

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN



**Control automático con interfaz Android para
un hábitat de reptiles**

TESIS

Que para obtener el título de:
Ingeniero Eléctrico Electrónico

PRESENTA

Oliver Silva Gómez

ASESOR DE TESIS

Dr. Ismael Díaz Rangel

Ciudad Nezahualcóyotl, Estado de México 2016



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos

Dedicado a Dios y a mis padres: Matilde Gómez Hernández y José Agustín Silva Oliver, quienes son mis héroes y continuamente estoy intentando alcanzar las virtudes y valores que, con paciencia y dedicación, intentan inculcarme.

Agradezco también a mis hermanos: Xóchitl y Agustín, que a pesar de las diferencias, aparentemente naturales en todos los hermanos, siempre he podido contar con su apoyo, ánimos y consejos incondicionales, saben que siempre podrán contar conmigo del mismo modo.

Quiero agradecer a mi iguana Edmond Ticolás II van der Garde conocido como Tico, por haber sido inspiración para esta propuesta experimental y haber despertado en mí, el sentido de responsabilidad y respeto a la naturaleza.

Así también, deseo agradecer a mi asesor de tesis doctor Ismael Díaz Rangel, por todo el apoyo y asesoría tanto a mí, como a mis compañeros de la facultad, además de haber sido el docente que más ha influenciado en mi desarrollo universitario y personal; y al profesor Luis Alejandro Sánchez Celis, por mostrarme un lado humano y engrandecer mis ambiciones culturales y éticas, así como de mi deseo de aprender de su capacidad de crítica subjetiva y profesional.

Gracias a la Universidad Nacional Autónoma de México por perseguir la excelencia en cada uno de sus alumnos y docentes, puesto que en ellos recae el peso del futuro de la nación, así como por brindar las herramientas culturales e intelectuales que forjan personas capaces e competitivas.

Este trabajo es la culminación de mi primer paso como profesionista, y es reflejo de tan inmenso sacrificio realizado por mis padres para brindarme el inimaginable lujo de una carrera universitaria, en la que sin duda es la institución académica más importante del país, y siempre será muy poco lo que puedo hacer o decir para expresar mi gratitud a mis padres, pues no puede ser comparado con todo lo que me han brindado en cada instante de mi vida.

Contenido

Índice de figuras	vi
Índice de tablas.....	xii
I. Introducción	1
1.1 Objetivo General	2
1.2 Objetivos Particulares	2
1.3 Motivación	2
1.4 Justificación.....	2
II. Marco Teórico	4
2.1 Fisiología de las iguanas	4
2.2 Características de terrarios.....	6
2.2.1 Dimensiones.....	6
2.2.2 Rayos UVA y UVB.....	6
2.2.3 Temperatura	7
2.2.4 Humedad Relativa.....	7
2.2.5 Iluminación	8
2.2.6 Ventilación.....	9
2.3 Sensores	9
2.3.1 Funcionamiento de Sensores de Humedad	13
2.3.2 Sensores de humedad.....	14
2.3.3 Sensores de temperatura.....	16
2.3.4 Sensores de nivel de agua	20
2.4 Tipos de fuentes de alimentación.....	24
2.4.1 Fuentes conmutadas	25

2.4.2 Fuentes lineales	26
2.5 Sistemas de calefacción.....	26
2.6 Diseño y programación de interfaz con Android	28
2.6.1 Importar aplicaciones al dispositivo Android ¹	33
2.6.1.1 Modo inalámbrico	33
2.6.1.2 Modo descarga en formato APK.....	35
2.7 Sistemas de Control	36
2.7.1 Sistemas de control de lazo abierto	37
2.7.2 Sistemas de control de lazo cerrado	38
2.7.2.1 Sistemas de control retroalimentados.....	39
2.7.3 Elementos y funciones de un lazo de control.....	40
2.7.4 Controladores	41
2.7.4.1 Programmable Logic Controller (PLC)	41
2.7.4.2 Microcontroladores	43
2.7.4.2.1 Tarjeta de desarrollo Arduino Uno	44
2.7.5 Actuadores.....	49
2.8 Sistemas de protecciones eléctricas y etapa de potencia.....	52
2.8.1 Protecciones eléctricas	52
2.8.2 Etapa de potencia	56
III. Desarrollo Experimental De La Propuesta.....	60
3.1 Algoritmo	60
3.1.1 Descripción del algoritmo en el compilador de la tarjeta de desarrollo.....	62
3.1.2 Descripción del algoritmo de comunicación entre la tarjeta de desarrollo y la interfaz Android	68
3.1.3 Interfaz de usuario mediante el display LCD (16x2).....	74

3.2 Diagramas esquemáticos y conexiones.....	76
3.2.1 Diagramas esquemáticos.....	77
3.2.2 PCB propuesta.....	85
3.3 Diseño y programación de la interfaz de usuario con Android.....	92
3.3.1 Diseño de la interfaz de usuario.....	93
3.3.2 Programación de la interfaz de usuario.....	99
IV. Pruebas y Resultados	104
4.1 Pruebas de conectividad Bluetooth.....	105
4.1.1 Resultados de conectividad Bluetooth.....	108
4.2 Pruebas de la interfaz Android.....	109
4.2.1 Resultados de las pruebas de la interfaz Android.....	111
4.3 Pruebas de emplazamiento de los sensores.....	112
4.3.1 Resultados de emplazamiento de los sensores.....	114
4.4 Desempeño de la etapa de potencia y protección.....	115
4.5 Desempeño de los actuadores.....	116
4.6 Desempeño del display LCD.....	118
Conclusiones.....	119
Bibliografía.....	122
Anexos.....	128
Código Arduino.....	128
Código MIT App Inventor 2.....	137

Índice de figuras

Figura 2.1-1 Regiones donde habitan las iguanas.	4
Figura 2.3-1 Curva de sensibilidad tensión contra magnitud.....	10
Figura 2.3-2 Curva no-linealidad salida contra magnitud física.	11
Figura 2.3-3 Curva histéresis salida contra magnitud física..	12
Figura 2.3.1 -1 Esquema básico de un sensor de humedad relativa y un circuito de potencia.	14
Figura 2.3.2-1 Sensor altamente sensible de humedad.	14
Figura 2.3.2-2 Sensor de humedad y temperatura SHT15.....	15
Figura 2.3.2-3 Diagrama de conexión sensor SHT15 y protocolo SPI.....	15
Figura 2.3.3-1 Sensor LM35 encapsulado DIP.....	16
Figura 2.3.3-2 Conexión sensor LM35.	16
Figura 2.3.3-3 Sensor TCN75A.	17
Figura 2.3.3-4 ConexiónTCN75A con protocolo I2C a un microcontrolador.....	18
Figura 2.3.3-5 Sensor DS18B20 DIP.....	18
Figura 2.3.3-6 Sensor DS18B20 sonda.	19
Figura 2.3.3-7 Sensor DHT11.....	19
Figura 2.3.4-1 Sensor de nivel de agua “Funduino”.....	21
Figura 2.3.4-2 Sensor HC-SRF-04.....	22
Figura 2.3.4-3 Sensor TCRT5000L.	23
Figura 2.3.4-4 Funcionamiento interruptor a prueba de agua.....	23
Figura 2.3.4-5 Interruptor a prueba de agua.....	24
Figura 2.4.1-1 Esquema fuente conmutada.....	26
Figura 2.4.2-1 Esquema fuente lineal.	26

Figura 2.5-1 Calefacción por convección o “suelo radiante”.	27
Figura 2.5-2 Esquema convección forzada.	28
Figura 2.5-3 Calefacción por sistema de radiadores.	28
Figura 2.6-1 Ventana Diseño.	30
Figura 2.6-2 Ventana Bloques.	32
Figura 2.6.1.1-1 Código QR MIT App Inventor 2.	33
Figura 2.6.1.1-2 Selección AI Companion.	34
Figura 2.6.1.1-3 Escaneo del código QR.	34
Figura 2.6.1.2-1 Selección descarga.	35
Figura 2.6.1.2-2 Descarga aplicación.	36
Figura 2.7-1 Diagrama a bloques de un sistema.	36
Figura 2.7-2 Diagrama a bloques de un sistema de control.	37
Figura 2.7.1-1 Sistema de control de lazo abierto.	38
Figura 2.7.2-1 Sistema de control de lazo cerrado.	39
Figura 2.7.2.1-1 Sistema de control retroalimentado.	39
Figura 2.7.2.1-2 Retroalimentación negativa.	40
Figura 2.7.2.1-3 Retroalimentación positiva.	40
Figura 2.7.4.1-1 Esquema de PLC.	41
Figura 2.7.4.2-1 Diagrama de bloques de un Microcontrolador.	43
Figura 2.7.4.2-2 Alteri basada en un PIC18F455 de montaje superficial.	44
Figura 2.7.4.2.1-1 Arduino UNO basado en un ATmega328P encapsulado DIP.	45
Figura 2.7.4.2.1-2 Selección puerto COM.	47
Figura 2.7.4.2-4 Botón de programación de la tarjeta Arduino.	48
Figura 2.7.4.2.1-3 Descripción de código en entorno de Arduino.	48
Figura 2.7.4.2.1-5 Barra de progreso de programación.	49

Figura 2.7.4.2.1-6 Programación realizada con éxito.	49
Figura 2.7.5-1 Calentador resistivo de inmersión.	51
Figura 2.7.5-2 Calentadorresistivo de inducción forzada.	51
Figura 2.7.5-4 Atomizador para cabello.	52
Figura 2.7.5-3 Bomba de agua 12Volts.	52
Figura 2.8.1-2 Interruptor termomagnético doméstico.	53
Figura 2.8.1-1 Diagrama corto circuito contra fase y contra tierra.	54
Figura 2.8.1-3 Fusibles domésticos.....	54
Figura 2.8.1-4 Contactor y diagrama de conexión.....	55
Figura 2.8.2-1 Símbolo diodo de potencia.	57
Figura 2.8.2-2 Símbolo SCR.....	57
Figura 2.8.2-3 Símbolo TRIAC.	57
Figura 2.8.2-4 Símbolo TBJ.....	58
Figura 2.8.2-5 Símbolo MOSFET.....	58
Figura 3.1-1 Esquema del algoritmo.	60
Figura3.1-2 Algoritmo desarrollado para el proyecto.....	61
Figura 3.1.1-1 Sensor DS18B20 tipo sonda sumergible.	63
Figura 3.1.1-2 Sensor DHT11.....	63
Figura 3.1.1-3 Segmento de código encargado de la adquisición de valores.	63
Figura 3.1.1-4 Código de control Humedad Relativa.	64
Figura 3.1.1-5 Código decisión del estado de la salida.....	66
Figura 3.1.1-6 Control temperatura.....	68
Figura 3.1.2-1 Código recepción de datos.	70
Figura 3.1.2-2 Código recepción de datos a guardar en la EEPROM.	71
Figura 3.1.2-3 Código envío de datos.	72

Figura 3.1.3-2 Subrutina Display().	74
Figura 3.1.3-1 Invocación de la subrutina Display().	74
Figura 3.1.3-3 Continuación subrutina Display().	75
Figura 3.2-1 Diagrama de bloques de la conexión de la etapa de control, sensores y actuadores.	76
Figura 3.2.1-1 Plug macho 2.1mm.	77
Figura 3.2.1-2 Alimentación Arduino UNO.	77
Figura 3.2.1-3 Conexión Arduino UNO y display LCD.	78
Figura 3.2.1-4 Conexión Arduino UNO y módulo HC-05.	79
Figura 3.2.1-5 Conexión Arduino UNO y reloj de tiempo real.	80
Figura 3.2.1-6 Conexión Arduino UNO y sensor DHT-11.	80
Figura 3.2.1-7 Conexión Arduino UNO y sensor DS18B20.	81
Figura 3.2.1-8 Conexión Arduino UNO, etapa de potencia y sensores.	82
Figura 3.2.1-9 Arduino UNO, etapa de potencia y sensores en protoboard.	84
Figura 3.2.2-1 PCB #1 superior cobre y emplazamiento.	86
Figura 3.2.2-2 PCB #1 superior cobre.	86
Figura 3.2.2-3 PCB #1 superior emplazamiento.	87
Figura 3.2.2-4 PCB #2 superior cobre y emplazamiento.	87
Figura 3.2.2-5 PCB #2 superior cobre.	88
Figura 3.2.2-6 PCB #2 superior emplazamiento.	89
Figura 3.2.2-7 PCB #3 superior cobre y emplazamiento.	89
Figura 3.2.2-8 PCB #3 superior cobre.	90
Figura 3.2.2-9 PCB #3 superior emplazamiento.	90
Figura 3.2.2-10 PCB #1 inferior.	91
Figura 3.2.2-11 PCB #2 inferior.	91

Figura 3.2.2-12 PCB #3 inferior.	92
Figura 3.3.1-1 Vista preliminar de la interfaz.	93
Figura 3.3.1-3 Propiedades de los elementos de la aplicación.	94
Figura 3.3.1-2 Botones Conectar y Prueba.	94
Figura 3.3.1-4 Propiedades de los Arrangement.	95
Figura 3.3.1-5 Vista previa Información Actual.	95
Figura 3.3.1-6 Vista previa Registro Histórico.	95
Figura 3.3.1-7 Localización de los elementos de la aplicación.	96
Figura 3.3.1-8 Sliders y botón Configurar.	97
Figura 3.3.1-9 Propiedades de los sliders.	97
Figura 3.3.1-10 Elementos ocultos.	99
Figura 3.3.2-1 Bloques de Emparejamiento.	99
Figura 3.3.2-3 Lista de dispositivos emparejados.	100
Figura 3.3.2-2 Bloques de ligación.	100
Figura 3.3.2-5 Bloques de conexión Bluetooth.	101
Figura 3.3.2-4 Botón Conexión exitosa/no exitosa.	101
Figura 3.3.2-6 Bloques de recepción e interpretación de los datos recibidos vía Bluetooth.	103
Figura 3.3.2-7 Vista de la aplicación ejecutada desde el teléfono móvil e interactuando con la tarjeta de desarrollo.	104
Figura 4.1-2 Botón de conexión.	106
Figura 4.1-1 Lista de dispositivos pareados.	106
Figura 4.1-3 Selección puerto COM.	107
Figura 4.1-4 Ejecución Monitor Serial.	107
Figura 4.1-5 Información recibida en computador vía Bluetooth.	108

Figura 4.2-1 Vista de la aplicación ejecutada desde el teléfono móvil e interactuando con la tarjeta de desarrollo.	109
Figura 4.2-2 Alerta nivel agua insuficiente.	110
Figura 4.2-3 Dialogo de confirmación.	110
Figura 4.2-4 Mensajes de error en la interfaz.	111
Figura 4.2.1-1 Confirmación de configuración EEPROM vista en el computador vía Bluetooth.	112
Figura 4.3-1 Sistema térmico.	113
Figura 4.3-2 Afectación por condensación.	114
Figura 4.4-1 Diagrama etapa de potencia con optoacoplador.	115
Figura 4.5-1 Esquema ducha eléctrica aterrizada utilizada en la UE.	118

Índice de tablas

Tabla 2.1-1 Descripción de la radiación UVA y UVB.	5
Tabla 2.7.5-1 Comparación y tipos de actuadores según su funcionamiento.	50
Tabla 3.1.1-1 Pin de salida de los actuadores y caracteres correspondientes impresos según el nivel o estado asignado por el programa.....	68
Tabla 3.1.2-1 Características de los datos enviados desde la EEPROM a través de la USART.....	73
Tabla 3.1.2-2 Características de los datos enviados por la memoria dinámica.....	73
Tabla 3.2.1-1 Actuadores y sus salidas con respecto a Arduino y PCB.	83
Tabla 4.1 Resultados de las pruebas de conexión vía Bluetooth.	105
Tabla 4.5-1 Desempeño de los actuadores en prueba de 48 horas con condiciones reales.	117

I. Introducción

En la actualidad, cada vez es más común la adquisición como mascotas de reptiles como tortugas, serpientes, lagartos e incluso cocodrilos, ya sea en negocios como restaurantes u hoteles, donde dichos animales están a la vista del público, o en hogares sin contacto con demasiada gente.

Los reptiles son criaturas con una gran capacidad de adaptación, que han evolucionado para ocupar una increíble variedad de ambientes. Se los encuentra en cada continente excepto la Antártida, y la mayoría de los países cuentan con al menos una especie de reptil terrestre.

Los reptiles viven en distintos ambientes, existe incluso una especie de lagarto marino, la iguana marina de las islas Galápagos, que se alimenta a menudo bajo el agua. Debido a que dependen del calor del ambiente, el área de influencia de los reptiles está limitada por la temperatura, y se hacen cada vez más raros a medida que se acercan a los polos.

Esta propuesta experimental ofrece una solución, utilizando electrónica, a la problemática de algunas especies de reptiles que no tienen las condiciones ambientales de temperatura y humedad adecuadas para una vida digna, y que requieran niveles de temperatura y humedad mayor de los que se tiene en el ambiente del área metropolitana de la república mexicana.

Este trabajo acercará las condiciones ambientales de temperatura y humedad, dentro del hábitat o terrario, a las condiciones existentes en las regiones tropicales donde habitan las iguanas verdes, pero fácilmente se puede incrementar o decrementar los parámetros de temperatura y humedad para adecuarse a otros reptiles o necesidades, como posiblemente una incubadora.

1.1 Objetivo General

Desarrollar un sistema que regule de manera automática la temperatura del ambiente y del agua, así como la humedad en el aire; además que pueda ser configurado mediante una interfaz en el sistema operativo Android.

1.2 Objetivos Particulares

- ◆ Investigar fisiología de las iguanas.
- ◆ Consultar características de terrarios.
- ◆ Investigar sensores de humedad.
- ◆ Investigar sensores de temperatura.
- ◆ Investigar tipos de fuentes de alimentación.
- ◆ Consultar sistemas de calefacción.
- ◆ Diseñar y programar una interfaz para sistema operativo Android.
- ◆ Investigar sistemas de control.
- ◆ Consultar e implementar sistemas de protecciones eléctricas y etapa de potencia.

1.3 Motivación

La realización de este trabajo fue para mejorar las condiciones de vida de una iguana, y también podrá ser utilizado con otros reptiles, como tortugas o algunos lagartos de clima tropical sin tener que modificar el proyecto en alambrado o algoritmo, además de que el sistema es fácil de utilizar y no dañará ni interferirá con la movilidad del reptil.

1.4 Justificación

Los principales beneficiados serán los reptiles en cautiverio, al mismo tiempo, facilitando el cuidado gracias a la información en tiempo real desplegada en la aplicación Android desarrollada para este trabajo.

El beneficio para las personas vendrá mayormente en comodidad, al tener información en tiempo real de las condiciones ambientales dentro del terrario o hábitat, mediante comunicación Bluetooth, lo que elimina la necesidad de estar monitoreando presencialmente el estado de los actuadores constantemente, esto permite a las personas realizar otras actividades sin tener que estar en contacto con el terrario, además de ser alertadas si existe algún malfuncionamiento mediante una notificación sonora emitida mediante la aplicación desarrollada..

Los reptiles tendrán mayor comodidad, pues la temperatura y humedad permanecerán cerca de un nivel óptimo, preestablecido o reconfigurado por el usuario, pues el metabolismo de los reptiles depende del calor ambiental, y este proyecto fue realizado en el área metropolitana de la República Mexicana, donde no existe en el ambiente condiciones ideales como en su hábitat natural en regiones tropicales del país.

II. Marco Teórico

Existen diferentes tópicos a investigar al realizar una propuesta experimental, para este proyecto, se ha requerido investigación respecto a la fisiología de las iguanas, etapa de potencia, protecciones eléctricas, e incluso desarrollo de aplicaciones para teléfonos inteligentes Android, además de consultar apuntes de electrónica y diversos diagramas eléctricos y electrónicos, a continuación se enlistan los tópicos requeridos.

2.1 Fisiología de las iguanas

Iguana es un género de lagartos herbívoros nativos de las zonas tropicales de México, América Central, varias islas de la Polinesia, y el Caribe (figura 2.1-1), donde la temperatura media anual es de 27°C - 28°C y la humedad ambiente es superior al 70%RH.



Figura 2.1-1 Regiones donde habitan las iguanas.

Las iguanas son animales diurnos, esto quiere decir que están activos únicamente por el día y duermen por la noche. Son animales que, a diferencia de algunas mascotas como perros y gatos, siguen una rutina diaria que suele ser invariable. Una iguana salvaje se despierta cuando sale el sol y busca un sitio soleado donde recostarse y calentarse hasta alcanzar la temperatura apropiada para moverse con facilidad y buscar alimento. Así permanecen más o menos hasta mediodía, es entonces cuando se dedican a la búsqueda de forraje, flores, hojas, brotes tiernos, frutos suaves, etc. Tras comer lo suficiente, buscan de nuevo un sitio soleado donde permanecer recostadas y digerir el alimento. Al atardecer, buscan un sitio seguro donde esperar a que llegue la noche. Y así durante los años que viven las iguanas salvajes.

Como las iguanas verdes sólo tienen unas pocas células Rod, tiene problemas de visión en condiciones de poca luz. Al mismo tiempo, tienen unas células llamadas "células de doble cono" que les dan la visión de color fuerte y les permiten ver las longitudes de onda ultravioleta. Esta habilidad es muy útil para poder tomar el sol y que el animal pueda asegurarse de que reciba suficiente luz solar en las formas de los rayos UVA y UVB para producir vitamina D3.

Tabla 2.1-1 Descripción de la radiación UVA y UVB.

Nombre	Abreviatura	Longitud de onda (nm)
Ultravioleta A (onda larga)	UVA	400 – 315
Ultravioleta B (onda media)	UVB	315 – 280

Las iguanas al igual que la mayoría de los reptiles, son de sangre fría, es decir, que el calor necesario para que su metabolismo funcione adecuadamente lo reciben del medio y no lo producen ellos mismos, como los mamíferos, por lo que con los primeros rayos de sol, suben a las ramas más altas para alcanzar la temperatura óptima para poder realizar sus actividades, aproximada de 28°C - 35 °C.

Las iguanas deben vivir a una temperatura de 24 °C - 32 °C y contar con una indispensable fuente de luz UVB y UVA. Sin esas radiaciones UVB pueden desarrollar enfermedades metabólicas de los huesos que serán fatales si no se trata, además de que pueden mantener el nivel de calcio y vitamina D3 a través de la absorción de luz, asimismo la humedad relativa del ambiente donde habitan de forma salvaje, en promedio esta entre 65%RH - 75%RH.

Una iguana en estado salvaje puede llegar a vivir entre 6 y 10 años, generalmente mueren por causas de los depredadores naturales, caza excesiva por los humanos o destrucción del hábitat; En cautiverio puede llegar a vivir entre 8 y 20 años, y fallecen a una edad avanzada para la especie, debido a problemas renales, pueden llegar a medir 2 metros de longitud, desde la nariz hasta la cola, y llegar a pesar alrededor de 15 kilogramos.

2.2 Características de terrarios.

En terrarios diseñados para iguanas, se pueden generalizar algunas condiciones como dimensiones, temperatura y humedad relativa interna, sin importar el material del que estén contruidos o recubiertos, a continuación se describen algunas características.

2.2.1 Dimensiones

Las medidas de largo y ancho del terrario, deben ser mínimo 2 veces el largo de la iguana. Y la medida de la altura del terrario debe ser mínimo 3 veces el largo de la iguana.

2.2.2 Rayos UVA y UVB

En un clima tropical, lo mejor será que se construya un terrario para exterior y se aproveche la temperatura y radiación solar de la localidad. Pero al contrario, en un clima templado o frío, lo mejor será que se construya un terrario para interior ya que en ese clima la temperatura en la noche desciende bastante. En la *Tabla 2.1-1 Descripción de la radiación UVA y UVB* se esquematizan las características de los rayos ultravioleta. Existen dos formas principales de obtener rayos UV-B:

- ◆ Sol: El sol debe incidir en la piel del reptil sin que pase a través de vidrio, plástico ni ningún material, porque de ser así estos materiales filtrarán el sol y no pasarán los rayos UV-B.
- ◆ Lámpara especial: De venta en las tiendas de mascotas. Las lámparas deben de emitir 5% de radiación UV-B. Hay en diferentes presentaciones, puede ser en tubo o bobilla convencional. Al igual que los rayos solares, no debe pasar a través de vidrio, ni plásticos y a diferencia del sol, tendrá que estar encendida 12 horas diarias, con el fin de emular condiciones naturales. Tendrá que cambiar por una lámpara nueva cada 6 - 8 meses ya que pierden su efectividad.

2.2.3 Temperatura

La temperatura es muy importante para las Iguanas ya que son reptiles de sangre fría. El calor es imprescindible para que una iguana viva, crezca, digiera la comida, luche contra las enfermedades, etc. El umbral de temperatura debe ser:

- ◆ Día: 30 a 35 °C
- ◆ Noche: 20 a 25 °C

En el terrario nunca debería haber un calor homogéneo, siempre debe tener una zona cálida, y una zona fresca, ya que las iguanas autorregulan su temperatura moviéndose entre sectores con altas y bajas temperaturas según lo necesiten, pero siempre respetando los límites de las temperaturas ideales.

Mantener vigilancia constante en la temperatura ya que si baja de 17°C o sube de 35°C puede llegar a ser muy peligroso para las iguanas y resultar fatal, debido a que algunas especies de reptiles, entre ellas las iguanas, no pueden regular su temperatura corporal (ni incrementar ni decrementar su temperaturacorporal).

2.2.4 Humedad Relativa

La humedad en el terrario debe oscilar entre el 60%RH y 90%RH. Una humedad por debajo ocasionará problemas de muda de piel en las iguanas y una humedad por arriba propiciará la aparición de hongos.

Algunos dispositivos bastantes comunes encargados de incrementar la humedad en el ambiente de los terrarios son:

- ◆ Rociadores: Se puede rociar agua dentro del terrario por medio de rociadores manuales (como los atomizadores que utilizan los estilistas para humedecer el cabello), o por rociadores automáticos (como los utilizados en jardinería), en invierno se debe rociar todo el terrario por la mañana y al atardecer, mientras que en las estaciones más calurosas debe rociarse con más frecuencia 5, 6 ó más en el día, dependiendo del tamaño del terrario.

- ◆ Humidificadores: De vapor frío. Combinados con un temporizador son lo mejor para mantener óptima la humedad. Estos crean una especie de neblina a partir de agua que se deposita en un tanque que incorpora el aparato, y una vez que se energiza el aparato, esta niebla empieza a salir subiendo bastante la humedad dentro del terrario, se utiliza en conjunto con temporizadores para que sea totalmente automático el proceso y se active varias veces en el día. Su precio suele ser bastante elevado.
- ◆ Piscina: Este elemento es imprescindible. Metiendo al terrario una pequeña piscina, el calor que hay en el terrario evaporará poco a poco esta agua lo que hará que la humedad aumente. Se puede lograr un mejor resultado si se pone una manta térmica, piedra térmica o cable calefactor por debajo de esta bandeja. Entre más grande sea la piscina, mayor será la humedad generada. Es importante que se cambie el agua diariamente ya que la iguana nadará, beberá y defecará aquí, por lo que podría ser un foco de infecciones para el reptil.

2.2.5 Iluminación

El fotoperiodo es el ciclo de luz-oscuridad, las iguanas necesitan de un correcto fotoperiodo ya que influye en su comportamiento y metabolismo, les ayuda a identificar la estación del año por la duración de horas luz en el día, les ayuda a saber el momento más apropiado para reproducirse, entrar en hibernación, controlar periodos de actividad, etc.

Así como es muy importante el periodo de luz/calor para mantener activo su metabolismo, también lo es el periodo de oscuridad/frío, ya que con este cambio de iluminación y temperatura en la noche, el metabolismo de la iguana baja su actividad y sus células se disponen a descansar mientras la iguana duerme. Si no existe en las noches este descenso de temperatura, se podrá ver la iguana dormida, pero su organismo, sus células, estarán en actividad debido a que su metabolismo solo descansa al estar a temperaturas menores, por lo tanto la iguana no descansará completamente.

2.2.6 Ventilación

Al igual que todos los vertebrados, las iguanas necesitan respirar oxígeno, por lo tanto, es necesario que el aire entre fácilmente dentro del terrario. Es importante también para la prevención de hongos y de enfermedades respiratorias, ya que si está en un terrario cerrado (por ejemplo los de cristal) el aire suele "viciarse" al no estarse renovando por lo que los olores que el terrario y la iguana producen se quedan atrapados, situación que afectará la salud del reptil. Para ventilar el terrario, basta abrir levemente la tapa del terrario, si es que la tiene, o habilitar en cada lado del terrario agujeros para que el aire pueda circular.

Debe haber cuidado con la ventilación, ya que si es excesiva, será imposible que la humedad se mantenga a los niveles adecuados.

2.3 Sensores

La necesidad de la humanidad, incluso podría considerarse capricho, para controlar su entorno, ha hecho evolucionar la manera con la que medimos magnitudes, sea por contacto o de alguna forma que, a cierta distancia, permita medir la magnitud, entre esta necesidad se encuentra la medición del ambiente en el que nos encontramos, como sucede en los aparatos actuales de aire acondicionado, donde la presencia de la electrónica es muy notoria al estar equipados con controles remotos, elementos que muestren los parámetros con los que interactúa, como display o indicadores LED, temporizadores para apagado automático, y hasta la capacidad de comunicación con teléfonos inteligentes.

Para realizar la medición de alguna magnitud, existen dos tipos de dispositivos, aunque a grandes rasgos, su principio de funcionamiento es muy distinto dependiendo de la magnitud a medir, en general es el principio de funcionamiento se generaliza y es esto lo que los divide en sensores y transductores.

Un transductor es el dispositivo que transforma una magnitud física, mecánica, térmica, magnética, eléctrica, óptica, entre otras, en otra magnitud, generalmente eléctrica.

Un sensor es el dispositivo que altera una señal eléctrica dependiendo de las características físicas o químicas de la magnitud a la que es sometido, es decir tiene una señal eléctrica de entrada que es modificada por una magnitud física, esta señal eléctrica modificada se presenta como señal de salida.

Las propiedades de los sensores y transductores son importantes, y se deben definir algunos conceptos de medición que son muy importantes al analizar un sensor o transductor.

- ◆ **Sensibilidad:** Se define como la pendiente de la curva (figura 2.3-1) que relaciona la salida eléctrica con la magnitud física a medir.

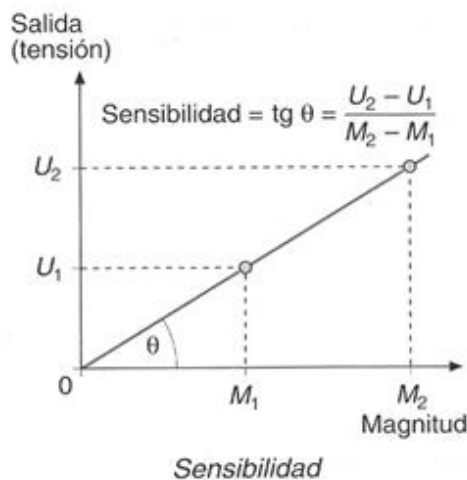


Figura 2.3-1 Curva de sensibilidad tensión contra magnitud.

- ◆ **Precisión:** Significa que existe o no una pequeña variación aleatoria en la medición de la variable. La dispersión en los valores de una serie de mediciones deberá ser mínima.
- ◆ **Resolución:** Es la variación detectable más débil.

- ◆ **No-linealidad:** Es la distancia mayor entre la curva de funcionamiento del **sensor** (en dirección ascendente) y la recta del punto inicial al final de funcionamiento como se aprecia en la figura 2.3-2.

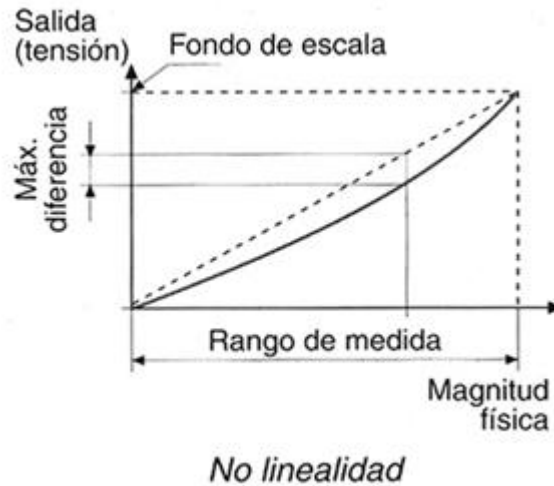


Figura 2.3-2 Curva no-linealidad salida contra magnitud física.

- ◆ **Repetitividad:** Cuando la medida se realiza varias veces, la gráfica que relaciona la magnitud con la señal eléctrica no siempre pasa por el mismo lugar. La máxima diferencia será el valor absoluto de la repetitividad.
- ◆ **Velocidad de respuesta:** El **transductor o sensor** debe ser capaz de responder a los cambios de la variable detectada en un tiempo mínimo. Lo ideal sería una respuesta instantánea.
- ◆ **Histéresis:** En algunos tipos de **transductores**, la gráfica (figura 2.3-3) que relaciona la magnitud con la señal eléctrica varía, y no tienen el mismo comportamiento en el aumento y en la disminución. Por ejemplo, el transductor no da la misma señal para una presión de 200 pascales en función de si esa presión se ha tomado de forma ascendente o de forma descendente. En la ecuación siguiente se describe la relación de la histéresis.

$$\text{Histéresis} = \frac{\text{Máxima diferencia}}{\text{Rango de medida}} 100$$

A continuación se describe el funcionamiento de algunos dispositivos creados específicamente para medir temperatura y humedad en nuestro entorno, pero para

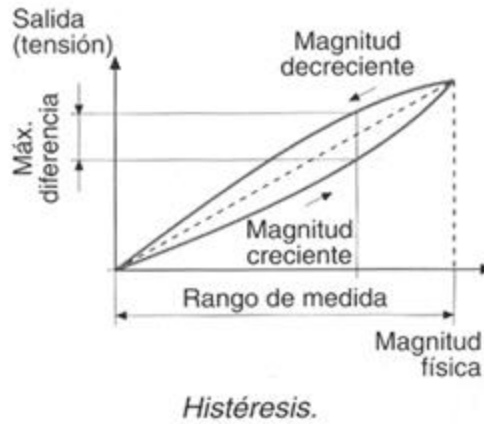


Figura 2.3-3 Curva histéresis salida contra magnitud física..

poder continuar se deben conocer dos conceptos fundamentales para este proyecto.

- ◆ **Humedad Relativa:** Es definida como la cantidad de agua en el aire en forma de vapor, comparada con la cantidad de agua que el aire puede “mantener o contener” a esa temperatura, es decir, la humedad relativa está en función de la temperatura. Cuando el aire no puede “mantener” toda la humedad, entonces se condensa como rocío (punto de rocío) y es equivalente a 100%RH. Su unidad es adimensional, expresada en porcentaje su símbolo es %RH. La evaporación del agua es el cambio del estado líquido al gaseoso a temperatura ambiente. A diferencia de la ebullición, ésta no sucede en toda la masa, sino en el agua superficial.
- ◆ **Temperatura:** Es la medida del calor de un cuerpo (y no la cantidad de calor que este contiene o puede almacenar). La temperatura no es energía sino una medida de ella; sin embargo, el calor sí es energía. Su unidad en el Sistema Internacional es el kelvin y su símbolo es K, en este proyecto se toma la unidad relativa grado Celsius y su símbolo es °C, su correspondencia con el kelvin es: $K=^{\circ}C+273.15$

2.3.1 Funcionamiento de Sensores de Humedad

Se puede clasificar en su principio de funcionamiento:

- ◆ Mecánicos: Aprovechan los cambios de dimensiones que sufren ciertos tipos de materiales en presencia de la humedad. Como por ejemplo, fibras orgánicas o sintéticas, el cabello humano.
- ◆ Basados en sales higroscópicas: Deducen el valor de la humedad en el ambiente a partir de una molécula cristalina que tiene mucha afinidad con la absorción de agua.
- ◆ Capacitivos: Se basan sencillamente en el cambio de la capacidad que sufre un condensador en presencia de humedad.
- ◆ Infrarrojos: Estos disponen de dos fuentes infrarrojas, al absorber parte de la radiación que contiene el vapor de agua, deducen el valor de humedad.
- ◆ Por conductividad: La presencia de agua en un ambiente permite que a través de unas rejillas de oro circule una corriente. Ya que el agua es buena conductora de corriente. Según la cantidad de corriente eléctrica se deduce el valor de la humedad y son ideales para medir humedad en el aire.
- ◆ Resistivos: Aplican un principio de conductividad de la tierra, es decir, cuanta más cantidad de agua hay en la muestra, más alta es la conductividad de la tierra, este tipo de sensores son exclusivamente para medir la humedad en materiales donde puedan ser insertados, como tierra, o alguna fibra.
- ◆ Los más comunes en el mercado agropecuario e industrial, su funcionamiento es por conductividad, y se basan en que el agua no es un material aislante como el aire, sino que tiene una conductividad eléctrica.

Por lo tanto un par de cables eléctricos desnudos (sin cinta aislante recubriéndolos) van a conducir una pequeña cantidad de corriente si el ambiente es húmedo, si se coloca un transistor en zona activa que amplifique esta corriente tenemos un detector de humedad, como se muestra en la figura 2.3.1-1.

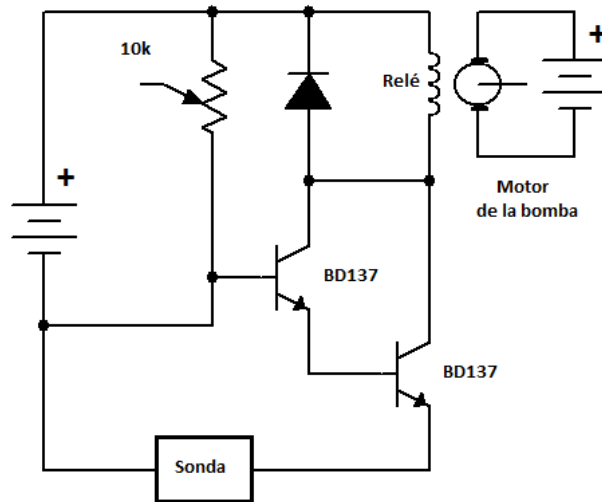


Figura 2.3.1 -1 Esquema básico de un sensor de humedad relativa y un circuito de potencia.

2.3.2 Sensores de humedad

Entre la gran variedad de sensores comerciales, existen algunos muy populares, ya que el umbral de humedad y su resolución es bueno en relación al costo de dicho sensor, a continuación se mencionan algunos de ellos.

High sensitivity moisture sensor, Sensor altamente sensible de humedad

Sensor simple, ideal para monitorear un jardín urbano, su recubrimiento de inmersión de oro protege el níquel de la oxidación. Puede detectar la humedad de un sólido si el agua lo rodea, este tipo de sensores solamente pueden medir temperatura en cuerpos sólidos, y no en el ambiente.

Consta de dos sondas (figura 2.3.2-1), las cuales hacen pasar una pequeña corriente a través del sólido y después lee la resistencia entre las sondas para poder obtener el nivel de humedad, mientras exista más agua, la resistencia será menor, mientras más seco mayor resistencia y una mejor conducción eléctrica.



Figura 2.3.2-1 Sensor altamente sensible de humedad.

Características del dispositivo:

- ◆ Alimentación: 5Vdc
- ◆ Corriente: <20mA
- ◆ Profundidad de sensibilidad: 37mm
- ◆ Temperatura de funcionamiento: 10°C – 30°C
- ◆ Señal de salida: Analógica
- ◆ Voltaje de señal de salida: 0Vdc – 4.2Vdc

Sensor de humedad y temperatura SHT15

Este dispositivo utiliza un sensor capacitivo para medir la temperatura relativa del aire mientras que la temperatura es medida por otro sensor de tipo **Band-Gap** (figura 2.3.2-2).

El principio del **Band-Gap** es que el voltaje de un diodo de silicón que era la base-emisor de un TBJ, y la temperatura afectara dicho voltaje. Este tipo de sensor es bastante adecuado para la propuesta del proyecto, pues su capacidad de censar la humedad relativa del aire lo hace adecuado para el propósito de la propuesta.



Figura 2.3.2-2 Sensor de humedad y temperatura SHT15.

Este sensor es digital, es decir, requiere un protocolo de comunicación conocido como **Serial Peripheral Interface** o por sus siglas **SPI**, es una comunicación síncrona serial por lo tanto requiere una señal de reloj, por lo que su conexión se observa en la figura 2.3.2-3.

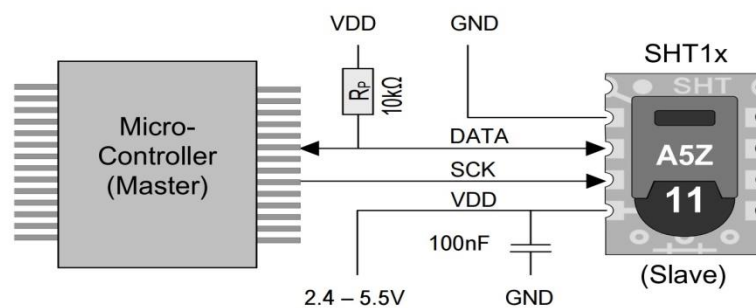


Figura 2.3.2-3 Diagrama de conexión sensor SHT15 y protocolo SPI.

Características:

- ◆ Alimentación: 3.3Vdc
- ◆ Corriente: 0.55mA
- ◆ Precisión humedad relativa: $\pm 2\%$ RH
- ◆ Respuesta humedad relativa: 8s
- ◆ Rango humedad relativa: 0-100%RH
- ◆ Resolución temperatura: $\pm 0.3^\circ\text{C}$
- ◆ Rango temperatura: $-40^\circ\text{C} - +123.8^\circ\text{C}$
- ◆ Respuesta temperatura: 5s – 30s

2.3.3 Sensores de temperatura

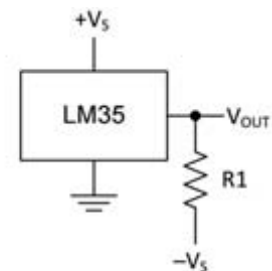
El principio básico de un sensor de temperatura se basa en los termistores y se trata de resistencias cuyo valor asciende con la temperatura (termistor **PTC** o **Positive Temperature Coefficient**) o bien disminuye con la temperatura (termistor **NTC** o **Negative Temperature Coefficient**), por lo tanto para este proyecto, depende de la temperatura que el termistor permita o no el paso de la corriente por el circuito de control del sistema.

A diferencia de la humedad, para esta propuesta, censar la temperatura de un cuerpo o sistema no requiere un sensor diseñado para sólidos y líquidos y otro diseñado para gases, pues el calor de la masa a medir se transmitirá mayormente por conducción al sensor, sea la masa de aire o de agua.

A continuación se presentan algunos sensores utilizados comúnmente en la electrónica:



Figura 2.3.3-1 Sensor LM35 encapsulado DIP.



$$R_1 = -V_S / 50 \mu\text{A}$$
$$V_{\text{OUT}} = 1500 \text{ mV at } 150^\circ\text{C}$$
$$V_{\text{OUT}} = 250 \text{ mV at } 25^\circ\text{C}$$
$$V_{\text{OUT}} = -550 \text{ mV at } -55^\circ\text{C}$$

Figura 2.3.3-2 Conexión sensor LM35.

Sensor de temperatura LM35

Es un dispositivo circuito integrado de precisión de temperatura e incluso está disponible en un encapsulado hermético a prueba de agua, su salida es voltaje linealmente proporcional a la temperatura, se observa en la figura 2.3.3-1.

Características:

- ◆ Alimentación: 4Vdc – 30Vdc
- ◆ Precisión @25°C: $\pm 0.5^{\circ}\text{C}$
- ◆ Rango: -50°C – $+150^{\circ}\text{C}$
- ◆ Salida: Analógica
- ◆ Factor de escala: $+10\text{mV}/^{\circ}\text{C}$

Requiere un arreglo simple con una resistencia para poder funcionar en su rango completo de trabajo de -50°C – $+150^{\circ}\text{C}$ como se observan en las figuras 2.3.3-2. En una etapa muy temprana de este proyecto, el sensor LM35 iría a ser utilizado para censar la temperatura del aire, pero fue sustituido por un sensor digital que elimina el problema de ser muy susceptible a caídas de tensión en los conductores que lo alimentan.

Sensor de temperatura TCN75A

Es un sensor digital de temperatura, en la figura 2.3.3-3, capaz de cesar temperaturas de -40°C a $+25^{\circ}\text{C}$. La información de la temperatura es medida desde un sensor de temperatura, después es convertido a un Word con la selección de 9 a 12 bit mediante un ADC (convertidor analógico-digital).



Figura 2.3.3-3 Sensor TCN75A.

Su comunicación es mediante el protocolo **I2C** o **I²C** (Inter-Integrated Circuit, Circuito Inter-Integrado), y está diseñado como un bus maestro-esclavo. La transferencia de datos es siempre inicializada por un maestro, por lo que el esclavo reacciona. Es posible tener varios maestros (**Multimaster-Mode**). En el modo multimaster pueden comunicarse dos maestros entre ellos mismos, de modo

que uno de ellos trabaja como esclavo. El arbitraje (control de acceso en el bus) se rige por las especificaciones, de este modo los maestros pueden ir turnándose.

El I2C precisa de dos líneas de señal: reloj (**SCL, Serial Clock**) y la línea de datos (**SDA, Serial Data**) como se observa en la figura 2.3.3-4.

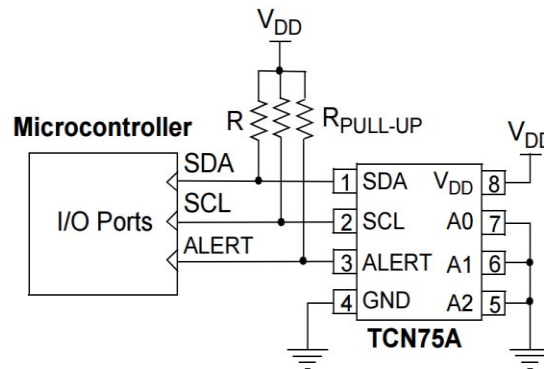


Figura 2.3.3-4 Conexión TCN75A con protocolo I2C a un microcontrolador.

Sensor de temperatura DS18B20

Es un sensor digital (figura 2.3.3-5) que provee la medición de temperatura en 9-bit a 12-bit, también disponible en encapsulado tipo sonda a prueba de agua, utiliza comunicación **1-Wire** que solo requiere una línea de datos, puede también, derivar energía directamente de la línea de datos (“**potencia o energía parasita**”) eliminando la necesidad de una fuente de alimentación.

Cada DS18B20 tiene un único código de serie de 64-bit, lo cual permite múltiples DS18B20 funcionen en el mismo bus **1-Wire**. Esto permite monitorear sistemas de temperatura dentro de construcciones, equipo o maquinaria, procesos de monitoreo y sistemas de control.

Características:

- ◆ Alimentación: 3.3Vdc – 5.5Vdc

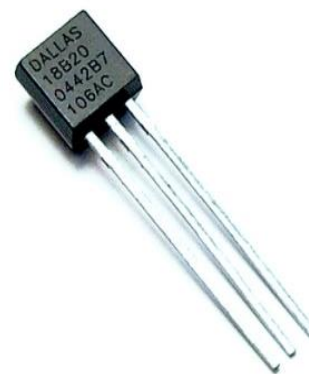


Figura 2.3.3-5 Sensor DS18B20 DIP

- ◆ Rango: $-55^{\circ}\text{C} - +125^{\circ}\text{C}$
- ◆ Precisión: $\pm 0.5^{\circ}\text{C}$
- ◆ Corriente: 1mA
- ◆ Salida: Digital

Por sus características y su popularidad, este sensor fue el elegido para monitorear la temperatura del estanque de agua de esta propuesta, debido a su fácil adquisición en tiendas de electrónica y su presentación de sonda (figura 2.3.3-6).



Figura 2.3.3-6 Sensor DS18B20 sonda.

Sensor de temperatura y humedad DHT11

Es un sensor de temperatura y humedad de salida digital (figura 2.3.3-7), está constituido por un sensor resistivo de humedad y un componente de medición de temperatura NTC.

Un formato de datos Single-Bus es usado para comunicación síncrona entre el microcontrolador y el sensor DHT11. Cada proceso de comunicación demora aproximadamente 4ms y cada transmisión de datos es de 40-bit, envía el bit más significativo primero. El formato de información es el siguiente: 8-Bit Valor RH entero + 8-bit Valor RH decimal + 8-bit Valor T entero + 8-bit Valor T decimal + 8-bit suma de comprobación.

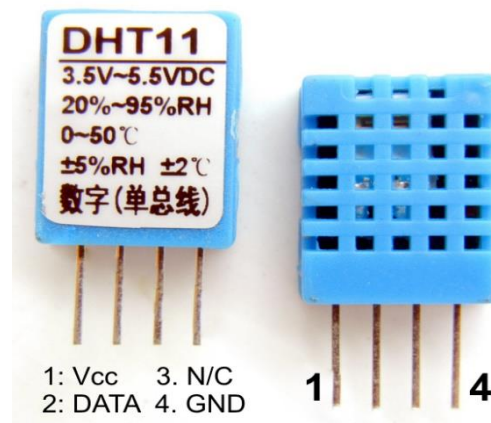


Figura 2.3.3-7 Sensor DHT11

Características:

- ◆ Alimentación: 5Vdc
- ◆ Corriente: 0.5mA
- ◆ Periodo de muestreo: 1s
- ◆ Salida: Digital

- ◆ Resolución Humedad Relativa: 1%RH
- ◆ Precisión Humedad Relativa: $\pm 4\%$ RH
- ◆ Rango Humedad Relativa @25°C: 20%RH – 90%RH
- ◆ Tiempo de respuesta: 6s-30s
- ◆ Resolución Temperatura Aire: 1°C
- ◆ Precisión Temperatura: $\pm 1^\circ\text{C}$
- ◆ Rango Temperatura: 0°C – 50°C

Debido a sus características y a su capacidad de integrar dos sensores (temperatura y humedad relativa) en un solo dispositivo, además de ser fácil de adquirir, pues es bastante popular, y su salida digital, que se es menos afectado por una atenuación de voltaje en comparación con una salida analógica, este sensor fue seleccionado para censar la temperatura ambiente y la humedad relativa en este proyecto.

2.3.4 Sensores de nivel de agua

Para asegurar algunos procesos que requiera un nivel mínimo de agua, se utilizan algunos métodos para medir el nivel, sea por un método mecánico como un flotador, o por un medio por el que se hace pasar una pequeña corriente a través del agua, o incluso algunos sensores de proximidad.

El termino nivel se utiliza para describir la horizontalidad de una determinada cosa. Por ejemplo, puede reconocerse a un nivel como la altura a la que llega algo o a la que ha sido ubicado. También se acepta como la altura máxima que alcanza la superficie de una sustancia líquida. A continuación se describen algunos sensores utilizados para medir el de nivel.

Sensor de Nivel de Agua “Funduino”

Este Sensor consiste en un PCB con una sección de pistas expuestas, cuando entran en contacto con el agua permiten la polarización de base de un transistor, lo que ocasiona una corriente proporcional al nivel del líquido en el *pin* (terminal) de salida, permitiendo así estimar en forma aproximada el nivel del líquido. En la figura 2.3.4-1 se observa dicho sensor. Es importante mencionar que dependiendo de los minerales o incluso la composición misma del líquido donde este sumergido, alterará el valor analógico de su salida, por lo que se debe calibrar, ya sea en la programación del microcontrolador, o añadiendo una etapa de acondicionamiento de señal.



Figura 2.3.4-1 Sensor de nivel de agua “Funduino”

Características:

- ◆ Voltaje de alimentación: 3Vdc – 5Vdc
- ◆ Corriente: 20mA.
- ◆ Salida: Analógica
- ◆ Temperatura de operación recomendada: 10°C – 30°C
- ◆ Área de detección: 40mm X 16mm
- ◆ Dimensiones 65mm X 20mm X 0.8mm

Este sensor originalmente sería utilizado para el proyecto, para activar el actuador que eleva la temperatura del agua, solamente si el nivel del agua es suficiente, pero durante las pruebas se observó que, la propiedad física del agua, capilaridad, hace subir

un poco de agua en las pistas, esto proporciona una lectura errónea del nivel de agua en un largo lapso de tiempo, tocar las pistas también afecta el nivel censado, además de no ser totalmente sumergible.

Sensor de proximidad ultrasónico HC-SRF-04

Este sensor consta de dos transductores en su concepto básico, un altavoz y un micrófono, su funcionamiento es semejante a un sonar, es decir, el altavoz dispara pulsos de sonido de frecuencia ultrasónica, y el micrófono recibe el eco, el tiempo diferido será proporcional a la distancia del objeto. No se ve afectado por la luz solar o el material negro como sensores de distancia ópticos, aunque acústicamente materiales suaves como telas pueden ser difíciles de detectar, además su vida útil se ve muy comprometida en exteriores.

La velocidad del sonido en el aire, a una temperatura de 20 °C, es de 343 m/s y se debe considerar por cada grado centígrado que sube la temperatura, la velocidad del sonido aumenta en 0,6 m/s. En la figura 2.3.4-2 se observa el sensor HC-SRF-04.

PinOut:

- ◆ Vcc: Alimentación
- ◆ Trigger: Disparo de entrada del sensor
- ◆ Echo: Eco de salida del sensor
- ◆ Gnd

Características:

- ◆ Alimentación: 5Vdc
- ◆ Corriente: 15mA
- ◆ Ángulo de medición: 30°
- ◆ Ángulo de medición efectivo: < 15°
- ◆ Rango de detección: 2cm – 400cm
- ◆ Resolución: 3mm
- ◆ Frecuencia de trabajo: 40KHz

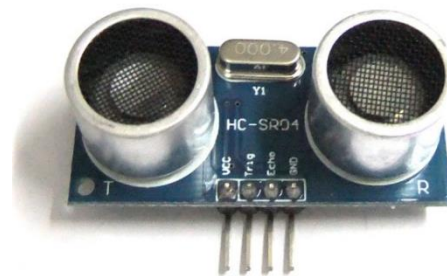


Figura 2.3.4-2 Sensor HC-SRF-04.

- ◆ Sensor óptico reflexivo TCRT5000

El TCRT5000L

Es un sensor reflexivo que incluye un emisor infrarrojo y un fototransistor, bloquea la luz visible. Este tipo de sensores emiten un tren de pulsos de luz infrarroja, mediante la ayuda de un microcontrolador, la luz rebota en los objetos y nuevamente llega al dispositivo, pero el fototransistor es quien se ve afectado por la presencia de la luz rebotada, en la figura 2.3.4-3 se observa el sensor óptico.



Figura 2.3.4-3 Sensor TCRT5000L.

Características:

- ◆ Alimentación: 5Vdc
- ◆ Rango de trabajo: 0.2mm – 15mm
- ◆ Corriente de salida: 1mA
- ◆ Filtro luz de sol
- ◆ Longitud de onda emitida: 950nm

Interruptor a prueba de agua

Se compone de un interruptor magnético y un imán en forma de anillo insertado en un material menos denso que el agua, lo que permite elevar el imán cuando es sumergido en agua, esto cierra el interruptor, como se observa en la figura 2.3.4-4. Estrictamente no se puede calcular el nivel de agua hasta que es inferior a la posición del interruptor. En la figura 2.3.4-5 se observa el interruptor.

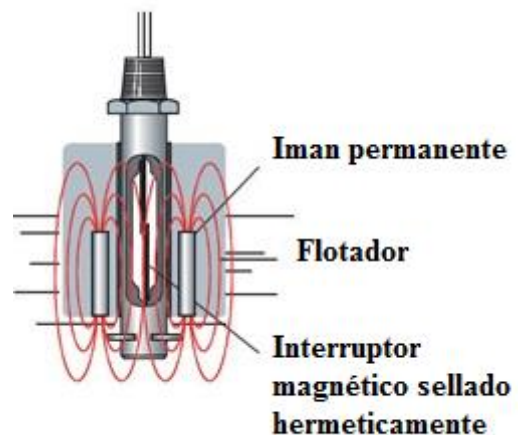


Figura 2.3.4-4 Funcionamiento interruptor a prueba de agua.

Características:

- ◆ Voltaje máximo: 110Vac
- ◆ Corriente máxima: 0.5A
- ◆ Temperatura -20 °C – 80°C

Este interruptor fue utilizado para este proyecto, pues no se ve afectado por la capilaridad del agua ni por otros factores y es menos susceptible a ser interferido por el reptil, como es el caso de los sensores ópticos o ultrasónicos, donde podrían medir la distancia entre los sensores y el reptil y no entre ellos mismos y el agua.



Figura 2.3.4-5 Interruptor a prueba de agua.

2.4 Tipos de fuentes de alimentación

Las fuentes de poder pueden ser **lineales** o **conmutadas**. Las fuentes lineales siguen el esquema de **transformador** (reductor de tensión), **rectificador** (conversión de voltaje alterno a onda completa), **filtro** (conversión de onda completa a continua) y **regulación** (mantenimiento del voltaje de salida ante variaciones en la carga). Las fuentes conmutadas, en cambio, convierten la energía eléctrica por medio de **conmutación de alta frecuencia** sobre transistores de potencia. Las fuentes lineales son típicamente de regulación ineficiente, comparadas con fuentes conmutativas de potencia similar. Las fuentes lineales son las más utilizadas cuando se requiere un diseño compacto y de bajo costo.

Las etapas esenciales que cumple la fuente son cuatro:

Transformación

Ahí se consigue reducir la tensión de entrada a la fuente (220 v o 125 v), que son las que suministra la red eléctrica. Participa un transformado, la salida de este dependerá de su devanado principal y secundario y del diseño requerido.

Rectificación

Tiene el objetivo de asegurar que no se produzcan oscilaciones de voltaje en el tiempo. Se intenta con esta etapa pasar de corriente alterna a corriente continua a través de un componente que se llama puente rectificador (constituido por cuatro diodos rectificadores). Esto permite que el voltaje no baje de 0 voltios, y siempre se mantenga por encima de este nivel.

Filtrado

En esta fase se aplana al máximo la señal, eso se consigue con uno o varios capacitores, que retienen la corriente y la dejan pasar lentamente, con lo que se logra, vista en un osciloscopio, una salida prácticamente plana.

Estabilización

Cuando se dispone ya de la señal continua y casi del todo es plana, el regulador o estabilizador es un circuito que se encarga de reducir el rizado y de proporcionar una tensión de salida de la tensión idealmente exacta.

2.4.1 Fuentes conmutadas

Su funcionamiento se puede simplificar en un diagrama de cuatro bloques, donde se puede apreciar la particularidad del controlador, que es el encargado de cambiar la frecuencia de la línea de 60Hz a alta frecuencia, disparando la base del transistor como se observa en la figura 2.4.1-1, cabe mencionar que los bloques no son estrictamente invariables, dependerá del diseñador del circuito.

Al ser energizadas por una toma doméstica, este tipo de fuentes se caracterizan por convertir AC a AC a posteriormente a DC, pero también pueden ser utilizadas como **inversores de corriente**, es decir, fuentes de DC a AC, como los inversores utilizados en automóviles para poder alimentar algún electrodoméstico de 127Vac cuando solo se cuenta con la energía eléctrica del automóvil de 12Vdc. Si se observa el esquema de la figura 2.4.1-1, se puede destacar que el controlador es alimentado por DC, y la salida del transformador es AC, aprovechando esta cualidad, las fuentes conmutadas presentan una gran ventaja a las fuentes lineales, y es la capacidad de convertir DC a AC a cualquier frecuencia que soporte el controlador.

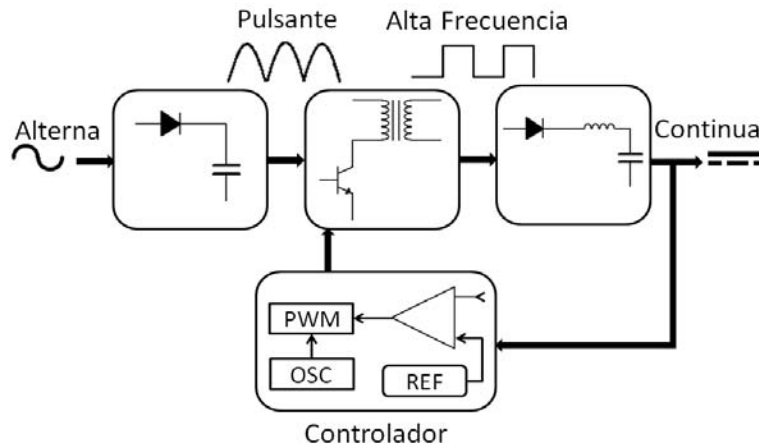


Figura 2.4.1-1 Esquema fuente conmutada.

2.4.2 Fuentes lineales

Su funcionamiento se puede simplificar en un diagrama de bloques, a diferencia de las fuentes conmutadas, las fuentes lineales son más sencillas en su construcción, y prácticamente es invariable el diseño aun cuando se demanda más corriente o más voltaje. En la figura 2.4.2-1 se observa en cada bloque un solo dispositivo, y posiblemente la totalidad de los elementos necesarios para su construcción, si se optó así. Las fuentes lineales pueden ser utilizadas en muchas ocasiones sin una etapa de **regulación** cuando el voltaje en el devanado primario de **transformación** es invariable, debido a que en la **filtración** se puede conseguir aplanar casi en la totalidad la señal, pudiendo volver aún más simple su construcción, pero esta omisión de la última etapa no debe ser realizada cuando el aparato a alimentar no tolera variaciones de voltaje, por lo que no es recomendable dicha omisión

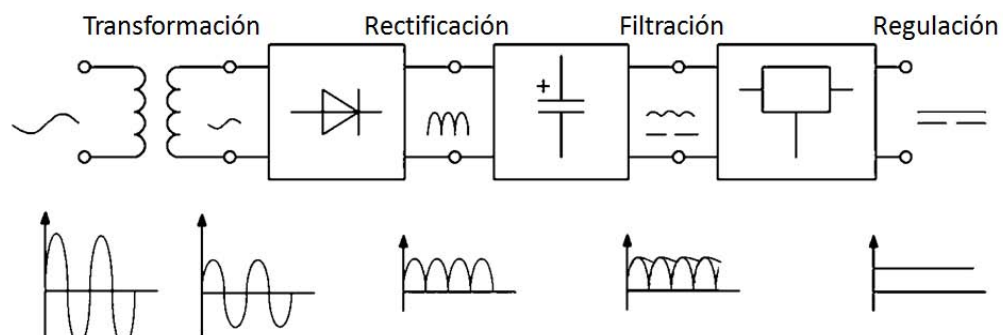


Figura 2.4.2-1 Esquema fuente lineal.

2.5 Sistemas de calefacción

Al hablar de calefacción, lo primero que pensamos es aire caliente, para obtener adecuadamente aire caliente para elevar la temperatura de un espacio cerrado, se debe considerar elementos que constituyen una maquina térmica. En principio dicha maquina termina debe ceder calor en el espacio cerrado y absorber calor en el exterior, el intercambio térmico sucederá según la maquina térmica empleada, a continuación se describe el funcionamiento de los métodos más utilizados en sistemas calefactores. El concepto de calefacción puede ser definido como un conjunto de aparatos que forman un sistema y sirven para calentar un lugar, especialmente un edificio o una parte de él.

Calefacción eléctrica

La **electricidad es convertida en calor a través de un aparato eléctrico**. Así, dentro de cada calefactor eléctrico hay una resistencia eléctrica por la que pasa la corriente. Esta corriente eléctrica convierte la electricidad en calor. Los aparatos de calefacción eléctrica siguen este principio, con la única diferencia en cuanto al método por el cual proporcionan el calor.

Sistemas de convección

Los sistemas que más se utilizan en lugares con inviernos fríos, son llamados **“radiación por suelo”** (figura 2.5-1) y **techo**. Funcionan mediante la electricidad o agua proveniente de una caldera. Al encenderlos, en el caso de los eléctricos, corriente eléctrica fluye mediante material conductor instalado debajo del piso o dentro de la pared, el conductor proporciona calor que a su vez calienta el suelo o pared. En suelo el calor se extiende en el aire, y después se extiende a través de la convección del aire. El calor comienza del suelo al techo consiguiendo crear una temperatura balanceada en la habitación.

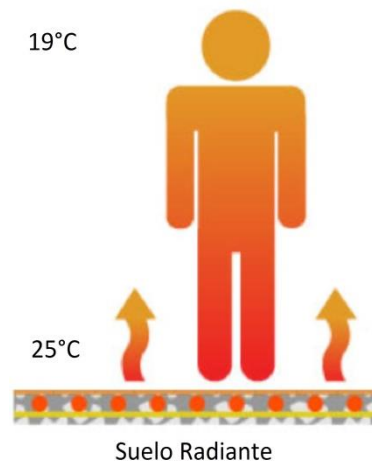


Figura 2.5-1 Calefacción por convección o “suelo radiante”.

Convección forzada

Calentadores de gas o eléctricos y la mayoría de calefactores portátiles suelen utilizar un ventilador que fuerza la entrada de aire al calentador, que en el caso de los eléctricos, ejemplo simple en la figura 2.5-2, constan de un arreglo de resistencias, el aire fluye a través de este arreglo, es calentado y después expulsado del aparato.

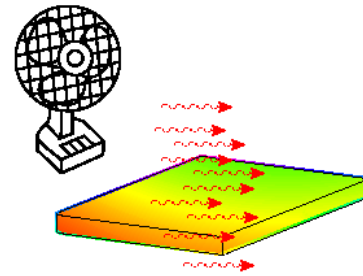


Figura 2.5-2 Esquema convección forzada.

Este tipo de calefacción fue la seleccionada para la propuesta, pues su capacidad de ser controlada mediante electricidad y no requerir de una instalación de ningún otro elemento, facilita el desarrollo del proyecto.

Sistema de radiadores

La caldera de gas calienta el agua que circula por el circuito de tuberías, del mismo modo que en sistema eléctrico, los radiadores irradian el calor. El agua que se enfría eventualmente vuelve a la caldera a través de una tubería de regreso. Así, repartidos por las habitaciones de la casa están colocados los radiadores, se observa un diagrama en la figura 2.5-3.

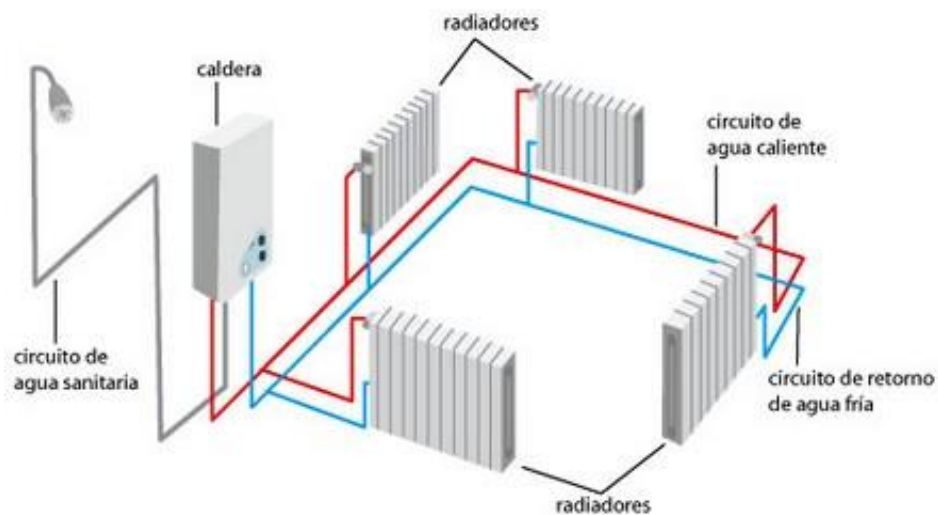


Figura 2.5-3 Calefacción por sistema de radiadores.

2.6 Diseño y programación de interfaz con Android

En la actualidad el diseño y creación de aplicaciones para teléfonos inteligentes ha llegado a manos de los usuarios y sin requerir conocimientos avanzados de programación en algún lenguaje de programación específico, como suele ser Java, los usuarios pueden desarrollar aplicaciones que pueden ser muy sencillas hasta muy complejas.

Un software libre para poder diseñar y programar aplicaciones en sistema operativo Android fue desarrollado en conjunto por el Massachusetts Institute of Technology o por sus siglas MIT y por Google, el software desarrollado fue nombrado MIT App Inventor 2, uno de sus objetivos es introducir a programadores principiantes en la creación de aplicaciones, que transforma un lenguaje complejo de programación en un lenguaje visual de bloques. Una interfaz simple de gráficos otorga, incluso a usuarios sin experiencia, la habilidad de crear un sencilla pero funcional aplicación en menos de una hora.

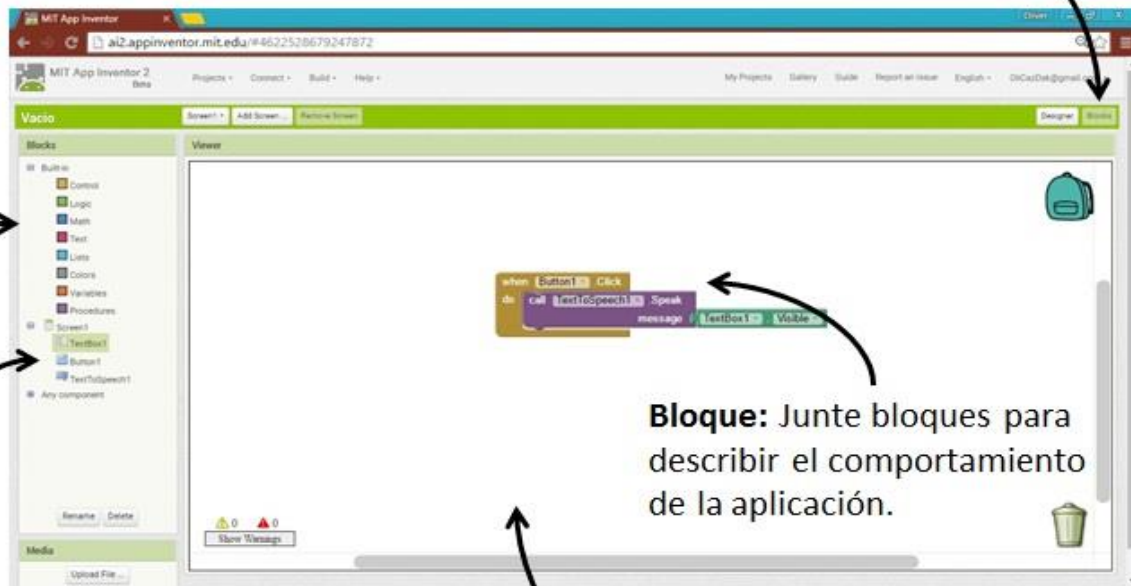
MIT App Inventor 2 consta de dos ventanas principales:

Ventana Diseño

En esta ventana (figura 2.6-1) se designa la posición y la apariencia visual de los elementos que conformaran la aplicación, pudiendo elegirse elementos visibles para el usuario como botones, listas, barras deslizantes, cuadros de texto entre otros, y elementos no visibles, como arreglos de columnas y filas para organizar los elementos visibles, Clocks (relojes) con intervalos de tiempo ajustables necesarios para algunos ciclos de la programación de la aplicación, alertas que pueden ser cuadros de texto o sonoras, disposición de algunos elementos de los celulares, como acelerómetro, altavoz, vibrador, Bluetooth, giroscopio, entre otros.

Bloques incorporados: Encuentre bloques para comportamientos generales que desee agregar a su aplicación, arrástrelos al Visor de Bloques.

Botón Bloque: Clic para ir a la ventana Bloques.



Bloque: Junte bloques para describir el comportamiento de la aplicación.

Visor de Bloques: Arrastre Bloques al Visor de Bloques para describir relaciones y comportamiento de la aplicación.

Componentes Específicos: Encuentre Bloques por comportamiento de Componentes específicos y arrástrelos al Visor de Bloques.

Figura 2.6-1 Ventana Diseño.

Ventana Bloques

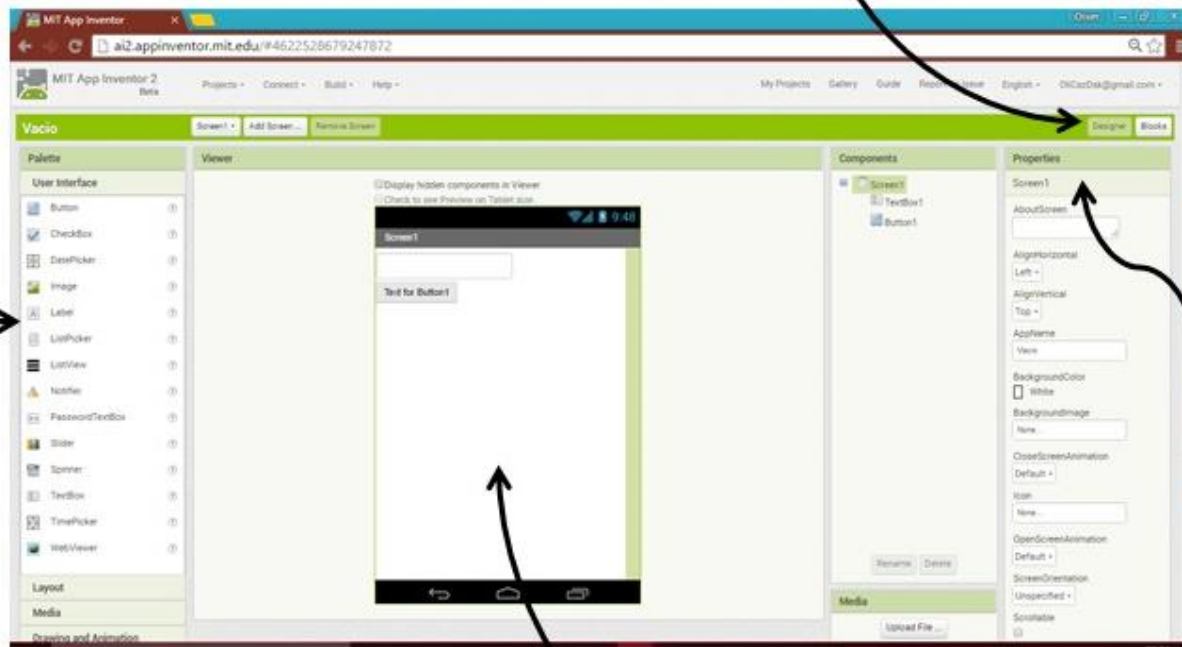
En esta ventana (figura 2.6-2) se realiza la programación de la aplicación, la programación visual facilita la descripción de algoritmos, dejando de lado un lenguaje de programación con sintaxis estricta. Esta ventana contiene los elementos generales, como control, condicionales, ciclos While o For, así como operadores lógicos, matemáticos, y algunas funciones que podrán modificar o funcionar texto.

Es importante resaltar que MIT App Inventor 2 discrimina Texto y Número, a diferencia de otros compiladores que relacionan los caracteres alfanuméricos a un valor de 7-bit llamado American Standard Code for Information Interchange, o ASCII por sus siglas, razón por la que se pueden observar Bloques de programación para Texto y Bloques de programación para Número, esto con la finalidad de poder manipular o describir, el algoritmo a programar, de diferente manera según la composición de las variables y resultados deseados.

Paleta: Encuentre los componentes y arrástrelos al Visor para añadirlos a la aplicación.

Botón Diseño: Clic para ir a la ventana Diseño.

Figura 2.6-2 Ventana Bloques.



Propiedades: Seleccione un componente de la lista para cambiar sus propiedades, color, tamaño, comportamiento.

Visor: Arrastre componentes desde la Paleta para ver como lucirán en la aplicación.

2.6.1 Importar aplicaciones al dispositivo Android¹

MIT App Inventor 2 es ejecutado desde un navegador de Internet, y para poder ejecutar las aplicaciones creadas y exportarlas al dispositivo móvil, los desarrolladores sugieren dos opciones mostradas a continuación:

2.6.1.1 Modo inalámbrico

Si se utiliza un dispositivo con sistema operativo Android y se tiene una conexión inalámbrica a Internet, se puede instalar App Inventor Companion App en el dispositivo Android, esta opción es recomendada por los desarrolladores, a continuación se describen los pasos a seguir.

Paso 1

Descargue e instale la aplicación MIT AI2 Companion App en su dispositivo Android. Abra el escáner de código Quick Response, o por sus siglas QR, en su dispositivo y escanee el siguiente código QR (figura 2.6.1.1-1) para descargar la aplicación MIT AI2 Companion App a su dispositivo Android.

También se puede descargar MIT AI2 Companion App desde el siguiente link:

<http://ai2.appinventor.mit.edu/companions/MITAI2Companion.apk>

Después de la descarga siga las instrucciones de instalación mostradas en la liga del link anterior, solo es necesario instalar esta aplicación una vez, y dejarla instalada en el dispositivo con sistema operativo Android.

Paso 2

Conecte su computadora y su dispositivo Android a la misma red WiFi.



Figura 2.6.1.1-1 Código QR
MIT App Inventor 2.

¹ <http://appinventor.mit.edu/explore/get-started?>

MIT App Inventor 2 mostrara automáticamente la aplicación creada, pero solo si la computadora y el dispositivo Android están conectados a la misma red WiFi.

Paso 3

Abra un proyecto en App Inventor y conecte su dispositivo. Seguidamente abra un proyecto en MIT App Inventor 2 y elija “Connect” y elija “AI Companion” desde el menú superior en el navegador MIT App Inventor 2 como se muestra en la figura 2.6.1.1-2.

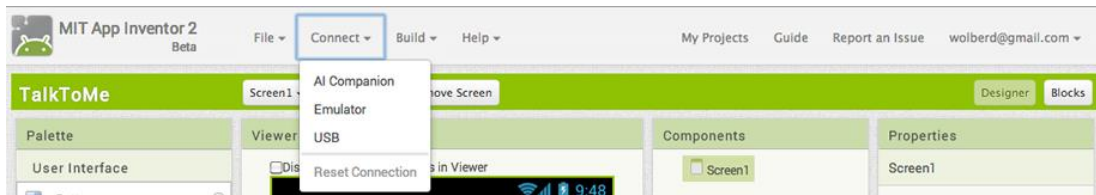


Figura 2.6.1.1-2 Selección AI Companion.

Un dialogo con un código QR (Figura 2.6.1.1-3) aparecerá en la pantalla de su computadora, en su dispositivo Android, ejecute la aplicación MIT App Companion como cualquier otra aplicación, después de clic en el botón “Scan QR code” y escanee el código en la ventana del navegador MIT App Inventor 2.

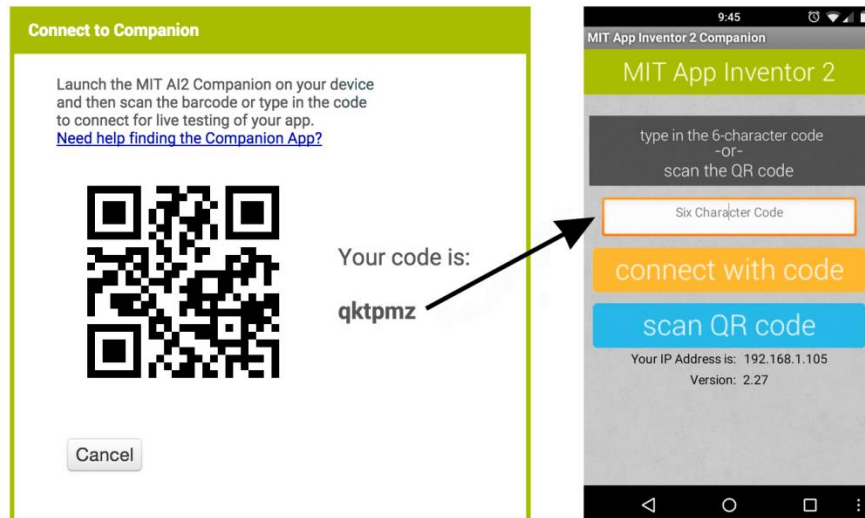


Figura 2.6.1.1-3 Escaneo del código QR.

Después de unos segundos la aplicación creada comenzara a instalarse en su dispositivo Android, podrá actualizarse cuando haga cambios en su diseño y bloques de programación gracias a un característica llamada “Live Testing”.

2.6.1.2 Modo descarga en formato APK

Los desarrolladores de MIT App Inventor 2 permiten la descarga de las aplicaciones creadas en formato Android Application Package, o por sus siglas APK, dicho paquete debe ser instalado manualmente por el usuario, para realizar este empaquetado se deben seguir los siguientes pasos:

Paso 1

Dar clic en el botón “Built” en la barra del navegador MIT App Inventor 2 como se muestra en la figura 2.6.1.2-1.

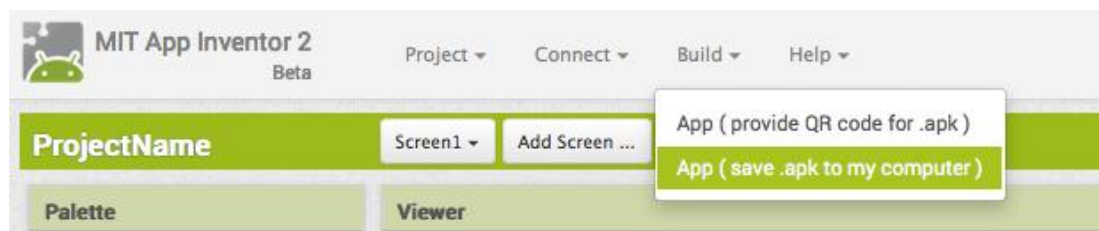


Figura 2.6.1.2-1 Selección descarga.

Paso 2

Seleccionar “save .apk to my computer”. Una ventana emergente alertara que la descarga ha comenzado (figura 2.6.1.2-2).

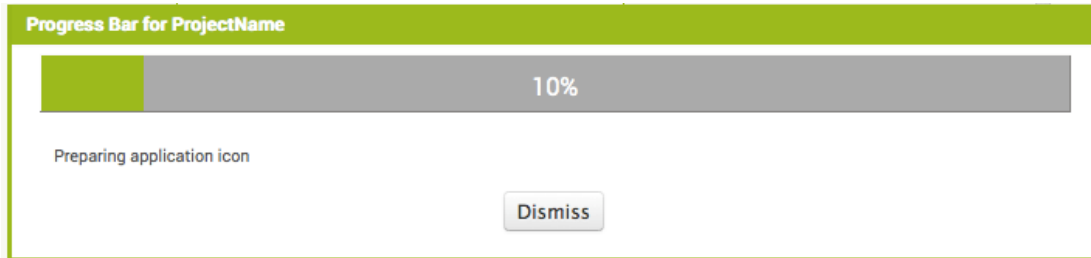


Figura 2.6.1.2-2 Descarga aplicación.

Paso 3

Debe copiarse el archivo .apk en el dispositivo Android o en una memoria extraíble para poder ser ejecutada desde el teléfono móvil.

Debido a que no requiere la instalación y a que facilita la programación de algoritmos este software fue elegido para el desarrollo del proyecto.

2.7 Sistemas de Control

Se define sistema como la combinación de componentes que actúan conjuntamente y cumplen un determinado objetivo (figura 2.7-1).



Figura 2.7-1 Diagrama a bloques de un sistema.

En el mundo actual industrializado, donde procesos complejos engloban sistemas mecánicos, neumáticos, electrónicos, y combinaciones de estos, es necesario recurrir a un análisis en su diseño, con el fin de desarrollar sistemas que a pesar de su complejidad o su velocidad de funcionamiento, sean fiables y eficientes.

El control de un sistema se efectúa mediante un conjunto de componentes mecánicos, hidráulicos, eléctricos y electrónicos que interconectados, recogen información acerca del funcionamiento, comparan este funcionamiento con datos

previos y, si es necesario, modifican el proceso para alcanzar el resultado deseado. Se puede definir como un conjunto de dispositivos que actúan para lograr un objetivo de control.

Otros conceptos que se deben describir para poder comprender la propuesta planteada se enlistan a continuación:

- ◆ Variable de entrada: Variable del sistema que una modificación de su magnitud o condición puede alterar el estado del sistema.
- ◆ Variable de salida: Variable del sistema cuya magnitud o condición se mide.
- ◆ Perturbación: Es una señal que tiende a afectar el valor de la salida de un sistema. Si la perturbación se genera dentro del sistema se la denomina interna, mientras que una perturbación externa se genera fuera del sistema y constituye una entrada.
- ◆ Estabilidad: Un sistema es estable si ante una entrada de magnitud limitada, responde con una salida también limitada.

Se ejemplifican estos conceptos en la figura 2.7-2. Existen dos clasificaciones de los sistemas de control, dependiendo de la configuración de sus elementos, sistemas de control de lazo abierto y sistemas de control de lazo cerrado.

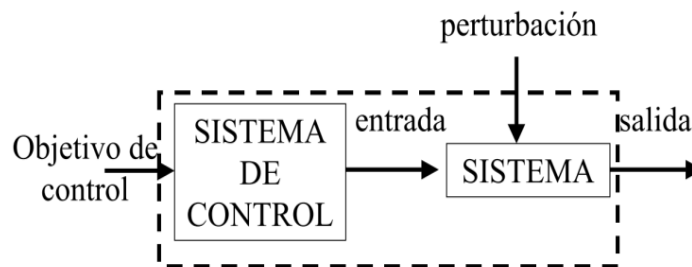


Figura 2.7-2 Diagrama a bloques de un sistema de control.

2.7.1 Sistemas de control de lazo abierto

Es el sistema en el que la variable de salida (variable controlada) no tiene efecto sobre la acción de control (variable de control), se ejemplifica en el diagrama a bloques de la figura 2.7.1-1.

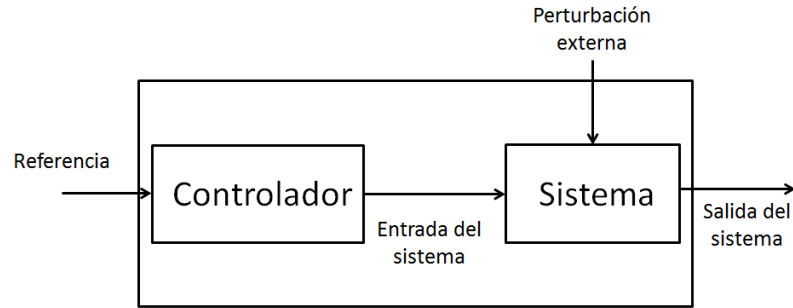


Figura 2.7.1-1 Sistema de control de lazo abierto.

Características:

- ◆ No se compara la salida del sistema con el valor deseado de la salida del sistema (referencia).
- ◆ Para cada entrada de referencia le corresponde una condición de operación fijada.
- ◆ La exactitud de la salida del sistema depende de la calibración del controlador.
- ◆ En presencia de perturbaciones estos sistemas de control no cumplen su función adecuadamente.

El control en lazo abierto suele aparecer en dispositivos con control secuencial, en el que no hay una regulación de variables sino que se realizan una serie de operaciones de una manera determinada. Esa secuencia de operaciones puede venir impuesta por eventos (event-driven) o por tiempo (timed-driven).

2.7.2 Sistemas de control de lazo cerrado

Se define como el sistema en que la señal de salida (variable controlada) tiene efecto directo sobre la acción de control (variable de control), como se ejemplifica en la figura 2.7.2-1.

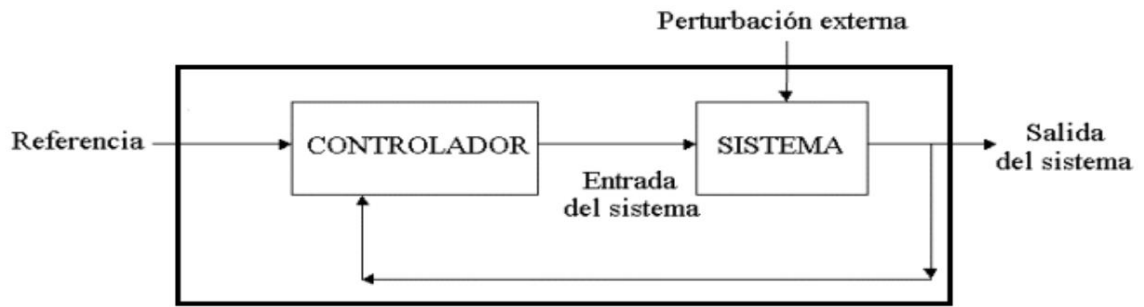


Figura 2.7.2-1 Sistema de control de lazo cerrado.

2.7.2.1 Sistemas de control retroalimentados

Los sistemas de control retroalimentados se clasifican como sistema de lazo cerrado. Se caracterizan por realizar operaciones que en presencia de perturbaciones tienden a

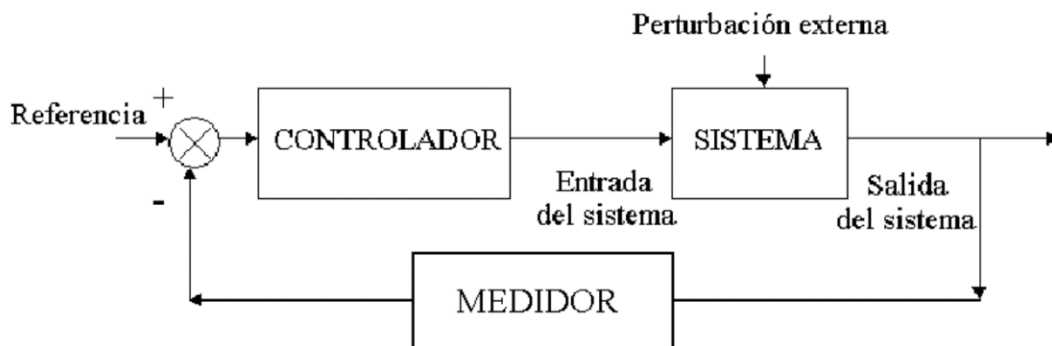


Figura 2.7.2.1-1 Sistema de control retroalimentado.

reducir la diferencia entre la salida de un sistema y alguna entrada de referencia. Esta reducción se logra manipulando alguna variable de entrada del sistema, siendo la magnitud de la variable de entrada, función de la diferencia entre la variable de referencia y la salida del sistema, descrito en la figura 2.7.2.1-1.

La forma en que se realimenta el sistema puede ser negativa o positiva. La retroalimentación negativa podrá decrementar (convergencia de la o las entradas con respecto a la o las salidas), y se dice que tiende a equilibrio, se ejemplifica en la figura 2.7.2.1-2.

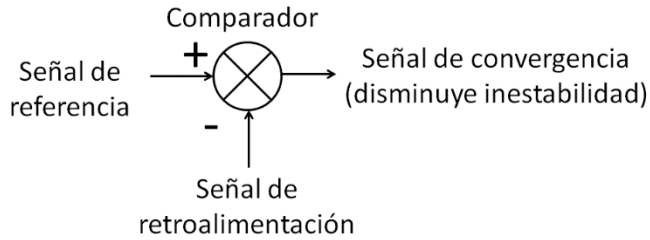


Figura 2.7.2.1-2 Retroalimentación negativa.

La retroalimentación positiva podrá acrecentar (divergencia de la o las entradas con respecto a la o las salidas) la señal de referencia, se ejemplifica en la figura 2.7.2.1-3.

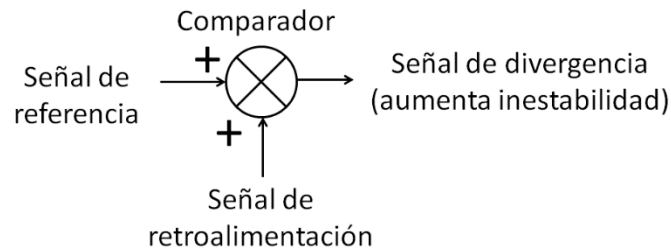


Figura 2.7.2.1-3 Retroalimentación positiva.

2.7.3 Elementos y funciones de un lazo de control

La constitución fundamental de un lazo de control se resume en el siguiente listado:

- ◆ Sistema a controlar: Sistema que se puede considerar aislado para su análisis de control.
- ◆ Controlador: Elemento de software o hardware que funciona de intermediario entre las entradas y salidas y el proceso para el que está diseñado a evaluar.
- ◆ Actuador (puede incluirse en el sistema a controlar): Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado.
- ◆ Medidor: Puede ser uno o varios sensores o transductores, que censan las magnitudes en el sistema para después ser devueltas como retroalimentación en lazo cerrado.

Las funciones de un lazo de control se enlistan a continuación:

- ◆ Medir el valor de la variable controlada (medición y transmisión).
- ◆ Detectar el error y generar una acción de control (decisión).
- ◆ Usar la acción de control para manipular alguna variable en el proceso de modo que tienda a reducir el error (manipulación).

2.7.4 Controladores

Desde la introducción de la tecnología electrónica a los sistemas de control se han creado diversos dispositivos conocidos como controladores o microcontroladores (al referirse a los chip que integran memorias, circuitos de entradas y salidas, circuitos de comunicación), y son los encargados de tomar decisiones y son mediadores entre los sensores y actuadores. Existen dispositivos diseñados específicamente para sistemas de control, como los Programmable Logic Controller o por sus siglas PLC, o se pueden adecuar algunos dispositivos de propósito general como los microcontroladores.

2.7.4.1 Programmable Logic Controller (PLC)

El Programmable Logic Controller, o por sus siglas PLC, es un dispositivo diseñado para controlar procesos secuenciales (una etapa después de la otra). El desarrollo actual de PLC incorpora relevadores en sus entradas y salidas como circuitos de protección, por lo que no requieren dispositivos adicionales para poder mediar entre los sensores y actuadores, además de poder

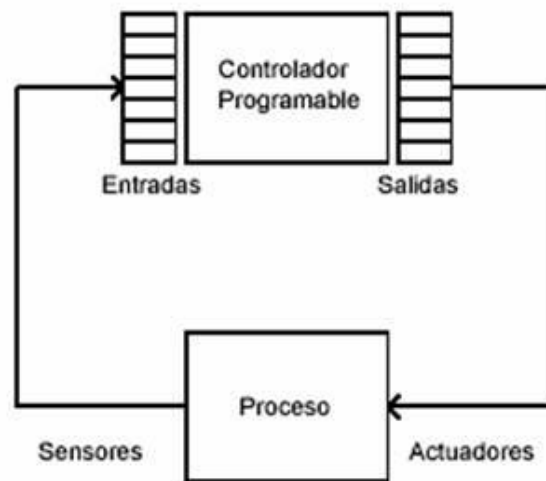


Figura 2.7.4.1-1 Esquema de PLC.

ser programados mediante una computadora o por una interfaz en el mismo PLC. Algunos cuentan con conexión a una red LAN para poder ser monitoreado a distancia,

incluso a través de una red WAN. En la figura 2.7.4.1-1 se observa un diagrama de bloques que describe un sistema de control con PLC.

Las características de los PLC cambiarán según el modelo y el fabricante, la mayoría de los PLC actuales cuentan con características de la siguiente lista:

- ◆ Alimentación: Puede ser 120Vac - 240 Vac o 12Vdc - 48 Vdc, dependerá del fabricante y modelo.
- ◆ Entradas/Salidas: En la mayoría de los PLC las entradas son el doble de las salidas, las más comunes en PLC sencillos son 16/8, 8/4, 4/2, algunas estas Entradas/Salidas pueden ser analógicas, y estarán señaladas por el fabricante.
- ◆ Timer: La mayoría de los PLC cuentan con Timer o temporizadores, estos pueden ser en unidades de tiempo en segundos, minutos, horas, días, y están enlazados a las entradas y salidas, lo que permite configurar el PLC para poder responder con un retardo a un cambio de estado en alguna entrada.
- ◆ Memoria: La memoria de los PLC puede ser volátil o no volátil, y dependerá del modelo. La memoria volátil permite almacenar información de los estados de las E/S mientras que la memoria no volátil permite almacenar registros de varias partes del PLC, como el temporizador o registros de las E/S, para continuar con el proceso en el último estado en caso de un corte de energía.
- ◆ Respaldo por batería: Incorporan baterías que evitan la pérdida de información de la programación del PLC, como puede ser la fecha y la hora, así sistemas que dependan de estos parámetros, no se verán afectados en caso de un corte de energía.

2.7.4.2 Microcontroladores

Nacen como dispositivos de propósito general e intentan reunir la mayor cantidad de circuitos en el menor espacio posible. Un microcontrolador es un circuito integrado, popularmente conocido como microchip y está constituido por una unidad central de procesamiento (sus siglas en inglés CPU), unidades de memoria volátil y no volátil, puertos de entrada y salida, comunicación serial y periféricos; En diagrama de bloques describe su diseño básico en la figura 2.7.4.2-1.

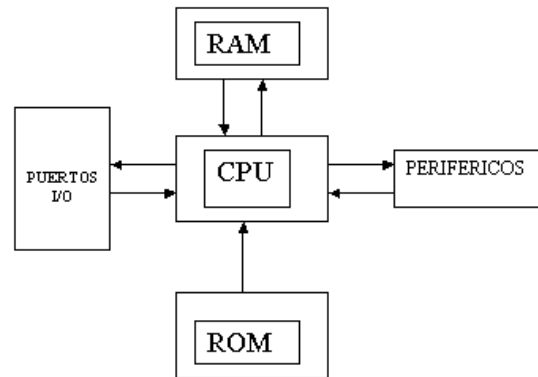


Figura 2.7.4.2-1 Diagrama de bloques de un Microcontrolador.

Toda microcontrolador requiere de un programa para que realice una función específica. Este se almacena normalmente en la memoria no volátil. El programa cargado también determinara el funcionamiento de sus pines y su comportamiento e interacción con los demás elementos del microcontrolador.

El propósito fundamental de los microcontroladores es el de leer y ejecutar los programas que el usuario le asigna, es por esto que la programación de un algoritmo es una actividad básica e indispensable cuando se diseñan circuitos y sistemas que los incluyan. La característica programable de los microcontroladores simplifica el diseño de circuitos electrónicos. Ya que un mismo circuito se puede utilizar para que realice diferentes funciones con solo cambiar el programa del microcontrolador lo hace ideal para sistemas de control.

Los microcontroladores más populares no cuentan con una etapa de potencia como protección en sus pines, por lo se han creado tarjetas de desarrollo para incrementar la versatilidad de los microprocesadores, adicionan reguladores de tención, cristales externos requeridos por los timer de los microcontroladores, botones con resistencias de pull-up (resistencias conectadas a los pines y de activación programable,

para la función reset, sockets en los pines de salida como divisor de corriente), alimentación y comunicación USB. Estas tarjetas de desarrollo eliminan la necesidad de tener que desmontar el microcontrolador, del circuito donde desempeña sus funciones como controlador, y montarlo en un grabador de microcontroladores, para después cargar el programa.

Entre estas tarjetas se encuentran Alteri (figura 2.7.4.2-2) basada en un PIC18F455 de la compañía Microchip, LaunchPad basada en un MSP432P401R de Texas Instruments, Wiring basada en un ATmega128 de Atmel, Propeller basada en el microcontrolador multitarea P8X32A-D40 de Parallax o Arduino Uno basada en un ATmega328p de Atmel.

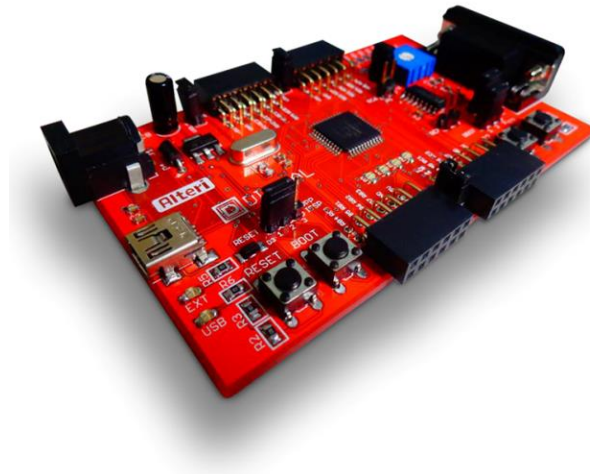


Figura 2.7.4.2-2 Alteri basada en un PIC18F455 de montaje superficial.

2.7.4.2.1 Tarjeta de desarrollo Arduino Uno²

Arduino Uno (figura 2.7.4.2.1-1) es un microcontrolador con 14 pines de Entradas/Salidas digitales, de las cuales 6 pueden ser usadas como salidas **PWM**, 6 como entradas analógicas, 2 pines que pueden ser configurados para ser utilizados por la **Universal Synchronous/Asynchronous Receiver/Transmitter** o por sus siglas **USART** y que permite comunicar el dispositivo con otros dispositivos que soporten el protocolo de comunicación, un conector de alimentación, un header **In-Circuit Serial**

² <https://www.arduino.cc/>

Programming, o por sus siglas **ICSP** (es la capacidad de algunos dispositivos programables, para ser programados mientras están instalados en un sistema, eliminando la necesidad de programar el dispositivo en un programador antes de ser instalado) y un botón reset para reiniciar el dispositivo.

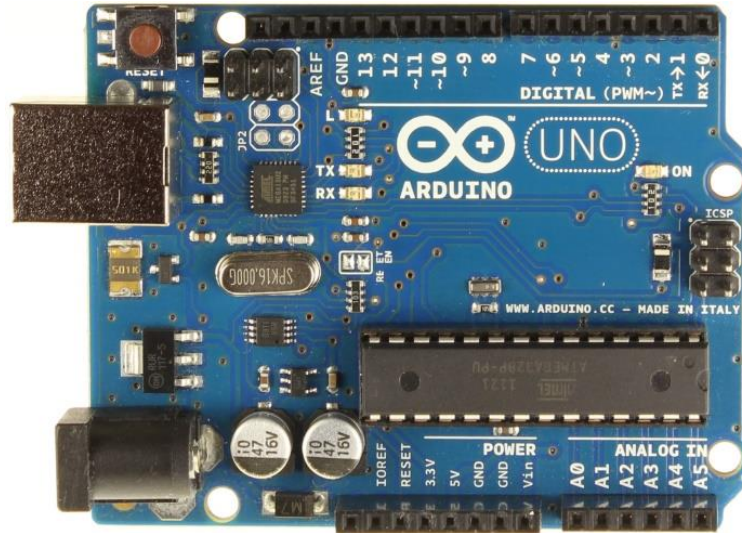


Figura 2.7.4.2.1-1 Arduino UNO basado en un ATmega328P encapsulado DIP.

El fabricante asegura que esta tarjeta de desarrollo contiene todo lo necesario para apoyar al microcontrolador, se conecta a la computadora mediante un cable USB, se puede alimentar con una fuente DC o mediante una batería en su conector de alimentación, se puede reemplazar el microcontrolador sin reemplazar toda la tarjeta de desarrollo.

Características:

- ◆ Microcontrolador: ATmega328P
- ◆ Alimentación del microcontrolador: 5Vdc
- ◆ Voltaje de alimentación (conector): 7Vdc – 12Vdc
- ◆ Voltaje de alimentación (límite): 6Vdc – 20Vdc
- ◆ Corriente por pin de E/S: 20mA
- ◆ Corriente en pin 3.3Vdc: 50mA

- ◆ Memoria Flash: 32KB, 0.5KB reservados para el programa de arranque
- ◆ SRAM (RAM estática): 2KB
- ◆ EEPROM (**ROM eléctricamente borrable programable**): 1KB
- ◆ Velocidad de reloj: 16MHZ

Para poder utilizar la tarjeta de desarrollo se debe instalar el compilador **ARDUINO 1.6.9** o alguna versión más actual, este software permite describir un algoritmo en lenguaje **Wiring**, y después el software de entorno **Processing**, compila dicho código para así programar el microcontrolador. Para programar el microcontrolador se deben seguir los siguientes pasos:

Paso 1

Descargar el software compilador de Arduino, debe instalarse según las instrucciones de los desarrolladores del software, del siguiente link:

<https://www.arduino.cc/en/Main/Software>

Paso 2

Conectar el Arduino Uno al compilador. Para poder programar el dispositivo, se debe seleccionar la tarjeta de desarrollo empleada, y el **puerco COM** (conocido como puerto serie o puerto serial es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente) en el que se conectó, para esto se debe dar clic en “Herramientas” y en “Puerto:” y seleccionar el puerco donde se conectó el dispositivo, como se muestra en la figura 2.7.4.2.1-2.

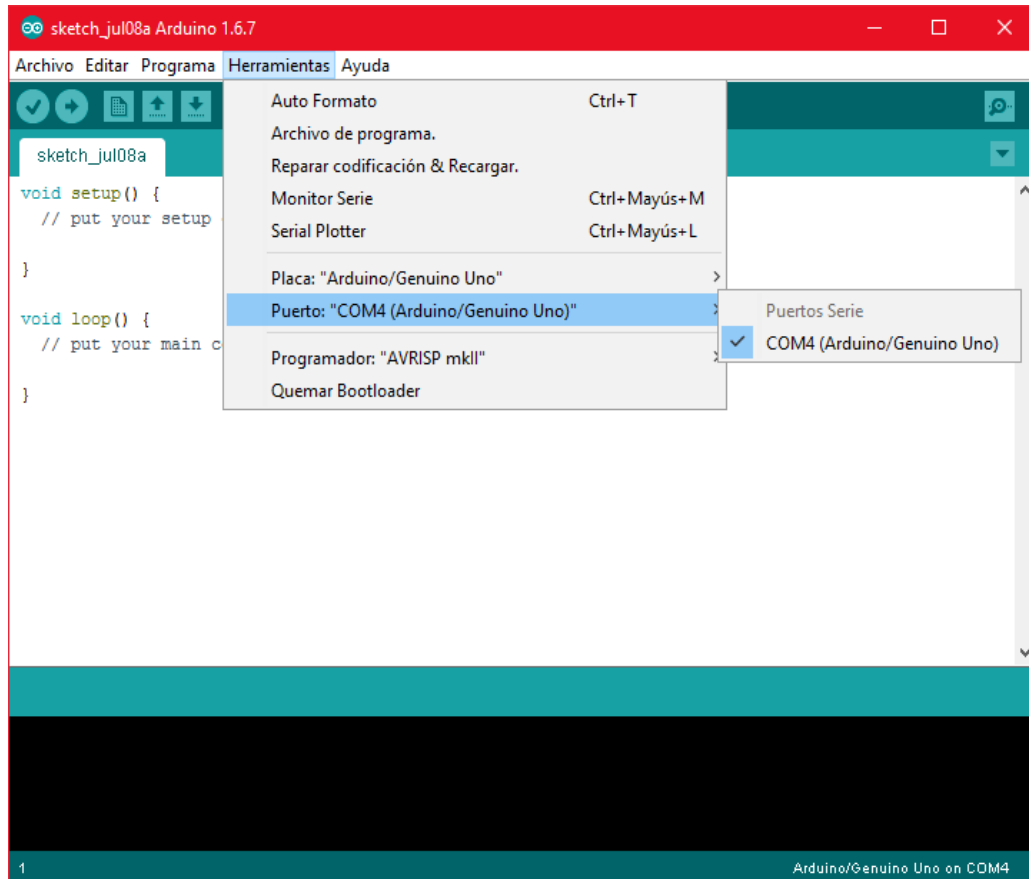


Figura 2.7.4.2.1-2 Selección puerto COM.

Paso 3

Describir el algoritmo a programar en la tarjeta de desarrollo. Al describir el algoritmo para programar la tarjeta de desarrollo, se utiliza lenguaje **Wiring**, se deben declarar las variables y el ciclo que será el programa del dispositivo, en otras palabras, se debe declarar los pines que se utilizarán, y si serán entradas o salidas en un segmento del código conocido como “void setup()”, seguidamente de las tareas que realizara el dispositivo, recordando que como todo microcontrolador, realizara estas tarea de forma cíclica y por tiempo indeterminado, a esto se le conoce como Loop, y se debe describir en un segmento del código conocido como “void Loop()”, en la figura 2.7.4.2.1-3 se muestra un ejemplo.

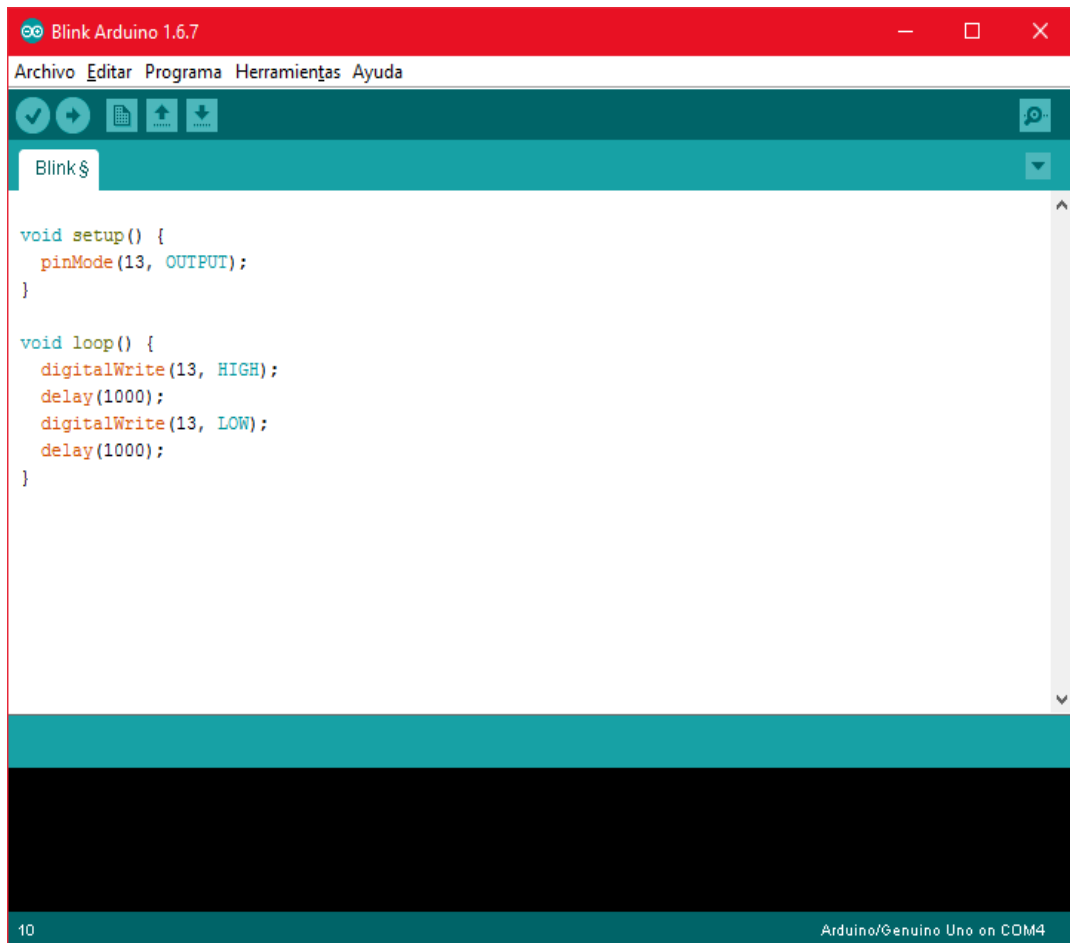


Figura 2.7.4.2.1-3 Descripción de código en entorno de Arduino.

Para este algoritmo, no es necesario alambrear ningún componente, pues la tarjeta de desarrollo tiene conectado un LED al pin 13, si se desea, se puede conectar en el pin 13 un LED de 5mm en serie con una resistencia de 330 Ohms conectando la terminal negativa a GND de la misma tarjeta de desarrollo.

Paso 4

Programar la tarjeta de desarrollo. La programación de la tarjeta de desarrollo es bastante simple, se debe dar clic en el botón “Subir” localizado en la parte superior izquierda de la ventana principal del software, figura 2.7.4.2-4, y el software compilara la



Figura 2.7.4.2-4 Botón de programación de la tarjeta Arduino.

descripción del algoritmo, mostrará una barra de progreso con el mensaje “compilando programa...” como se muestra en la figura 2.7.4.2.1-5.



Figura 2.7.4.2.1-5 Barra de progreso de programación.

Después de compilar el programa, automáticamente comenzara a grabar el programa en la tarjeta de desarrollo. Después de esperar unos momentos, finalizará y mostrara un mensaje como se muestra en la figura 2.7.4.2.1-6.

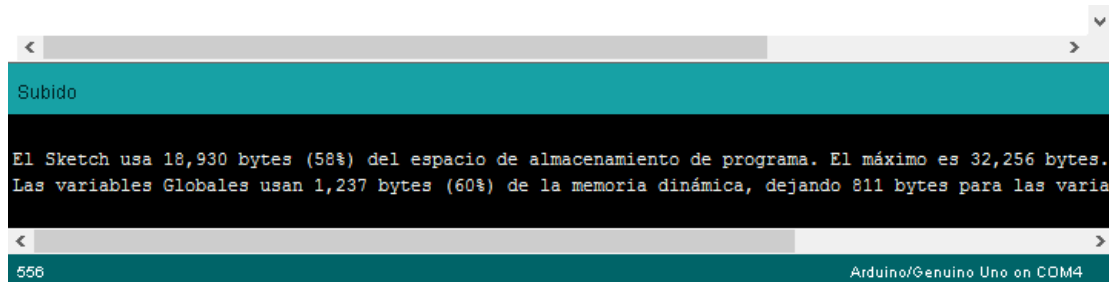


Figura 2.7.4.2.1-6 Programación realizada con éxito.

La tarjeta de desarrollo elegida para desempeñar como controlador del sistema de control para este proyecto, ha sido Arduino Uno, su facilidad de adquisición, su precio accesible, y gracias a que su entorno de compilación, llamado **Processing**, permite codificar fácilmente los algoritmos en lenguaje **Wiring**, permite al usuario enfocarse en el algoritmo en vez del lenguaje de programación.

2.7.5 Actuadores

El objetivo final de los sistemas de control consiste en iniciar, detener, regular o modificar alguna variable del proceso, según las entradas, para así obtener una salida deseada. En la actualidad existen diferentes dispositivos que transforman algún tipo de energía, a otro tipo de energía, para brindar funcionamiento a un sistema determinado, a estos dispositivos se les denomina actuadores.

Los actuadores se definen como dispositivos que brindan la posibilidad de transformar diferentes tipos de energía para generar algún funcionamiento dentro de un sistema determinado. Usualmente, los actuadores pueden generar una fuerza mecánica a partir de distintos tipos de energía, como puede ser eléctrica, neumática, o hidráulica. Los motores eléctricos, pistones neumáticos e hidráulicos, son ejemplo de actuadores, y según el tipo de energía con la que funcionan, tienen características como alto par de torsión o torque, alcanzar altas RPM, o en el caso de los hidráulicos, la capacidad de mover un gran peso, con una bomba hidráulica relativamente pequeña en comparación al peso que mueve el actuador hidráulico. A continuación se muestran en la tabla 2.7.5-1 las características más representativas de los actuadores según la energía que requieren para su funcionamiento.

Tabla 2.7.5-1 Comparación y tipos de actuadores según su funcionamiento.

	Neumáticos	Hidráulicos	Eléctricos
Energía	Aire a presión (5bar-10bar)	Aceite (50bar-100br)	Corriente eléctrica (AC o DC)
Opciones	Cilindros/pistones Compresor de tornillo Compresor de pistón	Cilindros/pistones Compresor de paletas Compresor de pistones axiales Compresor de pistón	Corriente continua Corriente alterna Motor de pasos Servomotor Solenoides
Ventajas	Rápidos Sencillos Robustos	Rápidos Alta relación peso-potencia Autolubricantes Alta capacidad de carga Estabilidad frente a cargas estáticas	Precisos Fiabiles Fácil control Instalación sencilla Silenciosos
Desventajas	Dificultad de control continuo Instalación especial (compresor, motor eléctrico, filtros) Ruidoso	Difícil mantenimiento Instalación especial (bomba hidráulica, motor eléctrico, filtros, purgado de aire) Frecuentes fugas	Potencia limitada por los conductores eléctricos Tiempo de operación

Costo

Para este proyecto, se utilizaron actuadores eléctricos, ya que elimina la necesidad de utilizar compresores neumáticos o bombas hidráulicas, además con referencia en el punto 2.1 Fisiología de las iguanas, se requiere modificar temperatura y humedad, por lo que los actuadores deberán poder modificar estos parámetros.

Calentador de inmersión

Este tipo de actuador de uso doméstico (figura 2.7.5-1), utiliza un arreglo de resistencias, para convertir la energía eléctrica en calor, y poder elevar la temperatura del líquido en donde se sumerge, y el volumen de líquido que es capaz de calentar, dependerá de la potencia eléctrica consumida, su funcionamiento es simple, y puede ser controlado con un interruptor y no requiere instalación de ningún otro dispositivo, por lo que fue idóneo para éste proyecto, la potencia promedio del actuador es de 400 Watts.



*Figura 2.7.5-1
Calentador resistivo de inmersión.*

Calentador de inducción forzada

Este actuador de uso doméstico (figura 2.7.5-2), utiliza un ventilador para forzar una masa de aire a través de un arreglo de resistencias, la masa de aire calentado, sale del aparato, para ceder calor en el entorno donde es expulsado, el volumen de aire que es capaz de calentar, será definido por el fabricante, y siempre será proporcional a la potencia eléctrica consumida. Algunos de estos aparatos integran un control de temperatura y protección de sobrecalentamiento en su sistema eléctrico, y no requieren de instalación ningún otro dispositivo, razón por lo que ha sido parte del desarrollo del proyecto presentado, la potencia promedio del actuador es cercana a 1000 Watts.



*Figura 2.7.5-2
Calentador resistivo de inducción forzada.*

Atomizador

El atomizador fue manufacturado en base a una bomba pequeña de agua de uso automotriz, y la boquilla de un atomizador para el cabello, dado que adquirir un atomizador de jardín o invernadero, además de ser de costo elevado como para adquirirlo, presenta el inconveniente de arrojar gotas de agua demasiado grandes que más bien, riegan la superficie en vez de incrementar la humedad con gotas muy finas, por ello se optó por utilizar una pequeña bomba de agua de 12Vdc conectada a una boquilla, obteniendo así mejores resultados y un control más fácil y simple que utilizar una electroválvula para poder utilizar el suministro de agua de una toma residencial por ejemplo.



Figura 2.7.5-3 Bomba de agua 12Volts.



Figura 2.7.5-4 Atomizador para cabello.

2.8 Sistemas de protecciones eléctricas y etapa de potencia

Al referirnos a dispositivos eléctricos siempre existe dos formas de suministrar energía eléctrica, en corriente alterna o corriente directa, y sin importar el propósito del dispositivo, ya sea un motor trifásico industrial alimentado con 440Vac o una compuerta lógica alimentada con 5Vdc, siempre es posible que puedan formar parte del mismo sistema e interactuar de forma segura sin importar la diferencia de potencial eléctrico y el tipo de energía eléctrica que utilizan, para ello se recurre a métodos para poder interactuar AC y DC, así como proteger a los dispositivos ante la diferencia de potencial existentes entre ellos.

2.8.1 Protecciones eléctricas

Toda instalación eléctrica tiene que estar dotada de una serie de protecciones que la hagan segura, tanto desde el punto de vista de los conductores y los aparatos a ellos conectados, como de las personas que han de trabajar o hacer uso de ella. Las fallas típicas de un sistema eléctrico residencial o comercial son:

Sobrecarga

Funcionamiento de un equipo excediendo su capacidad nominal de corriente, o de un conductor que excede su capacidad de conducción de corriente, cuando tal funcionamiento persiste por suficiente tiempo, puede causar daños o sobrecalentamiento peligroso.

Cortocircuito

Es una falla en un aparato o línea eléctrica por el cual la corriente eléctrica pasa directamente del conductor fase hacia al neutro o tierra. El cortocircuito se produce normalmente por fallas en los aislamientos de los conductores.

Falla a tierra

Ocurre cuando un aparato eléctrico es dañado o sus partes eléctricas están húmedas y el flujo de corriente eléctrica sale de los conductores del circuito, la corriente eléctrica encuentra continuidad en la parte exterior del aparato o en un elemento conductor.

Los dispositivos de protección en una instalación eléctrica son los interruptores termomagnéticos, interruptores de falla a tierra, los fusibles o una combinación de ellos, y sus propósitos fundamentales son:

- ◆ Proteger los conductores y el equipo instalado contra efectos excesivos de temperatura.
- ◆ Proteger de una sobrecorriente (cualquier corriente eléctrica en exceso, la cual puede ser causada por una sobrecarga, un cortocircuito o una falla a tierra).

El cortocircuito es una falla en la que el conductor de fase se pone en contacto directo con otra fase o con el hilo neutro del sistema eléctrico. Este contacto origina que súbitamente la corriente crezca a 10 o más veces su valor nominal (en la teoría se dice que se incrementa al infinito). En la figura

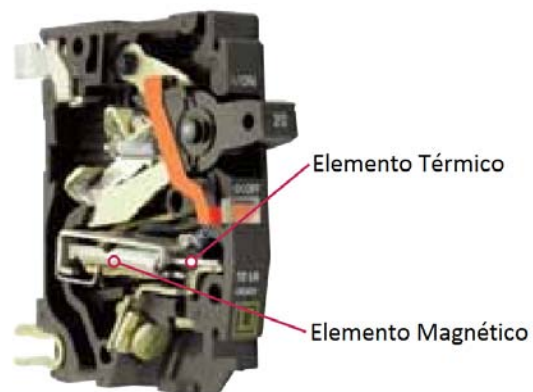


Figura 2.8.1-2 Interruptor termomagnético doméstico.

2.8.1-1 se esquematiza un cortocircuito. Para protegerse ante fallas de sobrecarga y cortocircuitos en los circuitos eléctricos, domésticos y comerciales, se utilizan principalmente los interruptores termomagnéticos (figura 2.8.1-2). En el interior del interruptor existe un elemento bimetálico que utiliza el efecto térmico de la sobrecarga para detectarla y mandar la apertura del interruptor para así proteger la instalación eléctrica y a sus usuarios.

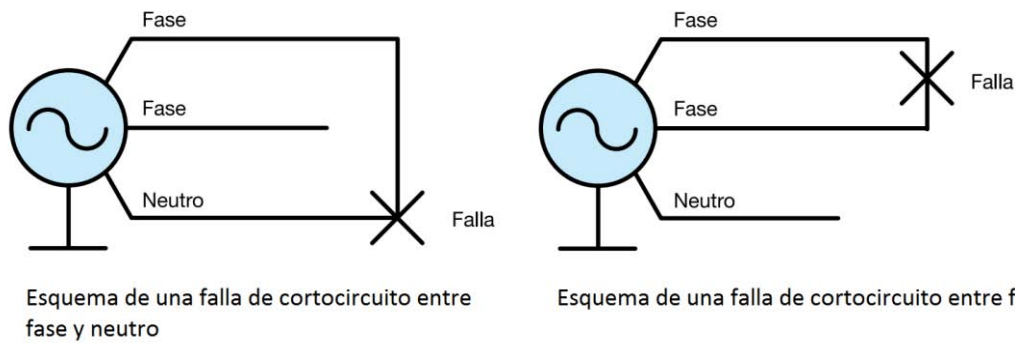


Figura 2.8.1-1 Diagrama corto circuito contra fase y contra tierra.

Existen otros dispositivos de protección, llamados fusibles, que por requerimiento de la compañía que suministra energía eléctrica, deben instalarse en todo sistema eléctrico doméstico, comercial e industrial.

Los fusibles o cortacircuitos, mostrados en la figura 2.8.1-3, son una sección de conductor más fino que los conductores normales, colocado en la entrada del circuito a proteger, para que al aumentar la corriente, debido a un cortocircuito, sea la parte que más se caliente, y por tanto la primera en fundirse. Una vez interrumpida la corriente, el resto del circuito ya no sufrirá daño alguno.



Figura 2.8.1-3 Fusibles domésticos.

En maquinarias robustas (bombas de agua a nivel residencial) o motores trifásicos (a nivel industrial) por ejemplo, la corriente es muy elevada para poder circular por un interruptor manual de uso doméstico, como solución a este problema, se utilizan dispositivos electromecánicos, conocidos como contactores, que puedan

soportar dicho transitorio y corrientes nominales bastante altas para un interruptor manual de uso doméstico.

Los contactores o relevadores o relés o releys, son componentes electromecánicos que tiene por objetivo establecer o interrumpir el paso de corriente, ya sea en el circuito de potencia o en el circuito de mando, están constituidos por una bobina, y una placa metálica que actúa como interruptor. La bobina es alimentada por una corriente de magnitud pequeña en relación de la corriente que conducirá la placa que actúa como interruptor, así se puede alimentar un dispositivo que requiera grandes cantidades de corriente para funcionar; Los contactores proporcionan la ventaja de poder utilizar múltiples niveles de tensión eléctrica en un mismo sistema, es decir, el dispositivo a alimentar puede funcionar con 127Vac y ser activado por un circuito de 5Vdc, el dispositivo intermediario que funcionara como interruptor a 127Vac será el contactor, en otras palabras un contactor es un interruptor de accionamiento eléctrico.

Según el fabricante, pueden ser activados por un rango de tensión en corriente alterna o directa. Un contactor y su diagrama se observa en la figura 2.8.1-4, donde en la terminal 1 y 2 se aprecie la bobina que generara el campo magnético para cambiar el interruptor de posición, es decir terminales 4 y 3 conduciendo, y las

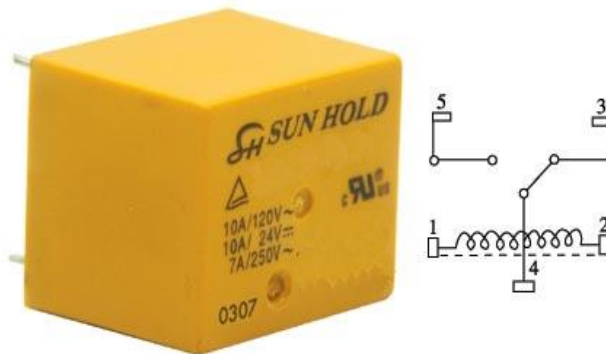


Figura 2.8.1-4 Contactor y diagrama de conexión.

terminales 4 y 5 en circuito abierto, a terminales 4 y 3 en circuito abierto y terminales 4 y 5 conduciendo. A esto se denomina interruptor Normalmente Abierto (su símbolo NA o en inglés NO), cuando el contactor está desenergizado y no conduce electricidad por el interruptor NA, sí conduce electricidad por el interruptor Normalmente Cerrado (su símbolo NC).

Este dispositivo ha sido elegido para este proyecto gracias a su capacidad de activar o desactivar los actuadores desde la tarjeta de desarrollo, al mismo tiempo, que los contactores aíslan circuitos eléctricos, es decir, el voltaje de la línea eléctrica de 127Vac no tiene una conexión eléctrica sólida (refiriendo un conductor eléctrico, como cable de cobre) con la tarjeta de desarrollo de 5Vdc, por lo que en caso de falla en el lado de 127Vac, la tarjeta de desarrollo no sufriría daño alguno. Según el fabricante, el contactor utilizado, puede conducir 10A a 120Vac y energizado a 12Vdc.

2.8.2 Etapa de potencia

La electrónica es conocida por la capacidad de involucrarse en sistemas complejos que desarrollan tareas y que muchas veces, realizan de manera más eficiente las tareas que máquinas mecánicas y con la ventaja de no sufrir descastes al no tener partes móviles, además de poder disminuir su tamaño, esto ha sucedido, por ejemplo, en los relojes, sistemas automotrices, electrodomésticos, por mencionar algunos; Estas máquinas han sido mejoradas por sistemas electrónicos, que reducen su tamaño, mejoran eficiencia, e incluso facilitan su uso gracias a interfaces de usuario más intuitivas y simples.

Electrónica

Es el estudio y aplicación del comportamiento de los electrones en diversos medios, como el vacío, los gases y los semiconductores, sometidos a la acción de campos eléctricos y magnéticos.

Dispositivo Eléctrico

Aparato que, para cumplir una tarea, utiliza energía eléctrica alterándola, ya sea por transformación, amplificación/reducción o interrupción

Para que los dispositivos electrónicos que funcionan con corriente directa, a niveles bajos de tensión y corriente, puedan interactuar con dispositivos eléctricos de corriente alterna, o incluso dispositivos mecánicos, hidráulicos o neumáticos, existen métodos que permiten esta interacción.

La **electrónica de potencia** se puede definir como la aplicación de la electrónica para el control y la conversión de la energía eléctrica, a niveles

significativos de tensión o corriente. Para estas aplicaciones se han desarrollado una serie de dispositivos semiconductores de potencia, todos los cuales derivan del diodo o el transistor. Entre estos se encuentran los siguientes:

Diodos de potencia

A diferencia de los diodos de baja potencia, estos se caracterizan por ser capaces de soportar una alta intensidad de corriente con una pequeña caída de tensión en estado de conducción. Su función es permitir el flujo de corriente en un solo sentido, de ánodo (A) a cátodo (K), al igual que un diodo rectificador, su símbolo y conexiones se observa en la figura 2.8.2-1.



Figura 2.8.2-1 Símbolo diodo de potencia.

Tiristor o Silicon Controlled Rectifier o por sus siglas SCR

Funciona al igual que un diodo rectificador, pero con la característica de poder controlar su conducción eléctrica, permitiendo circular la corriente en un solo sentido (de ánodo a cátodo). Mientras no se aplique ninguna tensión en la puerta (simbolizada con una letra G) del SCR no se inicia la conducción y en el instante en que se aplique dicha tensión, el tiristor comienza a conducir. Su símbolo y conexiones se observan en la figura 2.8.2-2.

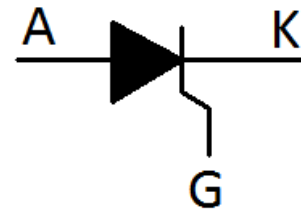


Figura 2.8.2-2 Símbolo SCR.

Triodo para corriente alterna o TRIAC

Se puede considerar como dos tiristores en paralelo, pero uno de estos conectado a la inversa. Al igual que el tiristor, el bloqueo o conducción se realiza por la aplicación de corriente en la puerta (G). A diferencia del SCR, el TRIAC puede conducir corriente alterna, cada terminal se simboliza por una letra A, y un índice que sirve para diferenciar cada terminal, como se observa en la figura 2.8.2-3, pues puede conducir electricidad en ambos sentidos en sus terminales.

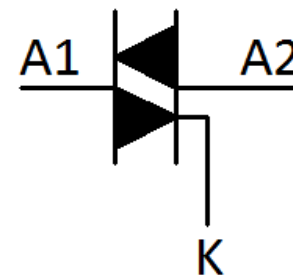


Figura 2.8.2-3 Símbolo TRIAC.

Transistores de juntura bipolar de potencia (TBJ)

Es un dispositivo electrónico de estado sólido consistente en dos uniones PNP o NPN (de materiales semiconductores generalmente de germanio o silicio y de polaridad Positiva o Negativa) que están muy cercanas entre sí, permite controlar el paso de la corriente a través de sus terminales. La denominación de bipolar se debe a que la conducción tiene lugar gracias al desplazamiento de portadores de dos polaridades (huecos positivos y electrones negativos), en otras palabras, al existir corriente eléctrica en la terminal Base (simbolizada con una letra B), se llenan los huecos entre las uniones NPN o PNP (según el tipo de transistor)

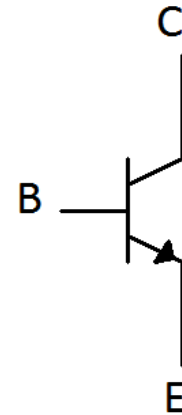


Figura 2.8.2-4
Símbolo TBJ.

permitiendo la conducción de la terminal Colector (simbolizada con la letra C) hacia la terminal Emisor (simbolizada con la letra E) en el caso de la juntura NPN simbolizado en la figura 2.8.2-4; En el caso de la juntura PNP la corriente será permitida del Emisor al Colector.

Los TBJ han sido seleccionados para conducir la corriente para cambiar el estado (abierto o cerrado) de los contactores en la etapa de potencia y aislamiento del proyecto propuesto, pues los contactores requieren cierta corriente para generar el campo magnético para ser activados, esta corriente no debe ser proporcionada por la tarjeta de desarrollo, para eso se utilizan los TBJ, estos proporcionaran la corriente de una fuente eléctrica distinta a la tarjeta de desarrollo, eliminando así el riesgo de dañar la tarjeta de desarrollo.

MOSFET

El Metal-oxide-semiconductor Field-effect transistor o MOSFET, es un dispositivo de cuatro terminales llamados Fuente (S, Source), Drenador (D, Drain), Puerta (G, Gate) y Sustrato (B, Body). Sin embargo, el Sustrato generalmente está conectado internamente al terminal de Fuente y por este motivo se pueden encontrar

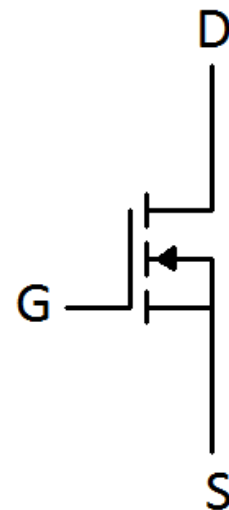


Figura 2.8.2-5
Símbolo MOSFET.

dispositivos MOSFET de tres terminales. En los transistores bipolares la corriente que circula por el Colector es controlada por la corriente que circula por la Base. Sin embargo en el caso de los transistores FET, la corriente de salida es controlada por una tensión de entrada (un campo eléctrico). En este caso no existe corriente de entrada. Tanto en el MOSFET de canal N o el de canal P, cuando no se aplica tensión en la Puerta no hay flujo de corriente entre el Drenador y la Fuente. Para que circule corriente en un MOSFET de canal N una tensión positiva se debe aplicar en la puerta. Así los electrones del canal N de la Fuente y el Drenador son atraídos a la Puerta y pasan por el canal P entre ellos. El movimiento de estos electrones, crea las condiciones para que aparezca un puente para los electrones entre el Drenaje y la Fuente. La amplitud o anchura de este puente (y la cantidad de corriente) depende o es controlada por la tensión aplicada a la Puerta. En la figura 2.8.2-5 se observa el símbolo de un MOSFET canal N.

III. Desarrollo Experimental De La Propuesta

Este proyecto fue pensado para facilitar los cuidados de reptiles en ambientes con temperatura y con humedad relativa inferior al hábitat natural de los reptiles; sin dejar de lado la seguridad para los usuarios, tanto personas como reptiles, por lo que durante el desarrollo, se experimentó y analizaron diferentes opciones sobre el algoritmo de funcionamiento, métodos de aislamiento eléctrico, actuadores, e incluso calibre de conductores. A continuación se describen los puntos importantes del desarrollo del proyecto así como una breve descripción de la realización de cada punto.

3.1 Algoritmo

Un algoritmo se puede definir como una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problemas. Para este proyecto se pensó y se perfecciono el algoritmo hasta donde fue posible, con las herramientas disponibles y procurando hacer eficiente el sistema.

Para comenzar la descripción, se simplificara el algoritmo, se observa en la figura 3.1-1. Las variables serán las entradas del sistema, recordando que deberán alcanzar niveles adecuados para el reptil en cuestión, estas variables serán censadas por sensores y posteriormente comparadas en la etapa de control.

La etapa de control comparará las variables censadas con la referencia establecida que será en niveles permisibles para el bienestar del reptil, y dependiendo de

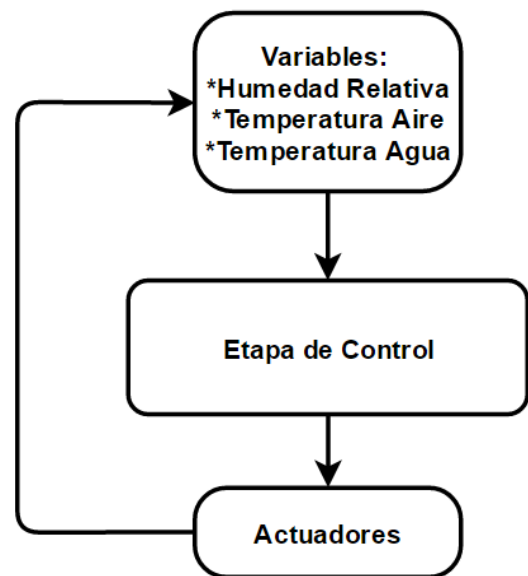


Figura 3.1-1 Esquema del algoritmo.

los resultados de la comparación, será el estado de los actuadores. Los actuadores incrementaran la temperatura o humedad, dependiendo si es requerido o no, por consecuencia se modificaran las variables y comenzara de nuevo el ciclo del algoritmo.

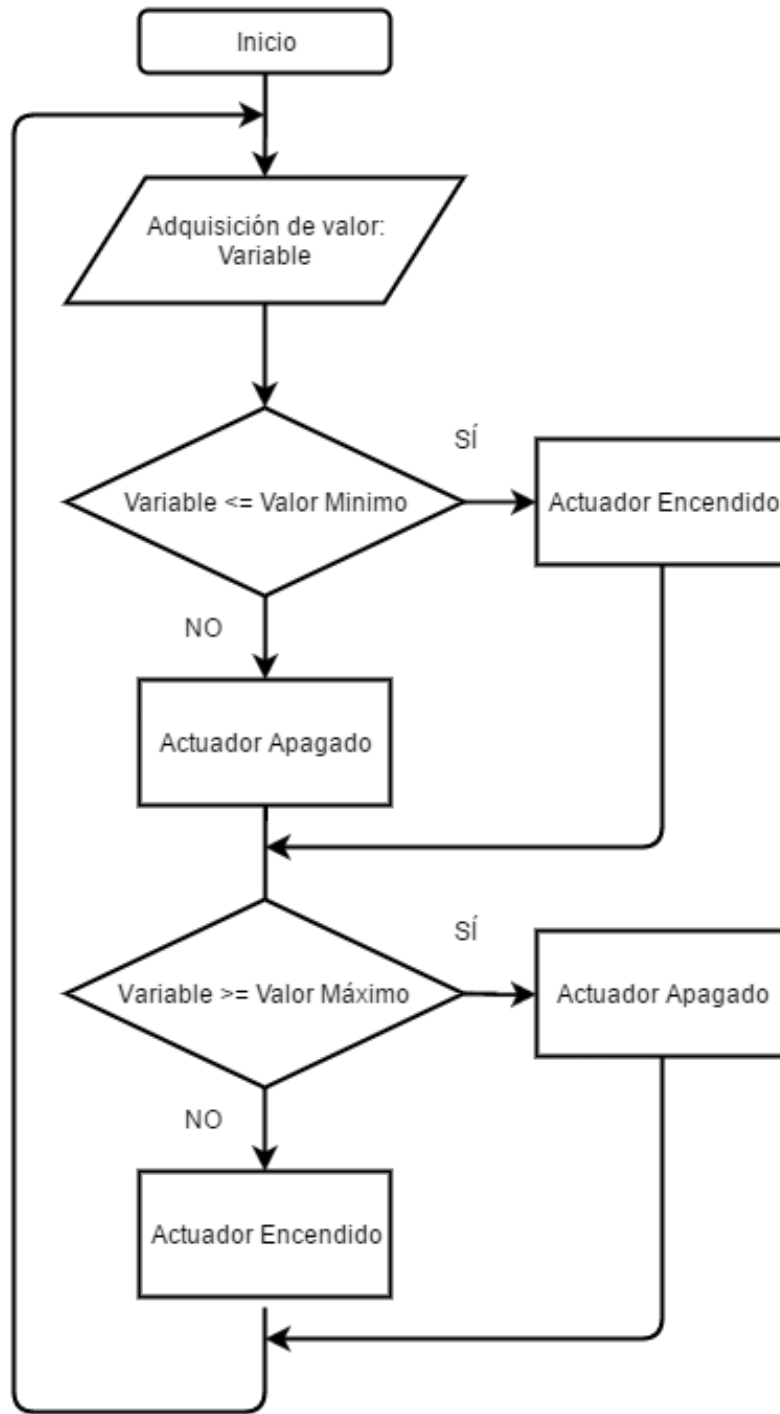


Figura3.1-2 Algoritmo desarrollado para el proyecto.

En la figura 3.1-2 se observa que el sistema está retroalimentado, y ya que los actuadores se condicionan en control todo o nada (encendido o apagado), se debe considerar un umbral de trabajo, un valor mínimo que será la referencia para encender los actuadores, y un valor máximo que será la referencia para apagar los actuadores. Los valores censados podrán estar entre ambos valores y los actuadores podrán estar encendidos o apagados según el diagrama de la figura 3.1-2, que representa el algoritmo que se programó en la tarjeta de desarrollo.

De forma genérica, *Variable* representa la humedad relativa, la temperatura del agua y la temperatura del aire, *Valor Mínimo* representa el valor inferior del umbral de trabajo, es decir, cuando el actuador deba activarse, *Valor Máximo* representa el valor superior del umbral de trabajo, es decir, cuando el actuador deba desactivarse.

Este algoritmo permite activar cada actuador cuando los sensores registren un valor igual o menor al *Valor Mínimo*, encender el actuador correspondiente y apagarlo hasta alcanzar el *Valor Máximo*. Una vez que se ha alcanzado el umbral de trabajo superior, el proceso sucede de manera contraria, si los sensores registran valores inferiores al *Valor Máximo*, el actuador correspondiente no se activara hasta ser igual o menor al *Valor Mínimo*.

3.1.1 Descripción del algoritmo en el compilador de la tarjeta de desarrollo

Para programar el algoritmo en la tarjeta de desarrollo, se utilizó el software **ARDUINO 1.6.9**, aprovechando que este software facilita la programación de un algoritmo utilizando lenguaje **Wiring** y entorno **Processing** y se pudo dedicar más tiempo a continuar con el desarrollo de otras partes del proyecto, como la etapa de potencia y protección o la comunicación con interfaz Android.

Para describir el algoritmo se declararon todas las variables y los pines de la tarjeta de desarrollo se declararon arbitrariamente según el dispositivo al que irían conectados y de los sensores a utilizar.

Para censar la temperatura del agua se utilizó el sensor DS18B20 en encapsulado aprueba de agua (figura 3.1.1-1), y su programación se puede realizar mediante bibliotecas. Para censar la temperatura ambiente y la humedad relativa se utilizó el sensor DHT11 (figura 3.1.1-2), proporciona ambos datos en un solo pin, por lo que no se requirieron un pin para cada dato.



Figura 3.1.1-1 Sensor DS18B20 tipo sonda

Las salidas que activaran los actuadores, fueron cuatro pines digitales, el primero para el actuador que incrementara la temperatura del aire, el segundo para el actuador que incrementara la temperatura del agua, el tercero para el actuador que incrementara la humedad relativa, y un cuarto reservado para trabajo futuro, posibilitando agregar algún otro actuador sin modificar el cableado del proyecto.

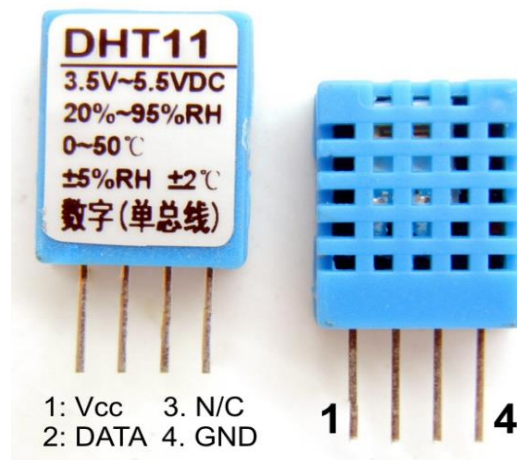


Figura 3.1.1-2 Sensor DHT11.

0 Una vez que en software se declararon las entradas y salidas, se describe la adquisición de datos, que en la figura 3.1.1-3, se puede observar que gracias a las

```
//Solicitando Temperatura Agua
sensors.requestTemperatures();
ta = sensors.getTempCByIndex(0); // se lee temp agua
hora = now.hour(); //se lee la hora
minu = now.minute(); // se lee los minutos
h = dht.readHumidity(); //se lee la humedad
t = dht.readTemperature(); // se lee la temperatura amb
```

Figura 3.1.1-3 Segmento de código encargado de la adquisición de valores.

bibliotecas, solo fue necesario una línea de código en **Wiring** por cada dato, los valores se asignaron a las variables mostradas.

Siguiendo el diagrama de flujo del algoritmo utilizado, se llega la toma de decisión sobre el estado de cada actuador, y para ello se describió el algoritmo, y con pocas modificaciones en él se pudo adecuar una réplica para cada variable censada. En la figura 3.1.1-4 se observa la descripción del algoritmo en lenguaje **Wiring**, y a la vez se imprime el resultado de dicha decisión en forma de texto con los caracteres “HOn” o “HOff” a través de la **USART** para posteriormente ser interpretado por la interfaz Android, dicha comunicación se describirá más adelante.

```
//Control Humedad

if(h<=hMin){ //Humedad menor a Humedad Minima
  if(h!=nan){
    digitalWrite(A3,1);
  }
  else{
    digitalWrite(A3,0);
  }
}
if(h>=hMax){ //Humedad mayor a Humedad Maxima
  digitalWrite(A3,0);
}
if(digitalRead(A3)== HIGH){
  Serial.println("HOn");
}
if(digitalRead(A3)== LOW){
  Serial.println("HOff");
}
```

Figura 3.1.1-4 Código de control Humedad Relativa.

Para este caso, la variable “h” representa el valor censado de la humedad relativa, y es comparada con la constante “hMin” y “hMax” cuyos valores son guardados en la memoria no volátil de la tarjeta de desarrollo y corresponden a los niveles mínimos y máximos de humedad relativa adecuados en el ambiente de las iguanas.

Si el resultado de la comparación, “h” es menor que la constante inferior del umbral, entonces se pondrá la salida en el “pin A3” en nivel alto, es decir, la salida estará energizada.

Se observa que la decisión siguiente es una desigualdad “**h!=nan**”, representando la variable “h”, la humedad relativa censada, es diferente a “**nan**”, estos caracteres provienen de la biblioteca correspondiente al sensor DHT11 y se asignan a la variable h cuando el valor censado no es un número, “**not a number**” y sus siglas “**nan**”, y esto representa una falla en el sensor, puede ser una desconexión o malfuncionamiento del mismo. Aprovechando esta característica, la desigualdad asegura que mientras no exista ninguna falla en el sensor, el actuador pueda encenderse, esto evitando que si en el último estado el actuador estaba encendido, al tomar nuevamente la decisión y el sensor falle, la toma de decisiones nunca podría evaluarse, pues no existirían valores numéricos para comparar, dejando así el actuador activado permanentemente.

Si el resultado de la comparación, “h” es mayor que la constante superior del umbral, entonces la salida se pondrá en nivel bajo, es decir desenergizada. Inmediatamente se toma otra decisión no tan importante para el algoritmo, pero si para conocer el estado del actuador, dicha decisión emplea ambos valores del umbral de trabajo, el superior “hMax” y el inferior “hMin”.

Al comenzar el ciclo del programa de la tarjeta de desarrollo, puede existir que la temperatura se encuentre entre el umbral de trabajo, lo que no efectuaría ninguna decisión, pues la variable censada no sería inferior a la variable mínima, y tampoco sería mayor a la variable máxima, lo que no requiere activar el actuador, pero el nivel lógico de la salida tampoco estaría definido, lo que no afectaría el algoritmo en la realización de sus tareas, pues no estaría energizada la salida, dado que la única condición para activar el actuador es proporcionado por la decisión, “h” es menor a “hMin”, y tampoco se activaría el actuador. Sin embargo, es necesario saber si el actuador estará o no encendido, para ello se efectúan las decisiones observadas al final de la figura 3.1.1-4.

Las decisiones mostradas en la figura 3.1.1-4, utilizan la función **digitalRead()** para leer el estado del pin seleccionado, en este caso el “pin A3”, una vez que se cumple alguna de las últimas dos condiciones, sea nivel alto, o nivel bajo en el pin de salida, se imprimen los caracteres “HOn” si el actuador esta energizado o “HOff” si el actuador esta desenergizado.

Estas decisiones se repiten de forma semejante para las otras dos variables restantes, temperatura del agua y temperatura del aire.

En la figura 3.1.1-5 se describe el control para la variable “ta”, a la que se le asignan el valor censado de por el sensor DS18B20, de igual manera se condiciona la salida en el “pin digital 9”, encargado de elevar la temperatura del agua; Esta condición se evaluara con la parte superior del umbral “taMax” y la parte inferior del umbral “taMin”. La primera decisión es efectuada por la comparación “ta” es menor o igual que “taMin”, es decir, la temperatura del agua censada es menor o igual que la temperatura mínima a la que deberá estar el agua, si el resultado es afirmativo, la salida será un nivel alto, activando el actuador.

```
//Control Temp Agua  
  
if(ta<=taMin){  
  if(ta!=-127){  
    if (nivel == LOW)  
    {  
      digitalWrite(9,1);  
    }  
    else{  
      digitalWrite(9,0);  
    }  
  }  
  else{  
    digitalWrite(9,0);  
  }  
}  
  
if(ta>=taMax){  
  digitalWrite(9,0);  
}  
  
if(digitalRead(9)== HIGH){  
  Serial.println("AOn");  
}  
  
if(digitalRead(9)== LOW){  
  Serial.println("AOff");  
}  
}
```

Figura 3.1.1-5 Código decisión del estado de la salida.

Nuevamente se asegura el funcionamiento con otra decisión, se comprara la variable “ta”, con una constante **-127**, que es proporcionada por la biblioteca del sensor, y se caracteriza por ser el valor retornado, o asignado a una variable, cuando el sensor no está conectado o por alguna razón, no interactúa con la tarjeta de desarrollo. Al realizar esta decisión con una desigualdad “ta!= -127” es decir, la temperatura censada

es diferente a **-127**, se asegura que el actuador nunca se encenderá cuando este falle o no tenga comunicación.

Si la variable “ta” es mayor que la temperatura máxima permisible, la salida tendrá un nivel bajo, desactivando así al actuador. En la figura 3.1.1-5 se describe en las últimas dos condiciones la lectura del nivel de la salida, si es alto, se envía por la **USART** los caracteres “AOn”, si el nivel es bajo, se envía por la **USART** los caracteres “AOff”.

Para el caso del valor censado de la temperatura del aire y asignado a la variable “t”, se describió el algoritmo mostrado en la figura 3.1.1-6, así se podrá determinar si la salida en el “pin digital 8”, tendrá nivel bajo o nivel alto.

El control del actuador nuevamente recurre a los niveles superior “tMax” e inferior “tMin”, se asigna el valor a la variable “t” proporcionado por el sensor DHT11 al censar la magnitud temperatura del aire, y posteriormente se compara con la constante mínima permitida o representada por “tMin”, si esta comparación, la variable t es menor que la temperatura mínima permitida, es afirmativa, entonces a la salida se le asigna un nivel alto el actuador será activado.

Nuevamente se asegura el algoritmo ante una falla en el sensor, con la desigualdad “t!=nan” o temperatura censada no es un número, y si no se cumple dicha desigualdad la salida tendrá un nivel bajo. Si la temperatura censada es mayor que el nivel máximo permitido, o “tMax”, a la salida se le asigna un nivel lógico.

Las últimas dos condiciones de la figura 3.1.1-6 describen la lectura del nivel lógico del “pin 8”, para poder enviar por la **USART** los caracteres correspondientes al nivel lógico. En el caso de nivel lógico alto, se enviarán los caracteres “BOn”, si el nivel lógico es bajo, se enviarán los caracteres “BOff”.

```
//Control Temp Ambiente
if (t<=tMin) {
if (t!=nan) {
digitalWrite(8,1);
}
else{
digitalWrite(8,0);
}
}
if (t>=tMax) {
digitalWrite(8,0);
}
if (digitalRead(8) == HIGH) {
Serial.println("BOn");
}
if (digitalRead(8) == LOW) {
Serial.println("BOff");
}
}
```

Figura 3.1.1-6 Control temperatura.

Todos los caracteres enviados por la **USART**, serán interpretados en la aplicación interfaz de usuario, para poder visualizar lo que acontece en la tarjeta de desarrollo. Así mismo, podrán ser visualizados en el monitor serial.

En la tabla 3.1.1-1 se describen los caracteres según el nivel lógico en cada pin de salida de la tarjeta de desarrollo.

Tabla 3.1.1-1 Pin de salida de los actuadores y caracteres correspondientes impresos según el nivel o estado asignado por el programa.

Actuador	Pin Arduino	Nivel Alto	Nivel Bajo
Aire	8	BOn	BOff
Agua	9	AOn	AOff
Sin asignar	10	-	-
Humedad	A3	HOn	HOff

3.1.2 Descripción del algoritmo de comunicación entre la tarjeta de desarrollo y la interfaz Android

Para realizar la interacción entre ambos dispositivos se debe comprender la forma en como interpretan los caracteres alfanuméricos. En el caso de la tarjeta de desarrollo

Arduino UNO, los caracteres alfanuméricos, son interpretados en su totalidad como números correspondientes al código **ASCII**, aunque se pueden declarar variables como **string** o **char** para interactuar con algunas funciones dentro del entorno de Arduino. Para la plataforma MIT App Inventor 2, existe la discriminación entre texto y números, aun si son declarados en una cadena (**string**) lo que en algunas circunstancias puede generar problema al codificar el algoritmo.

En la figura 3.1.2-1 se observa el segmento del código encargado de interpretar los datos recibidos en la **USART** desde la interfaz Android.

Con la función **Serial.peek()** se retorna o asigna el valor del primer byte disponible en la llegada de la comunicación serial. Utilizando esta función del entorno de Arduino, comienza la descripción de esta parte del código, evaluando los caracteres recibidos en la **USART**, dicho en lenguaje burdo, leyendo la información recibida, aunque estrictamente no estará leyendo la información, se someten los datos a comparación si el dato es iniciado por un salto de línea (**/n**) y continuado por la letra “T”, entonces realiza la acción descrita, en la figura 3.1.2-1, se observa la descripción de la prueba de los actuadores, que son encendidos por dos segundos, para poder verificar si existe algún mal funcionamiento en la etapa de potencia o cableado y facilite la resolución de un posible problema. La prueba de actuadores será realizada al

presionar el botón que muestra el texto “Prueba” en la interfaz de usuario, y mostrara en el display LCD texto con el estado de la tarjeta de desarrollo “Check actuadores”, así como el tiempo restante de los actuadores activados, terminada la prueba los actuadores se desactivaran, el display LCD ya no mostrara ningún mensaje y el algoritmo de control evaluara el estado en el que deberán estar.

En la figura 3.1.2-2 se observa la descripción del algoritmo encargado de escribir los valores del umbral de trabajo en localidades de memoria de la EEPROM de la tarjeta de desarrollo.

```

if(Serial.peek() == 'I'){
digitalWrite(8,0);
digitalWrite(9,0);
digitalWrite(10,0);
digitalWrite(A3,0);
digitalWrite(13,0);
Serial.println();
Serial.print("Prueba actuadores 2 segundos");
lcd.clear();
digitalWrite(backLight, HIGH);
lcd.setCursor(0,0);
lcd.print("Check actuadores");
lcd.setCursor(0,1);
lcd.print(" 2 segundos");
digitalWrite(8,1);
digitalWrite(9,1);
digitalWrite(A3,1);
digitalWrite(13,1);
delay(1500);
Serial.println();
Serial.print("Prueba actuadores 1 segundo");
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Check actuadores");
lcd.setCursor(0,1);
lcd.print(" 1 segundo");
delay(1000);
Serial.println();
Serial.print("Prueba actuadores terminada");
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Check actuadores");
lcd.setCursor(0,1);
lcd.print("Terminada");
lcd.clear();
delay(750);
digitalWrite(backLight, LOW);
digitalWrite(8,0);
digitalWrite(9,0);
digitalWrite(A3,0);
digitalWrite(13,0);
}

```

Figura 3.1.2-1 Código recepción de datos.

```
if (Serial.peek() == 'A') {  
  Serial.read();  
  taMax = Serial.parseInt();  
  EEPROM.write(0, taMax);  
  Serial.println();  
  Serial.print("Establecida Temp Agua Max: ");  
  Serial.println(taMax);  
}
```

Figura 3.1.2-2 Código recepción de datos a guardar en la EEPROM.

La descripción comienza nuevamente con la condición de la función **Serial.peek()**, para este caso, si los datos comienzan con un salto de línea y sucedido por la letra “A” la cual es recibida, entonces la función **Serial.read()** lee los datos que siguen a la letra “A”, y son asignados a la variable “taMax” mediante la función **Serial.parseInt()** que retorna o asigna el primer número entero del **buffer serial** y caracteres que no son enteros o negativos son omitidos, por lo que antes de ser transmitidos por la interfaz Android, se deben volver números enteros. Seguidamente mediante la función **EEPROM.write(dirección, valor)** se escribe, en la localidad de memoria asignada, el valor leído por la función **Serial.read()**, donde “dirección” corresponde a la localidad de memoria **EEPROM**, de 1024byte y “valor” será un número entero de **0 a 255bit**, para el microcontrolador AVR Atmega328. Finalmente se confirma la escritura imprimiendo el texto en la **USART** “Establecida Temp Agua Max:” y el valor asignado a la variable “taMax”.

Este proceso se repite para cada límite superior e inferior de las variables de temperatura del agua, temperatura del aire y humedad relativa.

Para poder visualizar información en la interfaz de usuario, es necesario enviar la información desde la tarjeta de desarrollo y es necesario poder diferenciar cada valor censado del resto, puesto que en ciertas circunstancias podrían existir valores numéricos idénticos, pero no corresponderían a la magnitud.

La solución encontrada fue incrementar cada valor antes de ser transmitido, para que pudiera ser separado en la interfaz de usuario por su rango de valor numérico, así se evita tener valores erróneos.

En la figura 3.2.2-3, se observa el código encargado de incrementar el valor de las variables al ser transmitidas, primeramente es llamada la variable, en el caso de las localidades de memoria no volátil, se utiliza la función **EEPROM.read(dirección)**, donde “dirección” es el número de la localidad, después arbitrariamente se le incrementa un valor, que corresponda a un rango numérico, finalmente es enviada por la **USART**. Para el caso de las variables almacenadas en la memoria dinámica, es decir datos que se crean y destruyen durante la ejecución del programa, primeramente se invoca la variable, es decir “t” o “ta” o

```
Serial.println();  
Serial.println(EEPROM.read(0)+10000);  
delay(100);  
Serial.println(EEPROM.read(1)+20000);  
delay(100);  
Serial.println(EEPROM.read(2)+30000);  
delay(100);  
Serial.println(EEPROM.read(3)+40000);  
delay(100);  
Serial.println(EEPROM.read(4)+50000);  
delay(100);  
Serial.println(EEPROM.read(5)+60000);  
while (Serial.available()>0) {  
  Serial.read();  
}  
Serial.println(sensors.getTempCByIndex(0)+1500);  
delay(100);  
Serial.println(t+2500);  
delay(100);  
Serial.println(h+3500);  
delay(100);  
Serial.println(taReg+4500);  
delay(100);  
Serial.println(taRegMax+5500);  
delay(100);  
Serial.println(tReg+6500);  
delay(100);  
Serial.println(tRegMax+7500);  
delay(100);  
Serial.println(hReg+8500);  
delay(100);  
Serial.println(hRegMax+9500);  
delay(100);  
Serial.println();  
delay(100);
```

Figura 3.1.2-3 Código envío de datos.

“h”, después se incrementa arbitrariamente, y finalmente es enviada por la **USART**. Es importante mencionar que después de cada envío de datos, se requiere un retardo, esto es para disminuir la posibilidad que en la interfaz, exista la omisión de ningún dato, por supuesto que al repetir el **Loop** o **ciclo del programa**, toda la información será enviada de nuevo.

En la tabla 3.1.2-1 se observan las características de los datos almacenados en la **EEPROM** y enviados por la **USART** a la interfaz de usuario.

Tabla 3.1.2-1 Características de los datos enviados desde la EEPROM a través de la USART.

Variable	Incremento	Localidad de Memoria	Carácter identificador	Preset	Variable código	Unidad
Temperatura del agua máxima	+10,000	0	A	30	taMax	°C
Temperatura del agua mínima	+20,000	1	a	25	taMin	°C
Temperatura del aire máxima	+30,000	2	B	30	tMax	°C
Temperatura del aire mínima	+40,000	3	b	25	tMin	°C
Humedad relativa máxima	+50,000	4	C	50	hMax	%RH
Humedad relativa mínima	+60,000	5	c	35	hMin	%RH

En la tabla 3.1.2-2 se observan las características de los datos almacenados en la memoria dinámica, es decir datos que se crean y destruyen durante la ejecución del programa, y posteriormente son enviados por la USART a la interfaz de usuario.

Tabla 3.1.2-2 Características de los datos enviados por la memoria dinámica.

Variable	Incremento	Variable código
Temperatura Agua Actual	+1,500	ta
Temperatura Aire Actual	+2,500	t
Humedad Relativa Actual	+3,500	h
Histórico temperatura del agua mínima	+4,500	taReg
Histórico temperatura del agua máxima	+5,500	taRegMax
Histórico temperatura del aire mínima	+6,500	tReg
Histórico temperatura del aire máxima	+7,500	tRegMax
Histórico humedad relativa mínima	+8,500	hReg
Histórico humedad relativa máxima	+9,500	hRegMax

3.1.3 Interfaz de usuario mediante el display LCD (16x2)

Para poder visualizar información sin requerir el teléfono móvil o una computadora, se incluyó una pequeña **subrutina** o **función** en el código del programa de la tarjeta de desarrollo llamada arbitrariamente **Display()**, ejecutada por el nivel alto proveniente de un push button

```
void Display () {  
  digitalWrite(backLight, HIGH);  
  lcd.clear();  
  lcd.setCursor(0,0);  
  Wire.begin();  
  RTC.begin();  
  DateTime now = RTC.now();  
  
  ta= sensors.getTempCByIndex(0);  
  hora = now.hour();  
  minu = now.minute();  
  h = dht.readHumidity();  
  t= dht.readTemperature();  
  
  for(int i=300; i>=0; i=i-1)  
  {  
    lcd.setCursor(0,0);  
    lcd.print("Hora:");  
    lcd.setCursor(0,1);  
    lcd.print(hora);  
    lcd.setCursor(2,1);  
    lcd.print(":");  
    lcd.setCursor(3,1);  
    lcd.print(minu);  
    lcd.setCursor(5,1);  
    lcd.print("  ");  
  }  
  
  lcd.clear();  
  for(int i=350; i>=0; i=i-1)  
  {  
    lcd.setCursor(0,0);  
    lcd.print("Temp Aire");  
    lcd.setCursor(0,1);  
    lcd.print(t);  
    lcd.setCursor(6,1);  
    lcd.print("C");  
  }  
}
```

Figura 3.1.3-2 Subrutina Display().

```
pin2=digitalRead(2);  
  if (pin2== HIGH)  
  {  
    Display();  
  }
```

Figura 3.1.3-1 Invocación de la subrutina Display().

(figura 3.1.3-1),

y el display LCD (**Liquid Crystal Display**) mostrará información, pero no podrá cambiar la configuración del programa, también podrá visualizarse alarmas como pérdida de comunicación de un sensor o nivel de agua inadecuado.

En la figura 3.1.3-2 se describe la función **Display()**, que únicamente será ejecutada cuando exista nivel alto en el push button, y comienza con encender la luz de fondo del display LCD, después invoca las bibliotecas de los sensores DS1B20, DHT11 y del reloj de tiempo real, posteriormente adquiere los valores las variables globales, temperatura del agua, temperatura del aire, humedad relativa, hora y minutos.

Dado que ciertas funciones del entorno de Arduino no se pueden ejecutar dentro de otras funciones, como es el caso de la función **delay** o **retardo**, para ello se utilizan otros métodos para realizar el **retardo** entre el flujo del programa, para este caso, se utilizó el **ciclo for**, su sintaxis

requiere establecer tres campos, **variable** y **asignación de su valor inicial** (por ejemplo: `int i=300`), **condición** (por ejemplo: `i>=0`) e **incremento** o **decremento** (en este caso un decremento `i=i-1`), este ciclo ejecutara el código contenido en él, hasta que se cumpla la condición (es decir `i>=0`). El contenido del ciclo **for**, se adecúa a simular

```
lcd.clear();
for(int i=3; i>=0; i=i-1)
{
    lcd.setCursor(0,0);
    lcd.print("Temp Agua");
    lcd.setCursor(0,1);
    lcd.print(ta);
    lcd.setCursor(6,1);
    lcd.print("C");
    sensors.requestTemperatures();
    ta= sensors.getTempCByIndex(0);
    lcd.setCursor(0,1);
    lcd.print(ta);
}

lcd.clear();
for(int i=350; i>=0; i=i-1)
{
    lcd.setCursor(0,0);
    lcd.print("Nivel de Agua");
    lcd.setCursor(0,1);
    lcd.print(textnivel);
}

lcd.clear();
for(int i=400; i>=0; i=i-1)
{
    lcd.setCursor(0,0);
    lcd.print("Temp Agua MinMax");
    lcd.setCursor(0,1);
    lcd.print(taReg);
    lcd.setCursor(6,1);
    lcd.print("C");
    lcd.setCursor(8,1);
    lcd.print(taRegMax);
    lcd.setCursor(14,1);
    lcd.print("C");
}

lcd.clear();
digitalWrite(backLight, LOW);
}
```

Figura 3.1.3-3 Continuación subrutina Display().

números y texto estático en el display LCD, para ello se utilizan bibliotecas del display LCD y se invocan a las variables a las que se les asignaron valores registrados por los sensores.

En la figura 3.1.3-2 se puede observar la invocación de variables, como “t” que corresponde al valor actual de la temperatura del aire, así también, se observa texto, como es “Hora” o “Temp Aire” y sus unidades “C”, notese la omisión del signo grados, pues el display LCD presenta problemas al mostrarlo, llenando un segmento de color negro, debido a eso se ha omitido dicho signo.

En la figura 3.1.3-3 (que es la continuación de la figura 3.1.3-2) se observa el ciclo **for** para mostrar la temperatura del agua, se observa que la constante” `i=3`” (y no “300” como en el ejemplo anterior) esto es debido a que el sensor DS18B20 demora varios instantes perceptibles para el ser humano, en censar la temperatura, y mediante pruebas se determinó que

con tres ciclos es suficiente tiempo para poder leer y comprender la información mostrada en el display.

Además de ser repetitivo el código para poder mostrar los valores actuales en el display, también se muestra el nivel del agua, pudiendo ser inadecuado o suficiente, también se muestra el registro histórico de los niveles máximos y mínimos censados. Esta función finaliza con limpiar los datos mostrados en el display LCD y apagando la iluminación del mismo, como cualquier subrutina regresa, mediante el **puntero de pila**, a la parte del programa donde fue invocada, para así continuar con el ciclo del programa principal.

3.2 Diagramas esquemáticos y conexiones

Para ejemplificar como los diversos dispositivos interactúan entre ellos, se puede representar su conexión, para ello se utilizan diagramas que simbolizan la conexión física que se usó durante el desarrollo de este proyecto, en la figura 3.2-1 se observa a grandes rasgos la conexión de los diferentes dispositivos, su interacción entre ellos y los parámetros en los que tienen efecto.

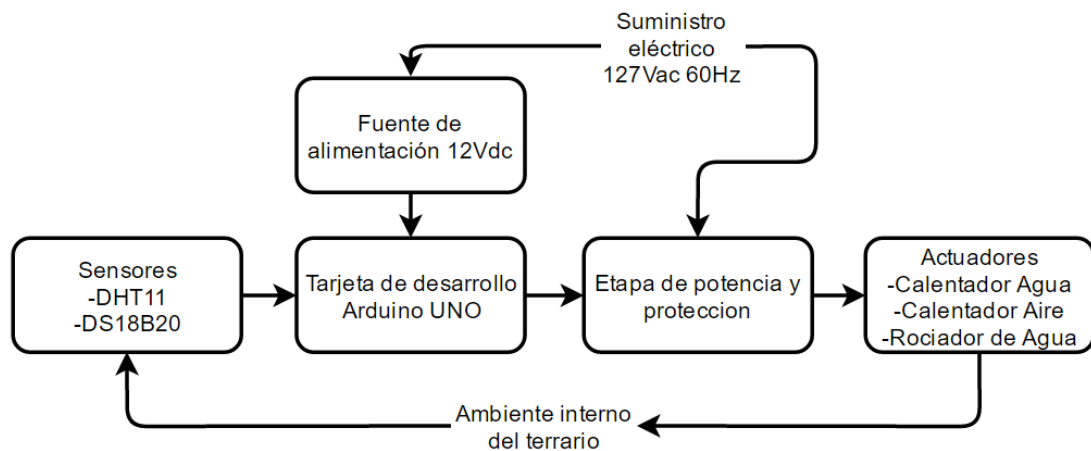


Figura 3.2-1 Diagrama de bloques de la conexión de la etapa de control, sensores y actuadores.

3.2.1 Diagramas esquemáticos

Los diagramas esquemáticos presentados a continuación, indican la conexión realizada entre los dispositivos eléctricos y electrónicos para esta propuesta, recordando que la mayoría de estas conexiones fueron de manera arbitraria, a excepción de aquellos dispositivos que requieren interactuar con algún periférico del microcontrolador y necesitan ser conectado a un pin en específico.

Para la conexión de la tarjeta de desarrollo a la fuente de alimentación, se utilizó una fuente lineal o regulada de 127Vac a 12Vdc a 500mA y mediante un conector macho de 2.1mm, como se muestra en la figura 3.2.1-1, se conectó a la entrada hembra de 2.1mm que está diseñada para voltajes de 7Vdc a 12Vdc, la tarjeta de desarrollo está equipada con dos reguladores de voltaje uno a 3.3Vdc y otro a 5Vdc.

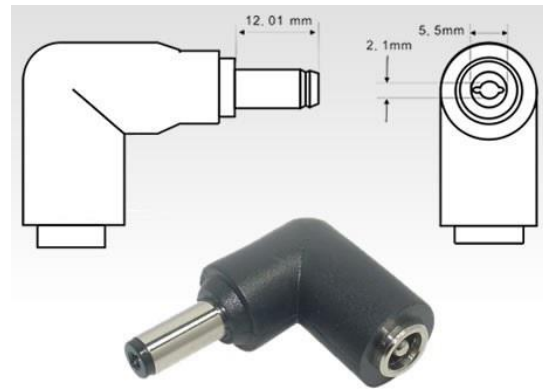


Figura 3.2.1-1 Plug macho 2.1mm.

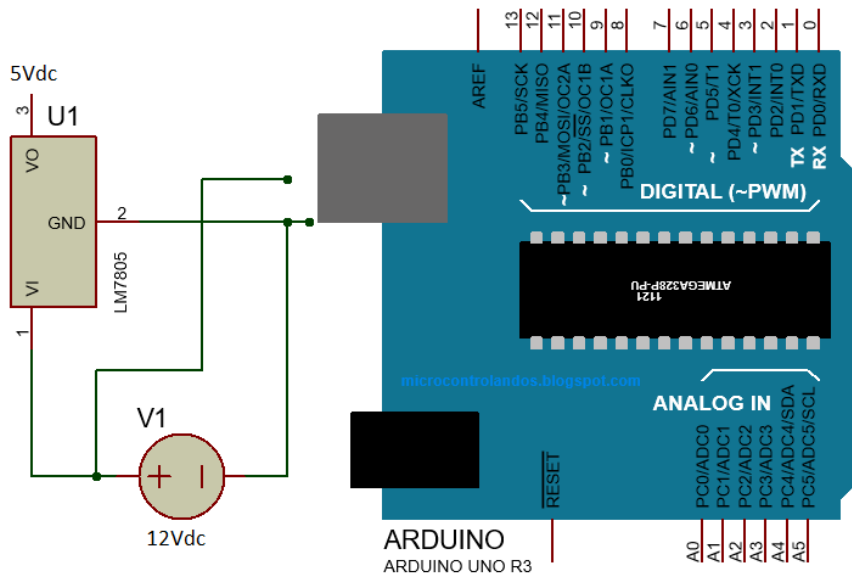


Figura 3.2.1-2 Alimentación Arduino UNO.

El diagrama de la figura 3.2.1-2 indica la conexión realizada entre la placa de desarrollo y la fuente de alimentación de 12Vdc a 500mA, y se toma una derivación al regulador de voltaje LM7805, dicho regulador se agregó al proyecto, para suministrar energía a los dispositivos adicionales, recordando que el fabricante no recomienda agregar mucha carga a las salidas de voltaje y periféricos de la tarjeta de desarrollo.

Para la conexión del display LCD que permite visualizar información en tiempo real, se eligió un display LCD caracteres, y su conexión se realizó siguiendo el pinout de la librería para dicho dispositivo. En la figura 3.2.1-3 se aprecia cómo se utiliza el regulador LM7805 como fuente de alimentación a 5Vdc, esto evitando que se haga pasar corriente a través de la tarjeta de desarrollo.

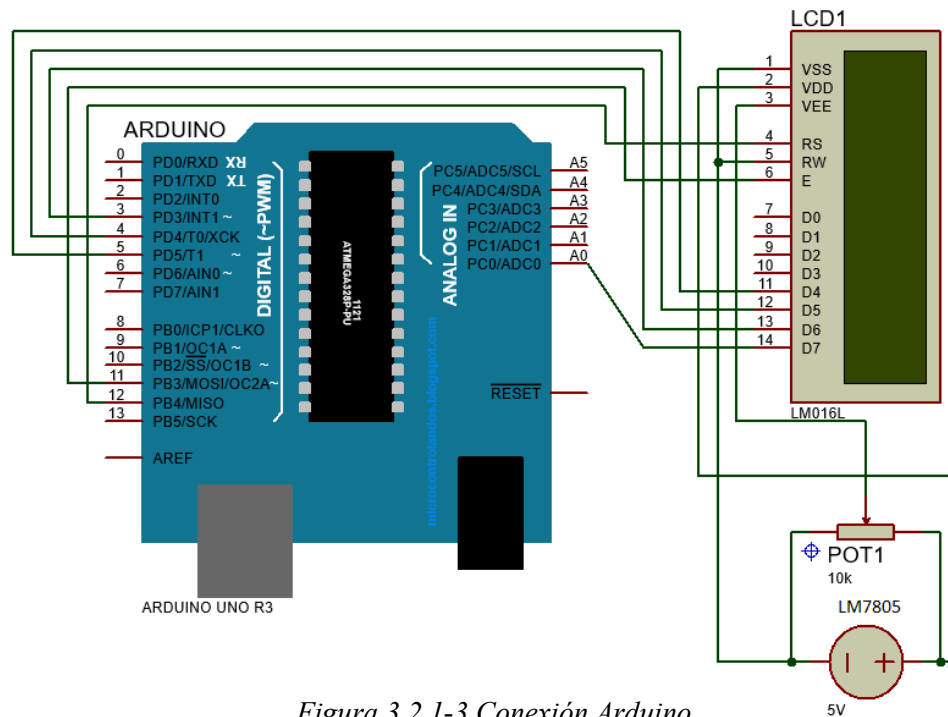


Figura 3.2.1-3 Conexión Arduino UNO y display LCD.

La comunicación con la interfaz de usuario es realizada vía Bluetooth, para ello se requiere la conexión de un dispositivo Bluetooth, recordando que se deben Cross-Conectar las terminales Tx y Tr de los dispositivos, es decir conectar la terminal de transmisión (Tx) de la USART de la tarjeta de desarrollo con la terminal de recepción (Rx) del módulo Bluetooth, y conectar la terminal de recepción (Rx) de la USART de

la tarjeta de desarrollo con la terminal de transmisión (Tx) del módulo Bluetooth. En la figura 3.2.1-4 se esquematiza esta conexión.

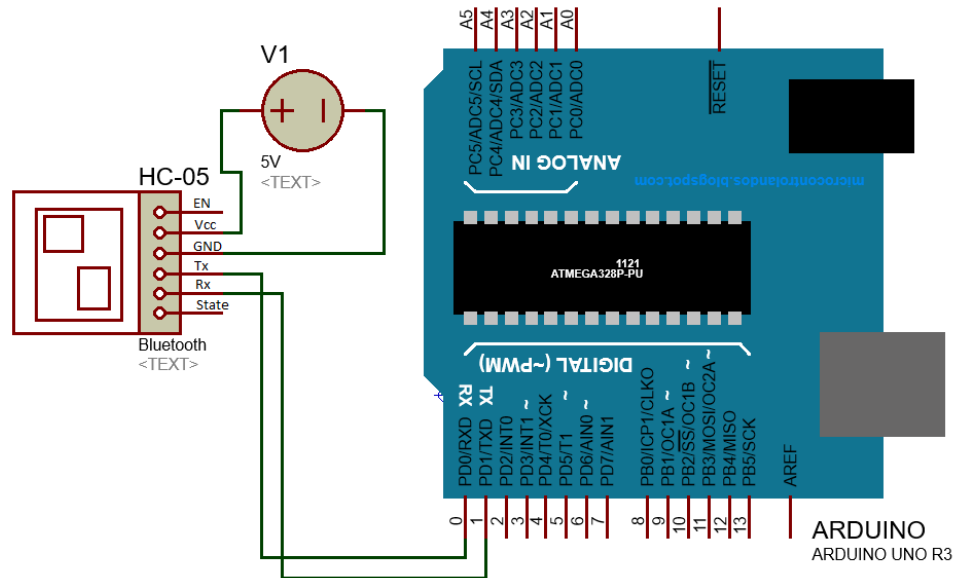


Figura 3.2.1-4 Conexión Arduino UNO y módulo HC-05.

Para poder diferenciar entre el día y la noche, lo que podrá modificar la activación de los actuadores durante el día, se recurrió al uso de un módulo reloj de tiempo real DS1307, denominado así pues su unidad es el segundo y no submúltiplos como el caso de los osciladores, dispositivo que mediante comunicación **I2C**, puede proporcionar la hora y fecha grabadas previamente en el dispositivo. Este dispositivo es una combinación de un contador y una **EEPROM**, además de contar con una batería que asegura continuar con el registro de fecha y hora aun cuando esta desenergizado. Su conexión se muestra en la figura 3.2.1-5.

En el diagrama de la figura 3.2.1-5 se esquematiza la conexión de las terminales del módulo, **SDA** y **SCL**, se conectan a los “pines A4 y A5” que corresponden a los pines de entrada analógica que son compartidos, en la tarjeta de desarrollo, por el **Analog Digital Converter** o por sus siglas **ADC**, y por el periférico de comunicación **I2C**.

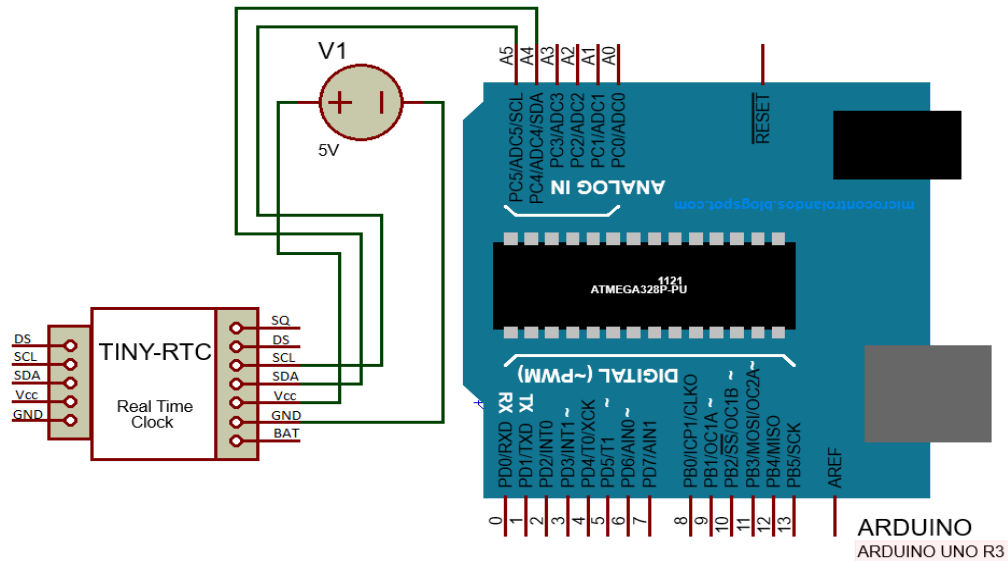


Figura 3.2.1-5 Conexión Arduino UNO y reloj de tiempo real.

La conexión con el sensor de temperatura del aire y humedad relativa, solo precisaba de un pin, y un arreglo con una resistencia para comunicarse con la tarjeta de desarrollo, como se muestra en el esquema de la figura 3.2.1-6.

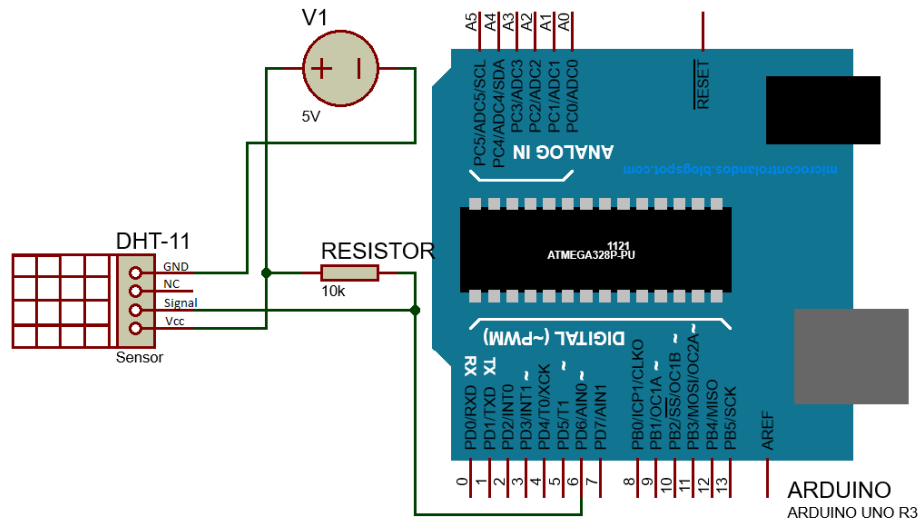


Figura 3.2.1-6 Conexión Arduino UNO y sensor DHT-11.

La conexión del sensor de temperatura sumergible, requiere una conexión semejante, y se puede observar en la figura 3.2.1-7.

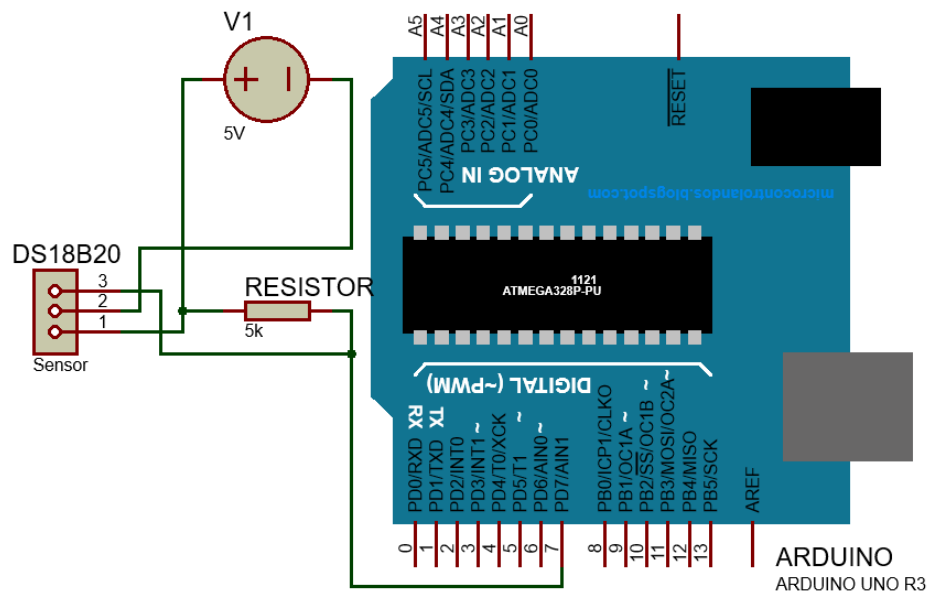


Figura 3.2.1-7 Conexión Arduino UNO y sensor DS18B20.

Para la conexión de la etapa de potencia y protección, se utilizaron transistores 2N2222, la Base fue conectada en serie con una resistencia de $1k\Omega$ a la salida de la tarjeta de desarrollo asignada para activar o desactivar los actuadores, el Colector se conectó al común del arreglo de alimentación, y el Emisor en serie con la bobina del contactor y al positivo de la fuente de alimentación, de esta forma se consigue aislar eléctricamente la tarjeta de desarrollo y los actuadores. En la figura 3.2.1-8 se describen las conexiones realizadas.

Para poder apreciar si las salidas están activadas o desactivadas se han conectado a una barra de LED, proporcionando así la posibilidad de percibir el estado de los transistores, de los contactores y por consecuencia de los actuadores. Los Bornes 0 al 3, en la figura 3.2.1-8, deben ser conectados en serie con los actuadores a 127Vac o al voltaje que requieran (siempre que sea inferior a 127Vac, pues el fabricante especifica el máximo a 127Vac), pues los relevadores funcionan en corriente alterna y corriente directa.

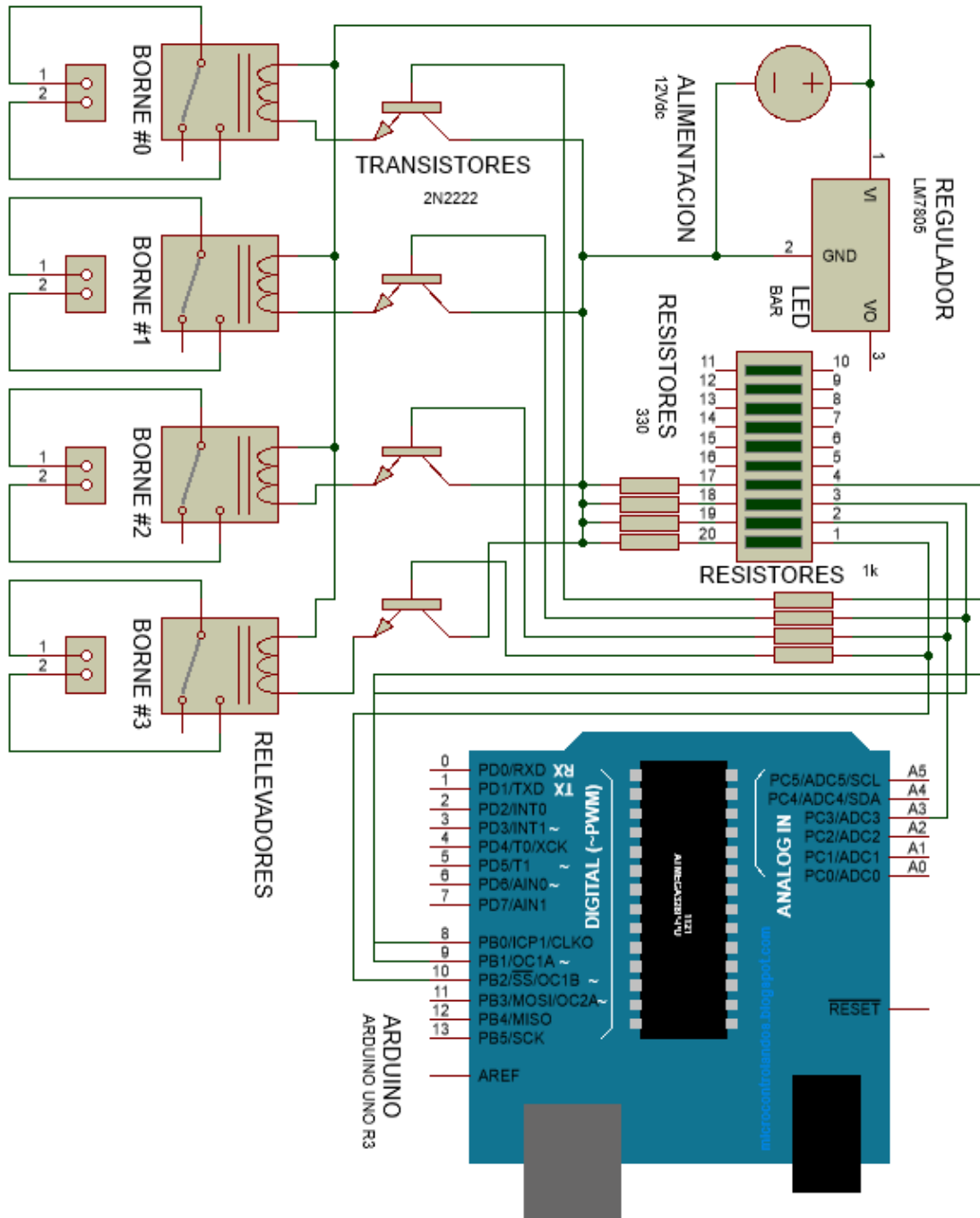


Figura 3.2.1-8 Conexión Arduino UNO, etapa de potencia y sensores.

En la tabla 3.2.1-1, se muestran la asignación de pines y conexión en los bornes de la PCB propuesta para este proyecto.

Tabla 3.2.1-1 Actuadores y sus salidas con respecto a Arduino y PCB.

Actuador	Tipo	Pin Arduino	Pin Borne
Aire	Resistivo/Motor	8	0
Agua	Resistivo	9	1
Humedad	Motor(Bomba)	A3	2
Sin Asignar	-	10	3

En la figura 3.2.1-9 se aprecia el circuito ensamblado en la protoboard, el acotamiento representa cada conexión:

- A. Reloj de tiempo real.
- B. Display LCD o de 16x2 caracteres.
- C. Barra de LED, resistencias, y push button para activar el display.
- D. Sensor DHT-11 temperatura y humedad relativa.
- E. Sensor DS18B20 temperatura sumergible.
- F. Arreglo transistor 2N2222 y contactor.
- G. Modulo Bluetooth HC-05.
- H. Tarjeta de desarrollo Arduino UNO
- I. Regulador de voltaje LM7805.

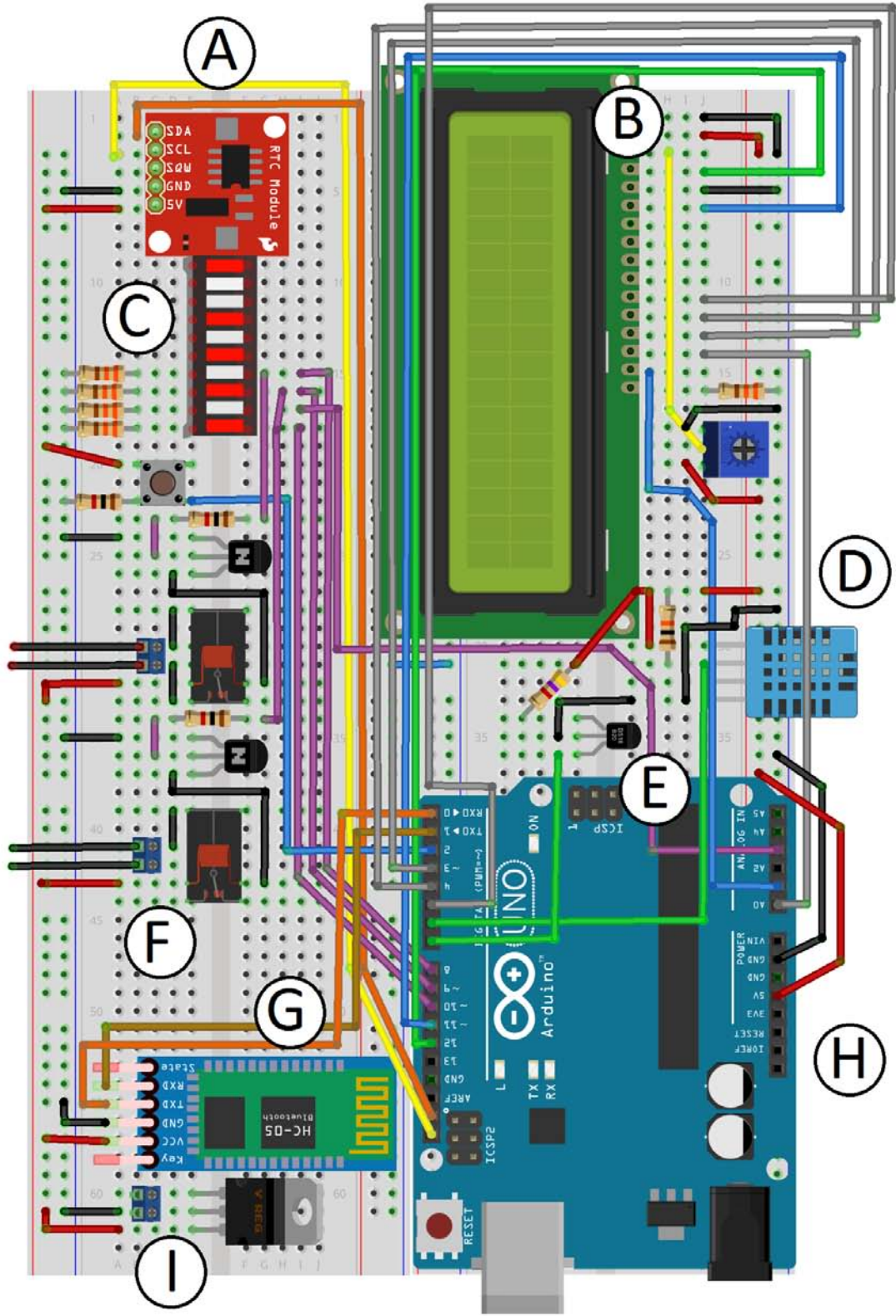


Figura 3.2.1-9 Arduino UNO, etapa de potencia y sensores en protoboard.

3.2.2 PCB propuesta

Para crear la **Printed Circuit Board** o **PCB** por sus siglas, es necesario trasladar los diagramas esquemáticos diagramas que ejemplifiquen las rutas en conductor de cobre y el emplazamiento donde deberá ser soldado o instalado mediante pin headers el dispositivo, esto con la finalidad de poder reemplazarlo en caso de falla. Para ello se propone el circuito basado en tres placas de doble cara conductora de 100mm x 100mm, donde cada placa tiene asignado contener ciertos circuitos dependiendo de sus características, dichos esquemas se realizaron en el software Ares Proteuss 8.

La primera placa contiene una interfaz de usuario simple, el display LCD caracteres, potenciómetro requerido para ajuste del contraste del display LCD, el módulo Bluetooth HC-05 y el push button para activar el display LCD y bornes de alimentación en 5Vdc, escudos de la UNAM y de la Facultad de Estudios Superiores Aragón.

La segunda placa contiene la etapa de potencia y aislamiento, que consta de contactores, transistores 2N222, borne de alimentación a 127Vac, bornes de alimentaciones en 12Vdc y 5Vdc, borne para tres actuadores más uno en reserva para posible trabajo futuro con etiquetas ACT#0, ACT#1, ACT#2 y ACT#3, borne de entradas digitales para activar los contactores, jumpers para selección de voltaje de salida por el borne ACT#2 de 127Vac o 12Vdc, LED indicadores del estado de las salidas.

La tercera placa contiene la etapa de control, la tarjeta de desarrollos Arduino UNO, bornes de alimentaciones en 12Vdc y 5Vdc, regulador de voltaje LM7805, bornes de conexión de sensores, Bluetooth, display LCD de 16x2, switch nivel del agua, bornes de salidas digitales, reloj de tiempo real, barra indicadora LED, jumper para programar la tarjeta de desarrollo.

La primera placa lado superior o placa 1 superior se muestra en la figura 3.2.2-1, en pistas de cobre y superposición de los componentes, recordando que sus dimensiones son de **100mm X 100mm** y están ajustadas para **hoja tamaño carta** y deben ser impresas en **espejo** para ser replicadas.

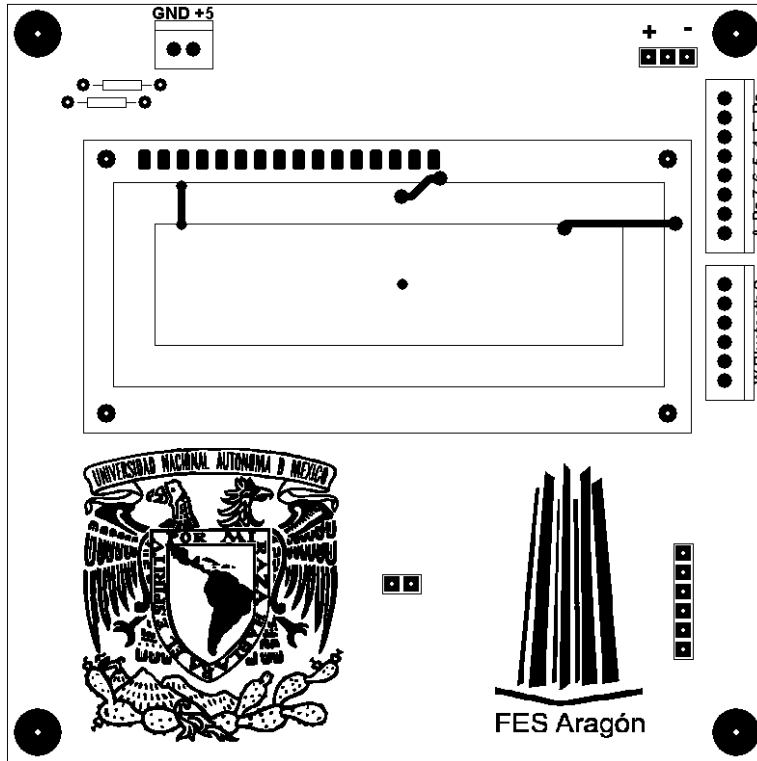


Figura 3.2.2-1 PCB #1 superior cobre y emplazamiento.

En la figura 3.2.2-1, se observa que el conductor de cobre esta superpuesto con las líneas que describen el contorno de los dispositivos en la placa 1 superior, para poder reproducir dicha placa, a continuación se muestra las placas sin superposición de dispositivos y conductor de cobre.

En la figura 3.2.2-2 se observa la placa número 1 superior solamente con conductores de cobre y los escudos de la UNAM y la Facultad de Estudios Superiores Aragón que estarán en cobre.

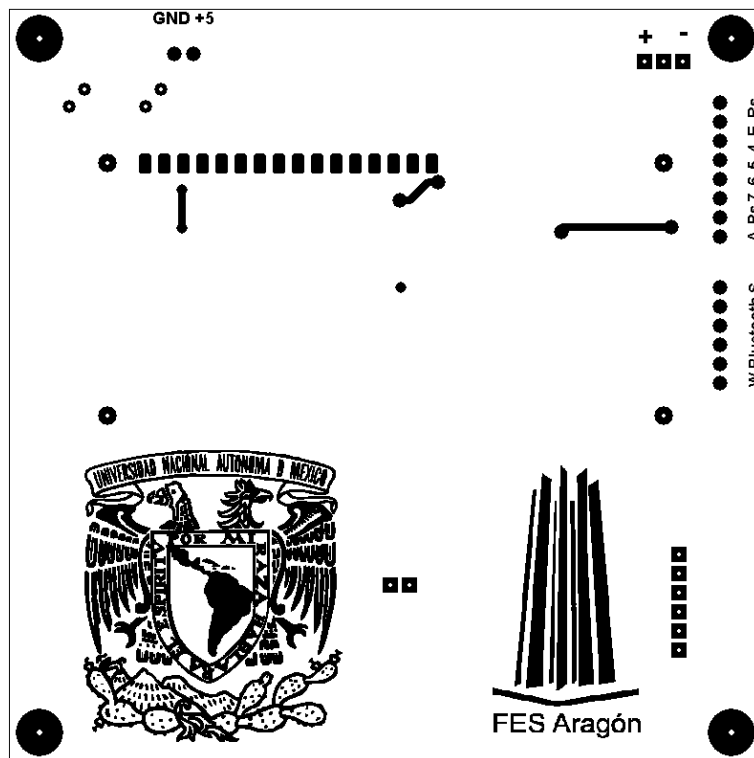
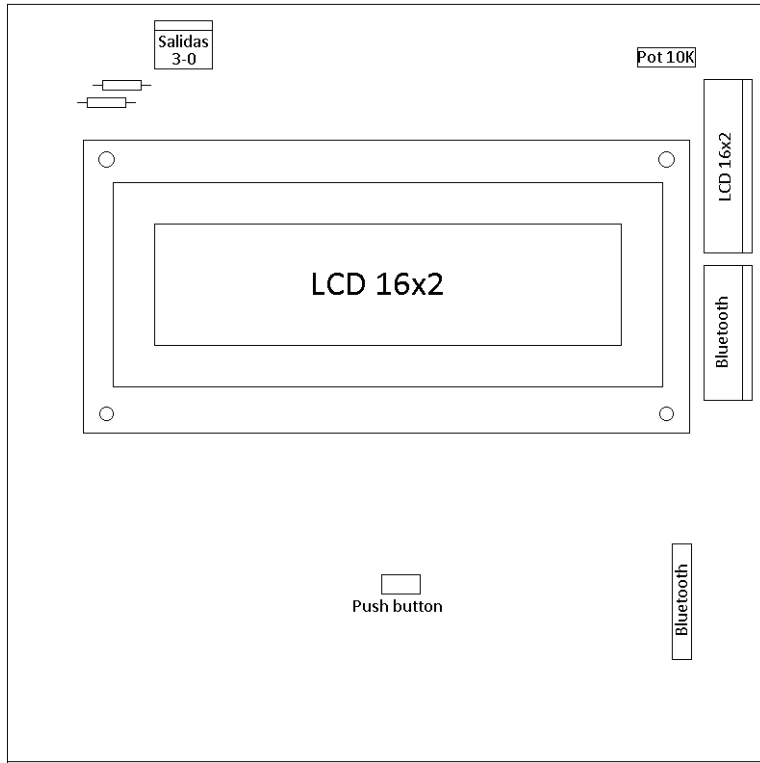


Figura 3.2.2-2 PCB #1 superior cobre.



En la figura 3.2.2-3 se observa el contorno de los dispositivos que se deben instalar en la placa 1 superior.

Figura 3.2.2-3 PCB #1 superior emplazamiento.

La segunda placa superior o placa 2 superior contiene la etapa de potencia y protección, en la figura 3.2.2-4 se ilustra el emplazamiento de los dispositivos y los conductores de cobre.

Nótese que los bornes para conectar los actuadores, o bornes número 0 al número 3, de salida están etiquetados por la nomenclatura

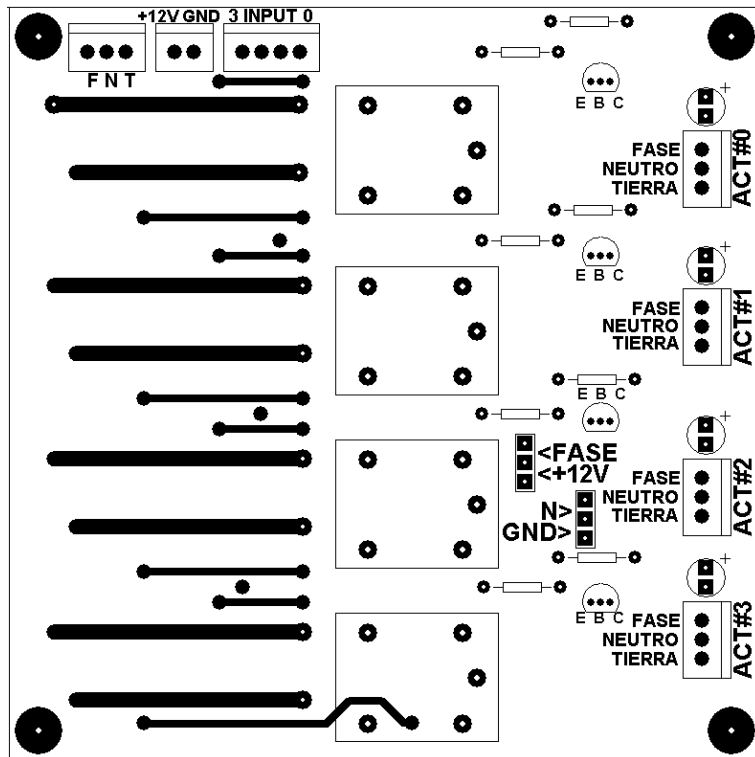


Figura 3.2.2-4 PCB #2 superior cobre y emplazamiento.

ACT#3, ACT#2, ACT#1 y ACT#0 correspondiendo la entrada digital Input 3 a Input 0.

En la figura 3.2.2-5 se ilustra la placa 2 superior solamente con los conductores de cobre así como algunas etiquetas correspondiente a la conexión de los bornes de entrada y salida, la polaridad de los transistores PN2222 y la posición de los jumpers selectores de 127Vac o 12Vdc para la salida ACT#2 o borne de salida número 2.

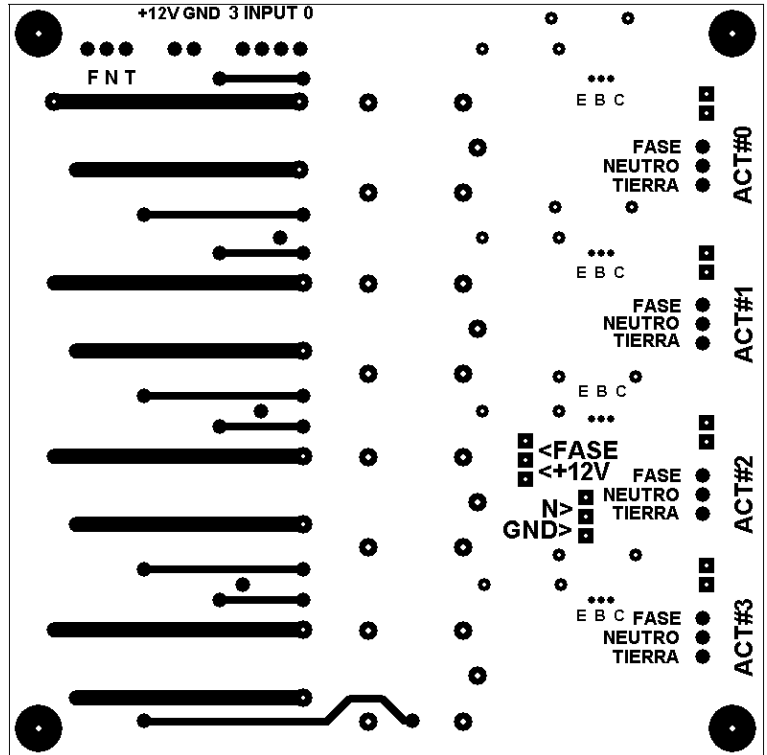
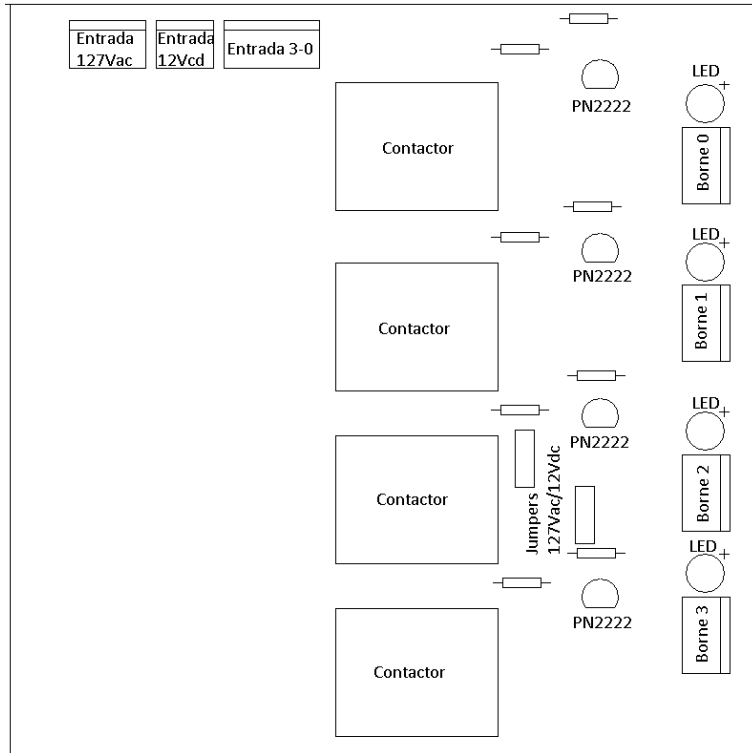


Figura 3.2.2-5 PCB #2 superior cobre.



En la figura 3.2.2-6 se ilustra la placa 2 superior con el emplazamiento de los dispositivos.

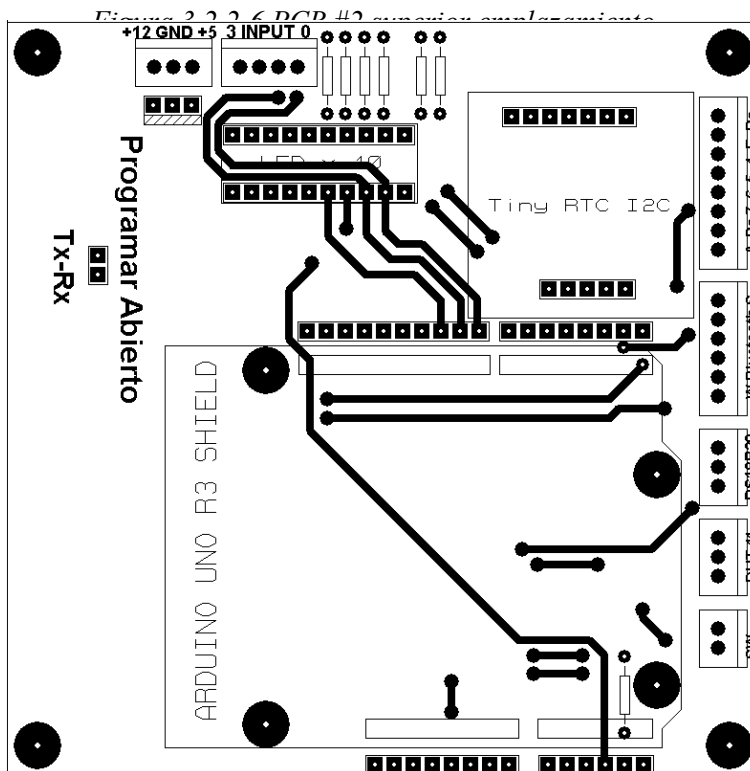


Figura 3.2.2-7 PCB #3 superior cobre y emplazamiento.

En la figura 3.2.2-7 se ilustra en conjunto los conductores de cobre y el emplazamiento de cada dispositivo y su correspondencia a la tercera placa superior o placa 3 superior.

En la figura 3.2.2-8 se ilustra la placa 3 superior solamente con los conductores de cobre.

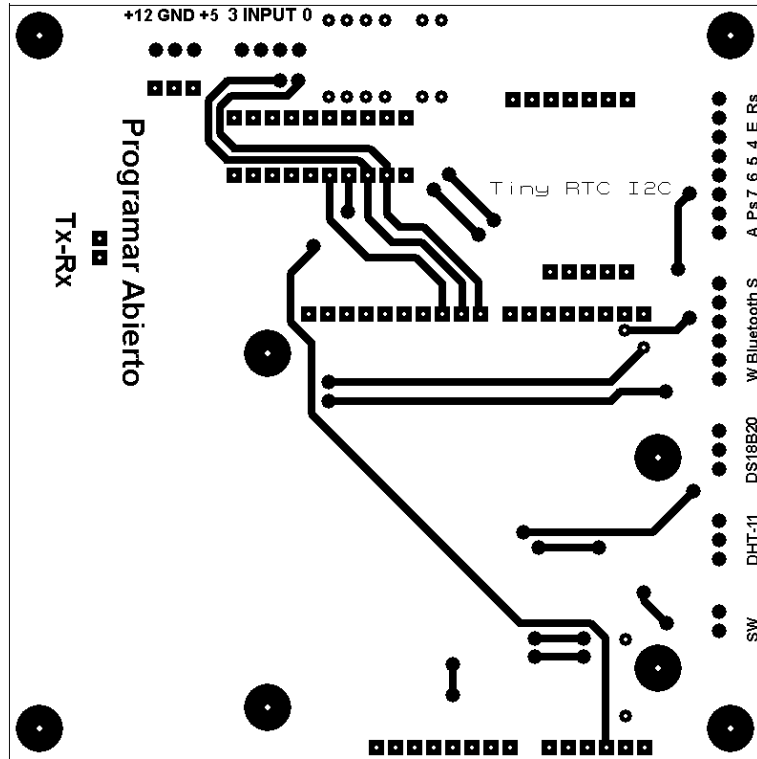


Figura 3.2.2-8 PCB #3 superior cobre.

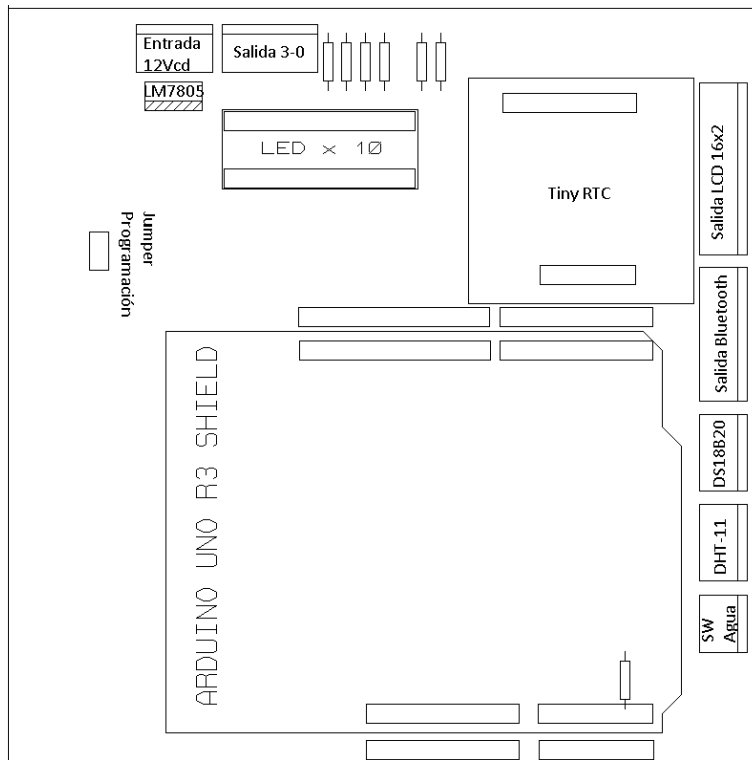


Figura 3.2.2-9 PCB #3 superior emplazamiento.

En la figura 3.2.2-9 se ilustra la placa 3 superior con el emplazamiento de los dispositivos.

En el caso del lado inferior de las placas, solo se requiere ilustrar las pistas de cobre, pues todos los dispositivos están colocados en el lado superior. En la figura 3.2.2-10 se ilustra la primera placa lado inferior, o placa 1 inferior.

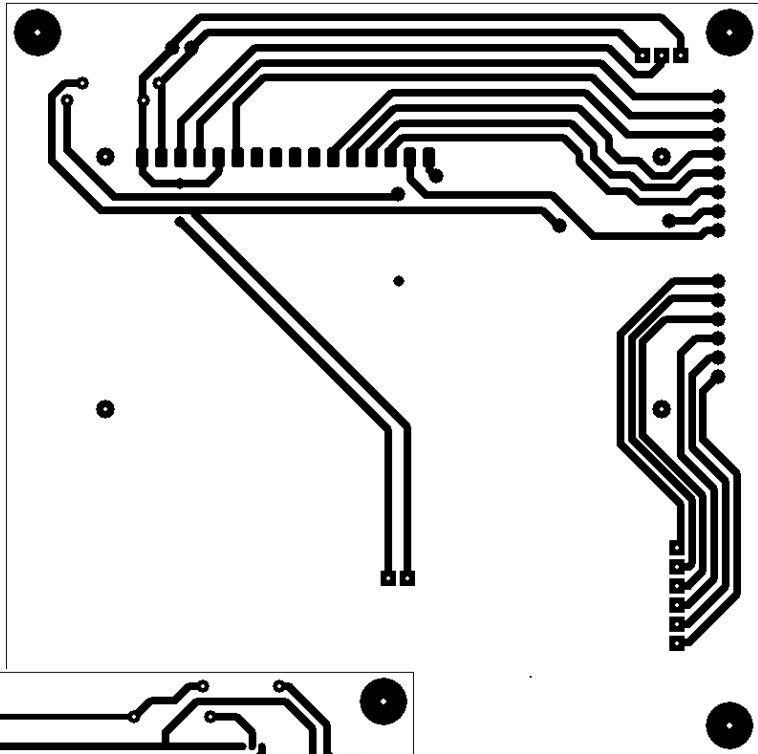


Figura 3.2.2-10 PCB #1 inferior.

El lado inferior de la segunda placa o placa 2 inferior se ilustra en la figura 3.2.2-11.

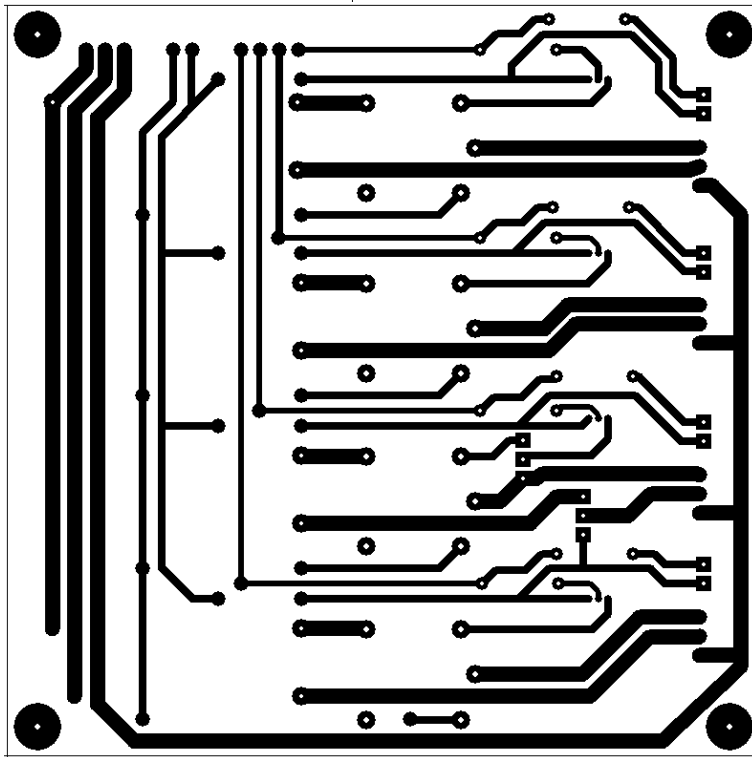


Figura 3.2.2-11 PCB #2 inferior.

El lado inferior de la tercera placa o placa 3 inferior se ilustra en la figura 3.2.2-12.

Es importante recordar que para reproducir estas placas se deben imprimir en espejo y que las dimensiones deben corresponder a 100mm x 100mm.

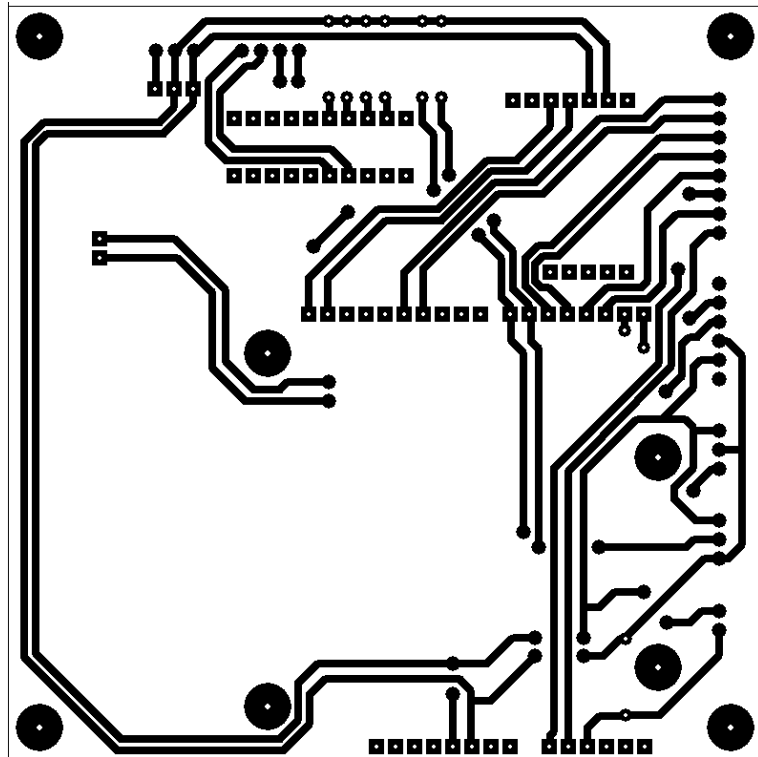


Figura 3.2.2-12 PCB #3 inferior.

3.3 Diseño y programación de la interfaz de usuario con Android

Habiendo recurrido a MIT App Inventor 2, se creó una sencilla aplicación para teléfono celular, que permite modificar ciertos valores vía Bluetooth, sin necesidad de requerir una computadora para programar la tarjeta de desarrollo Arduino UNO o tener una conexión alámbrica entre el teléfono celular y la tarjeta de desarrollo, con esto se logra simplificar la interacción entre un usuario no familiarizado con el uso de microcontroladores o programación de los mismos, y el sistema de control, así mismo se aprovecha la adquisición de información en tiempo real de manera remota, el concepto se conoce como **telemetría**, y literalmente significa medición a distancia.

3.3.1 Diseño de la interfaz de usuario

La plataforma MIT App Inventor 2, utiliza elementos organizados en columnas y renglones, utilizando elementos llamados **Arrangement** consistentes en arreglos de columnas y renglones dentro del orden predeterminado, se creó un arreglo de 4 tablas principales, mostradas en la figura 3.3.1-1, entre ellas se colocaron, botones para conectar el dispositivo móvil a la tarjeta de desarrollo, datos actuales, un registro historio de valores máximos y mínimos alcanzados, y un conjunto de barras deslizantes para guardar los niveles de umbral de trabajo mínimo y máximo en la **EEPROM** de la tarjeta de desarrollo, evitando así que con un corte del suministro eléctrico, dichos valores se pierdan y no se existan en el código del programa.

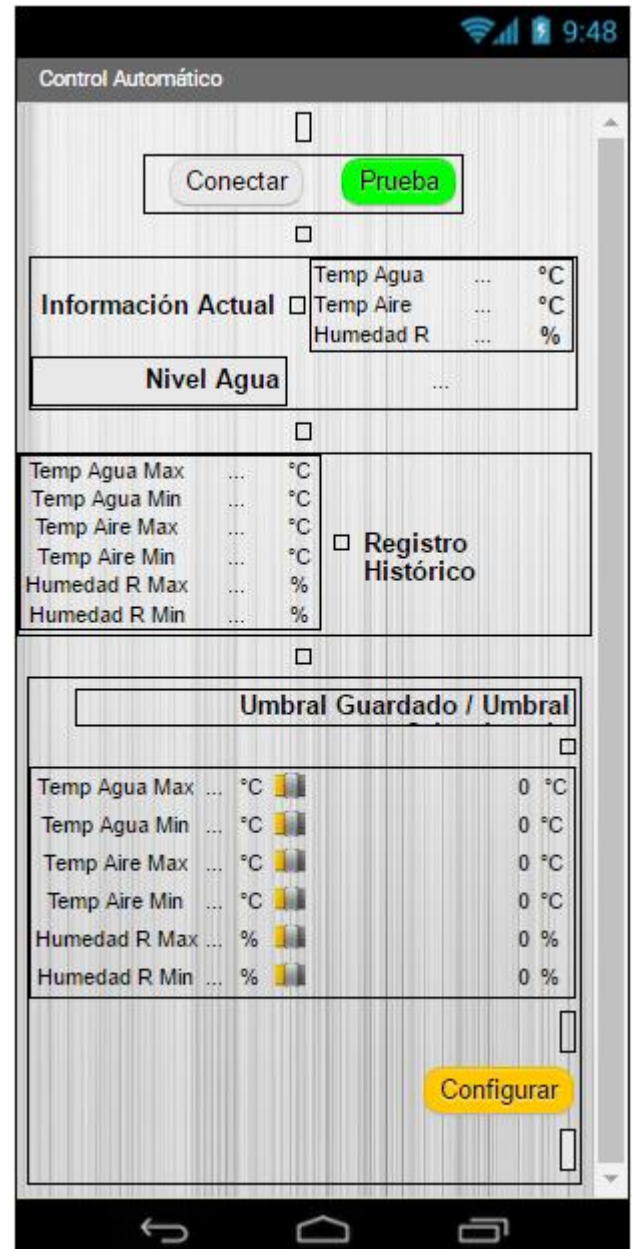


Figura 3.3.1-1 Vista preliminar de la interfaz.

Recordando que para agregar un elemento a la vista preliminar de la aplicación, es suficiente con arrastrar los elementos desde la Paleta al Visor.

En la figura 3.3.1-2 se observa el emplazamiento de los botones “Conectar” y “Prueba”, así como un arreglo vacío en la parte superior e inferior para agregar un espacio en el diseño de la aplicación, estos últimos, al no contener elementos, no serán visibles en la aplicación vista en el teléfono celular.

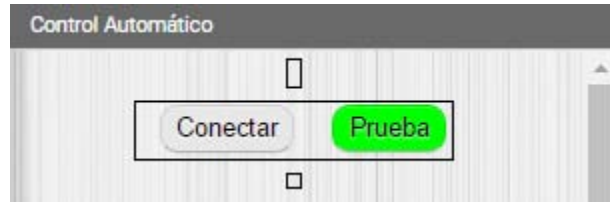


Figura 3.3.1-2 Botones Conectar y Prueba.

Los elementos arrastrados tienen un nombre predeterminado, y texto que pueden mostrar, en el caso del botón “Conectar”, su nombre es “Ligar” y el texto visible en la aplicación es “Conectar”, en el caso del botón “Prueba”, su nombre es “BotonTest” y el texto visible es “Prueba”, esto es para poder diferenciar los elementos en la programación de la aplicación, aunque muestren el mismo texto o

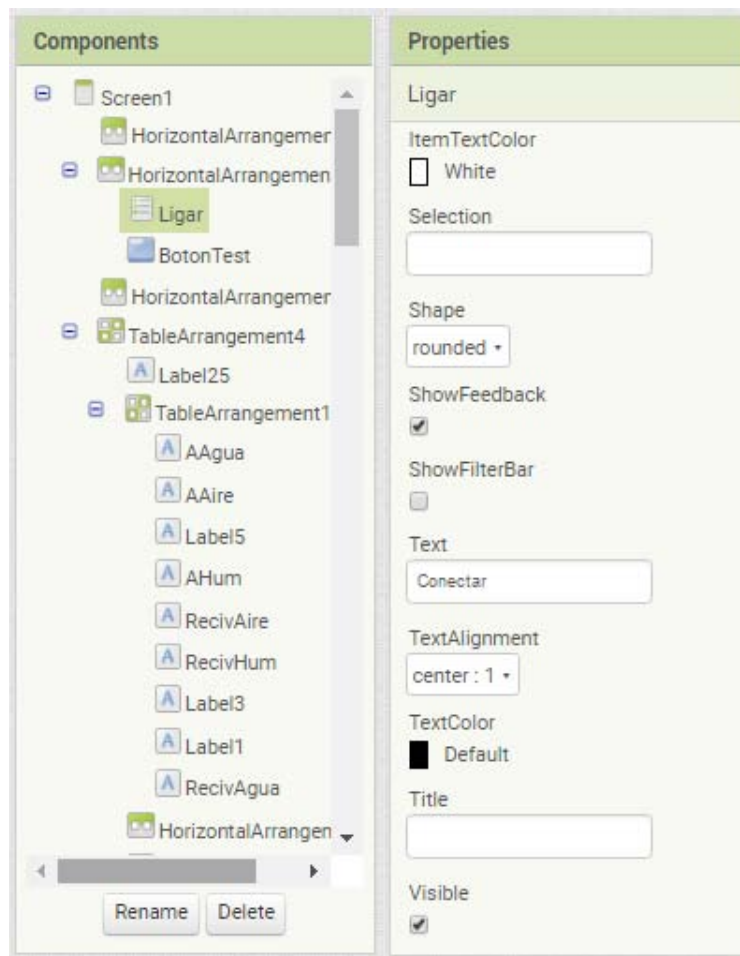


Figura 3.3.1-3 Propiedades de los elementos de la aplicación.

muestre un elemento sin texto y pueda ser modificado por alguna variable del

programa; En Propiedades de los elementos, se pueden modificar las características según el elemento.



Figura 3.3.1-4 Propiedades de los Arrangement.

En la figura 3.3.1-3 se observan elementos como “Label1”, “Label 3” o “Label5”, que conservan el nombre predeterminado por el MIT App Inventor 2, y otros elementos como las características del botón “Conectar”, que se muestra seleccionado en la imagen, sombreado, entre las características, se muestra color de texto, **shape** o forma del elemento, y si es visible o no.

Algunas características de los **Arrangement** son determinar el número de columnas y renglones, altura y anchura en pixeles, y si es visible o no, como se muestra en la figura 3.3.1-4.

es

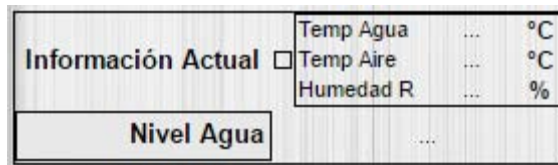
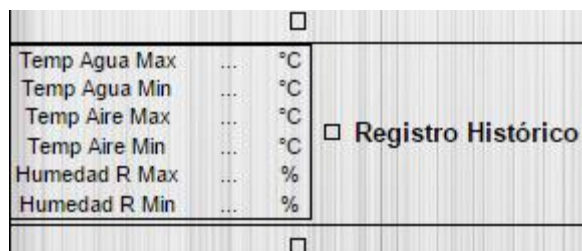


Figura 3.3.1-5 Vista previa Información Actual.

si el



en la

Figura 3.3.1-6 Vista previa Registro Histórico.

Dentro de cada sección de los **Arrangement**

posible colocar otro **Arrangement**, o lo que es lo mismo, en la celda de un tabla, colocar otra tabla, sin importar número de columnas y renglones de cada arreglo, y se muestra la segunda tabla de la sección **Información Actual** aplicación desarrollada en la figura 3.3.1-5.

Para mostrar la tercera tabla principal de la interfaz de usuario, se hizo nuevamente un **Arrangement** dentro de otro **Arrangement**, con lo que se consigue agrupar y mostrar los elementos en los y diferenciarlos con un encabezado, al igual que las demás tablas principales, en la

figura 3.3.1-6 se muestra dicha tabla y dos arreglos vacíos colocados arriba y abajo para agregar espacio entre las tablas.

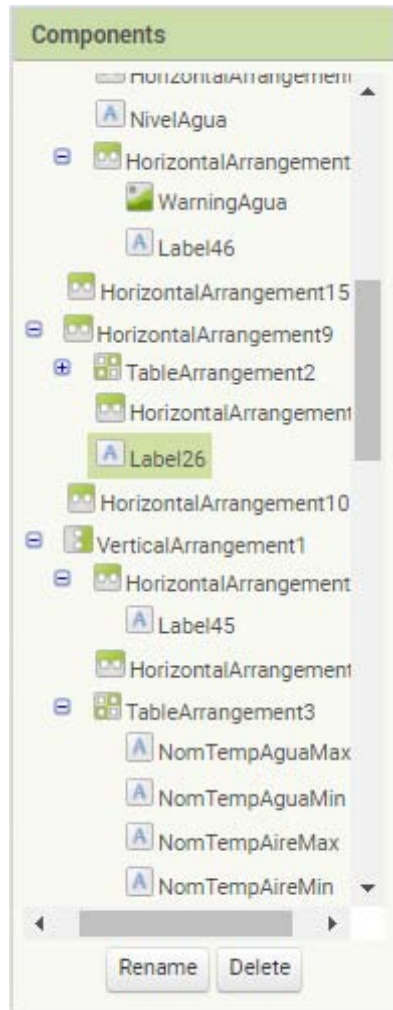


Figura 3.3.1-7 Localización de los elementos de la aplicación.

La tabla tercera tabla, es decir, donde se muestra el registro histórico, muestra los valores máximos y mínimos alcanzados durante todo el tiempo que la tarjeta de desarrollo este encendida, y automáticamente serán mostrados en el orden asignado y acompañados por las unidades de medición correspondientes.

Mediante la adquisición de datos de forma remota, las etiquetas que muestran en texto tres puntos suspensivos (...) serán reemplazadas por los valores numéricos adquiridos, esto nos hace retomar la idea que el nombre de los elementos puede no corresponder al texto que muestran.

En la figura 3.3.1-7 se muestra las distribuciones de la tabla en la sección Componentes, y se observa la distribución de los **Arrangement**, nuevamente se utilizaron estos arreglos vacíos con dimensiones arbitrarias en pixeles para separar arbitrariamente las tablas principales.

Para poder cambiar los niveles máximos y mínimos permitidos, se utilizaron barras deslizantes o **sliders**, con un valor mínimo y un valor máximo determinado, evitando así asignar valores peligrosos según los parámetros a recrear del hábitat de las iguanas, además de estar posicionados predeterminadamente en un valor sugerido.

También se colocó un botón “Configurar”, para confirmar el cambio de valores y un cuadro de dialogo para confirmar dicho cambio. En la figura 3.3.1-8 se observa la distribución de los sliders y el botón que muestra el texto “Configurar”, así como **Arrangement** para separar elementos entre sí.



Figura 3.3.1-8 Sliders y botón Configurar.

Se colocaron etiquetas donde se mostraran los valores guardados, al lado izquierdo, y valores seleccionados en el **slider**, al lado derecho, así como sus unidades. Al botón “Configurar” se le asignó un color amarillo para resaltar y evitar tocarlo por accidente.

También se observan a varias etiquetas que muestran el texto “Temp Agua Max” y otras muestran el texto tres puntos suspensivos (...) y que cambian a valores numéricos que estarán dependiendo de los valores censados por los sensores.

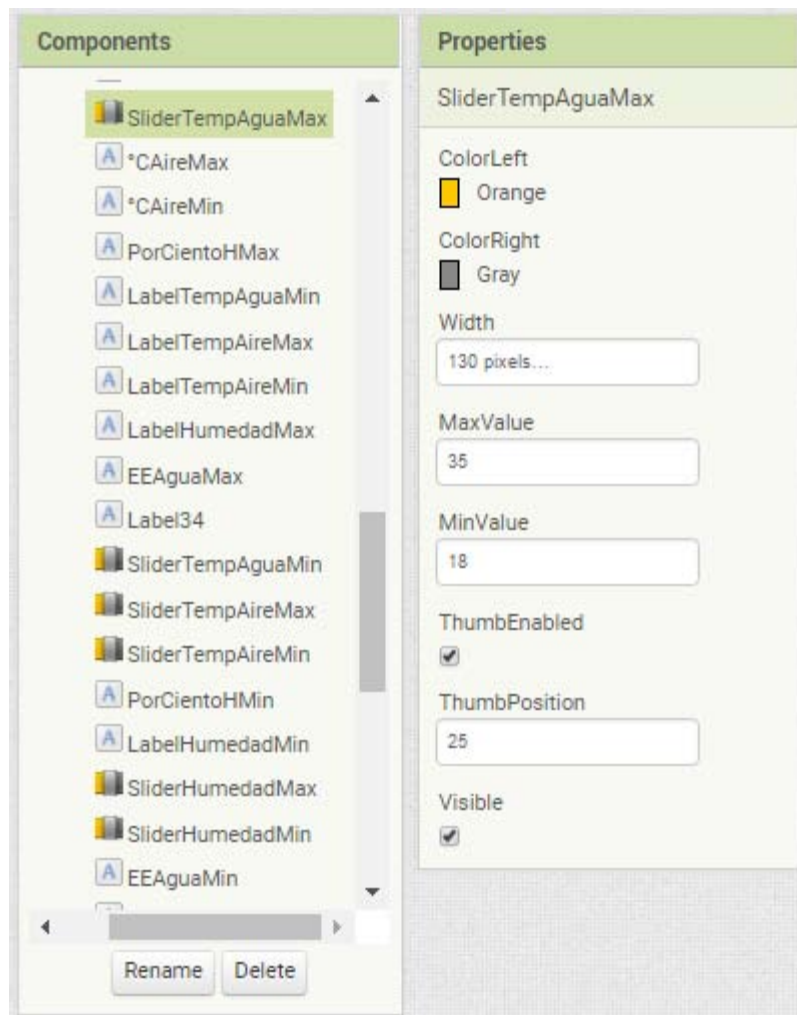


Figura 3.3.1-9 Propiedades de los sliders.

En la figura 3.3.1-9 se observa las características del primer **slider**, como color de la barra, ancho determinado a 130 pixeles, valor máximo 35 y mínimo 18, para este caso representan los valores permisibles para la temperatura del agua, es decir el umbral tendrá un mínimo de 18°C y el máximo será 35°C, y una **ThumbPosition**, es decir, posición de la barra, en 25, que representa una sugerencia a 25°C como valor de temperatura máximo predeterminado, así como elemento visible.

Estos valores son diferentes para el caso de humedad relativa, los cuales son 70% como humedad relativa máxima, 30% como humedad relativa mínima, y 40% como humedad sugerida.

Otros elementos no visibles en la aplicación aparecen en la parte inferior de la ventana Visor, donde se muestra una versión previa de la aplicación finalizada, entre los que se utilizaron para esta propuesta, se encuentran los siguientes:

- ◆ BluetoothClient1: Cliente de Bluetooth.
- ◆ Clock1: Reloj utilizado para actualizar la posición de los slider con su etiqueta.
- ◆ Clock2: Reloj utilizado para actualizar los valores censados por la tarjeta de desarrollo y sus etiquetas correspondientes en la interfaz.
- ◆ DelayClock: Reloj utilizado como timer de retardo.
- ◆ Notifier1: Alerta para notificar la prueba de los actuadores.
- ◆ Notifier2: Alerta utilizada para confirmar la configuración de los valores del umbral de trabajo.
- ◆ Sound1: Utilizado por MIT App Inventor 2 para acceder al vibrador del teléfono celular.
- ◆ Sound2: Utilizado para emitir una alarma sonora en caso de malfuncionamiento de algún sensor en la el sistema de control.

En la figura 3.3.1-10 se observan los elementos no visibles pero que interactúan con la interfaz y su programación y no se debe confundir con la propiedad de los elementos al ser visibles o invisibles.

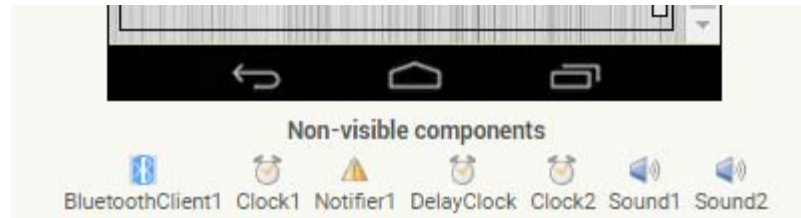


Figura 3.3.1-10 Elementos ocultos.

3.3.2 Programación de la interfaz de usuario

Para conectar vía Bluetooth la interfaz de usuario en el teléfono móvil con el módulo HC-05 conectado a la tarjeta de desarrollo Arduino UNO, se requiere establecer una conexión propia de los dispositivos que funcionan con Bluetooth, se conoce como **Parear** o **Emparejar**, esta ligación se realiza activando el Bluetooth en el teléfono móvil, y después seleccionando el módulo HC-05 para poder ser aceptada la comunicación entre el teléfono móvil y el módulo HC-05.

Una vez pareados ambos dispositivos, se puede comunicar la aplicación con los protocolos del teléfono móvil, para ello se programó el siguiente algoritmo que puede ser utilizado para establecer cualquier conexión entre dispositivos Bluetooth y las aplicaciones desarrolladas por la plataforma MIT App Inventor 2, en la figura 3.3.2-1 se observa el comienzo de la comunicación entre la aplicación y el módulo HC-05, donde se detecta si está o no activada la comunicación Bluetooth del dispositivo móvil, dado que se debe parear la comunicación entre el teléfono móvil y el módulo HC-05, por lo que solo es necesario encargarse de la comunicación entre la aplicación y el módulo HC-05.



Figura 3.3.2-1 Bloques de Emparejamiento.

En la figura anterior, se describe el algoritmo que al estar activada la comunicación Bluetooth, se solicitan las direcciones y nombres de los dispositivos pareados en el teléfono móvil.

En la figura 3.3.2-2 se describe el algoritmo para ligar la comunicación de la aplicación a través del dispositivo móvil, y el módulo HC-05.

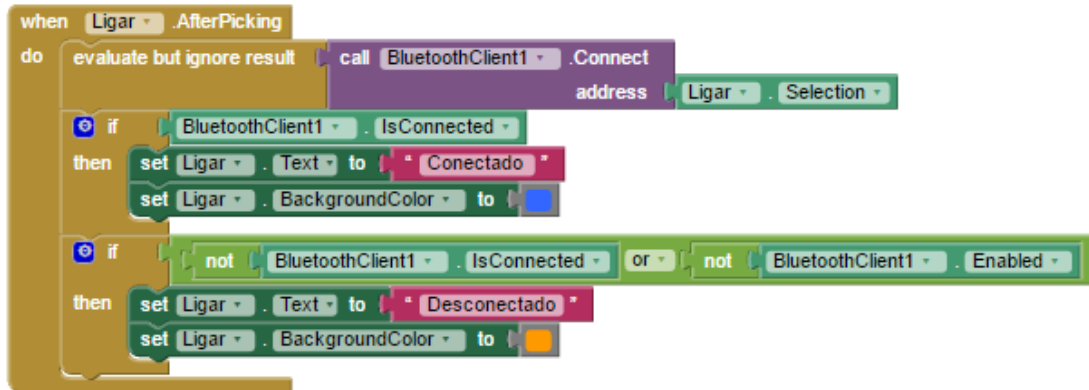


Figura 3.3.2-2 Bloques de ligación.

El algoritmo descrito en la figura anterior, solo se ejecuta después de haber tocado o presionado el botón que muestra el texto “Conectar”, su nombre es “Ligar”, y lo primero que realiza es evaluar si está disponible una conexión entre el dispositivo móvil y el módulo HC-05, si existe dicha posibilidad de conexión aparecerá en la pantalla del dispositivo móvil, una lista con los nombres de los dispositivo previamente pareados (mostrada en la figura 3.3.2-3), y se selecciona HC-05, es importante mencionar que esta lista es ejecutada sin necesidad de crear un código para realizarla, es decir, está programada por la plataforma MIT App Inventor 2.

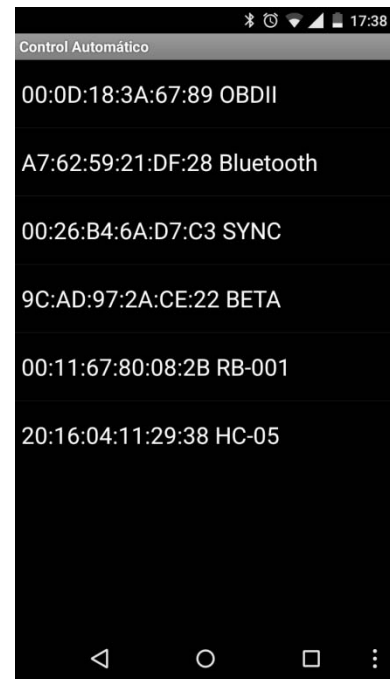


Figura 3.3.2-3 Lista de dispositivos emparejados.

Seguidamente evalúa dos condiciones, si esta conexión fue exitosa, entonces el botón con texto “Conectar” y de nombre “Ligar”, cambiara el texto mostrado a “Conectado” y el botón se tornara de color gris a azul, si no fue exitosa dicha conexión, el texto cambiara a “Desconectado” y el botón se tornara de color gris a anaranjado como se muestra en la figura 3.3.2-4.

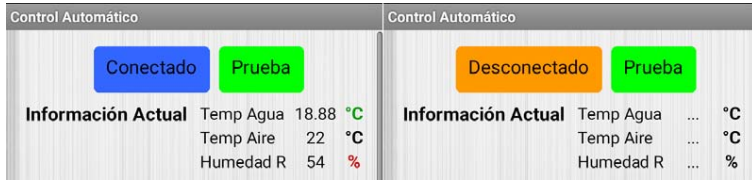


Figura 3.3.2-4 Botón Conexión exitosa/no exitosa.

Para actualizar los valores de información actual, se describe un código utilizando un **reloj o timer**, en este caso el Clock2, esto es para actualizar la información cada 100 milisegundos o el intervalo de tiempo establecido en el panel de configuración del elemento Clock2 en la pestaña Diseño.

El algoritmo descrito en la figura 3.3.2-5 comienza con la ejecución del **timer** Clock2, después una toma de decisión, que evalúa la conexión Bluetooth, si esta existe, entonces guarda toda la información en un etiqueta no visible llamada **Data**, esta etiqueta no importa su ubicación en la pestaña Diseño, pues no es visible para el usuario y solo almacena la información recibida, después evalúa un rango numérico, y lo muestra en el nombre de la etiqueta correspondiente, es decir, para el caso de la figura anterior, la Temperatura del Agua Máxima, es enviada a través de la **USART** de la tarjeta de desarrollo acrecentándole 1000 al valor censado, es decir si registra 22°C el valor será enviado como el número 1022, después es transmitida por el módulo HC-05 y es recibida por el teléfono móvil y guardada en la etiqueta Data, después se evalúa si esta entre 1000 y 1999, si está en ese rango, se le decrementará 1000 y será el texto mostrado por la etiqueta correspondiente a la Temperatura del Agua máxima, es decir el valor 1022, se le decrementara 1000 y

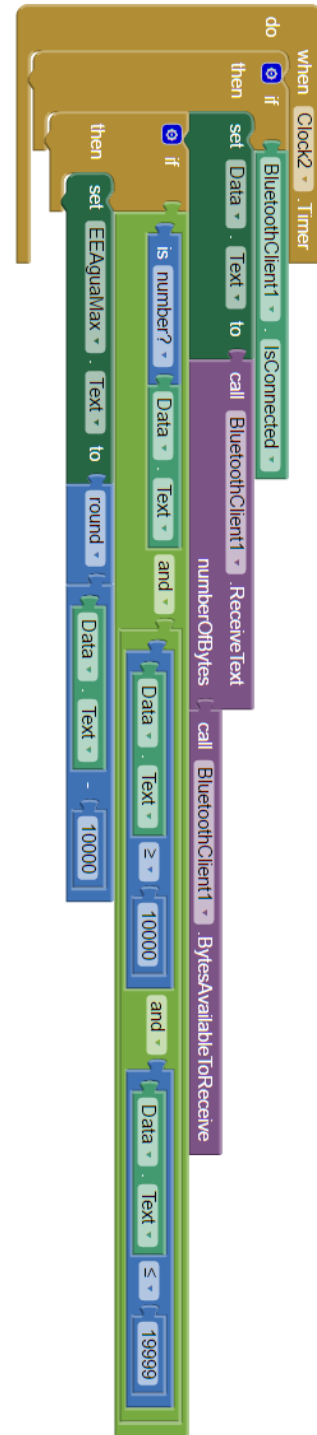


Figura 3.3.2-5 Bloques de conexión Bluetooth.

será mostrado como valor numérico 22, y es precedido por la etiqueta TempAguaMax y seguido por sus unidades °C.

Dicho proceso es repetido para cada valor mostrado en la interfaz, y cada valor leído o guardado en la **EEPROM** de la tarjeta de desarrollo, y antes de su transmisión ha sido incrementado para poder ser seleccionado del resto de números recibidos en la aplicación.

En la figura 3.3.2-6 se describe el algoritmo para almacenar valores en la **EEPROM** de la tarjeta de desarrollo Arduino UNO, el código mostrado comienza al confirmar la Notifier1, que es un cuadro de diálogo para confirmar o rechazar, mediante botón “Sí” y botón “No”, la escritura de los valores tomados de los sliders (mostrados en la figura 3.3.1-8) para posteriormente enviarlos a la tarjeta de desarrollo, encabezados por una letra que corresponde a una localidad de memoria arbitraria.

Si existe una conexión Bluetooth tomará cada valor de los sliders y los asignará a sus etiquetas correspondientes, después, en la función call Bluetooth, unirá la letra correspondiente a la localidad de memoria EEPROM y el valor de la etiqueta mediante la función join, y finalizará con un salto de línea (representado por los caracteres: \n), el código toma cuatro delay o retardos de un segundo para esperar la escritura en la tarjeta de desarrollo, y continúa repitiendo los mismos pasos con cada una de los valores guardados en las etiquetas de los sliders.

La visualización final de la aplicación ejecutada desde el teléfono móvil y conectada al módulo HC-05 es la mostrada en la figura 3.3.2-7.

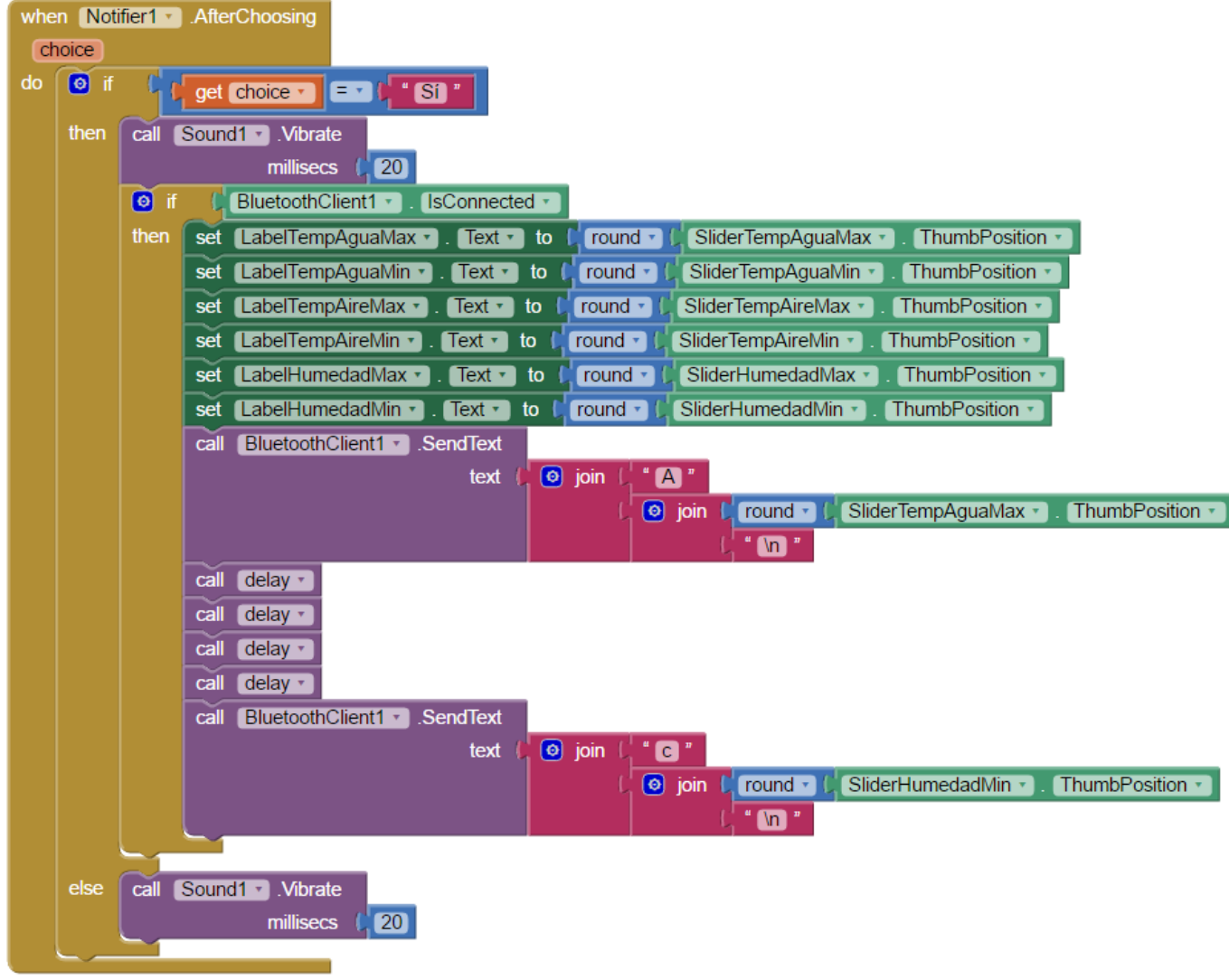


Figura 3.3.2-6 Bloques de recepción e interpretación de los datos recibidos via Bluetooth.



Figura 3.3.2-7 Vista de la aplicación ejecutada desde el teléfono móvil e interactuando con la tarjeta de desarrollo.

IV. Pruebas y Resultados

En esta sección se describen las pruebas realizadas para comprobar el desempeño de todas las etapas del sistema, obteniendo resultados que permiten comprobar el

desempeño del mismo, finalmente se comentan los resultados y conclusiones de este proyecto, así como el trabajo futuro que se puede realizar.

4.1 Pruebas de conectividad Bluetooth

Para evaluar el desempeño de la conectividad vía Bluetooth, se realizaron pruebas en una residencia, y se evaluaron diferentes dispositivos, como una computadora portátil conectándolos al módulo HC-05, y aunque no se realizó una interfaz para computadora, se pueden observar la información enviada a través del monitor serial y asignando el puerto **COM** (puerto de comunicación serial, empleado en computadoras y periféricos) correspondiente, los resultados de conexión exitosa fueron los mostrados en la tabla 4.1-1.

Tabla 4.1 Resultados de las pruebas de conexión vía Bluetooth.

Dispositivo	Distancia Aproximada	Obstáculos
Teléfono móvil	<10 m	Ninguno
Teléfono móvil	4 m	Muro 20cm
Computador	>10 m	Ninguno
Computador	6 m	Muro 20cm

Para realizar dicha conexión, basta con alimentar la tarjeta de desarrollo, y habilitar la conexión Bluetooth en el dispositivo móvil, seguidamente se accede a la interfaz desarrollada para este proyecto, y se debe tocar el botón “Conectar”, y seleccionar el módulo HC-05 como se muestra en la figura 4.1-1, si la conexión es exitosa el botón mostrara el texto “Conectado”, y se tornara de color gris a color azul, si la conexión no fue exitosa, el botón mostrara el texto “Desconectado”, y se tornara de color gris a color naranja como se observa en la figura 4.1-2.

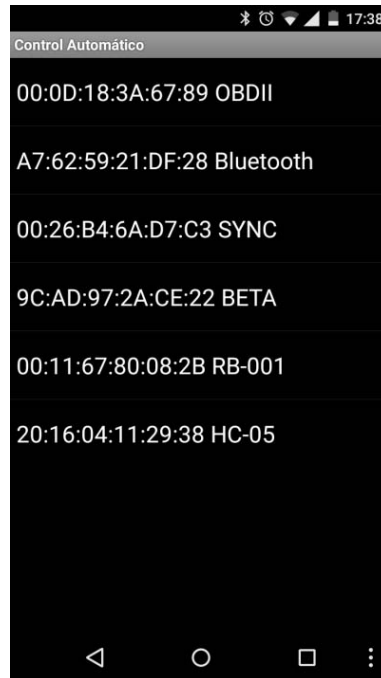


Figura 4.1-1 Lista de dispositivos pareados.

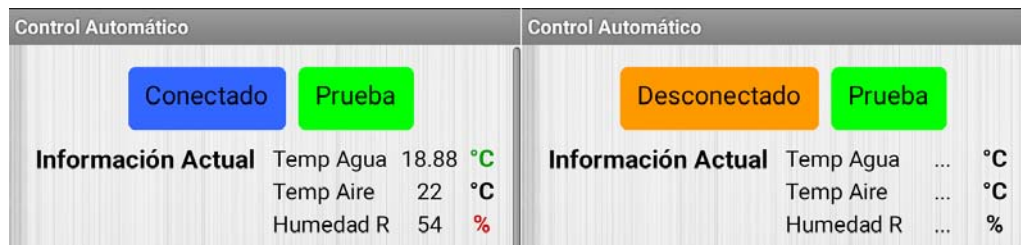


Figura 4.1-2 Botón de conexión.

Para conectar el computador se debe habilitar la comunicación vía Bluetooth, abrir el compilador de Arduino, seleccionar el puerto COM que será diferente para cada dispositivo, como se muestra en la figura 4.1-3, y dar clic en el botón “Monitor Serial”, lo que abrirá una nueva ventana donde se podrá enviar y recibir información, como se muestra en la figura 4.1-4.

Si la comunicación es exitosa, se podrán observar todos los datos recibidos vía Bluetooth, así como mantener una conexión sin interrupción, además de poder enviar caracteres y poder interactuar con la tarjeta de desarrollo, en la figura 4.1-5 se observa el Monitor Serial recibiendo información de la tarjeta de desarrollo.

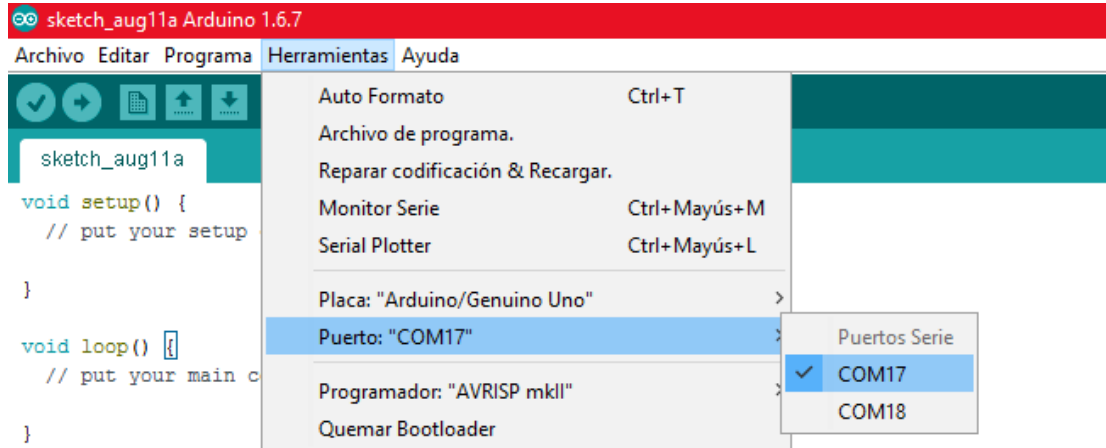


Figura 4.1-3 Selección puerto COM.

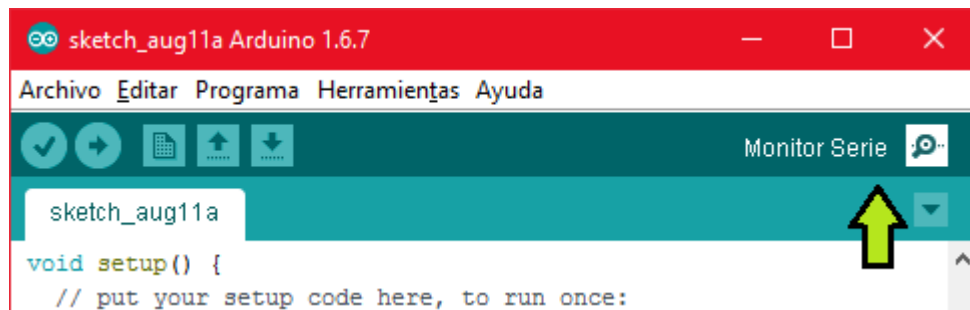


Figura 4.1-4 Ejecución Monitor Serial.

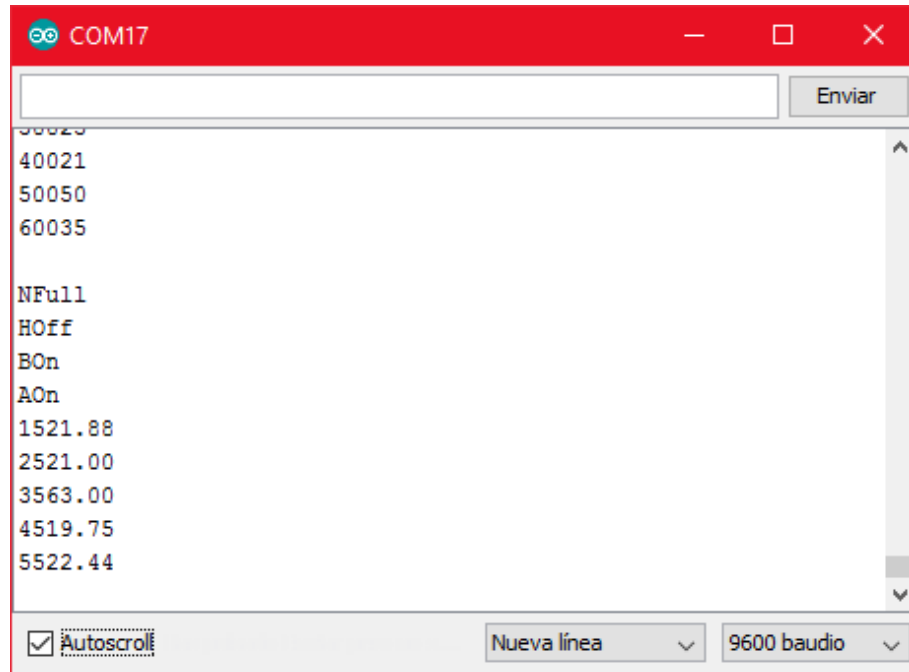


Figura 4.1-5 Información recibida en computador vía Bluetooth.

4.1.1 Resultados de conectividad Bluetooth

Habiendo tenido una conexión estable vía Bluetooth, se puso a prueba la capacidad de mantener dicha conexión incrementando la distancia entre el módulo HC-05 y el dispositivo, siendo notable la incapacidad de mantener una conexión al existir muros entre ambos dispositivos, así como elementos metálicos de tamaño considerable como puertas.

La respuesta de estos dispositivos en general fue satisfactoria, pues puede existir una interacción entre usuario y el sistema de control sin requerir un cable, o estar en contacto directo, siendo posible estar alejado algunos metros, o en una habitación adjunta. Aunque la distancia no es demasiada, es suficiente para este proyecto, siendo una posibilidad para trabajo futuro, adicionar elementos de mayor alcance como **WiFi** (conexión inalámbrica a una red LAN), **XBee** (comunicación por radio) o inclusive un sistema **GSM** (Global System for Mobile communications) que permita utilizar una red de telefonía celular para poder adquirir y proporcionar información entre la tarjeta de desarrollo y el teléfono celular. Se podría afirmar que este sistema tiene una distancia aproximada de 5 metros donde la conexión es estable, aunque dependiendo

de los obstáculos que existan será su efectividad y no requiere de conocimientos en comunicaciones para poder utilizar esta interfaz. **Para poder programar la tarjeta de desarrollo el “pin Rx” debe ser desconectado del “pin Tx” del módulo HC-05.**

4.2 Pruebas de la interfaz Android

La interfaz de usuario puede realizar dos tareas principales, visualización de la adquisición de información en la tarjeta de desarrollo, y poder configurar los niveles máximos y mínimos de los rangos de trabajo de los actuadores. Para ello es necesario haber conectado vía Bluetooth ambos dispositivos, seguidamente se observa que la aplicación desarrollada adquiere información como el estado de los actuadores, parámetros actuales, registro histórico, umbral almacenado en la **EEPROM**.

Al inicializar la aplicación se observa en la figura 4.2-1 que existe un botón de color verde con el texto “Prueba”, dicho botón, al ser tocado, envía el carácter “T”, que es interpretado por la tarjeta de desarrollo, y activa los actuadores durante 2 segundos, esto es para poder detectar posibles problemas en la etapa de potencia o actuadores y si es requerido reemplazar algún dispositivo.



Figura 4.2-1 Vista de la aplicación ejecutada desde el teléfono móvil e interactuando con la tarjeta de desarrollo.

Los símbolos de las unidades de medida en la sección Información Actual, al inicio de la ejecución de la aplicación son color gris oscuro, al adquirir información sobre el estado de los actuadores, se tornaran color verde al estar encendidos o color rojo al estar apagados, así se puede conocer el estado de los actuadores, a la vez que se visualiza información en tiempo real. Recordando que el nivel de agua también está vinculado al comportamiento del actuador encargado de elevar la temperatura del agua, en caso que el nivel del agua sea inferior al mínimo, es decir, que el actuador no este sumergido totalmente, el actuador nunca se energizara y se mostrara un icono de color ámbar con un signo de exclamación, así como el nivel de agua mostrara el texto “Insuficiente”, se observa en la figura 4.2-2.

En la sección Registro Histórico se visualiza los niveles máximos alcanzados durante el tiempo que la tarjeta de desarrollo haya adquirido información de los sensores, es decir, el tiempo que el sistema de control permanezca encendido, permitiendo así, observar el desempeño de la propuesta experimental.



Figura 4.2-2 Alerta nivel agua insuficiente.

En la sección última, se aprecian los sliders correspondientes a cada nivel superior e inferior de las tres variables, temperatura del aire, temperatura del agua y humedad relativa, los sliders están antecidos por “Umbral Guardado”, estos valores están almacenados en la **EEPROM** de la tarjeta de desarrollo, y seguidos por “Umbral Seleccionado”, que corresponde a los valores de los sliders, si se toca el botón amarillo con el texto “Configurar”, los valores de los sliders serán almacenados en la **EEPROM**, así se puede modificar el umbral de trabajo, es decir, los niveles mínimos y máximos sin requerir reprogramar la tarjeta de desarrollo. Un cuadro de dialogo

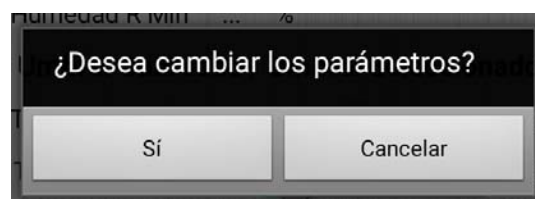


Figura 4.2-3 Dialogo de confirmación.

aparecerá después de tocar el botón “Configurar”, mostrando un mensaje de confirmación como se observa en la figura 4.2-3.

Habiendo confirmado almacenar los valores nuevos, demorara unos segundos en completar la escritura en la **EEPROM**, el fabricante proporciona el tiempo de escritura y demora 3.3 ms en ser completada, pero debido a la estructura del código se debe aguardar aproximadamente 20 segundos en completar la escritura de los seis valores, esto es debido a que al finalizar la escritura del primer valor, el flujo de programa ejecuta todos las decisiones antes de poder volver a escribir en la **EEPROM**, cada **Loop** del programa demora aproximadamente 3 segundos si no es demandada otra acción por el usuario, como probar los actuadores, por lo que para completar la escritura el programa debe completar 6 Loop.

En caso de la pérdida de comunicación con un sensor, la tarjeta de desarrollo transmite caracteres “nan” o “-127” que son interpretados en la interfaz de usuario como condiciones que emiten una alarma sonora y hace vibrar el teléfono celular a la vez que muestra un mensaje textual en la pantalla del teléfono celular, como se observa en la figura 4.2-4 al lado derecho de la misma, el mensaje desplegado para la falta de sensor DS18B20 y a la derecha el DHT11.



Figura 4.2-4 Mensajes de error en la interfaz.

4.2.1 Resultados de las pruebas de la interfaz Android

En general las pruebas resultaron fueron positivas, en algunos casos la actualización de la información demora 20 a 30 segundos, así como el cambio de color en los símbolos centígrado y porciento, que representan el estado del actuador demora unos 6 segundos. Así mismo se observa un problema con la alarma sonora emitida al no tener comunicación con algún sensor, pues tras salir del rango de alcance del Bluetooth, en

ocasiones es emitida la alarma a pesar de no estar conectado al módulo HC-05, esto se soluciona desactivando la comunicación Bluetooth del teléfono celular.

La escritura de la **EEPROM** en todas las pruebas realizadas fue satisfactoria y se comprobó la escritura en las localidades de memoria mediante una línea de texto retornada al escribir satisfactoriamente, esta línea de texto solo se puede observar en el monitor serial. En la figura 4.2.1-1 se observa dicha línea de texto retornada al escribir satisfactoriamente en la localidad de memoria 0 correspondiente al nivel máximo de temperatura del agua. En la aplicación también se puede comprobar la escritura, ya que si fue satisfactoria, se cerrará automáticamente el cuadro de texto de confirmación de escritura, después de haber sido tocado el botón “Configurar”. Por segunda vez la escritura se puede comprobar si exitosa al ser actualizados los valores en la sección Umbral Guardado, estos valores son leídos de la **EEPROM** en cada **Loop** realizado por la tarjeta de desarrollo.

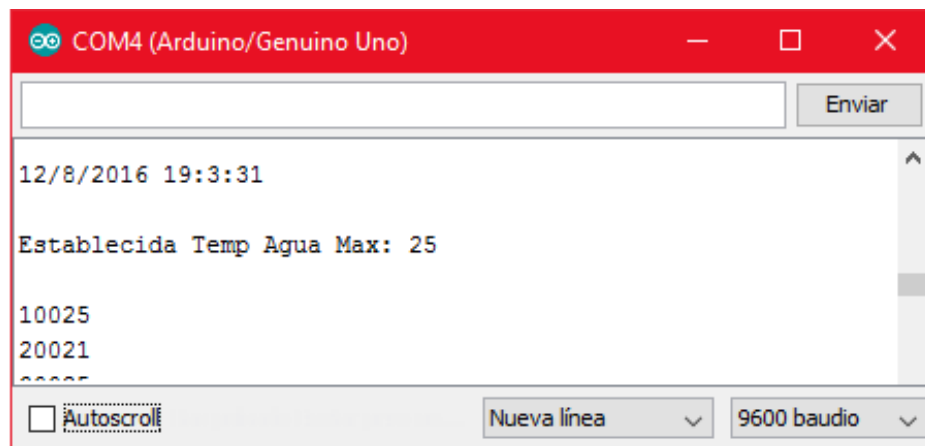


Figura 4.2.1-1 Confirmación de configuración EEPROM vista en el computador vía Bluetooth.

4.3 Pruebas de emplazamiento de los sensores

El emplazamiento de los sensores afectara la magnitud máxima del medio afectada por el actuador correspondiente, debido a que la temperatura podría no ser homogénea instantáneamente, es decir, parte del aire o agua podría estar a diferente temperatura y demoraría tiempo cesar el valor que apagaría el actuador, siendo la magnitud máxima censada superior a la que debe apagar el actuador, fundamentado en el equilibrio térmico o ley cero de la termodinámica, que enuncia: “Dos sistemas A y B que separadamente están en equilibrio térmico con un tercer sistema C, están en equilibrio

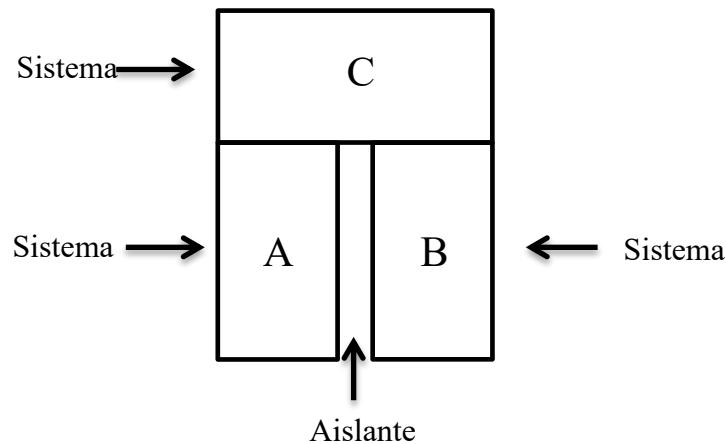


Figura 4.3-1 Sistema térmico.

térmico también entre sí”, en este caso A será el actuador, B el sensor y C el medio, ejemplificado en la figura 4.3-1.

Para el caso del sensor sumergible DS18B20, se colocó en el lado opuesto de donde se colocó el calentador del agua, permitiendo así censar la temperatura en el lado más frío del agua, puesto que si estuviese cerca del actuador la temperatura censada incluso podría ser la temperatura del actuador, que podría superar los 100°C.

Para el caso del sensor DHT-11, se colocó adherido en una pared del terrario, a una distancia central en relación a la longitud y altura del terrario, esto provocó problemas, pues condensación se acumulaba en la pared, del mismo modo que condensación se acumula en el muro del baño al tomar una ducha, o en el parabrisas de los automóviles en días lluviosos. Esta condensación es provocada por la diferencia de temperatura entre el aire y la pared, la humedad en el aire se condensa formando gotas de agua, estas gotas de agua se deslizaron por la pared y entraron en el sensor DHT-11 provocando sobrecalentamiento, dañando parcialmente el recubrimiento plástico del sensor, posteriormente se optó por alejar unos centímetros un nuevo sensor DHT-11 de la pared utilizando los cables de alimentación y señal del sensor como estructura de sujeción, como se muestra en la figura 4.3-2, con esto se consiguió evitar que la condensación afectara el funcionamiento del sensor y por consecuencia, el sistema de control del proyecto.

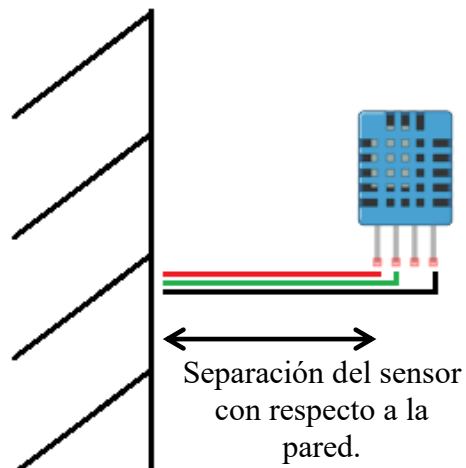


Figura 4.3-2 Afectación por condensación.

La distancia entre los sensores y tarjeta de desarrollo, fue menor a un metro y medio, y posiblemente afectase la adquisición de información, debido a la atenuación de voltaje, en este proyecto ambos sensores tiene salida digital, por lo que la atenuación de voltaje no afecta demasiado la adquisición de datos además de no haberse registrado falla o interferencia alguna. En caso que fuese analógica, debería considerarse un acondicionamiento de señal.

4.3.1 Resultados de emplazamiento de los sensores

Pese a que durante las pruebas un sensor DHT-11 fue dañado por la condensación y habiendo emplazado en el mejor lugar los sensores, el desempeño del sistema de control fue bastante bueno, la temperatura del agua, la temperatura del aire y la humedad relativa, estuvieron entre el rango de trabajo y en las pruebas realizadas los

actuadores se activaban al alcanzar el umbral mínimo, y se apagaban en el umbral máximo.

El único problema a considerar, es la temperatura en el agua que en ocasiones no es homogénea, por lo que al censar 25°C (o la temperatura del umbral máximo seleccionado) el actuador se apagaba, pero al pasar unos minutos, la temperatura en el agua se volvía homogénea siendo un par de grados centígrados más elevada que el umbral máximo.

4.4 Desempeño de la etapa de potencia y protección

En un comienzo, se utilizaron optoacopladores y TRIAC para aislar la corriente alterna de la toma eléctrica residencial (domestica) de la corriente eléctrica empleada por la tarjeta de desarrollo, el diagrama se muestra en la figura 4.4-1, la señal de entrada de 12Vdc es suministrada por un TBJ 2N2222 donde su base está conectada a un pin de salida de la tarjeta de desarrollo.

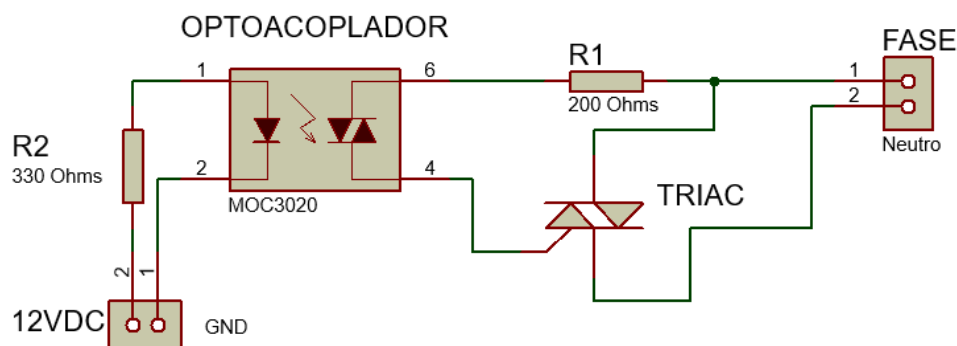


Figura 4.4-1 Diagrama etapa de potencia con optoacoplador.

En el diagrama el TRIAC corresponde a un BT139F600F con las características de poder conducir hasta 600V, una corriente máxima de 16A y un costo aproximado de 16.00 pesos mexicanos, inicialmente fue el diseño de la etapa de potencia y protección para esta propuesta experimental, pese la capacidad de conducir una gran corriente, haber energizando el actuador calefactor de aire por 40min continuos sin problemas y poder utilizar **Pulse-Width Modulation** (o modulación por ancho de pulsos y sus siglas **PWM**) para poder controlar la velocidad del ventilador o el calor producido por las resistencias de la calefacción, este diseño no fue el utilizado

finalmente, debido a que requiere disipador de calor para enfriar el TRIAC cuando conduce electricidad, y el calor podría ser excesivo para el sistema de control que no está refrigerado, añadir más elementos de refrigeración tornaría más complejos los circuitos pues se debe ubicar los disipadores de calor junto un sistema de ventilador o ventiladores, además de requerir una rejilla que permita la expulsión del aire caliente.

El segundo diseño de etapa de potencia, aprovechando que las salidas que controlan el estado de los actuadores es On/Off, no requiere compatibilidad con una señal **PWM**, se optó por utilizar contactores, el diagrama se aprecia en el capítulo 3.2.1 Diagramas esquemáticos, dicho contactor puede conducir 10A a 120Vac y energizado a 12Vdc, no requiere disipador de calor y su costo es aproximadamente 25.00 pesos mexicanos, por lo que es mucho más económico que el diseño con optoacoplador, pues se debe considerar el costo del MOC3020 o MOC3021, el TRIAC, el disipador de calor, y el posible uso de ventilador o ventiladores.

La etapa de potencia y protección a pesar de utilizar un sistema electromecánico para control, es bastante fiable, en la industria es muy común utilizar contactores como elementos de potencia, y para este proyecto ha resultado la mejor opción pues se puede conectar directamente a los transistores TBJ 2N2222 utilizados como protección de las salidas de la tarjeta de desarrollo.

4.5 Desempeño de los actuadores

El desempeño de los actuadores será proporcional al volumen de aire o agua en el que tendrán efecto, en las pruebas iniciales los actuadores fueron evaluados sin estar en el mismo sistema, o con alguno de ellos no energizado, permitiendo así observar el desempeño de cada uno, en el caso del actuador que calentó el agua. El actuador encargado de elevar la temperatura inicialmente fue de 100 Watts, pero tras varias pruebas, se determinó que no era capaz de elevar la temperatura en el volumen de 20 L, dando como consecuencia mantener el actuador encendido permanentemente,

generando un gasto eléctrico permanente. Se optó por utilizar otro actuador de mayor potencia, esta vez de 400 Watts, que finalmente pudo elevar la temperatura.

Los actuadores para calentar el aire y el atomizador para incrementar la humedad relativa, se evaluaron en 3 metros cúbicos de aire, los resultados se muestran en la tabla 4.5-1.

Tabla 4.5-1 Desempeño de los actuadores en prueba de 48 horas con condiciones reales.

	Valor inicial	Valor final	Volumen	Tiempo Aproximado
Aire	21 °C	25 °C	3 m ³	10 min
Agua	19 °C	25 °C	20 L	45 min
Humedad	40 %RH	70 %RH	3 m ³	-

En la tabla 4.5-1 se observa que el tiempo aproximado en la humedad relativa no está determinado, debido a que dependerá y de la evaporación del agua y en algunas pruebas, la humedad era cercana a 60%RH y el valor registrado al finalizar era 60%RH. Al tratarse de un ambiente evaluado aislado por una pared de hule espuma, la humedad relativa se ve afectada por el volumen de agua y la temperatura que alcanza, la evaporación se produce a temperatura ambiente y afecta a las moléculas que se encuentran en la superficie libre del líquido, por lo que al calentarse el agua, algunas moléculas se evaporan hasta que el aire (a cierta temperatura) ya no puede contener más agua y se condensa como gotas de agua, esto se conoce como punto de rocío, y afecta a este sistema, por lo que en ocasiones no es requerido el atomizador para incrementar la humedad, dado que el ambiente evaluado aislado del ambiente exterior, puede contener demasiada humedad relativa.

Esto crea un problema en el control de la humedad, dado que no se desarrolló ningún sistema deshidratador del aire, y a mayor temperatura del aire, mayor cantidad de moléculas de agua puede contener, mientras que a menor temperatura del aire menor cantidad de moléculas de agua puede contener. Considerando lo anterior, se explica la razón por la que al decrementar la temperatura del aire, se forma condensación muy rápido, lo que aumenta la humedad relativa en el sistema, es decir, una vez que se

alcanza la temperatura del umbral máximo de trabajo tanto del aire como del agua, al decrementar la temperatura la humedad relativa aumenta.

Como resultado de las evaluaciones se debe destacar que la mayor parte del tiempo de evaluación el atomizador no fue requerido para el sistema, pues la humedad relativa se mantuvo entre el 50%RH y el 75%RH lo cual es un rango permisible, dado que el hábitat natural de las iguanas se encuentra idealmente en 65%RH a temperatura promedio del día y noche de entre 20°C y 35°C.

Al utilizar un actuador para elevar la temperatura del agua descrito en el capítulo 2.7.5 Actuadores se utilizó un sistema correcto de tierra, dado que no es un actuador aislado eléctricamente, la jaula donde se realizaron las pruebas era metálica, el sistema de tierra, es conocido por igualar la diferencia de potencial eléctrico en un punto común, igualando así el voltaje existente entre el volumen de agua, la jaula y el

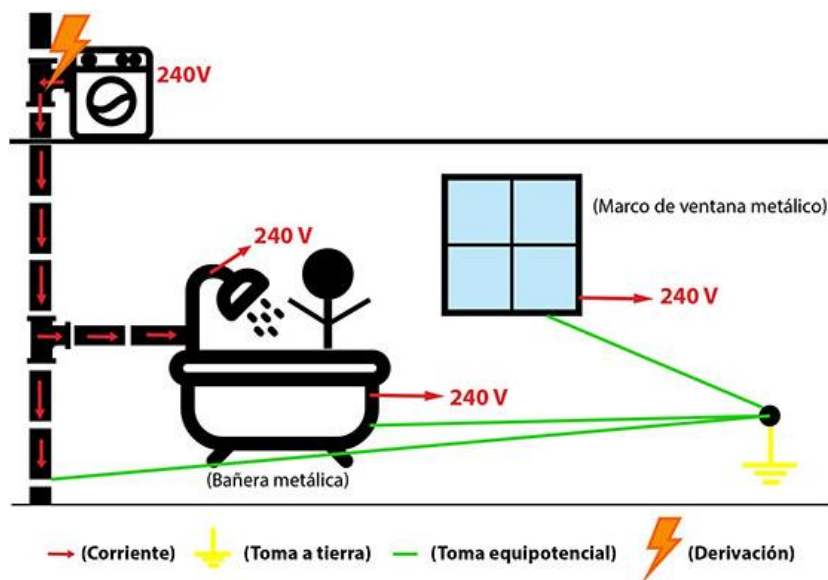


Figura 4.5-1 Esquema ducha eléctrica aterrizada utilizada en la UE.

suelo, también es conocido en duchas eléctricas de 240V, muy comunes en países donde es más barata la electricidad que usar combustible para calentar el agua, en la figura 4.5-1 se ilustra un esquema de una ducha eléctrica, el mismo principio fue requerido para este proyecto.

4.6 Desempeño del display LCD

Para poder visualizar la información en tiempo real y el registro histórico sin recurrir a un teléfono celular o un computador, se puede presionar el push button que ejecutara una función o subrutina en la tarjeta de desarrollo, en ella se ilumina el display y mediante texto, muestra la temperatura actual del agua y del aire, y la humedad relativa actual, también los niveles máximos y mínimos alcanzados durante el tiempo que la tarjeta de desarrollo esté censando datos desde los sensores.

El mayor inconveniente de esta subrutina es que durante el tiempo que es ejecutada, otras funciones no pueden ser realizadas, debido a que la subrutina momentáneamente sale del programa principal, durante ese lapso de tiempo menor a 25 segundos, las otras funciones tanto de control como las solicitadas por la aplicación Android, no pueden ser realizadas, en el caso de la prueba de actuadores será ejecutada hasta que el flujo del programa retorne al programa principal mientras que la configuración de los niveles del umbral de trabajo, no es llevada a cabo en su totalidad, pues en la Aplicación se envían los datos a escribir, mientras que en la tarjeta de desarrollo solo con almacena el ultima dato de llegada en la **USART**, siendo los caracteres “/n” que es interpretado como salto de línea.

Debido a que la duración que requiere esta subrutina relativamente es pequeña, no se detectó ningún inconveniente en el funcionamiento de control del sistema, pese a que podría retardar el encendido o apagado de los actuadores por un lapso de 25 segundos, no se detectaron problemas en los niveles superiores del umbral de trabajo. La comunicación vía Bluetooth es la más afectada por esta subrutina, debido a que la aplicación no fue diseñada para confirmar la ejecución de los datos enviados, y aparentemente son realizados si se visualiza la aplicación, contrario a lo que sucede en la tarjeta de desarrollo.

Conclusiones

El objetivo de esta propuesta experimental fue desarrollar un “Control automatizo con interfaz Android para un hábitat de reptiles” que facilite los cuidados para iguanas o

posiblemente alguna otra especie de reptil que requiera condiciones de temperatura y humedad más elevadas que las registradas en el área metropolitana de la república mexicana, así como realizar pesquisas sobre elementos de control que se pudieran adecuar a un caso práctico con elementos doméstico o que su adquisición fuera posible para todo público.

Así pues la aportación principal de este trabajo, está en la descripción de un sistema simple que resuelva un problema haciendo uso de electrónica, sin requerir elementos costosos, y que pueda ser utilizado o replicado por casi cualquier persona.

Esta propuesta experimental evaluó el desempeño del sistema desarrollado en condiciones exteriores reales, primeramente elevando y manteniendo la temperatura del agua, seguidamente elevando y manteniendo la temperatura del aire, finalmente los actuadores fueron evaluados en conjunto, de esta forma se evaluó el desempeño de cada actuador y en conjunto. Haciendo con esto, un sistema fiable y confiable para los usuarios, evitando la posibilidad de una descarga eléctrica. El resultado de este trabajo es por tanto, el diseño de tres bloques fundamentales:

- ◆ Interfaz de usuario
- ◆ Etapa de potencia y aislamiento
- ◆ Etapa de control

Todos los circuitos y etapas fueron elaborados en la protoboard, elaborar el circuito PCB requirió dedicar tiempo, pero los diagramas son proporcionados para poder ser reproducidas fácilmente (recordando que deben medir 100mm X 100mm), pero sin dudas, las pruebas realizadas fueron la parte de mayor enriquecimiento, dado que todos los problemas fueron motivo para mejorar el sistema presentado, así como implementar mejoras en la adquisición de datos o telemetría como posible trabajo futuro.

Con respecto a trabajo futuro, es importante resaltar que en la actualidad muchos sistemas se apoyan en el internet para compartir información, por otro lado, muchos sistemas utilizan comunicación satelital para poder tener alcance en los

sistemas donde no existe comunicación telefónica o móvil para poder emplearla como medio de transmisión, para este caso es posible utilizar un sistema GSM para poder enviar y recibir información de manera remota sin requerir estar a unos metros de distancia. También se puede realizar investigaciones sobre el aislamiento térmico del terrario mejorando así la eficiencia del sistema, evitando ceder calor al medio ambiente donde se encuentre dicho terrario. También podría utilizarse el pin sin asignar, que en los diagramas PCB ya se ha realizado la etapa de potencia y aislamiento basta con realizar el código correspondiente, para poder utilizar otro actuador, en una etapa muy temprana se había propuesto utilizar dicho pin para iluminación artificial, pero fue descartada esta idea, tras haber investigado que se requiere, para el bienestar del reptil en cuestión, radiaciones propias de la luz natural y replicarlas o bien podría ser costoso pues dichos dispositivos como bombillas ultravioletas, o eléctricamente representarían un gasto energético muy alto, inadecuado para un uso doméstico. Así mismo durante la realización de la parte final de este proyecto, se pensó en utilizar el reloj de tiempo real para poder activar o no determinados actuadores según la hora del día, sin importar si las condiciones climáticas eran las mejores, esto para poder optimizar el funcionamiento del sistema, dado podrían censarse niveles inferiores a los mínimos del umbral durante un instante y activar los actuadores pese a que o fuese totalmente requerido, se pensó en llamarlo “Modo Nocturno” o “Modo Ahorro de Energía” e implementar un botón en la aplicación que permitiese activarlo o desactivarlo, así como conocer el estado de dicho modo.

Los conocimientos adquiridos durante la realización de este proyecto como una concepción muy básica, que duró poco más de un año desde la tutoría del Dr. Ismael Díaz Rangel, se debió superar varios problemas, como consultas de practicarios del laboratorio de Electrónica de Potencia, hasta idear algoritmos que tuvieran como función principal, asegurar el bienestar del usuario y seguidamente realizar un sistema eficiente. Inicialmente el proyecto comenzó como un proyecto de fin de curso de la asignatura impartida por el Dr. Ismael Díaz Rangel, como una tarjeta de desarrollo Arduino UNO con un sensor DS18B320 y un LED que representaba el actuador energizado o no, para finalizar poco más de un año, con un sistema que puede

comunicarse con el teléfono móvil de manera inalámbrica y emplear actuadores con una potencia eléctrica significativa, si recurrimos a los electrodomésticos como referencia.

Personalmente me siento satisfecho con el desempeño del proyecto, pues facilita los cuidados de una iguana, aunque no es un sistema totalmente autónomo, que pueda realizar limpieza del terrario o que pueda alimentar automáticamente al reptil, proporciona la temperatura adecuada en estaciones del año que no son adecuadas para las iguanas, así como facilita el monitoreo del sistema, además de poder ser fácilmente configurable para que actué en el umbral que considero más adecuado en base a la sección 2.1 Fisiología de las iguanas. Este proyecto requirió de mucho tiempo, en el he puesto mucho empeño y esfuerzo, no solo a que es una trámite para titulación, sino también a que es un sistema que utilizare y que el reptil podrá gozar de tener calefacción y agua caliente las 24 horas del día en invierno.

Bibliografía

- Actuadores

Díaz, M. (2013). ¿Qué son los actuadores electrónicos?

<https://www.330ohms.com/blogs/blog/85020868-que-son-los-actuadores-electronicos>

Gonzales, V. (2002). Actuadores.

http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/actuadores.htm

- Arduino UNO

Arduino. (2016). Características Arduino UNO y Geniuno UNO.

<https://www.arduino.cc/en/Main/ArduinoBoardUno>

Arduino. (2016). EEPROM Write

<https://www.arduino.cc/en/Tutorial/EEPROMWrite>

Arduino. (2016). analogWrite().

<https://www.arduino.cc/en/Reference/AnalogWrite>

Arduino. (2016). map(value, fromLow, fromHigh, toLow, toHigh).

<https://www.arduino.cc/en/Reference/Map>

- Conductores

Viakon Conductores Monterrey (2012). Cables para la Construcción (Baja Tensión).

<http://www.viakon.com/pdf/categorias/24.pdf>

- Conceptos medición e instrumentación

Rodríguez, M. (2015). Transductores y sensores: propiedades y características técnicas.

<http://revistadigital.inesem.es/gestion-integrada/caracteristicas-tecnicas-transductores-sensores/>

- Contactores

Naranjo, F. (2013). Proyecto - Relé control lámpara.

<http://electroprofano.blogspot.mx/2013/01/proyecto-relee-control-lampara.html>

- EEPROM

Marone, J. (s/f). Microcontroladores de Atmel.

http://www.exa.unicen.edu.ar/catedras/tmicrocon/Material/3_Overview_Microcontroladores_ATMEL.pdf

Pérez. A. (s/f). Arduino: Tipos de memoria en el microcontrolador de Arduino, Flash, SRAM y EEPROM.

<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=571>

- Fisiología de las iguanas

Anatomía y Morfología de la Iguana Verde. (s/f).

<http://todoiguanasverdes.jimdo.com/01-qu%C3%A9-es-una-iguana/c-anatom%C3%ADa-y-morfolog%C3%ADa/>

Pérez, E. (s/f). La iguana: Clasificación y anatomía.

<http://villaiгуana.blogspot.mx/2011/12/la-iguana-clasificacion-y-anatomia.html>

Pinacho, B., Arcos-García, J., López-Pozos, R., Mendoza, G. & Plata, F. (2010). Parámetros reproductivos de la iguana verde (*Iguana iguana*) en condiciones de cautiverio en Oaxaca, México. *Revista Científica*, vol. XX, núm. 5, septiembre-octubre, (paginas 467-472) Universidad del Zulia, Maracaibo, Venezuela.

Temperatura. (s/f).

<http://todoiguanasverdes.jimdo.com/04-el-terrario-y-sus-accesorios/c-puntos-importantes-del-terrario/temperatua/>

- Fuentes de alimentación

Departamento de Electricidad, Electrónica y Computación. (2014). Regulador de tensión conmutado (fuente switching). Facultad de Ciencias Exactas y Tecnología de la Universidad Nacional de Tucumán, Tucumán, Argentina.

[http://www.herrera.unt.edu.ar/de/material/apuntes/20%20y%2021%20clase%20ftes%20conm\(switching\).pdf](http://www.herrera.unt.edu.ar/de/material/apuntes/20%20y%2021%20clase%20ftes%20conm(switching).pdf)

IHateMyself. (2013). Tipos de fuentes de alimentación.

<http://tiposdefuentesdealimentacion.blogspot.mx/2011/05/tipos-de-fuentes-de-alimentacion.html>

Microcontroladores

¿Qué es un microcontrolador? (s/f).

<http://www.electronicaestudio.com/microcontrolador.htm>

Microcontrolador. (s/f).

<http://microcontroladores-e.galeon.com/>

- MIT App inventor 2

Massachusetts Institute of Technology (2012). Getting Started with MIT App Inventor 2. <http://appinventor.mit.edu/explore/get-started.html>

- Optoacoplador

Prácticas de Laboratorio de Electrónica de Potencia, (s/f). FES Aragón, UNAM.

<http://chuviscar.angelfire.com/Potencia/Siete.pdf>

- PLC

EkoPLC. (2003). QUÉ ES PLC.

<http://www.ekoplac.net/que-es-plc/index.htm>

- Protecciones/Instalaciones

Carrillo, G. (2007). Protecciones Eléctricas.

<http://ocw.uis.edu.co/ingenieria-electrica/protecciones-electricas/material-de-clase-1/libroproteccionesgcc.pdf>

Escuela Universitaria de Ingeniería Técnica Agrícola de Ciudad Real. (s/f).

Protecciones eléctricas.

https://www.uclm.es/area/ing_rural/Instalaciones/Protecciones.pdf

Protección de los circuitos eléctricos. (s/f).

http://www.profesormolina.com.ar/electromec/prot_circ_elect.htm

Schneider Electric. (2009). Protección de las instalaciones eléctricas. *ElectriQO Actualizando al profesional electricista*. Instituto Schneider de Schneider Electric México S.A de C.V. México.

http://www.schneider-electric.com.mx/documents/soporte/electriqo-magazine/electri-qo_vol02.pdf

▪ Sensores

Arduino. (2016). Dallas Semiconductor's 1-Wire Protocol.

<http://playground.arduino.cc/Learning/OneWire>

Arduino One Wire Digital Temperature Sensor DS18B20. (2016).

<http://www.hobbytronics.co.uk/ds18b20-arduino>

D-Robotics. (2010). DHT11 Humidity & Temperature Sensor.

<http://www.micropik.com/PDF/dht11.pdf>

González F. (2014). LM335: Sensor de temperatura para Arduino.

<http://panamahitek.com/lm335-sensor-de-temperatura-para-arduino/>

High Sensitivity Moisture Sensor. (s/f).

http://www.fut-electronics.com/wp-content/plugins/fe_downloads/Uploads/moisture-sensor-arduino.pdf

Maxim Integrated. (2015). DS18B20 Programmable Resolution 1-Wire Digital Thermometer.

<https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

Microchip. (2016). TCN75A.

<http://www.microchip.com/wwwproducts/en/TCN75A>

Reloj con Arduino y el módulo Tiny RTC I2C. (2013).

<http://zygzax.com/reloj-con-arduino-y-el-modulo-tiny-rtc-i2c/>

Sensor óptico reflectivo Infrarrojo TCRT5000L. (2016).

<http://electronilab.co/tienda/tcrt5000l-sensor-optico-reflectivo-infrarrojo/>

Studt, J. (2007). OneWire Library.

https://www.pjrc.com/teensy/td_libs_OneWire.html

Texas Instruments. (2016). LM35 Precision Centigrade Temperature Sensors.

<http://www.ti.com/lit/ds/symlink/lm35.pdf>

Vishay. (2009). Reflective Optical Sensor with Transistor Output.

<http://www.vishay.com/docs/83760/tcrt5000.pdf>

- Sistemas de calefacción

Devis, R. (2014). Tipos de calefacción: gas, eléctrica y bomba de calor.

<http://www.rankia.mx/blog/luz-y-gas/2552235-tipos-calefaccion-gas-electrica-radiante-bomba-calor>

- Sistemas de control

Acebes, L. (s/f). Introducción a los sistemas de control.

http://www.isa.cie.uva.es/~felipe/docencia/ra12itielec/tema1_trasp.pdf

Sistemas de control. (s/f).

<http://www.tecnologia-tecnica.com.ar/sistemadecontrol/index%20sistemasdecontrol.htm>

Universidad de Alcalá. (s/f). Estabilidad de los sistemas de control.

<https://www.depeca.uah.es/depeca/repositorio/asignaturas/20613/ApendiceTema1.pdf>

- Transistor 2N2222

Fairchild. (2004). PN2222A.

<https://www.fairchildsemi.com/datasheets/PN/PN2222A.pdf>

- TRIAC

Philips Semiconductors. (1997). Triacs BT139 series.

<http://pdf.datasheetcatalog.com/datasheet/philips/BT139-600.pdf>

Anexos

Código Arduino

```
#include <EEPROM.h>  
#include "DHT.h"  
#include <Wire.h>  
#include "RTCLib.h"
```

```
#include <OneWire.h>  
#include <DallasTemperature.h>  
#include <LiquidCrystal.h>  
#define DHTPIN 6
```

```
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
#define ONE_WIRE_BUS 7
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
RTC_DS1307 RTC;
LiquidCrystal lcd(12, 11, 5, 4, 3, A0);

int nivel=A2;
int hMax;
int hMin;
int tMax;
int tMin;
int taMax;
int taMin;
int SetM;
int SetH;
char* textnivel;
int pin2=2;
int backLight = A1;
int vala;
int val;
int valh;
float X;
float Y;
float Z;
int leeprom;
int a;
float taRef;
int tRef;
int hRef;
float taCom;
int tCom;
int hCom;
float taReg;
float tReg;
float hReg;
float ta;
int hora;
int minu;
int segu;
float h;
float t;
float taComMax;
int tComMax;
int hComMax;
float taRegMax;

float tRegMax;
float hRegMax;
float nan;

void setup()
{
//librerias del DHT 11
/* LCD RS pin to digital pin 12
 * LCD Enable pin to digital pin 11
 * LCD D4 pin to digital pin 5
 * LCD D5 pin to digital pin 4
 * LCD D6 pin to digital pin 3
 * LCD D7 pin to digital pin 2
 */
// set up the LCD's number of columns
and rows:
pinMode(13, OUTPUT);
pinMode(A3, OUTPUT);
pinMode(10,OUTPUT);
pinMode(9, OUTPUT);
pinMode(8, OUTPUT);
pinMode(2, INPUT);
pinMode(backLight, OUTPUT);

hMax=EEPROM.read(4);
hMin=EEPROM.read(5);
tMax=EEPROM.read(2);
tMin=EEPROM.read(3);
taMax=EEPROM.read(0);
taMin=EEPROM.read(1);
SetM=0;
SetH=7;
nivel=0;
nan=nan;
textnivel=textnivel;

lcd.begin(16, 2);
lcd.clear();
digitalWrite(backLight, HIGH);
// Print a message to the LCD.
lcd.print("Chales mijos!");
delay(2000);
lcd.clear();
digitalWrite(backLight, LOW);
Serial.begin(9600); //Se inicia la
comunicación serial
```

```
dht.begin(); //Se inicia el sensor DHT11
//Serial.println("Proyecto
Automatizacion Terrario");
sensors.begin(); //sensor sumergible

//RTC
Wire.begin();
RTC.begin();
//Si quitamos el comentario de la linea
siguiente, se ajusta la hora y la fecha con
la del ordenador
//RTC.adjust(DateTime(__DATE__,
__TIME__));
taReg=50;
tReg=50;
hReg=110;
taRegMax=0;
tRegMax=0;
hRegMax=0;

}

void loop(){
// Enviar datos solo cuando recibe datos:

    //if (Serial.available() > 0) {
    if(Serial.peek() == 'T'){
        digitalWrite(8,0);
        digitalWrite(9,0);
        digitalWrite(10,0);
        digitalWrite(A3,0);
        digitalWrite(13,0);
        Serial.println();
        Serial.print("Prueba actuadores
        2 segundos");
        lcd.clear();
        digitalWrite(backLight, HIGH);
        lcd.setCursor(0,0);
        lcd.print("Check actuadores");
        lcd.setCursor(0,1);
        lcd.print(" 2 segundos");
        digitalWrite(8,1);
        digitalWrite(9,1);
        digitalWrite(A3,1);
        digitalWrite(13,1);
        delay(1500);
        Serial.println();

        Serial.print("Prueba actuadores
        1 segundo");
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Check actuadores");
        lcd.setCursor(0,1);
        lcd.print(" 1 segundo");
        delay(1500);
        Serial.println();
        Serial.print("Prueba actuadores
        terminada");
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Check actuadores");
        lcd.setCursor(0,1);
        lcd.print("Terminada");
        delay(1500);
        lcd.clear();
        digitalWrite(backLight, LOW);
        digitalWrite(8,0);
        digitalWrite(9,0);
        digitalWrite(A3,0);
        digitalWrite(13,0);
    }
    if(Serial.peek() == 'A'){
        Serial.read();
        taMax = Serial.parseInt();
        EEPROM.write(0, taMax);
        Serial.println();
        Serial.print("Establecida Temp
        Agua Max: ");
        Serial.println(taMax);
    }
    if(Serial.peek() == 'a'){
        Serial.read();
        taMin = Serial.parseInt();
        EEPROM.write(1, taMin);
        Serial.println();
        Serial.print("Establecida Temp
        Agua Min: ");
        Serial.println(taMin);
    }
    if(Serial.peek() == 'B'){
        Serial.read();
        tMax = Serial.parseInt();
        EEPROM.write(2, tMax);
        Serial.println();
    }
}
```

```

Serial.print("Establecida Temp
Ambiente Max: ");
Serial.println(tMax);
}
if(Serial.peek() == 'b'){
Serial.read();
tMin = Serial.parseInt();
EEPROM.write(3, tMin);
Serial.println();
Serial.print("Establecida Temp
Ambiente Min: ");
Serial.println(tMin);
}
if(Serial.peek() == 'C'){
Serial.read();
hMax = Serial.parseInt();
EEPROM.write(4, hMax);
Serial.println();
Serial.print("Establecida
Humedad Max: ");
Serial.println(hMax);
}
if(Serial.peek() == 'c'){
Serial.read();
hMin = Serial.parseInt();
EEPROM.write(5, hMin);
Serial.println();
Serial.print("Establecida
Humedad Min: ");
Serial.println(hMin);
}
if(Serial.peek() == 'D'){
Serial.read();
SetH = Serial.parseInt();
Serial.println();
Serial.print("Establecida Hora:
");
Serial.println(SetH);
}
if(Serial.peek() == 'd'){
Serial.read();
SetM = Serial.parseInt();
Serial.println();
Serial.print("Establecidos
Minutos: ");
Serial.println(SetM);
}

if(Serial.peek() == 'e'){
Serial.read();
a = Serial.parseInt();
Serial.println();
Serial.print("Lectura EEPROM
con direccion: ");
Serial.println(a);
if(a==20){
leeprom=EEPROM.read(20);
Serial.println (leeprom);
leeprom=EEPROM.read(21);
Serial.println (leeprom);
leeprom=EEPROM.read(22);
Serial.println (leeprom);
}
else if(a==30){
leeprom=EEPROM.read(30);
Serial.println (leeprom);
leeprom=EEPROM.read(31);
Serial.println (leeprom);
leeprom=EEPROM.read(32);
Serial.println (leeprom);
}
else{
leeprom=EEPROM.read(a);
Serial.println (leeprom);
}
}

//if(Serial.peek() == 'E'){
Serial.println();
Serial.println(EEPROM.read(0)+10000);
delay(100);
Serial.println(EEPROM.read(1)+20000);
delay(100);
Serial.println(EEPROM.read(2)+30000);
delay(100);
Serial.println(EEPROM.read(3)+40000);
delay(100);
Serial.println(EEPROM.read(4)+50000);
delay(100);
Serial.println(EEPROM.read(5)+60000);

//Serial.println("Lectura EEPROM");
//}

while (Serial.available(>0){
Serial.read();

```



```

}
Serial.println();
//-----}

DateTime now = RTC.now();

pin2=digitalRead(2);
if (pin2== HIGH)
{
  Display();
}

//Solicitando Temperatura Agua
sensors.requestTemperatures();
ta = sensors.getTempCByIndex(0); // se
  lee temp agua
hora = now.hour(); //se lee la hora
minu = now.minute(); // se lee los
  minutos
segu = now.second(); // se lee los
  minutos
h = dht.readHumidity(); //se lee la
  humedad
t = dht.readTemperature(); // se lee la
  temperatura amb
nivel=digitalRead(A2);

//Nivel Agua
if (nivel == HIGH){
  textnivel = "Insuficiente";
  Serial.println("NEmp");
  digitalWrite(backLight, HIGH);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Nivel Agua");
  lcd.setCursor(0,1);
  lcd.print("Insuficiente");
  delay(400);
  digitalWrite(backLight, LOW);
  delay(100);
  digitalWrite(backLight, HIGH);
  delay(400);
  lcd.clear();
  digitalWrite(backLight, LOW);
}
if (nivel == LOW){
  textnivel = "Adecuado";
  Serial.println("NFull");
}

//Adquisicion Referencia
taRef=ta;
tRef=t;
hRef=h;

//Control Humedad

if(h<=hMin){ //Humedad menor a
Humedad Minima
  if(h!=nan){
    digitalWrite(A3,1);
  }
  else{
    digitalWrite(A3,0);
  }
}
if(h>=hMax){ //Humedad mayor a
Humedad Maxima
  digitalWrite(A3,0);
}
if(digitalRead(A3)== HIGH){
  Serial.println("HOn");
}
if(digitalRead(A3)== LOW){
  Serial.println("HOff");
}

//Control Temp Ambiente

if(t<=tMin){
if(t!=nan){
  digitalWrite(8,1);
}
}
else{
  digitalWrite(8,0);
}
}
if(t>=tMax){
  digitalWrite(8,0);
}
}
if(digitalRead(8)== HIGH){
  Serial.println("BOn");
}
}
if(digitalRead(8)== LOW){

```

```

Serial.println("BOff");
}
//Control Temp Agua

if(ta<=taMin){
  if(ta!=-127){
    if (nivel == LOW)
    {
      digitalWrite(9,1);
    }
    else{
      digitalWrite(9,0);
    }
  }
  else{
    digitalWrite(9,0);
  }
}
if(ta>=taMax){
  digitalWrite(9,0);
}
if(digitalRead(9)== HIGH){
  Serial.println("AOn");
}
if(digitalRead(9)== LOW){
  Serial.println("AOff");
}

//Control Iluminacion
if(hora<SetH && minu<SetM ){
  digitalWrite(13,1);
}
if(hora<SetH && minu>SetM ){
  digitalWrite(13,0);
}
if(hora>SetH && minu<SetM ){
  digitalWrite(13,0);
}
if(hora>SetH && minu>SetM ){
  digitalWrite(13,0);
}

pin2=digitalRead(2);
if (pin2== HIGH)

```

```

{
  Display();
}

if (((h/h)!=1 ) || ((t/t!=1)))
{
  digitalWrite(A3, LOW);
  Serial.println("HOff");
  digitalWrite(8,LOW);
  Serial.println("BOff");
  Serial.println("Perdida comunicacion
sensor temperatura/humedad");
  digitalWrite(backLight, HIGH);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Sensor sin");
  lcd.setCursor(0,1);
  lcd.print("comunicacion");
  delay(300);
  digitalWrite(backLight, LOW);
  delay(100);
  digitalWrite(backLight, HIGH);
  delay(300);
  lcd.clear();
  digitalWrite(backLight, LOW);
}

if (ta== -127)
{
  Serial.println("Perdida comunicacion
sensor sumergible DS18B20");
  digitalWrite(backLight, HIGH);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Sensor sin");
  lcd.setCursor(0,1);
  lcd.print("comunicacion");
  delay(300);
  digitalWrite(backLight, LOW);
  delay(100);
  digitalWrite(backLight, HIGH);
  delay(300);
  lcd.clear();
  digitalWrite(backLight, LOW);
}

```

```

//Adquisicion Comparacion
sensors.requestTemperatures();
ta = sensors.getTempCByIndex(0); //
se lee temp agua
h = dht.readHumidity(); //se lee la
humedad
t = dht.readTemperature(); // se lee la
temperatura amb

//Registro Datos Minimios
taCom=ta;
if(ta>-127){
  if(taRef<taReg)
  {
    if (taRef<taCom)
    {
      taReg=taRef;
    }
    if (taRef>taCom);
    {
      taReg=taCom;
    }
  }
}

tCom=t;
if(t>0){
  if(tRef<tReg)
  {
    if (tRef<tCom)
    {
      tReg=tRef;
    }
    if (tRef>tCom);
    {
      tReg=tCom;
    }
  }
}

hCom=h;
if(h>0){
  if(hRef<hReg)
  {
    if (hRef<hCom)
    {
      hReg=hRef;
    }
    if (hRef>hCom);
    {
      hReg=hCom;
    }
  }
}

pin2=digitalRead(2);
if (pin2== HIGH)
{
  Display();
}

//Registros Datos Maximios
taComMax=ta;
if(taRef>taRegMax)
{
  if (taRef>taComMax)
  {
    taRegMax=taRef;
  }
  if (taRef<taComMax);
  {
    taRegMax=taComMax;
  }
}

tComMax=t;
if(tRef>tRegMax)
{
  if (tRef>tComMax)
  {
    tRegMax=tRef;
  }
  if (tRef<tComMax);
  {
    tRegMax=tComMax;
  }
}

hComMax=h;
if(hRef>hRegMax)

```

```

{
  if (hRef>hComMax)
  {
    hRegMax=hRef;
  }
  if (hRef<hComMax);
  {
    hRegMax=hComMax;
  }
}

//Serial.print("A");

Serial.println(sensors.getTempCByIndex(0
)+1500);
delay(100);
//Serial.print("B");
Serial.println(t+2500);
delay(100);
//Serial.print("C");
Serial.println(h+3500);
delay(100);
Serial.println(taReg+4500);
delay(100);
Serial.println(taRegMax+5500);
delay(100);
Serial.println(tReg+6500);
delay(100);
Serial.println(tRegMax+7500);
delay(100);
Serial.println(hReg+8500);
delay(100);
Serial.println(hRegMax+9500);
delay(100);
Serial.println();
delay(100);

//Imprimi el dia
Serial.print(now.day(), DEC);
Serial.print('/');
//Imprimir el mes
Serial.print(now.month(), DEC);
Serial.print('/');
//Imprimir el año
Serial.print(now.year(), DEC);
Serial.print(' ');
//Imprimir la hora

Serial.print(now.hour(), DEC);
Serial.print(':');
//Imprimir los minutos
Serial.print(now.minute(), DEC);
Serial.print(':');
//Imprimir los segundos
Serial.print(now.second(), DEC);
Serial.println();
}

void Display (){
digitalWrite(backLight, HIGH);
lcd.clear();
lcd.setCursor(0,0);
Wire.begin();
RTC.begin();
DateTime now = RTC.now();

ta= sensors.getTempCByIndex(0);
hora = now.hour();
minu = now.minute();
h = dht.readHumidity();
t= dht.readTemperature();

//LCD imprime temperatura

for(int i=300; i>=0; i=i-1)
{
  lcd.setCursor(0,0);
  lcd.print("Hora:");
  lcd.setCursor(0,1);
  lcd.print(hora);
  lcd.setCursor(2,1);
  lcd.print(":");
  lcd.setCursor(3,1);
  lcd.print(minu);
  lcd.setCursor(5,1);
  lcd.print(" ");
}

lcd.clear();
for(int i=350; i>=0; i=i-1)
{
  lcd.setCursor(0,0);

```

```

        lcd.print("Temp Aire");
        lcd.setCursor(0,1);
        lcd.print(t);
        lcd.setCursor(6,1);
        lcd.print("C");
    }

    lcd.clear();
    for(int i=3; i>=0; i=i-1)
    {
        lcd.setCursor(0,0);
        lcd.print("Temp Agua");
        lcd.setCursor(0,1);
        lcd.print(ta);
        lcd.setCursor(6,1);
        lcd.print("C");
        sensors.requestTemperatures();
        ta= sensors.getTempCByIndex(0);
        lcd.setCursor(0,1);
        lcd.print(ta);
        lcd.noCursor();
    }

    lcd.clear();
    for(int i=350; i>=0; i=i-1)
    {
        lcd.setCursor(0,0);
        lcd.print("Humedad");
        lcd.setCursor(0,1);
        lcd.print(h);
        lcd.setCursor(6,1);
        lcd.print("%");
    }

    lcd.clear();
    for(int i=350; i>=0; i=i-1)
    {
        lcd.setCursor(0,0);
        lcd.print("Nivel de Agua");
        lcd.setCursor(0,1);
        lcd.print(textnivel);
    }

    lcd.clear();
    for(int i=400; i>=0; i=i-1)
    {
        lcd.setCursor(0,0);
        lcd.print("Temp Agua MinMax");
        lcd.setCursor(0,1);
        lcd.print(taReg);
        lcd.setCursor(6,1);
        lcd.print("C");
        lcd.setCursor(8,1);
        lcd.print(taRegMax);
        lcd.setCursor(14,1);
        lcd.print("C");
    }

    lcd.clear();
    for(int i=350; i>=0; i=i-1)
    {
        lcd.setCursor(0,0);
        lcd.print("Temp Aire MinMax");
        lcd.setCursor(0,1);
        lcd.print(tReg);
        lcd.setCursor(6,1);
        lcd.print("C");
        lcd.setCursor(8,1);
        lcd.print(tRegMax);
        lcd.setCursor(14,1);
        lcd.print("C");
    }

    lcd.clear();
    for(int i=350; i>=0; i=i-1)
    {
        lcd.setCursor(0,0);
        lcd.print("Humedad MinMax");
        lcd.setCursor(0,1);
        lcd.print(hReg);
        lcd.setCursor(6,1);
        lcd.print("%");
        lcd.setCursor(8,1);
        lcd.print(hRegMax);
        lcd.setCursor(14,1);
        lcd.print("%");
    }

    lcd.clear();

    digitalWrite(backLight, LOW);
    }
}

```

Código MIT App Inventor 2

```
when Screen1.Initialize
do
  set LabelTempAguaMin.Text to round(SliderTempAguaMin.ThumbPosition)
  set LabelTempAguaMax.Text to round(SliderTempAguaMax.ThumbPosition)
  set LabelTempAireMax.Text to round(SliderTempAireMax.ThumbPosition)
  set LabelTempAireMin.Text to round(SliderTempAireMin.ThumbPosition)
  set LabelHumedadMax.Text to round(SliderHumedadMax.ThumbPosition)
  set LabelHumedadMin.Text to round(SliderHumedadMin.ThumbPosition)
  if not BluetoothClient1.Enabled
  then
    call Notifier1.ShowAlert
    notice "Bluetooth No Activado"

initialize global delaytime to 0

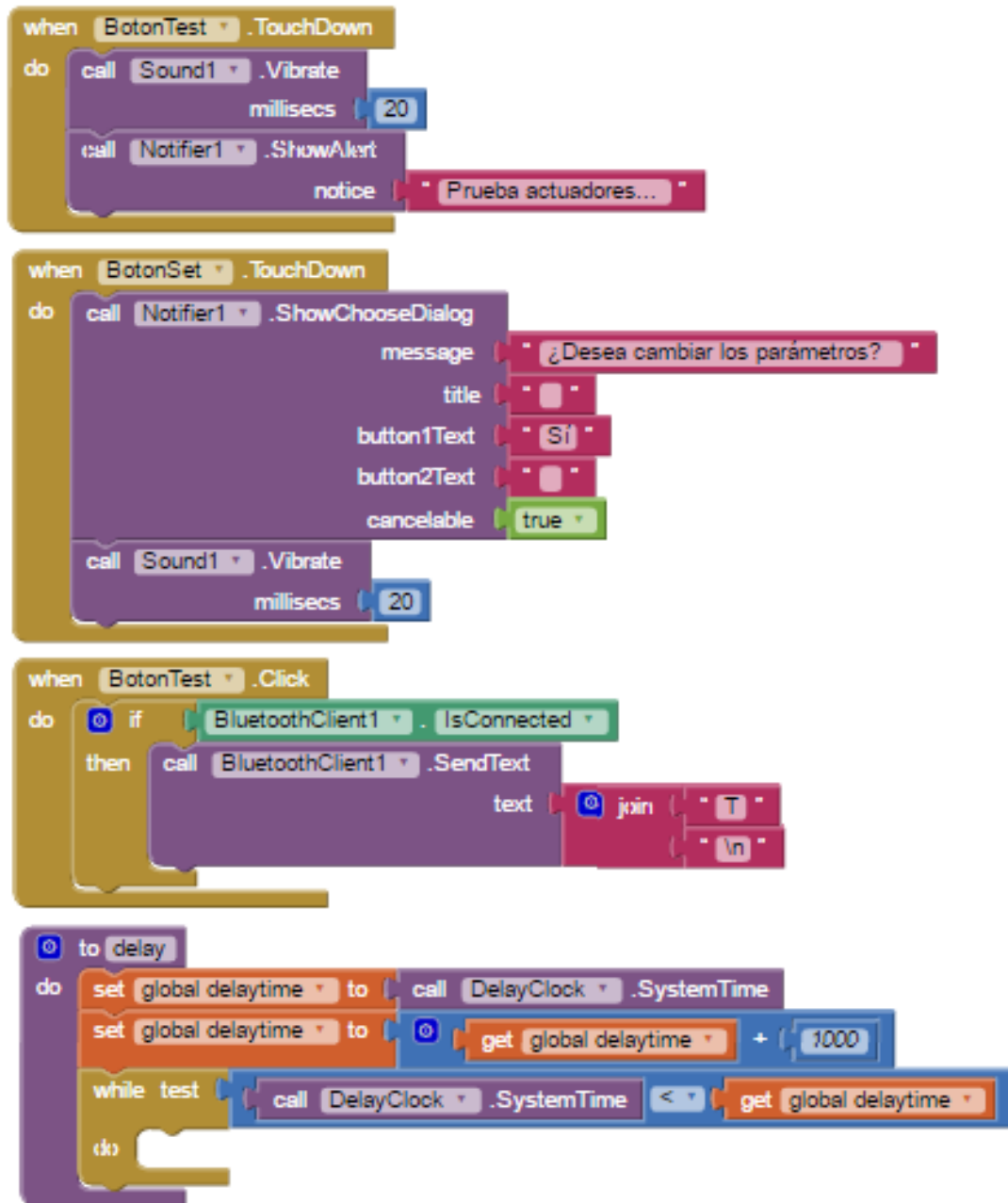
when Ligar.BeforePicking
do
  if BluetoothClient1.Available
  then
    set Ligar.Elements to BluetoothClient1.AddressesAndNames
```

Inializacion sliders, delay y conexión Bluetooth.

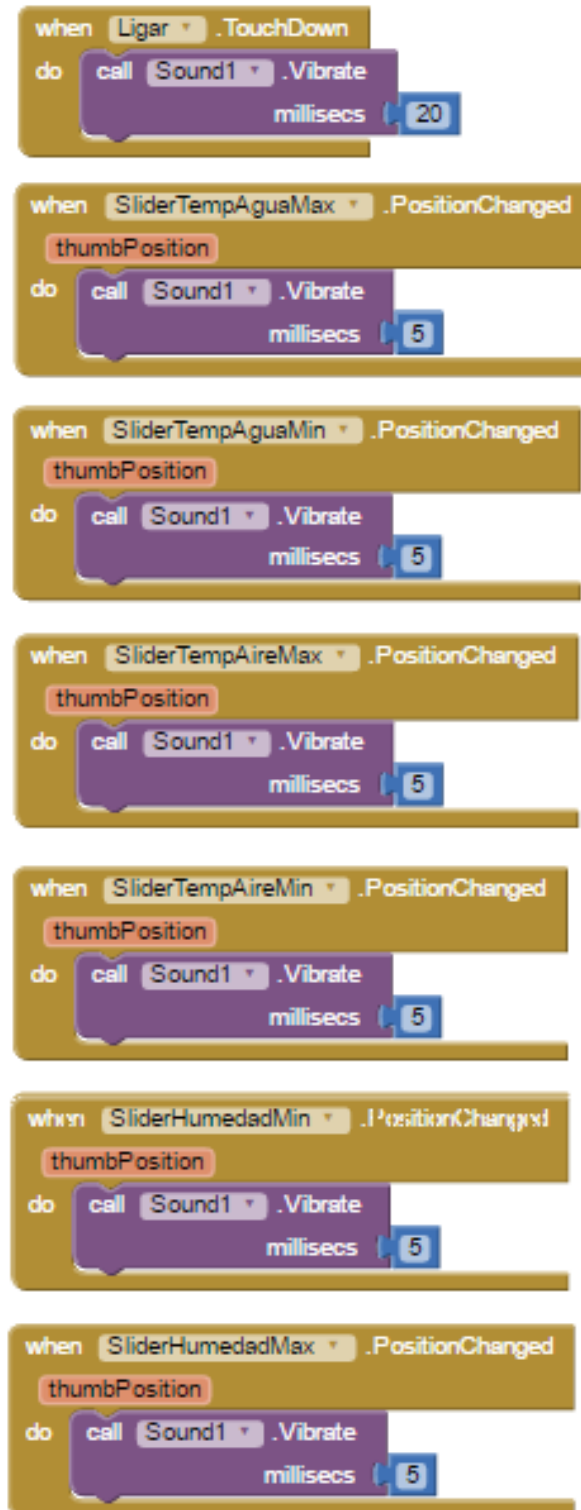
```
when Ligar .AfterPicking
do
  evaluate but ignore result call BluetoothClient1 .Connect
  address Ligar . Selection
  if BluetoothClient1 . IsConnected
  then
    set Ligar . Text to "Conectado"
    set Ligar . BackgroundColor to blue
  if not BluetoothClient1 . IsConnected or not BluetoothClient1 . Enabled
  then
    set Ligar . Text to "Desconectado"
    set Ligar . BackgroundColor to orange

when Clock1 .Timer
do
  set LabelTempAguaMin . Text to round SliderTempAguaMin . ThumbPosition
  set LabelTempAguaMax . Text to round SliderTempAguaMax . ThumbPosition
  set LabelTempAireMax . Text to round SliderTempAireMax . ThumbPosition
  set LabelTempAireMin . Text to round SliderTempAireMin . ThumbPosition
  set LabelHumedadMax . Text to round SliderHumedadMax . ThumbPosition
  set LabelHumedadMin . Text to round SliderHumedadMin . ThumbPosition
```

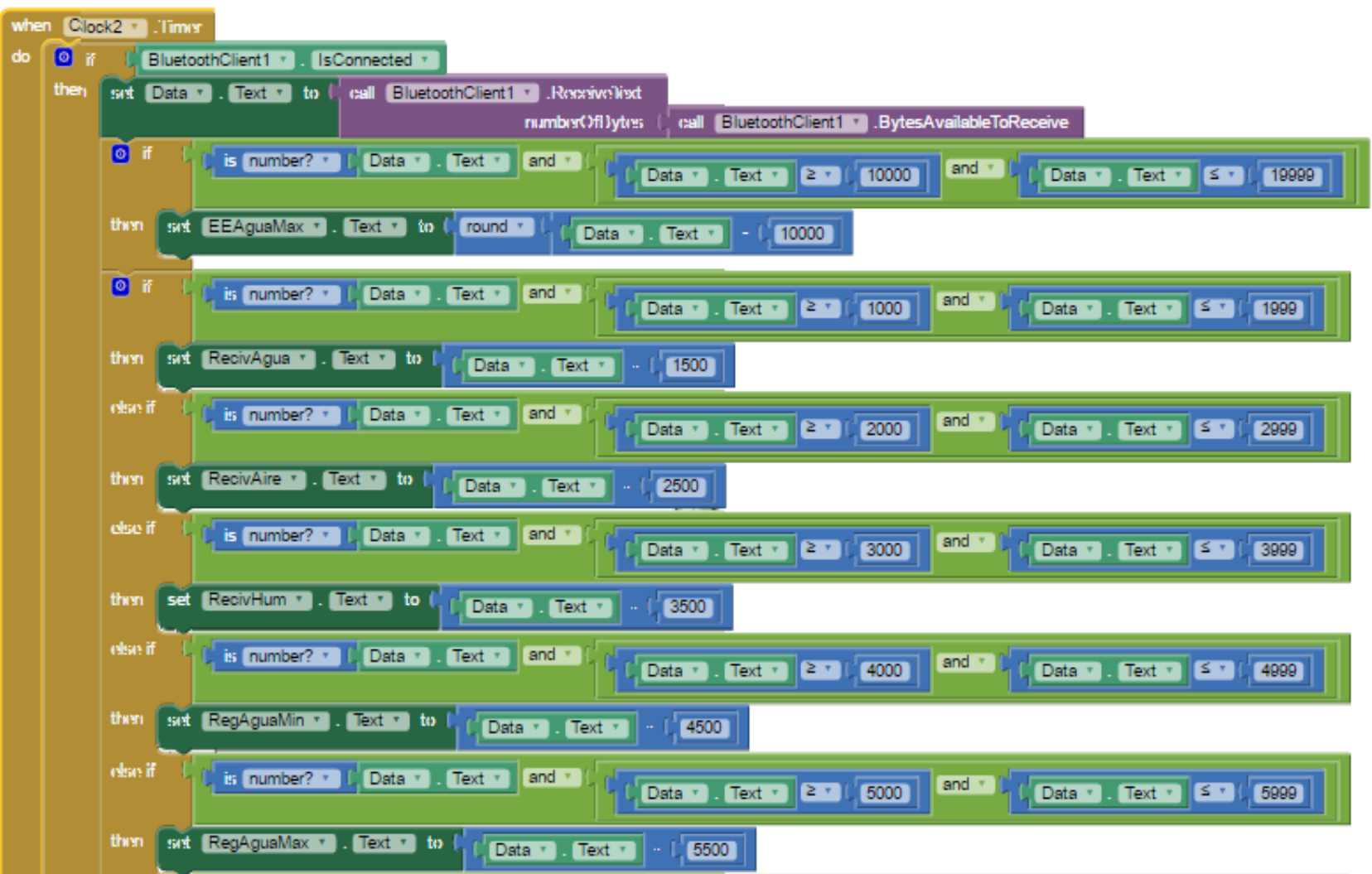
Ligación Bluetooth y adquisición valores sliders.



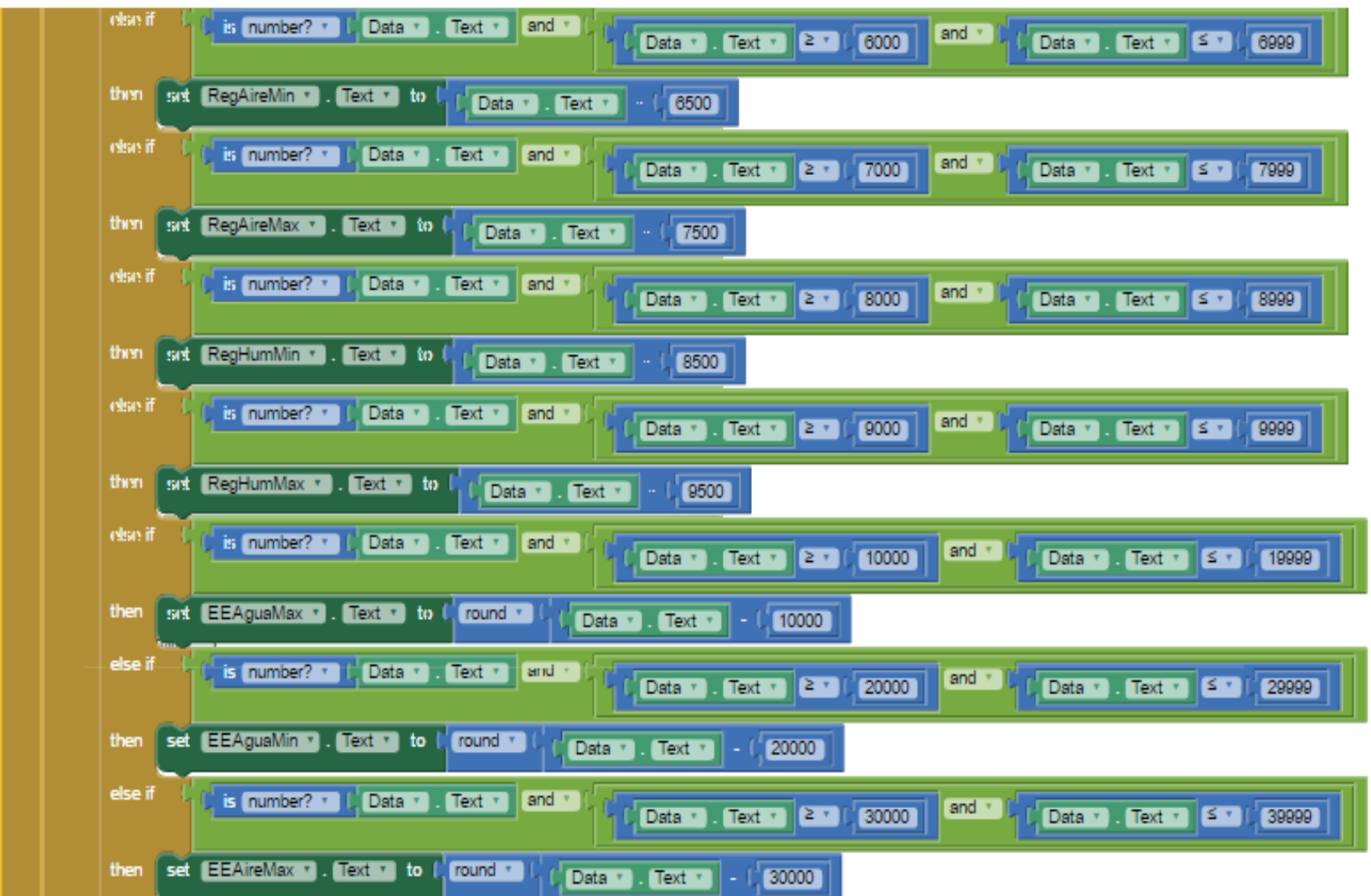
Prueba actuadores, vibración al presionar el botón prueba, cuadro de dialogo de confirmación, ejecución delay.

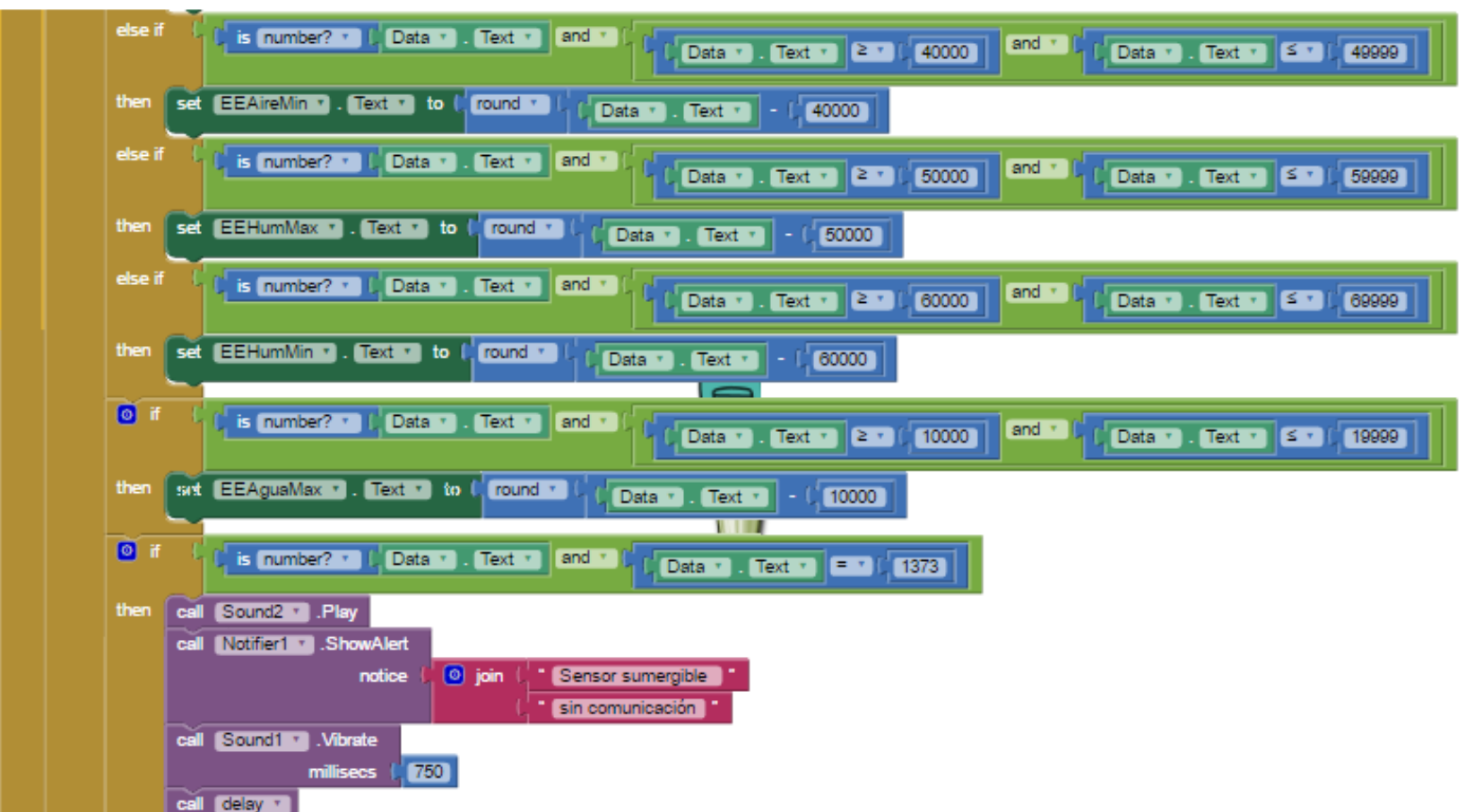


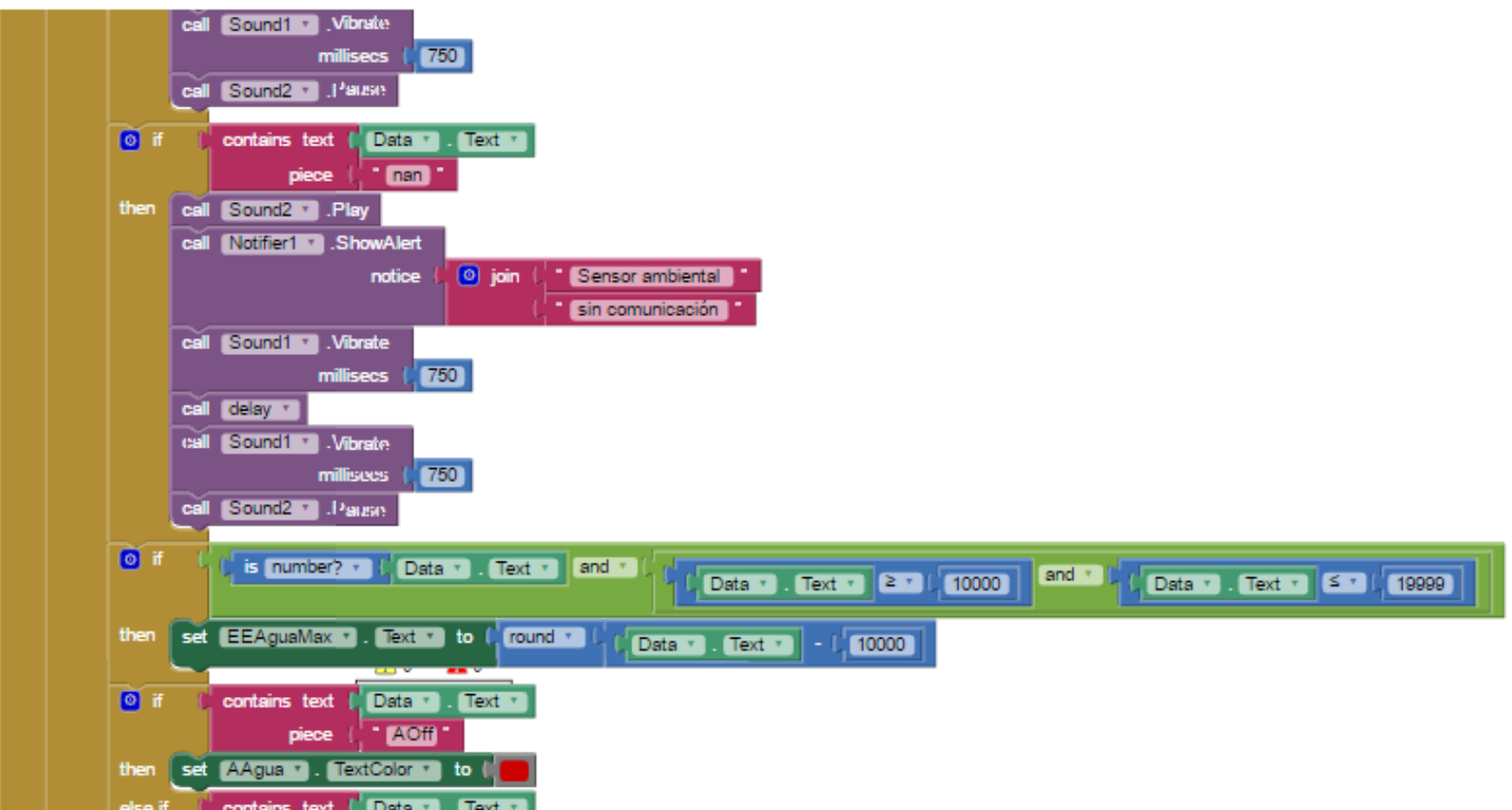
Vibración sliders.



Recepción e discriminación de datos de llegada, parte 1.







Recepción e discriminación de datos de Llegada, parte 4.


```

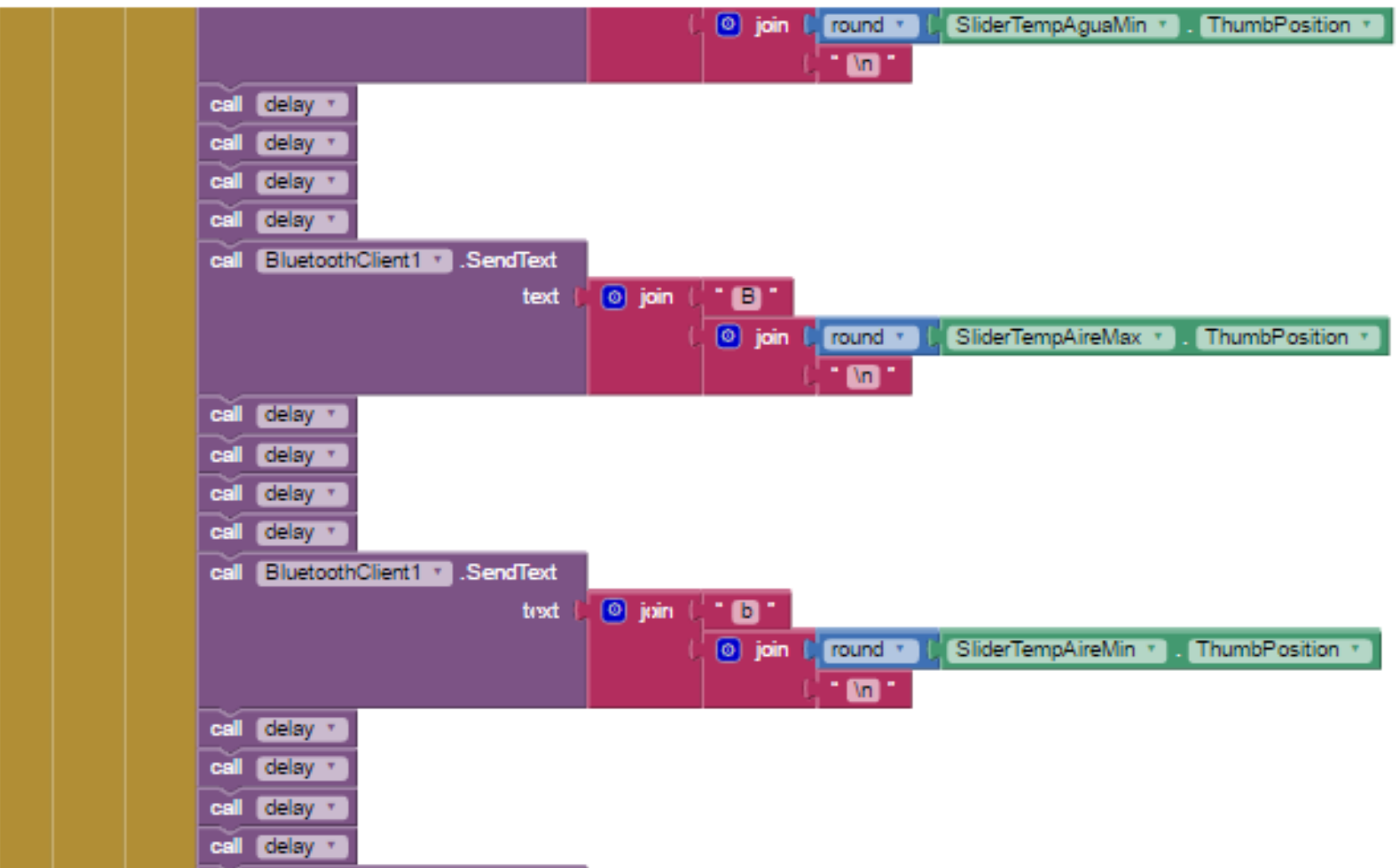
then set AAgua . TextColor to green
else if contains text Data . Text piece " BOff "
then set AAire . TextColor to red
else if contains text Data . Text piece " BOn "
then set AAire . TextColor to green
else if contains text Data . Text piece " HOff "
then set AHum . TextColor to red
else if contains text Data . Text piece " HOn "
then set AHum . TextColor to green
if contains text Data . Text piece " NEmp "
then set WarningAgua . Visible to true
set NivelAgua . Text to " Insuficiente "
else if contains text Data . Text piece " NFull "
then set NivelAgua . Text to " Adecuado "
set WarningAgua . Visible to false

```

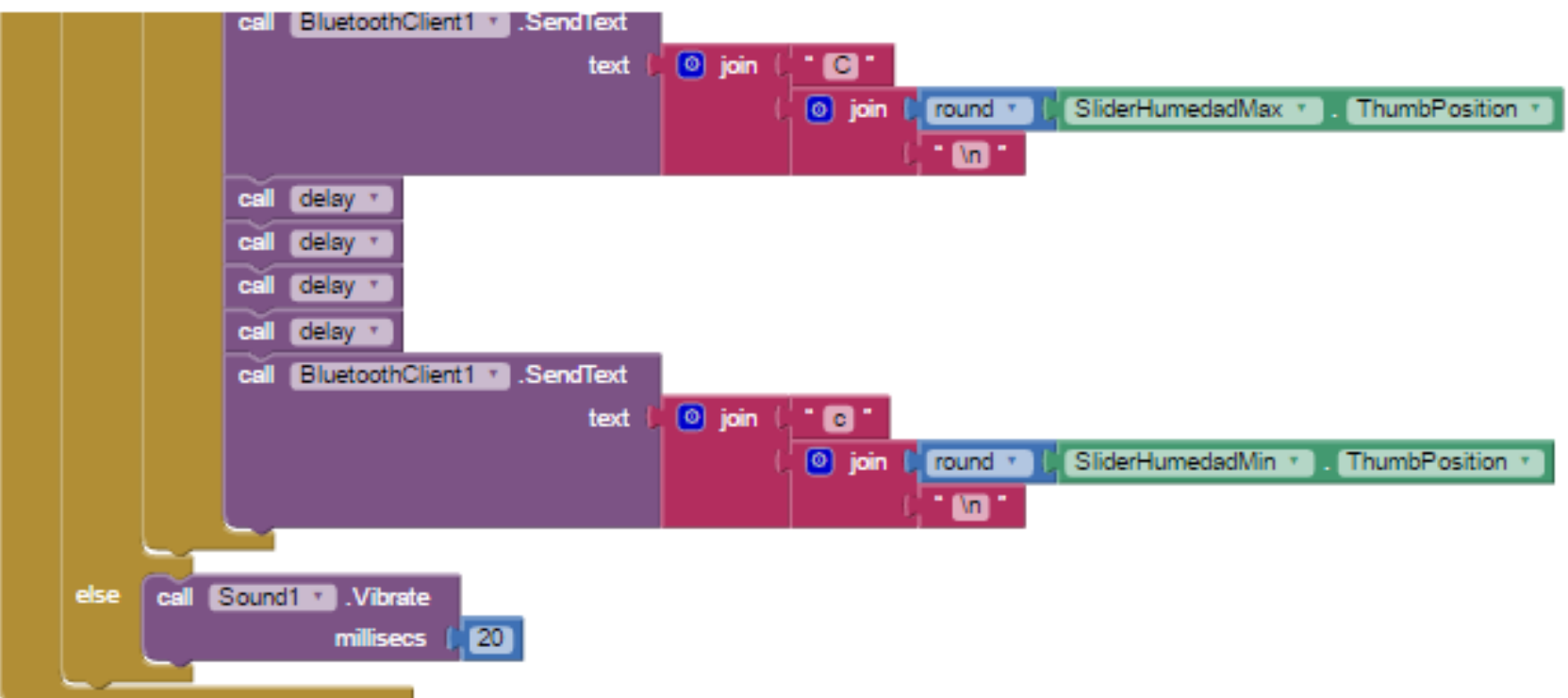
Recepción e discriminación de datos de llegada, parte 4.


```
when Notifier1 .AfterChoosing
  choice
  do
    if (get choice = "S1")
      then
        call Sound1 .Vibrate
          millisecs 20
        if (BluetoothClient1 .IsConnected)
          then
            set LabelTempAguaMax .Text to round (SliderTempAguaMax .ThumbPosition)
            set LabelTempAguaMin .Text to round (SliderTempAguaMin .ThumbPosition)
            set LabelTempAireMax .Text to round (SliderTempAireMax .ThumbPosition)
            set LabelTempAireMin .Text to round (SliderTempAireMin .ThumbPosition)
            set LabelHumedadMax .Text to round (SliderHumedadMax .ThumbPosition)
            set LabelHumedadMin .Text to round (SliderHumedadMin .ThumbPosition)
            call BluetoothClient1 .SendText
              text join ("A"
                join (round (SliderTempAguaMax .ThumbPosition)
                  "\n")
            )
            call delay
            call delay
            call delay
            call delay
            call BluetoothClient1 .SendText
              text join ("a")
```

Envío datos e incremento, parte 1.



Envío datos e incremento, parte 2.



Envío datos e incremento, parte 3.