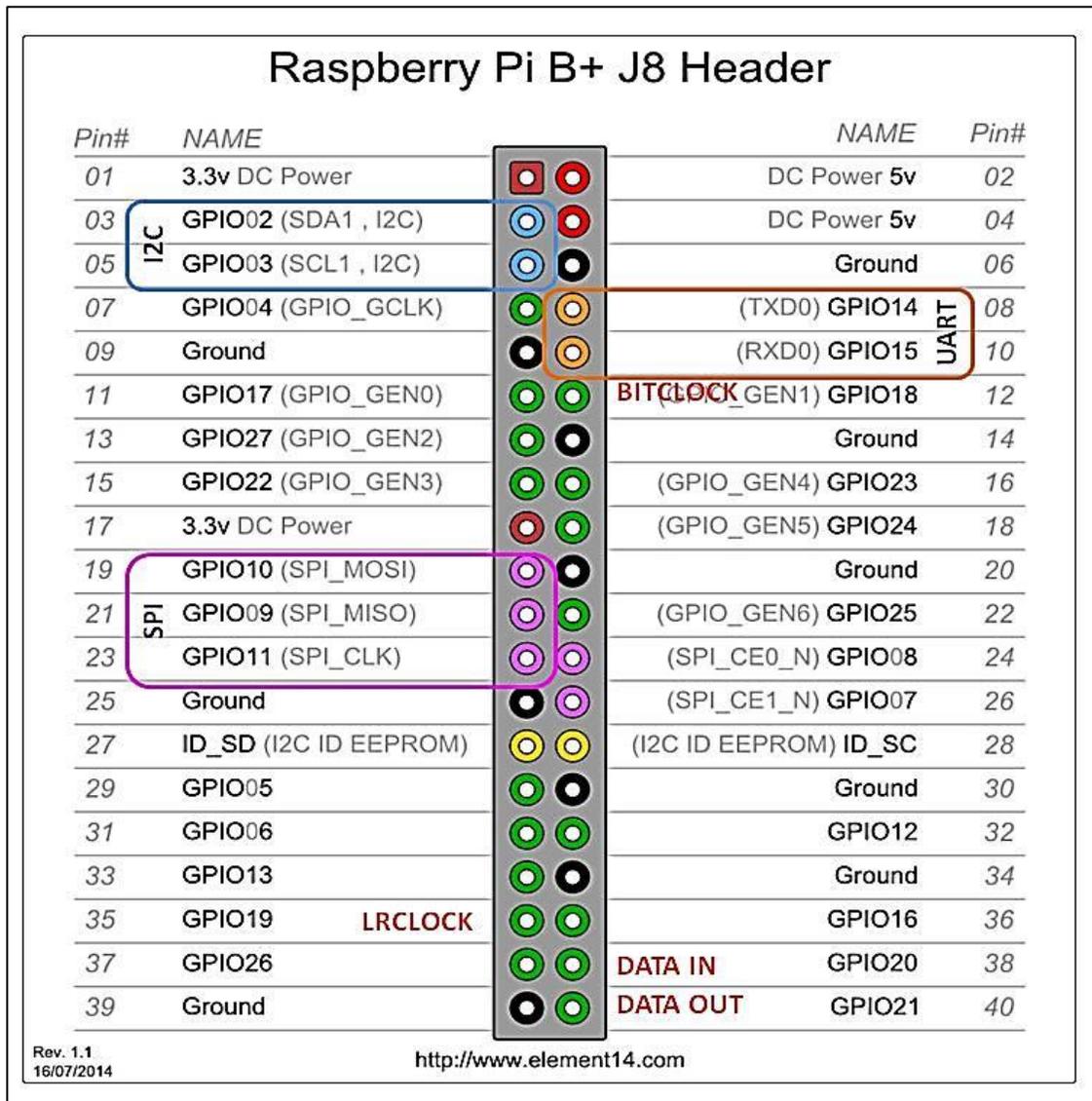
3.1.2. GPIO

Puesto que la Raspberry Pi se utilizará como un sistema de control, esta necesita entradas de retroalimentación. Es necesario configurar los pines de entrada y salida de señales que en la Raspberry se conocen como *General purpose input/output (GPIO)*.

Estas entradas y salidas serán la base de la comunicación entre periféricos y controlado PID.

A continuación se presenta un diagrama general de los puertos de entrada y salida de la Raspberry Pi b+:

Figura 21. Diagrama de pines GPIO



Fuente: *GPIO*. <https://hifiduino.files.wordpress.com/2014/11/i2spins-001.jpg>.

Consulta: julio de 2015.

3.2. Software y conceptos de control para la Raspberry Pi b+

Para unificar todos los circuitos y realizar la teoría de control es necesario desarrollar un *software* que se encargue tanto del control como del monitoreo de las señales. La herramienta que mejor sirve para este propósito es un controlador PID.

3.2.1. Control proporcional integral diferencial

Los controladores PID son usados en una gran cantidad de aplicaciones, entre estas el control industrial. El control proporcional integral diferencial se define así por sus parámetros; el proporcional que controla el error en tiempo real, el integral que se vale de los errores pasados para corregir una señal y el diferencial que puede predecir los errores futuros.

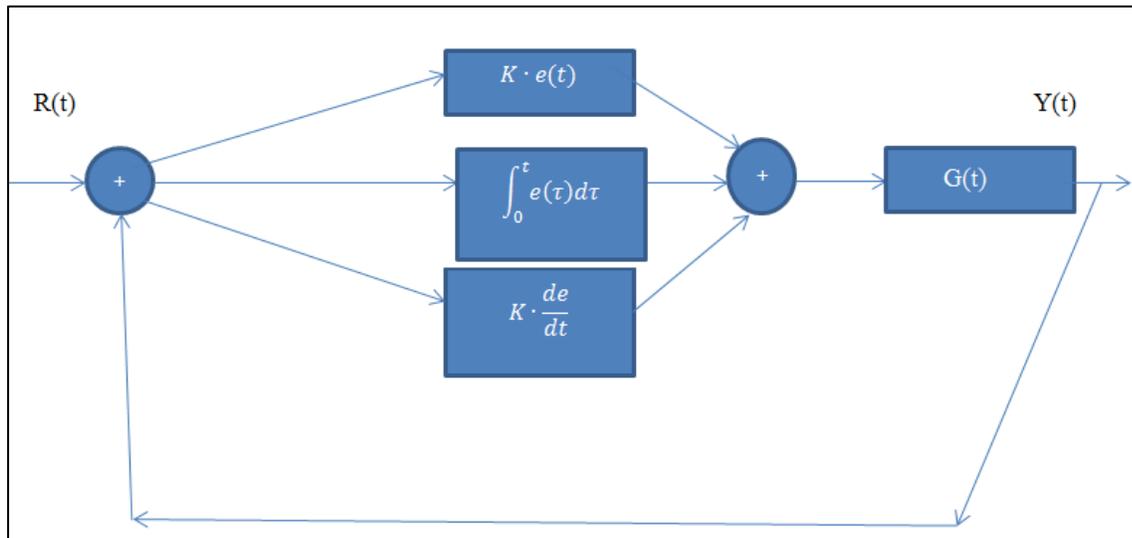
El principio de funcionamiento de este controlador es tomar una señal y compararla contra una señal de referencia. Esta señal de referencia está dada por la suma de los tres parámetros antes mencionados.

Esta metodología de control es la más apropiada para este sistema de control, puesto que responde muy bien a los procesos desconocidos. Debido a que el control de temperatura en un ambiente con seres vivos puede ser muy aleatorio, este sistema va a responder muy bien a la hora de corregir temperatura de manera rápida y exacta.

Otra ventaja del control PID es que es bastante versátil y puede ser implementado fácilmente.

Para implementarlo se describe su estructura:

Figura 22. Estructura control PID



Fuente: elaboración propia, empleando AutoCAD.

Posee una señal de entrada $R(s)$ que es la que se compara con la señal de error producida por la señal de retroalimentación $H(s)$. Esta al ser comparada pasa por los parámetros integral, proporcional, diferencial, y por una ganancia adicional que provee a la retroalimentación de mayor estabilidad.

El resultado es una salida controlada $Y(s)$ que se cambia y monitorea por la retroalimentación.

- Control proporcional: este pilar del sistema de control define el error como una constante proporcional de la variable controlada, es decir que el cambio en la señal es un múltiplo del porcentaje de cambio de medición.

- Control integral: este parámetro elimina el error estacionario, suma todos los errores proporcionales y realiza la corrección. Es más lento que el proporcional.
- Control diferencial: trata de corregir los errores futuros midiendo la oscilación y controlando que no haya un exceso de la misma.

La salida de estos tres parámetros se suma y da como resultado la salida controlada del controlador PID. Este se describe de la siguiente forma:

$$Y(t) = Ke(t) + K \int_0^t e(\tau) d\tau + K \frac{de}{dt}$$

3.2.2. Creación de las rutinas de control y entrada/salida para el programa principal

Para utilizar la computadora como controlador PID es necesario desplegar el paquete de gráficos de computador, dentro del programa de Python. Este permitirá utilizar la clase PIDController() que creará un objeto de control PID con el que se define la temperatura por controlar. La primera etapa del despliegue es descargar el paquete de desarrollo cgkit 2.0.0 para Python. Este software es *open source* de la licencia Mozilla public licence. Por ello se usa cómodamente dentro del desarrollo; al ser *open source* no presenta ningún costo y posee bastante soporte.

Figura 23. Rutinas de control y entrada/salida para el programa principal

```
41 #Controlador PID
42 class PIDController(component.Component):
43     """PID controller.
44     """
45
46     exec slotPropertyCode("input")
47     exec slotPropertyCode("output")
48     exec slotPropertyCode("setpoint")
49     exec slotPropertyCode("max")
50     exec slotPropertyCode("min")
51     exec slotPropertyCode("integr")
52     exec slotPropertyCode("deriv")
53     exec slotPropertyCode("propor")
54
55     def __init__(self,
56                 name = "PIDController",
57                 setpoint = 0.0,
58                 propor = 0.0,
59                 integr = 0.0,
60                 deriv = 0.0,
61                 max= 999999,
62                 min = -999999,
63                 auto_insert = True):
64         """Constructor.
65         """
66
67         component.Component.__init__(self, name, auto_insert)
68
69         self.input = DoubleSlot()
70         self.setpoint= DoubleSlot(setpoint)
71         self.max= DoubleSlot(maxout)
72         self.min= DoubleSlot(minout)
73         self.propor= DoubleSlot(Kp)
74         self.integr = DoubleSlot(Ki)
75         self.deriv = DoubleSlot(Kd)
76
77         self.output = ProceduralDoubleSlot(self.computeOutput)
78
79         self.addSlot("input", self.input)
80         self.addSlot("setpoint", self.setpoint)
81         self.addSlot("output", self.output)
82         self.addSlot("max", self.max)
83         self.addSlot("min", self.min)
84         self.addSlot("propor", self.propor)
85         self.addSlot("integr", self.integr)
86         self.addSlot("deriv", self.deriv)
87
88         self.input.addDependent(self.output)
89         self.setpoint.addDependent(self.output)
90         self.max.addDependent(self.output)
91         self.min.addDependent(self.output)
92         self.propor.addDependent(self.output)
93         self.integr.addDependent(self.output)
94         self.deriv.addDependent(self.output)
95
```

Continuación de la figura 23.

```
100     self._integraltotal = 0.0
101     self._error_previo = 0.0
102
103     eventmanager.eventManager().connect(events.STEP_FRAME, self)
104     eventmanager.eventManager().connect(events.RESET, self)
105
106
107     def onStepFrame(self):
108         err = self.setpoint-self.input
109         dt = getScene().timer().timestep
110         self._integraltotal += dt*err
111
112     def onReset(self):
113         self._integraltotal = 0.0
114         self._error_previo = 0.0
115
116     def computeOutput(self):
117
118         err = self.setpoint-self.input
119         dt = getScene().timer().timestep
120         I = self._integraltotal
121         D = (err-self._prev_err)/dt
122         # print "D:",D
123         res = self.propor*err + self.integral*I + self.deriv*D
124
125         self._error_previo = err
126
127         maximo = self.max
128         minimo = self.min
129         if res>maximo:
130             res = maximo
131         elif res<minimo:
132             res = minimo
133
134
135         return res
136
```

Fuente: elaboración propia, empleando Python y el editor de texto NotePad ++.

Una vez instalado se utilizan las propiedades de la clase para crear un objeto temperatura dentro del archivo TEMPPID.py creado para controlar la temperatura.

Figura 24. Controlador de temperatura

```
26 #controlador de temperatura
27
28 TEMPCON=PIDController() #se define la clase PIDController
29 var_controlada= TEMPCON.input #se define la variable controlada
30 setp= TEMPCON.setpoint #se define el setpoint
31
32 integral= TEMPCON.propor #constante proporcional
33 proporcional=TEMPCON.integr #constante integral
34 derivativo=TEMPCON.deriv # constante derivativa
35 TEMPCON.output_slot=salidaPID #salida
36
37 #se actualiza el valor de la salida constantemente en un ciclo.
38 while True: TEMPCON.output=salidaPID
39
```

Fuente: elaboración propia, empleando NotePad ++.

Después de haber creado el objeto de control es necesario unir el sistema a las salidas analógicas y digitales del ordenador. Esto para comunicar los datos al sistema de *hardware*. Para esto es necesario descargar la librería GPIO para Python de la misma manera que se descarga cgkit.

Al hacer esto se añade la rutina de control al archivo TEMPPID.py. Lo anterior espera que se pueda controlar la modulación de ancho de pulsos en el sistema, por medio del controlador previamente realizado.

3.2.2.1. PWM

La modulación de ancho de pulsos es un concepto que sirve para traducir valores binarios al equivalente de una señal analógica, por medio de modulación de voltaje. Esto se logra haciendo un tren de pulsos cuadrados más

ancho o más angosto para que devuelva un ciclo de trabajo promedio que puede ser traducido como un voltaje analógico. Este se utiliza para modular motores y variables que cambien con un voltaje analógico

La modulación de ancho de pulsos puede ser hecha en la Raspberry Pi. Esto por medio de librerías de *open source* similares a las del grupo de gráficos solo que esta controlará los pines de entrada y salida. Es necesario definir el ciclo de trabajo como la salida del controlador PID. Para que esto funcione, se hace de la siguiente manera:

Figura 25. **Salida del controlador PID**

```
6 import component
7 import eventmanager, events
8 from scene import getScene
9 from slots import *
10 from cgtypes import *
11 import _core #se importan las librerias necesarias para el controlador PWM
12 import RPi.GPIO as GPIO #se importa la libreria de entradas y salidas
13
14
15 #modulacion de ancho de pulsos
16
17
18 GPIO.setmode(GPIO.BCM) #se llama a la libreria a utilizar
19 GPIO.setup(18, GPIO.OUT) #se define el pin de salida 18
20 salida_regulada_temperatura = GPIO.PWM(18, 50) #se define la salida como modulacion de ancho de pulsos
21 salida_regulada_temperatura.start(salidaPID)#se une la temperatura controlada a la salida del pwm
22
```

Fuente: elaboración propia, empleando NotePad ++.

Una vez teniendo el sistema de control es necesario tener las variables de referencia proporcionadas por el sensor de humedad y de temperatura. Esto para el ds18b20 y agregarlo a la programación.

Al definir definido las variables de referencia es necesario hacer el diseño de la interfaz y la base de datos.

3.3. Creación de la interfaz de control como página web

Un servidor es un ordenador o computadora capaz de brindar un servicio especializado. La mayoría de servidores proveen almacenamiento de páginas web, resolución de dominios, sincronización de tiempo o transferencia de archivos.

3.3.1. Creación de un servidor LAMP

Para establecer un servicio de almacenamiento e interpretación de páginas web, es necesario crear un servicio que entienda instrucciones que el usuario pueda introducir fácilmente. Estos pueden ser botones o casillas, y que la rutina de control pueda utilizar. Esto se hace mediante una base de datos.

Básicamente, se desarrolla una aplicación que pueda comunicar bidireccionalmente las necesidades del cliente hacia la rutina de control y el comportamiento del sistema. Esto cuando ya tiene parámetros dados para su funcionamiento.

Un servidor LAMP es el que une todas estas características LAMP, y significa:

- Linux (plataforma operativa)
- Apache (servidor HTTP)
- Mysql (servidor de base de datos)
- PHP (lenguaje de programación utilizado para conectar los servidores)

Está diseñado para prestar un servicio de interpretación de instrucciones y comunicación vía HTTP. Cada parte del acrónimo es una aplicación destinada a un propósito en específico.

3.3.1.1. Linux

Se ha descrito en este capítulo la programación y la rutina de control hecha en Python para el control del sistema, más no al sistema operativo encargado de almacenar y correr todas las aplicaciones que se desarrollan.

GNU/Linux es una plataforma libre y de código abierto. Esto permite amplias posibilidades a la hora de desarrollar *software*. La RaspberryPi funciona con una distribución de GNU/Linux llamada Raspbian, la cual está basada en una distribución robusta llamada Debian Linux.

La plataforma Raspbian provee soporte y herramientas para compilar las aplicaciones de control e interfases de salida manejando todo dentro de un solo procesador, haciendo la comunicación entre sistema y periféricos bastante fácil. Los servidores que a continuación se describirán son todos programas compilados para esta distribución de Linux. Es un sistema operativo bastante intuitivo diseñado para el aprendizaje. Esto representa una ventaja puesto que hay varios servicios, como el de base de datos, que podrían complicarse.

Al ser un sistema operativo completo este cuenta con las aplicaciones necesarias para que el usuario no solo pueda operar el sistema de control sino también lo pueda cambiar a gusto. Las aplicaciones utilizadas en el sistema son las siguientes:

- Python (compilador)
- Midori (explorador web)
- Terminal (interpretar instrucciones y comandos)

El código de las aplicaciones se programa en texto plano y se compila por separado para cada aplicación.

3.3.1.2. Apache

Ya teniendo un sistema operativo, se necesita un servicio que sea capaz de enviar páginas web, estáticas o dinámicas, a la aplicación de explorador web (midori en este caso). Apache es un paquete de *software* que permite almacenar e interpretar código de páginas web en distintas plataformas, el hecho de que sea una aplicación modular, multiplataforma y de código abierto la hace bastante robusta e ideal para la creación de un servidor web.

Este servidor es la base de la interfaz vía web del sistema de control. Sin la plataforma del servidor apache no sería posible manejar el sistema de control como una página web.

3.3.1.3. Mysql

Es un servidor gestor de bases de datos, al instalarse en el sistema operativo Linux es capaz de crear tablas de almacenamiento de datos. Este permite a distintas aplicaciones extraer la información de dichas tablas.

Posee su propio lenguaje para hacer consultas y para comunicarse con otras aplicaciones. Todos los programas y todas las aplicaciones utilizadas para

el desarrollo de este sistema tienen soporte y comunicación con Mysql haciéndolo idóneo para el proyecto.

Es una base de datos muy rápida y bastante fácil de usar.

3.3.1.3.1. Creación de la base de datos para temperatura y humedad

Para el sistema de control se crea una base de datos sencilla sobre la plataforma de Mysql. es necesario definirla ahora puesto que después será ampliamente utilizada en el diseño de la página web.

Existen varias maneras de crear una base de datos en Mysql, la más fácil es crear un archivo texto que contendrá los índices y características de la tabla de datos. Una vez más se programa en texto plano con la extensión *.sql.

La base de datos por utilizar en este sistema de control se llama temperatura SQL. Esta base contiene los datos que el PID estará utilizando y modificando y, así mismo, contiene los datos de humedad del sistema. (ver apéndice 1)

Referencia a la foto de la tabla de base de datos.

Cuando ya se tiene la base de datos con la tabla, se llama al archivo desde otras aplicaciones y se introducen datos a la tabla dinámicamente desde otra aplicación. Esto es exactamente lo que se busca, que la interfaz web haga consultas a la base de datos y la pueda modificar con llamadas a la tabla, lo cual se puede hacer con el lenguaje PHP.

3.3.1.4. PHP

Luego de crear la tabla con un lenguaje propio y con la capacidad de almacenar los datos que se van a utilizar dentro de la aplicación, es momento de desarrollar un entorno de manejo global. Esto para unificar cada servicio y lograr con un lenguaje que abarque todo lo mencionado previamente y el soporte HTTP. Ese lenguaje es PHP.

PHP es un lenguaje de programación orientado a servidores, a desarrollo web y edición de contenido dinámico. PHP es uno de los lenguajes más flexibles que existen hasta ahora y uno de los más sólidos también. El servicio de PHP no solo viene con un compilador de código fuente, viene también con una línea de comandos para interpretar distintas aplicaciones y comandos.

PHP tiene compatibilidad con Mysql, Linux, Apache y también permite hacer llamadas a rutinas hechas en el programa Python. Por ello, representa la base ideal para desarrollar la página web de interfaz que se va a llamar `SERVIDOR_TEMPERATURA.html`

PHP cuenta con la capacidad de interpretar lenguaje html para modelado de páginas web. Por ello se puede hacer un entorno visual bastante amigable para el usuario (ver apéndice 2).

3.3.1.5. HTML

HTML es el estándar para modelado de páginas web. Es un lenguaje que provee estructuración para contenido de páginas web, permitiendo así el despliegue de contenido dinámico como videos o tablas dentro de una página web.

Puesto que HTML es un estándar, cualquier explorador es capaz de leer e interpretar las instrucciones dentro del texto programado. HTML es el vínculo entre PHP y el explorador web.

3.3.2. Diseño de la página web

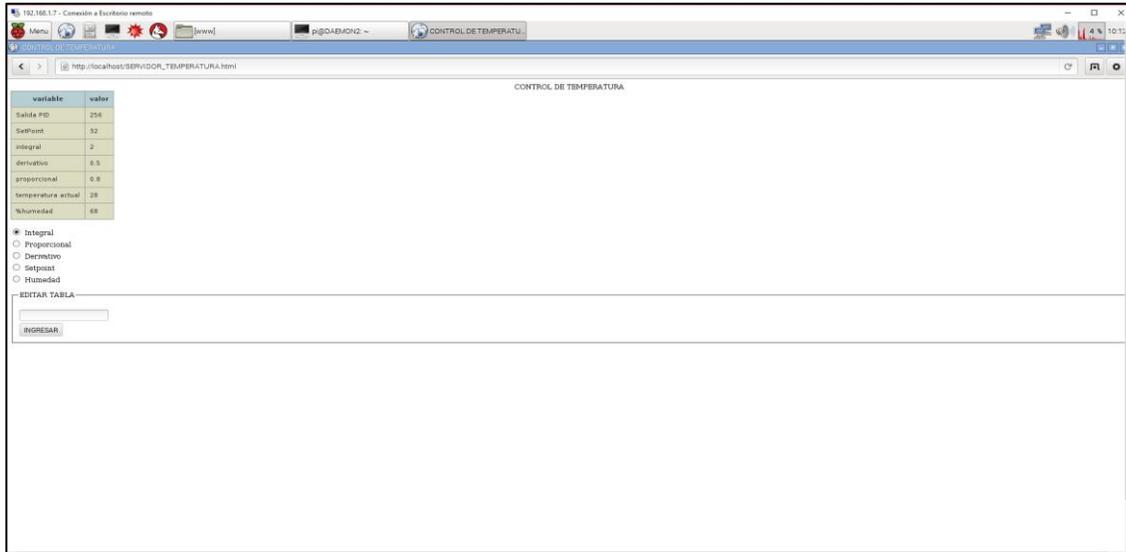
Una vez definidos los fundamentos de programación y las herramientas necesarias, para el diseño web, se procederá a crear la página. Esto con el contenido necesario para el funcionamiento del programa.

La página contiene el código para llamar a la base de datos. Así como la tabla de despliegue de información, el botón para editar y posee también las llamadas a la rutina de control.

Unido mediante el estándar HTML se puede dar un formato a la página, y direccionar las rutinas que se encuentran en la misma carpeta que la página web.

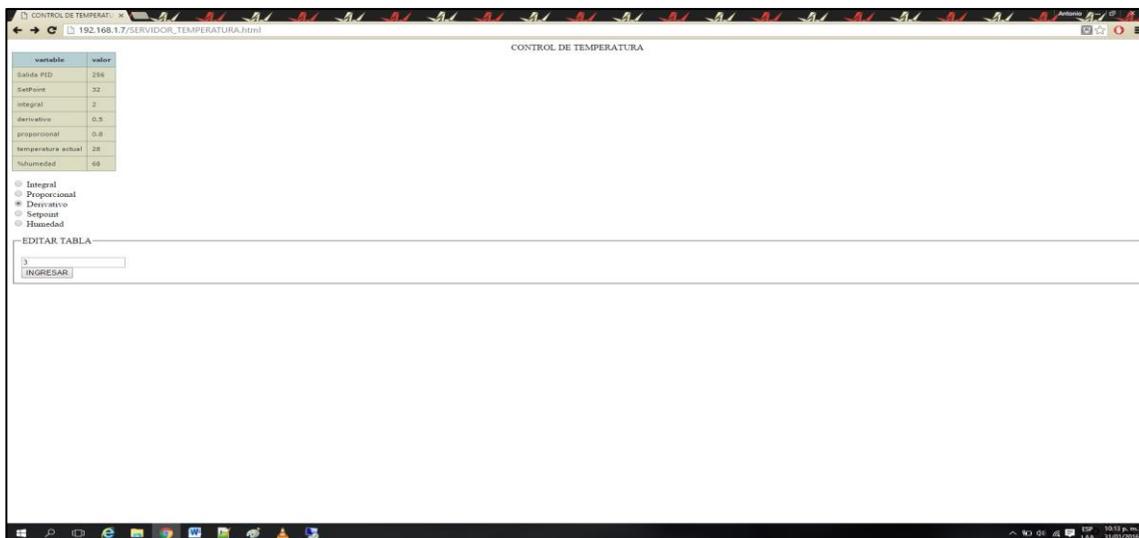
Se accede a la página desde un cliente mediante la dirección 192.168.1.7/SERVIDOR_TEMPERATURA.html, o desde el servidor web localhost/var/www/SERVIDOR TEMPERATURA.html

Figura 26. Raspberry Pi



Fuente: elaboración propia, empleando Midori y PHP.

Figura 27. Cliente remoto



Fuente: elaboración propia, empleando Chrome y Apache Web server.

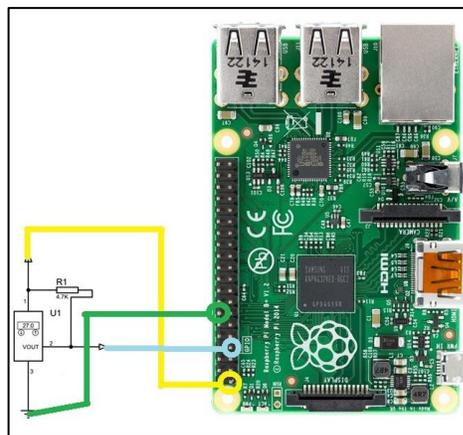
Conexión de los módulos de medición y ajuste al sistema de control

3.3.3. Entradas

Al haber desarrollado el código fuente para la conexión del sistema, es necesario establecer las conexiones físicas y el hardware para que todo funcione adecuadamente.

El primer módulo por conectar es el de medición de temperatura, el DS18B20. Este sensor va a pin 2 para alimentación de cinco voltios y nueve para tierra los bits de información se envían hacia el pin 4.

Figura 28. **Medición de temperatura DS18B20**

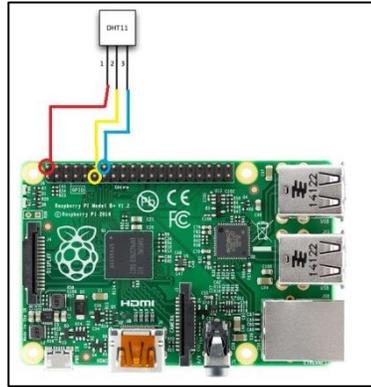


Fuente: *Medición de temperatura*. <https://hifiduino.files.wordpress.com/2014/11/i2spins-001.jpg>.

Consulta: julio de 2015.

El segundo módulo por conectar es el sensor de humedad DHT11. Este va conectado al pin 11 para la entrada de datos. Este sensor requiere de 3,3 voltios de alimentación por lo que va conectado al pin 1 y al pin 6 para tierra.

Figura 29. **Sensor de humedad DHT11**



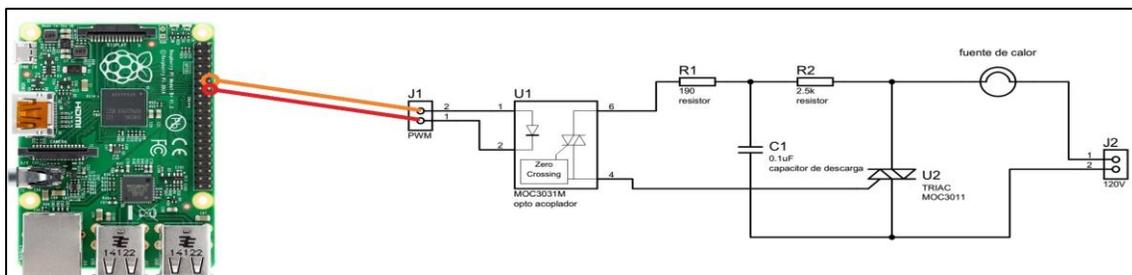
Fuente: *Sensor de humedad*. <https://hifiduino.files.wordpress.com/2014/11/i2spins-001.jpg>.

Consulta: julio de 2015.

3.3.4. Salidas

El primer módulo de salida que se conectará a la Raspberry Pi comparte tierras con el módulo DHT11 en el pin 6 y la salida PWM. Este programa en el pin 12 esto es lo único necesario para acoplar el módulo a la placa de control.

Figura 30. **Conexión de salida**

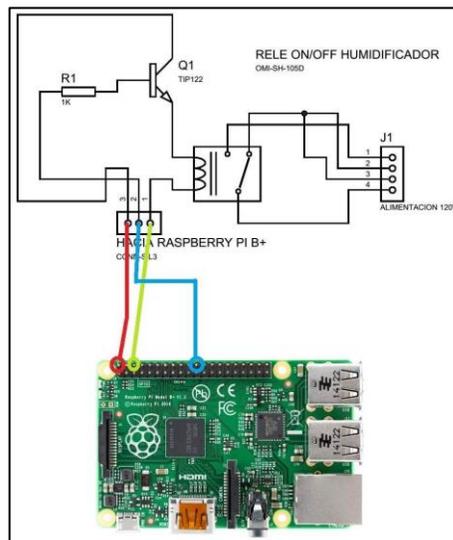


Fuente: *Conexión de salida*. <https://hifiduino.files.wordpress.com/2014/11/i2spins-001.jpg>.

Consulta: julio de 2015.

El segundo módulo de salida es el del humidificador. Este circuito solo es una salida a *relay*. Es por lo que se utilizará la misma alimentación de cinco voltios en el pin 2, la tierra en el pin 4 y el pin de salida a relé en el pin 22.

Figura 31. **Humificador**



Fuente: *Humificador*. <https://hifiduino.files.wordpress.com/2014/11/i2spins-001.jpg>.

Consulta: julio de 2015.

4. ESCALABILIDAD DEL DISEÑO

Este sistema de control está diseñado para controlar un solo hábitat por cada sistema. Lo cual no detiene la escalabilidad del diseño, puesto que cada uno es una página web que tiene varias páginas abiertas para cada sistema de control en diferentes ambientes. Esto con base en el protocolo IP/Ethernet el cual facilita la comunicación del sistema y evita el diseño de sistemas complicados todos los datos pueden ser leídos desde cualquier computadora. Esto no solo hace el sistema escalable, sino también universal. Lo cual permite tener una red de sistemas de control tan grande como se desee diseñar.

4.1. El protocolo IP/Ethernet (IP)

Es el que se encarga de direccionar paquetes de datos a diferentes direcciones asignadas a cada usuario o cliente, esto a través de una o varias redes conectadas por medio del estándar Ethernet.

Las direcciones proporcionadas por el protocolo son tanto lógicas como físicas. Las lógicas para direccionar paquetes y las físicas para identificar clientes que puedan recibirlos.

4.1.1. El Modelo OSI

El protocolo está basado en el modelo *Open System Interconnection (OSI)* creado por la organización internacional de estándares que superó al modelo TCP/IP. Este modelo permite la separación lógica de las diferentes funciones de una red facilitando la segmentación de dispositivos y paquetes.

El modelo consta de siete capas:

- Aplicación: esta provee al usuario acceso a las aplicaciones y a las aplicaciones acceso a la red.
- Presentación: convierte la información a un formato ya estandarizado. Provee servicios de encriptación.
- Sesión: abre y cierra sesiones de puertos, también la separa lógicamente.
- Transporte: define los números de puerto para transmisión y el tipo de transmisión (confiable/no confiable).
- Red: provee direccionamiento lógico a los paquetes.
- Enlace: provee direccionamiento físico y corrige errores.
- Física: transmite los datos sobre medios eléctricos.

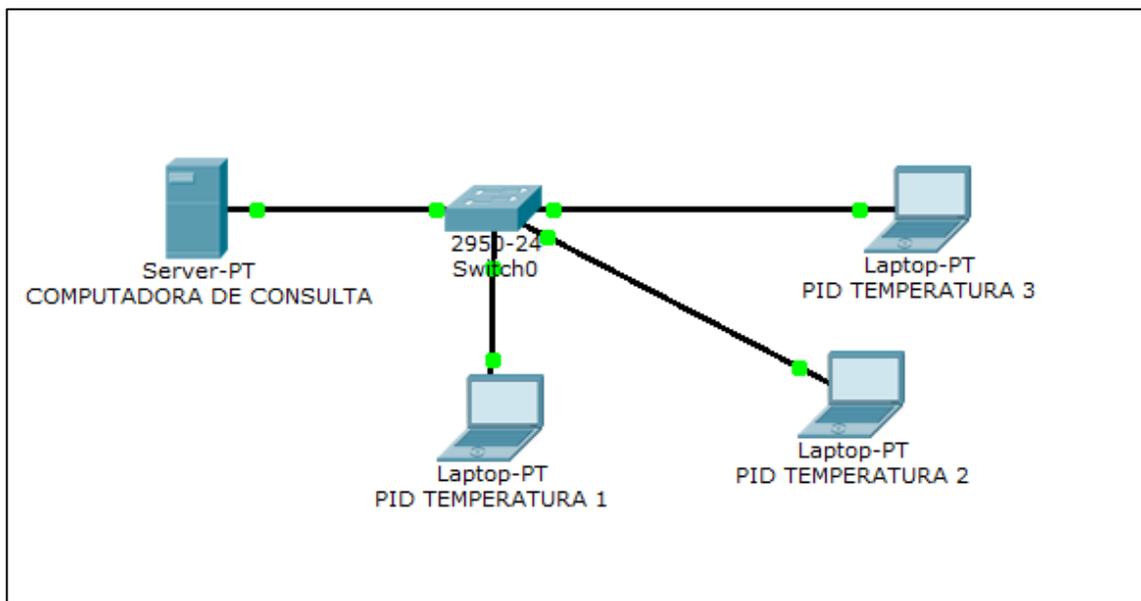
Este modelo es estudiado con fines académicos. La implementación del modelo varía dependiendo de las aplicaciones y de las plataformas de *software* el servidor LAMP maneja todas estas capas. Cada una es aplicada en uno de los diferentes servicios, pero el propósito final es el mismo y sirve perfectamente sin importar el sistema operativo.

4.2. Conectividad física

Para conectar físicamente los sistemas de control se hará uso de un *conmutador* de capa 2 encargado de direccionar físicamente los paquetes y conectar el cliente o *host* a los diferentes sistemas de control por medio de páginas web.

Para este tipo de aplicación no hay necesidad de administrar el *conmutador*.

Figura 32. Topología de red



Fuente: elaboración propia, empleando PACKET TRACER 6.1.

CONCLUSIONES

1. Existen varias especies del reino animal que necesitan de ambientes controlados para vivir.
2. La electrónica aplicada y los sistemas de control pueden utilizarse para controlar y modificar ambientes naturales, no solo procesos industriales.
3. El control de temperatura del ambiente puede mejorar la calidad de vida de los organismos que viven en el hábitat.
4. La construcción del sistema con componentes accesibles y de bajo costo permite que el diseño y la implementación sea fácil y rápida.
5. El diseño modular permite agregar o quitar periféricos que no se necesiten, el diseño completamente *open source* deja al usuario con la libertad de acomodar el sistema a sus condiciones y términos.
6. El servidor LAMP permite que el sistema sea universal, dando el uso de diferentes sistemas operativos, sin ninguna dificultad para la integración.
7. Desarrollar el sistema de control en lenguaje Python permite controlar fácilmente el *hardware* mediante librerías *open source* ya desarrolladas. Esto permite también la comunicación nativa entre base de datos y servidor web mediante PHP.

8. El uso de MySQL, como servicio de base de datos, permite la comunicación de manera nativa entre todos los demás servicios. Hace posible acceder a datos mucho más rápido y sin complicaciones.
9. Utilizar el protocolo Ethernet/IP facilita la conexión física de todos los dispositivos, al ser un sistema estandarizado los dispositivos pueden volverse *plug and play*.
10. El sistema está pensado para crecimiento futuro, y todos los componentes tanto de *hardware* como de *software* permiten replicar una y otra vez el sistema de control. Esto permite un crecimiento rápido y garantiza la compatibilidad de todos los sistemas de control.

RECOMENDACIONES

1. Es necesario estudiar las condiciones de vida óptimas de los organismos antes de hacer el despliegue del sistema de control. No se debe experimentar con condiciones aleatorias, puesto que puede dañar todo el ecosistema o incluso acabar con la vida de los organismos.
2. Buscar la manera más natural de implementar la electrónica dentro del hábitat.
3. Tratar de conservar el ambiente natural de todos los organismos
4. Aislar toda la circuitería de poder para no causar fatalidades en los organismos que habitan el ecosistema controlado.
5. Supervisar el sistema constantemente, todo sistema de control necesita monitoreo.

BIBLIOGRAFÍA

1. ALLABOUTCIRCUITS. *The triac*. [en línea]. <<http://www.allaboutcircuits.com/textbook/semiconductors/chpt-7/the-triac/>>. [Consulta: julio de 2015].
2. *Creating a LAMP server (web server – Linux Apache Mysql PHP) with the Raspberry Pi*. [en línea]. <<http://www.penguintutor.com/linux/raspberrypi-webserver>>. [Consulta: julio de 2015].
3. CURTIS. *Humificador*. [en línea]. <<http://www.curtisbiologia.com/node/1359>>. [Consulta: julio de 2015].
4. HILL, Richard. *Fisiología animal*. [en línea]. <https://books.google.com.gt/books?id=HZaC45m9IMMC&pg=PA277&lpg=PA277&dq=termorregulacion+insectos+reproduccion&source=bl&ots=jpHZHsbXDN&sig=efctV0ggfWqASaguzKmW70ufBqs&hl=es419&sa=X&ei=VfqQU8fcLK6_sQSnIYH4DQ#v=onepage&q=termorregulacion%20insectos%20reproduccion&f=false>. [Consulta: julio de 2015].
5. MYSQLTUTORIAL. *Call MySQL Stored Procedures*. [en línea]. <<http://www.mysqltutorial.org/php-calling-mysql-stored-procedures/>>. [Consulta: julio de 2015].
6. RASPBERRYPI. *GPIO setup for LM35 temperature sensor*. [en línea]. <<https://www.raspberrypi.org/forums/viewtopic.php?f=26&t=3464>>. [Consulta: julio de 2015].

7. SENDRA. *Radiación infrarroja*. [en línea]. <http://www.sld.cu/galerias/pdf/sitios/rehabilitacionfis/radiacion_infrarroja.pdf>. [Consulta: julio de 2015].
8. STACKOVERFLOW. *HTML form submit to PHP script*. [en línea]. <<http://stackoverflow.com/questions/4559474/html-form-submit-to-php-script>>. [Consulta: julio de 2015].
9. STEVENS, Charles. *Medición de temperatura*. [en línea]. <<https://hifiduino.files.wordpress.com/2014/11/i2spins-001.jpg>>. [Consulta: julio de 2015].

APÉNDICES

Apéndice 1. Código fuente main. PHP

```
1 <html>
2 <head>
3 <title>CONTROL DE TEMPERATURA </title>
4 </head>
5 <body>
6 <center>CONTROL DE TEMPERATURA</center>
7
8
9
10
11 <?php
12 $servername = "localhost";
13 $username = "username";
14 $password = "password";
15 $dbname = "temperatura";
16
17 // Create connection
18 $conn = new mysqli($servername, $username, $password, $dbname);
19 // Check connection
20 if ($conn->connect_error) {
21     die("Connection failed: " . $conn->connect_error);
22 }
23
24
25
26 <!-- creando tabla para desplegar datos de control -->
27
28 <style type="text/css">
29 table.imagetable {
30     font-family: verdana,arial,sans-serif;
31     font-size:11px;
32     color:#333333;
33     border-width: 1px;
34     border-color: #999999;
35     border-collapse: collapse;
36 }
37
38 table.imagetable th {
39     background:#b5cfd2 url('cell-blue.jpg');
40     border-width: 1px;
41     padding: 8px;
42     border-style: solid;
43     border-color: #999999;
44 }
45
46 table.imagetable td {
47     background:#d0d0d0 url('cell-grey.jpg');
48     border-width: 1px;
49     padding: 8px;
50     border-style: solid;
51     border-color: #999999;
52 }
53 </style>
54 <!-- la tabla va en el cuerpo del documento -->
55 <table class="imagetable">
56 <tr>
57 <th>variable</th> <th>valor</th>
58 </tr>
59 <tr>
60 <td>Salida PID</td><td> echo $fila["salida"]</td>
61 </tr>
62 <tr>
63 <td>SetPoint</td><td>echo $fila["setpoint"]</td>
64 </tr>
65 <tr>
66 <td>integral</td><td>echo $fila["integral"]</td>
67 </tr>
68 <tr>
69 <td>derivativo</td><td>echo $fila["diferencial"]</td>
70 </tr>
71 <tr>
72 <td>proporcional</td><td>echo $fila["proporcional"]</td>
73 </tr>
74 <tr>
75 <td>temperatura actual</td><td>echo $fila["temperatura"]</td>
76 </tr>
77 <tr>
78 <td>humedad</td><td>echo $fila["humedad"]</td>
79 </tr>
80 </table>
81
82 <input type="radio" name="variable" value="Integralpid" checked="checked" /> Integral<br>
83 <input type="radio" name="variable" value="Proporcionalpid" /> Proporcional<br>
84 <input type="radio" name="variable" value="Derivativopid" /> Derivativo <br>
85 <input type="radio" name="variable" value="setpointpid" /> Setpoint<br>
86 <input type="radio" name="variable" value="humedadpid" /> Humedad<br>
87 </form>
88
89 <form action="MAINPHP.php">
90 <fieldset>
91 <legend>EDITAR TABLA</legend>
92 <div>
93 <input type="text" value="Valor a introducir" name="VALOREDIT" /><br>
94 <input type="submit" value="INGRESAR" />
95 </div>
96 </fieldset>
97 <div>
98 <input type="text" value="" name="VALOREDIT" />
99 </div>
100
101 <?php
102 $sql = "INSERT INTO temperatura (integral,diferencial,proporcional,var_controlada,salida,setpoint ,humedad) VALUES ($_GET['Integralpid'],$_GET['Proporcionalpid'],$_GET['Derivativopid'],$_GET['setpointpid'],$_GET['Humedadpid'])";
103 $conn->query($sql);
104
105 if ($conn->query($sql) === TRUE) {
106     echo "Nuevo registro creado";
107 } else {
108     echo "Error al crear registro";
109 }
110
111 $sql = "SELECT * FROM temperatura";
112 $result = $conn->query($sql);
113
114 if ($result->num_rows > 0) {
115     while($fila = $result->fetch_assoc()) {
116         echo "id: " . $fila["id"] . " - Name: " . $fila["Firstname"] . " " . $fila["Lastname"] . "<br>";
117     }
118 } else {
119     echo "0 resultados";
120 }
121
122 $ruta = "Ejecutar.py";
123 $command = "python " . $ruta;
124 $output = shell_exec($command);
125 echo $output;
126
127 </?php>
128 </form>
129 </body>
130 </html>
```

Fuente: elaboración propia, empleando Midori y PHP.

Apéndice 2. Temperatura SQL

```
1
2
3
4 CREATE DATABASE /*!32312 IF NOT EXISTS*/`classicmodels` /*!40100 DEFAULT CHARACTER SET latin1 */;
5
6 USE `classicmodels`;
7
8 /*estructura para la tabla de datos */
9
10 DROP TABLE IF EXISTS `temperaturas`;
11
12 CREATE TABLE `temperaturas` (
13   `integral` int(11) NOT NULL,
14   `diferencial` int(11) NOT NULL,
15   `proporcional` int(11) NOT NULL,
16   `var_controlada` varchar(50) NOT NULL,
17   `salida` varchar(50) NOT NULL,
18   `setpoint` varchar(50) NOT NULL,
19   `humedad` varchar(50) NOT NULL,
20 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
21
22 /*datos para la tabla */
23
24
25 LOCK TABLES `temperaturas` WRITE;
```

```
1
2 #TEMPPID.py TRABAJO DE TESIS HECHO POR LIONEL MAZARIEGOS.
3
4
5
6 import component
7 import eventmanager, events
8 from scene import getScene
9 from slots import *
10 from cgtypes import *
11 import core #se importan las librerias necesarias para el controlador PWM
12 import RPi.GPIO as GPIO #se importa la libreria de entradas y salidas
13 import re, os, rrdtool, time
14
15
16
17 #modulacion de ancho de pulsos
18
19
20 GPIO.setmode(GPIO.BCM) #se llama a la libreria a utilizar
21 GPIO.setup(18, GPIO.OUT) #se define el pin de salida 18
22 salida_regulada_temperatura = GPIO.PWM(18, 50) #se define la salida como modulacion de ancho de pulsos
23 salida_regulada_temperatura.start(salidaPID)#se une la temperatura controlada a la salida del pwm
24
25
26
27
28 #controlador de temperatura
29
30 TEMPCON=PIDController() #se define la clase PIDController
31 var_controlada= TEMPCON.input #se define la variable controlada
32 setp= TEMPCON.setpoint #se define el setpoint
33
34 integral= TEMPCON.propor #constante proporcional
35 proporcional=TEMPCON.integr #constante integral
36 derivativo=TEMPCON.deriv # constante derivativa
37 TEMPCON.output_slot=salidaPID #salida
38
39
40
41 #humedad
42
43
```

Fuente: elaboración propia, empleando Midori y PHP.

Apéndice 3. TEMPPID.py

```
40
41 #humedad
42
43
44
45 def bin2dec(string_num):
46     return str(int(string_num, 2))
47
48 data = []
49
50 GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_UP)
51
52 for i in range(0,500):
53     data.append(GPIO.input(17))
54
55 contador = 0
56
57 contador2= 0
58 humedadpid1 = ""
59
60
61
62 try:
63     while data[contador2] == 1:
64         tmp = 1
65         contador2= contador2+ 1
66
67
68     for i in range(0, 32):
69         contador = 0
70
71         while data[contador2] == 0:
72             tmp = 1
73             contador2= contador2+ 1
74
75         while data[contador2] == 1:
76             contador = contador + 1
77             contador2= contador2+ 1
78
79         if contador > 3:
80             if i>=0 and i<8:
81                 humedadpid1 = humedadpid1 + "1"
82
83         else:
84             if i>=0 and i<8:
85                 humedadpid1 = humedadpid1 + "0"
86
```

Continuación del apéndice 3.

```
88 except:
89     print "error de rango"
90     exit(0)
91
92
93 try:
94     for i in range(0, 8):
95         contador = 0
96
97         while data[contador2] == 0:
98             tmp = 1
99             contador2= contador2+ 1
100
101         while data[contador2] == 1:
102             contador = contador + 1
103             contador2= contador2+ 1
104
105         if contador > 3:
106             crc = crc + "1"
107         else:
108             crc = crc + "0"
109 except:
110     print "ERR_RANGE"
111     exit(0)
112
113
114 Humedad = bin2dec(humedadpid1)
115
116
117 if int(Humedad) - int(bin2dec(crc)) == 0:
118     return humedad
119
120 else:
121     print "ERR_CRC"
122
123
124
125 humedad = 0 #se define la variable de humedad como 0
126 GPIO.setup(25,GPIO.OUT) #se define el pin de salida de la Rpi para encender o apagar el relay del humidificador
127
128
129 if humedad < humedadpid #se define la condicion de humedad extrayendo la variable de la pagina web
130
131     GPIO.output(25,1)
132
```

```
139 #se actualiza el valor de la salida constantemente en un ciclo.
140 while True: TEMPCON.output=salidaPID
141
142
143
144
145 #leer temperatura del sensor
146 tfile = open("/sys/bus/w1/devices/10-000802824e58/w1_slave")
147 text = tfile.read()
148 tfile.close()
149 secondline = text.split("\n")[1]
150 temperaturadato = secondline.split(" ")[9]
151 tempdigital = float(temperaturadato[2:])
152 tempdigital = tempdigital / 1000
153 tempdigital= var_controlada
154
155
156
157
158
159 #Controlador PID
160 class PIDController(component.Component):
161     """PID controller.
162     """
163
164     exec slotPropertyCode("input")
165     exec slotPropertyCode("output")
166     exec slotPropertyCode("setpoint")
167     exec slotPropertyCode("max")
168     exec slotPropertyCode("min")
169     exec slotPropertyCode("integr")
170     exec slotPropertyCode("deriv")
171     exec slotPropertyCode("propor")
172
173     def __init__(self,
174                 name = "PIDController",
175                 setpoint = 0.0,
176                 propor = 0.0,
177                 integr = 0.0,
178                 deriv = 0.0,
179                 max= 999999,
180                 min = -999999,
181                 auto_insert = True):
182         """Constructor.
183         """
184
```

Continuación del apéndice 3.

```
185         component.Component.__init__(self, name, auto_insert)
186
187         self.input = DoubleSlot()
188         self.setpoint= DoubleSlot(setpoint)
189         self.max= DoubleSlot(maxout)
190         self.min= DoubleSlot(minout)
191         self.propor= DoubleSlot(Kp)
192         self.integr = DoubleSlot(Ki)
193         self.deriv = DoubleSlot(Kd)
194
195         self.output = ProceduralDoubleSlot(self.computeOutput)
196
197         self.addSlot("input", self.input)
198         self.addSlot("setpoint", self.setpoint)
199         self.addSlot("output", self.output)
200         self.addSlot("max", self.max)
201         self.addSlot("min", self.min)
202         self.addSlot("propor", self.propor)
203         self.addSlot("integr", self.integr)
204         self.addSlot("deriv", self.deriv)
205
206         self.input.addDependent(self.output)
207         self.setpoint.addDependent(self.output)
208         self.max.addDependent(self.output)
209         self.min.addDependent(self.output)
210         self.propor.addDependent(self.output)
211         self.integr.addDependent(self.output)
212         self.deriv.addDependent(self.output)
213
214
215
216
217
218         self._integraltotal = 0.0
219         self._error_previo = 0.0
220
221         eventmanager.eventManager().connect(events.STEP_FRAME, self)
222         eventmanager.eventManager().connect(events.RESET, self)
223
224
225     def onStepFrame(self):
226         err = self.setpoint-self.input
227         dt = getScene().timer().timestep
228         self._integraltotal += dt*err
229
230
231     def onReset(self):
232         self._integraltotal = 0.0
233         self._error_previo = 0.0
234
235     def computeOutput(self):
236
237         err = self.setpoint-self.input
238         dt = getScene().timer().timestep
239         I = self._integraltotal
240         D = (err-self._prev_err)/dt
241         # print "D:",D
242         res = self.propor*err + self.integral*I + self.deriv*D
243
244         self._error_previo = err
245
246         maximo = self.max
247         minimo = self.min
248         if res>maximo:
249             res = maximo
250         elif res<minimo:
251             res = minimo
252
253         return res
254
```

Fuente: elaboración propia, empleando Midori y PHP.

