

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE TECNOLOGÍA
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES



**“DISEÑO DE UN SISTEMA DE AUTOMATIZACIÓN PARA LA
FACULTAD DE TECNOLOGIA (EDIFICIO BLOQUE “B”) POR
MEDIO DE DISPOSITIVOS ELECTRÓNICOS”**

Proyecto de Grado presentado para obtener el Grado de Licenciatura

Por: Marcelo Chavez Chavez

Tutor: Lic. Juan Carlos Valencia Tarqui

La Paz- Bolivia

Diciembre, 2018

DEDICATORIA

“A Dios, mis Padres, mis Hermanos y Hermana, Amigos, Docentes que me acompañaron durante todo este proceso, de la realización del Proyecto de Grado, también durante todo mi estudio.... Gracias por su apoyo Incondicional”

AGRADECIMIENTO

“A mi tutor Juan Carlos Valencia Tarqui, que sin su ayuda y conocimientos no hubiese sido posible realizar este Proyecto de Grado”

“A Dios, mis Padres, mis Hermanos y Hermana, quienes con abnegación y mucha paciencia me apoyaron en todo momento de mi vida”

Índice

	Página
RESUMEN	1
 CAPÍTULO I	
1.1. Introducción	2
1.2. Planteamiento del Problema	3
1.3. Objetivos	4
1.3.1. Objetivo General	4
1.3.2. Objetivos Específicos	4
1.4 Justificación	5
1.5 Delimitación	5
1.6 Metodología	6
 CAPÍTULO II	
2 Marco Teórico	7
2.1 Antecedentes	7
2.1.1 Domótica	7
2.2 Microcontrolador	9
2.3 Arduino	11
2.4 Servidor WEB	14
2.4.1 Funcionamiento de un Servidor Web	16

2.5	HTML	18
2.6	Android	20
2.6.1	Estructura de Android	22
2.7	App Inventor	24
2.7.1	Características y funciones	25
2.8	Tratamiento Avanzado de Strings	26
2.9	Sistema de Control	27
CAPÍTULO III		
3	Desarrollo Práctico/Experimental	29
3.1	Diagrama de Bloques del Funcionamiento del Sistema	29
3.2	Relevamiento de los Sistemas	30
3.2.1	Aplicabilidad del Sistema	30
3.3	Hardware	33
3.3.1	Fuente de Alimentación	33
3.3.2	Arduino Mega	33
3.3.3	Arduino Nano	35
3.4	Diseño del Sistema	36
3.5	Construcción de las Placas Electrónicas	50
3.6	Software	61
3.6.1	IDE Arduino	61
3.6.1.1	Estructura de un Programa	62
3.6.1.2	Lenguaje de programación Arduino	63

3.6.2	Aplicación Android	63
3.6.3	Diseño de la Aplicación en App Inventor	65
3.6.4	Envío de Datos	69
3.7	Pruebas y Resultados de Funcionamiento del Sistema	72
4	Costos	78
5	Cronograma de Actividades	79
6	Conclusiones	80
7	Recomendaciones	81
8	Bibliografía	82
	ANEXO 1 Programación Aplicación Android App Inventor	85
	ANEXO 2 Código Control de Luces	107
	ANEXO 3 Código Control de Temperatura	118
	ANEXO 4 Código Control de Acceso a Laboratorios	131

Índice Figuras

	Pagina
Figura 2.1 Domótica	9
Figura 2.2 Microcontrolador	11
Figura 2.3 Arduino	13
Figura 2.4 Servidor WEB	15
Figura 2.5 Funcionamiento Servidor Web	18
Figura 2.6 Etiquetas Básicas o Mínimas	20
Figura 2.7 Plataforma Android	24
Figura 2.8 Sistema de control	27
Figura 2.9 Control de lazo abierto	28
Figura 2.10 Control lazo cerrado	28
Figura 3.1 Diagrama Luces, Estados de las Puertas, Control de Acceso, y Control de Temperatura	29
Figura 3.2 Facultad de Tecnología Edificio Bloque “B” (Av. 6 de Agosto)	30
Figura 3.3 Facultad de Tecnología Edificio Bloque “B” (Av. Arce)	31
Figura 3.4 Arduino Mega	34

Figura 3.5 Arduino Nano	36
Figura 3.6 Circuito de activación de relé desde la salida de un microcontrolador (3.3V o 5V)	37
Figura 3.7 Transistor de Potencia TIP31C	39
Figura 3.8 Modulo Ethernet ENC28J60	40
Figura 3.9 Conexión Modulo Ethernet ENC28J60	41
Figura 3.10 Final de Carrera	42
Figura 3.11 Conexión Luces, Sensor Final de Carrera	43
Figura 3.12 Conexión Sensor de Temperatura	45
Figura 3.13 Conexión LCD Y Modulo I2C - Arduino Nano	46
Figura 3.14 Circuito Control de Temperatura – Arduino Mega y Nano	47
Figura 3.15 Modulo RFID-RC522	48
Figura 3.16a Plano Impreso – Control de Luces y Estado de las Puertas	52
Figura 3.16b Placa Electrónica Control de Luces y Estado de las Puertas (Lado A)	53
Figura 3.16c Placa Electrónica Control de Luces y Estado de las Puertas (Lado B)	54
Figura 3.17a Plano Impreso – Control de Temperatura	55
Figura 3.17b Placa Electrónica Control de Temperatura (Lado A)	56
Figura 3.17c Placa Electrónica Control de Temperatura (Lado B)	57

Figura 3.18a Plano Impreso – Control de Acceso Modulo RFID	58
Figura 3.18b Placa Electrónica Control de Acceso Modulo RFID (Lado A)	59
Figura 3.18c Placa Electrónica Control de Acceso Modulo RFID (Lado B)	60
Figura 3.19 Entorno de Programación IDE Arduino	62
Figura 3.20 Entorno Diseño App Inventor 2	64
Figura 3.21 Entorno Bloques App Inventor 2	64
Figura 3.22 Diseño de la pantalla principal	65
Figura 3.23 Código de pantalla principal	66
Figura 3.24 Diseño de submenú utilizado en cada piso	66
Figura 3.25a Código submenú utilizado en cada piso	67
Figura 3.25b Código submenú utilizado en cada piso	67
Figura 3.25c Código submenú utilizado en cada piso	68
Figura 3.25d Código submenú utilizado en cada piso	68
Figura 3.26a Diagrama de Flujo del programa principal Luces parte 1	69
Figura 3.26b Diagrama de Flujo del programa principal Luces parte 2	70
Figura 3.27a Diagrama de Flujo del programa principal Temperatura parte 1	70
Figura 3.27b Diagrama de Flujo del programa principal Temperatura parte 2	71

Figura 3.28a Diagrama de Flujo del programa principal Control de Acceso parte 1	71
Figura 3.28b Diagrama de Flujo del programa principal Control de Acceso parte 2	72
Figura 3.29a Tablero Demostrativo	73
Figura 3.29b Ventana Principal Aplicación Android	73
Figura 3.30a Ventana Planta Baja Aplicación Android	74
Figura 3.30b Indicador Led Encendido de Luz	74
Figura 3.30c Encendido de Luz Mediante la Aplicación Android	75
Figura 3.31a Indicador de Temperatura Aplicación Android	75
Figura 3.31b Indicador Pantalla LCD	76
Figura 3.32a Circuito Control de Acceso	76
Figura 3.32b Indicador Control de Acceso	77

Índice Tablas

	Pagina
Tabla 2.1 Tipos de Arduino	13
Tabla 3.1 Lugares de Instalación Edificio Bloque “B”	31
Tabla 3.2 Especificaciones Arduino Mega	34
Tabla 3.3 Especificaciones Arduino Nano	35
Tabla 3.4 Especificaciones Sensor de Temperatura	45
Tabla 3.5 Conexión Módulo RFID	50
Tabla 4.1 Costos	78

RESUMEN

El proyecto de grado se realizó para el Edificio Bloque “B” de la Facultad de Tecnología que consta de 7 pisos. Cada piso y cada aula se diseñó el control luces, estado de puerta (abierto o cerrado), control de temperatura y control de acceso. Durante el desarrollo de este proyecto se realizó la investigación de cada sensor del hardware a utilizar y la aplicación que controla los distintos dispositivos. Luego del análisis se realizaron las pruebas que consisten en el software (aplicación Android) que se instaló en una Tablet y un celular con el sistema operativo Android.

El diseño del sistema de Automatización para la Facultad de Tecnología (Edificio Bloque “B”), ayudará a tener mejor control del encendido de luces y apagado de luces, así mismo ayudará a reducir el consumo discriminado de energía eléctrica por medio de Dispositivos Electrónicos.

El control de acceso a laboratorios ayudará a que sólo el personal autorizado pueda ingresar a los laboratorios teniendo mejor control y cuidado de los materiales y equipos existentes en los mismos, así mismo en la Aplicación Android se muestra el estado de las puertas (Abierto o Cerrado).

El control de temperatura fue diseñado para activar los ventiladores que se pondrán en funcionamiento cuando los sensores registren temperaturas mayores a los 20°C, los datos de la temperatura son mostrados en una pantalla LCD y en la Aplicación Android, estos sensores son instalados en los laboratorios que producen mayores temperaturas.

Por último, se propuso una recomendación formal sobre el uso del sistema de automatización.

DISEÑO DE UN SISTEMA DE AUTOMATIZACION PARA LA FACULTAD DE TECNOLOGIA (EDIFICIO BLOQUE “B”) POR MEDIO DE DISPOSITIVOS ELECTRONICOS

CAPÍTULO I

1.4. Introducción

Desde hace muchos años, el hombre ha demostrado interés por los robots y todos aquellos aparatos que le permitan simplificar su estilo de vida. Hoy existen productos y técnicas capaces de automatizar o robotizar las actividades domésticas, con el fin de aumentar el confort y la seguridad de los habitantes.

La domótica permite integrar y comunicar interactivamente todas las funciones y permite al usuario final interactuar con el sistema de forma sencilla, y está generando gran impacto en la sociedad y que, en un futuro gracias a las ventajas que ofrece, será la gran tendencia mundial. Pero ¿qué es la domótica? Se podría definir como el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda y/o edificios, que permite una gestión eficiente del uso de la energía además de aportar seguridad, confort, y comunicación entre el usuario y el sistema.

Aunque el ser humano todavía no está arraigado a las propiedades que ofrece la domótica es un hecho que en un futuro estará instalada en cualquier vivienda y/o edificios. Y no solo la domótica se está posicionando fuerte, en nuestra sociedad hay un cambio tecnológico por así decirlo, en cuanto al uso de celulares inteligentes y tabletas.

Los Smartphone, como se conocen los celulares inteligentes, tienen gran impacto a nivel mundial, donde una parte considerable de la humanidad posee este tipo de dispositivo electrónico. Basándose en una tendencia que alrededor del 50% de las personas que tienen un Smartphone poseen con sistema operativo android, en cuanto a su preferencia por sistema operativo, que tiene como fin combinar el uso de estos dispositivos electrónicos con la automatización de ambientes creando un prototipo de sistema de control que el cual permitirá controlar luces, alarmas, puertas, con elementos de campo (detectores, sensores, captadores, actuadores, etc.), transmitirán las señales a una unidad central inteligente que tratará y elaborará la información recibida. En función de dicha información y de una determinada programación, la unidad central actuará sobre determinados circuitos de potencia relacionados con las señales recogidas por los elementos de campo correspondientes.

1.5. Planteamiento del Problema

Se dice que un edificio tiene una automatización cuando incluye una infraestructura propia del cableado y equipos necesarios para brindar servicios avanzados, optimizando a la vez las funciones a medida que el mundo va avanzando y los seres humanos evolucionando, todo aquello que conocemos también va cambiando y la tecnología de automatización no es la excepción porque hoy en día se está implementado el software como otro medio para este tipo de procedimientos que incorporan enlaces a medios inalámbricos como el wifi.

¿Cómo podría ayudar el diseño de un sistema de automatización para la Facultad de Tecnología (Edificio Bloque “B”) a estos problemas?

La Facultad de Tecnología (Edificio Bloque “B”) tiene una insuficiente seguridad en el ingreso a las aulas y laboratorios, el uso indiscriminado de energía y laboratorios con entornos

lentos de equipamiento (computadoras, instrumentación, etc) que producen calor, por lo cual es necesario impulsar la incorporación de dispositivos de automatización, especialmente por los beneficios que brindan al control de ingreso, luces y temperatura, proporcionando la seguridad necesaria y ahorrando el consumo de energía que serán controlados mediante un teléfono celular o una Tablet con Sistema Operativo Android.

Por consiguiente, es necesario desarrollar un sistema de control y seguridad para la estructura de la Facultad de Tecnología (Edificio Bloque “B”), utilizando herramientas y materiales de bajo costo.

1.6. Objetivos

1.6.1. Objetivo General

- Diseñar un sistema de automatización, por medio de dispositivos electrónicos, para la Facultad de Tecnología (Edificio Bloque “B”), con las condiciones adecuadas y necesarias, proporcionando el control de los distintos sensores, optimizando el funcionamiento de aparatos eléctricos o electrónicos en las aulas y laboratorios con el fin de incrementar el ahorro de la energía y recursos.

1.6.2. Objetivos Específicos

- Diseñar un circuito regulador de temperatura.
- Diseñar un circuito de encendido y apagado de luces internas y exteriores.
- Planificar sistemas de control de apertura de puertas.

- Diseñar e implementar por medio de Software el sistema de información que permita al usuario el control y la configuración de los dispositivos electrónicos utilizando herramientas de bajo costo y máximo desempeño.
- Realizar un prototipo que mostrara el funcionamiento de los distintos sensores a utilizar en la Facultad de Tecnología Edificio Bloque “B”.

1.7 Justificación

El diseño del sistema permitirá automatizar actividades en la Facultad de Tecnología y brindará un ahorro económico y energético, al instalar los sistemas de encendido, apagado de luces, apertura de puertas en laboratorios y control de temperatura mediante el encendido de ventiladores en el edificio Bloque “B”, que contará con tecnología inteligente.

El diseño del sistema se justifica precisamente en el hecho de presentar de forma investigativa la teoría básica relacionada con los sistemas de automatización y a partir de los conocimientos adquiridos durante la investigación para la realización del presente proyecto de grado, poder realizar un diseño para el control de los distintos sensores a ser instalados en la Facultad de Tecnología Edificio Bloque “B”.

1.8 Delimitación

Se realizará el diseño para la Facultad de Tecnología Edificio Bloque “B”, tomando como parámetros sus características, pruebas y la utilización de recursos. En base al análisis comparativo y a un conjunto de requerimientos necesarios para el diseño.

1.9 Metodología

El método será “Descriptiva”, porque se utiliza la estadística para obtener las actividades y procesos.¹

Mediante este tipo de investigación, que utiliza el método de análisis, se logra caracterizar un objeto de estudio o una situación concreta, señalar sus características y propiedades. Combinada con ciertos criterios de clasificación sirve para ordenar, agrupar o sistematizar los objetos involucrados en el trabajo indagatorio, a través de la descripción exacta de las actividades y procesos durante el desarrollo del trabajo.

Por el propósito o finalidades perseguidas en el presente proyecto, durante el desarrollo del mismo se enfocará en la Metodología de Investigación Aplicada, el cual se caracteriza porque busca la aplicación o utilización de los conocimientos que se adquieren y los que ya están pre-establecidos, como son el conjunto de tecnologías aplicadas al control y automatización de viviendas denominado domótica. La investigación aplicada se encuentra estrechamente vinculada con la investigación básica, pues depende de los resultados y avances de esta última, como ser en el área tecnológica con la aplicación de plataformas de hardware libre como ser Arduino, sistemas operativos basados en el núcleo Linux como el Android y dispositivos transductores que cada vez son actualizados en sus propiedades y aplicaciones con mayor sensibilidad y mejores beneficios; esto queda aclarado si se advierte de toda investigación aplicada requiere de un marco teórico. Sin embargo, en una investigación, lo que interesa, primordialmente, son las consecuencias prácticas que se obtendrán con el cumplimiento de ciertos servicios, tales como el ahorro energético, seguridad confort y comunicación.

¹ (<http://metodologiadelainvestigaciis.blogspot.com>, s.f.)

2 Marco Teórico

8.1 Antecedentes

8.1.1 Domótica

El término domótica viene de la unión de las palabras “domus” (que significa casa en latín) y “tica” (de automática, palabra en griego, “que funciona por sí sola”). La domótica es el conjunto de tecnologías aplicadas al control y la automatización de la vivienda tratando de buscar ciertos servicios, tales como el ahorro energético, seguridad, confort y comunicación.

Los primeros pasos de la domótica se remontan a los años 80 a nivel comercial, pero cuando realmente empezó a utilizarse en el ámbito doméstico fue en los años 90, cuando empezaron a realizarse pequeñas prestaciones, coincidiendo con la evolución y despliegue de internet. En los primeros países donde comenzó el uso fue en Japón, Estados Unidos y algunos países del norte de Europa.

A medida que fue pasando el tiempo fueron desarrollándose nuevos sistemas domóticos domésticos. Estas mejoras eran totalmente autónomas, sin buscar en ningún momento la comunicación con otros dispositivos, cada uno hacia lo suyo, ni con el exterior, lo que conllevó un desarrollo de un mercado puramente vertical.

En esa época se puede decir que se buscaba la tecnología por la tecnología, pero a raíz de la entrada de internet en la mayoría de los hogares se cambió la forma de interpretar la domótica, buscando a partir de ahora el beneficio de los usuarios, buscando satisfacer sus necesidades. De

este modo se ha pasado de un mercado estrictamente vertical a otro que también se desarrolla horizontalmente. Este mercado horizontal trata de asegurar la capacidad de comunicación entre todos los equipos de una vivienda u edificio.

Los servicios que ofrece la domótica se pueden agrupar en diferentes categorías: ahorro energético, seguridad, confort y comunicaciones. A menudo existen situaciones que es difícil clasificar un servicio estrictamente en una sola porque puede englobar a varias a la vez.

El ahorro energético es algo que en un primer momento puede no notarse, pero a largo plazo sí que es una aplicación de la domótica a tener en cuenta para un uso razonado de los recursos naturales del planeta, así como para entre otras cosas ahorro económico. Para que esta aplicación surta efecto, no es necesario cambiar los aparatos que están instalados en el hogar, sino que es necesario un uso adecuado tanto en tiempo como en el uso de la potencia que tienen que utilizar para la labor a desarrollar.

Cuando se habla de seguridad se refiere tanto en el sentido de evitar los robos como minimizar los daños en la vivienda o edificio ante cualquier percance. Este tipo de situaciones pueden referirse tanto a detección de escapes de gas, agua, etc. como a otro tipo de acciones que pueden provocar inseguridades al usuario dentro de la vivienda o edificio.

Refiriendo a las comunicaciones lo buscado en la domótica es la capacidad para establecer conexiones para saber el estado de los aparatos electrónicos dentro de la vivienda o edificio sin necesidad de estar en ella y, llegado el caso, poder modificarlas.²

² (<http://www.academia.edu>, s.f.) (<http://www.monografias.com>, s.f.)



Figura 2.1 Domótica
(Fuente: domotizados.com)

8.2 Microcontrolador

Un microcontrolador (abreviado μC , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Algunos microcontroladores pueden utilizar palabras de cuatro bits y funcionan a velocidad de reloj con frecuencias tan bajas como 4 kHz, con un consumo de baja potencia (mW o microwatts). Por lo general, tendrá la capacidad de mantenerse a la espera de un evento como pulsar un botón o de otra interrupción; así, el consumo de energía durante el estado de reposo (reloj de la CPU y los periféricos de la mayoría) puede ser sólo de nanowatts, lo que hace que muchos de ellos sean muy adecuados para aplicaciones con batería de larga duración. Otros microcontroladores pueden servir para roles de rendimiento crítico, donde sea necesario actuar

más como un procesador digital de señal (DSP), con velocidades de reloj y consumo de energía más altos.

Cuando es fabricado el microcontrolador, no contiene datos en la memoria ROM. Para que pueda controlar algún proceso es necesario generar o crear y luego grabar en la EEPROM o equivalente del microcontrolador algún programa, el cual puede ser escrito en lenguaje ensamblador u otro lenguaje para microcontroladores; sin embargo, para que el programa pueda ser grabado en la memoria del microcontrolador, debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento.

En la figura 2.2, se distingue al microcontrolador dentro de un encapsulado de circuito integrado, con su procesador (CPU), buses, memoria, periféricos y puertos de entrada/salida. Fuera del encapsulado se ubican otros circuitos para completar periféricos internos y dispositivos que pueden conectarse a los pines de entrada/salida. También se conectarán a los pines del encapsulado la alimentación, masa, circuito de completamiento del oscilador y otros circuitos necesarios para que el microcontrolador pueda trabajar.

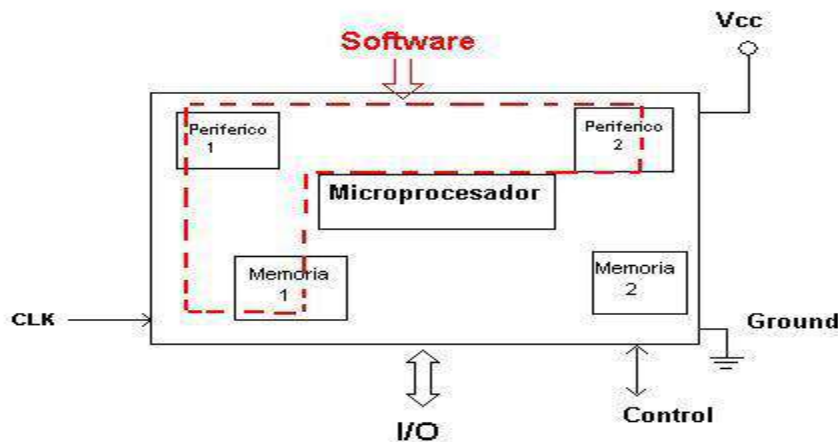


Figura 2.2 Microcontrolador
(Fuente: es.wikipedia.org)

8.3 Arduino

Al ser Arduino una plataforma de hardware libre tanto su diseño como su distribución puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. Por eso existen varios tipos de placa oficiales, las creadas por la comunidad Arduino o las no oficiales creadas por terceros, pero con características similares. En la placa Arduino es donde conectaremos los sensores, actuadores y otros elementos necesarios para comunicarnos con el sistema.

El hardware consiste en una placa de circuito impreso con un microcontrolador, usualmente Atmel AVR, y puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (shields) que expanden las características de funcionamiento de la placa Arduino.

Por otro lado, el software consiste en un entorno de desarrollo (IDE) basado en el entorno de Processing y lenguaje de programación basado en Wiring, así como en el cargador de

arranque (bootloader) que es ejecutado en la placa. El microcontrolador de la placa se programa a través de un computador, haciendo uso de comunicación serial mediante un convertidor de niveles RS-232 a TTL serial.³

La primera placa Arduino fue introducida en el 2005, ofreciendo un bajo costo y facilidad de uso para novatos y profesionales buscando desarrollar proyectos interactivos con su entorno mediante actuadores y sensores. A partir de octubre del año 2012, se incorporaron nuevos modelos de placas de desarrollo que hacen uso de microcontroladores CortexM3, ARM de 32 bits, que coexisten con los originales modelos que integran microcontroladores AVR de 8 bits. ARM y AVR no son plataformas compatibles a nivel binario, pero se pueden programar y compilar bajo el IDE clásico de Arduino sin ningún cambio.

Las placas Arduino están disponibles de forma ensamblada o en forma de Kits, los esquemáticos de diseño del Hardware están disponibles bajo licencia Libre, permitiendo a cualquier persona crear su propia placa arduino sin necesidad de comprar una prefabricada. Adafruit Industries estimó a mediados del año 2011 que alrededor de 300,000 placas arduinos habían sido producidas comercialmente, y en el año 2013 estimó que alrededor de 700,000 placas oficiales de la empresa Arduino estaban en manos de los usuarios.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data. Una tendencia tecnológica es utilizar Arduino como tarjeta de adquisición de datos desarrollando interfaces en software como JAVA, Visual Basic y LabVIEW.⁴

³ (<http://www.monografias.com>, s.f.)

⁴ (Arduino, 2017)



Figura 2.3 Arduino
(Fuente: www.arduino.cc)

Existen diversos tipos de Arduino lo que varía de uno a otro son las características del microcontrolador, el número de patillas digitales, entradas analógicas disponibles y cantidad de memoria.

Tabla 2.1:

Tipos de Arduino (Fuente: store.arduino.cc)

Modelo de placa Arduino	Características
UNO	ATmega328P a 16Mhz, puerto USB, 5v, SRAM de 2Kb, 32 de flash, 1kb de EEPROM, 14 pines digitales y 6 con PWM, 6 salidas analógicas.
MEGA	ATmega2560 a 16Mhz, 5v, 4Kb de EEPROM, 8Kb de SRAM y 256Kb de flash. Las entradas digitales 54, 14 con PWM y 16 analógicas.

8.4 Servidor WEB

Es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web. Para la transmisión de todos estos datos se utiliza algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del Modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.⁵

Sus servicios son:

- **Internet:** Internet es una Red de Ordenadores conectados en toda la extensión del Globo Terráqueo, que ofrece diversos servicios a sus usuarios como pueden ser el Correo electrónico, el Chat o la Web. Todos los servicios que ofrece Internet son llevados a cabo por miles de ordenadores que están permanentemente encendidos y conectados a Internet, esperando que los usuarios les soliciten los servicios y sirviéndolos una vez son solicitados.
- **Página Web:** Documento o fuente de información, generalmente en formato HTML y que puede contener hiperenlaces a otras Páginas Web. Dicha página, podrá ser accesible desde un dispositivo físico, una intranet, o Internet.
- **Navegador Web:** Para establecer conexiones con los servidores Web, y obtener la información y los servicios que estos prestan, el usuario necesita tener instalado en su equipo un programa cliente capaz de comunicarse con ellos. Estos programas son los

⁵ (Wikipedia, servidor Web, 2017)

llamados Navegadores Web. Los Navegadores Web, también llamados Visores de Web o Browsers, son aplicaciones que permiten ver en pantalla texto con formato (con palabras en negrita, y con distintas fuentes tipográficas, tamaños y colores) y presentar imágenes en línea. También permiten visualizar secuencias de vídeo y escuchar ficheros de sonido.

⁶

- Servidor: Un Servidor es un tipo de Software que suministra servicios a los usuarios o terminales que lo solicitan. Por ejemplo, en una típica Arquitectura Cliente-servidor, el cliente podría ser un ordenador que realiza peticiones de información a través de un programa de correo (Outlook Express por ejemplo) y, el servidor le entrega los datos en forma de correos electrónicos en respuesta a su solicitud. Un servidor no es necesariamente una máquina de última generación de grandes proporciones, no es necesariamente un superordenador; un servidor puede ser desde una computadora vieja, hasta una máquina sumamente potente (ej. Servidores web, bases de datos grandes, etc., procesadores especiales y hasta varios gigabytes de memoria). ⁷



Figura 2.4 Servidor WEB

(Fuente: blog.infranetworking.com/que-es-un-servidor-web/)

⁶ (Wikipedia, Navegador Web, 2017)

⁷ (Masa adelante, 2017)

8.4.1 Funcionamiento de un Servidor Web

La Web funciona siguiendo el Modelo cliente-servidor. Un Servidor se encarga de prestar el servicio, y un cliente que es quien recibe dicho servicio.

Cliente Web: Es un programa mediante el cual el usuario solicita a un Servidores Web el envío de información. Esta información se transfiere mediante el Protocolo HTTP. Información que recibe: La información que se recibe es un conjunto de documentos de texto codificados en lenguaje HTML. ⁸

El Cliente Web debe interpretar estos documentos para mostrárselos al usuario en el formato correspondiente. Cuando la información recibida no es un documento de texto, sino un objeto multimedia que el cliente no sabe interpretar, el propio cliente Web debe activar una aplicación externa encargada de gestionarlo. Clientes Web más habituales: Los clientes Web más habituales son Microsoft Internet Explorer, Mozilla Firefox y Netscape Navigator.

- a) **Arquitectura Modelo Cliente – Servidor** Diversas aplicaciones se ejecutan en un entorno Cliente/servidor. Esto significa que los equipos clientes (equipos que forman parte de una red) contactan a un servidor, un equipo generalmente muy potente en materia de capacidad de entrada/salida, que proporciona servicios a los equipos clientes. Estos servicios son programas que proporcionan datos como la hora, archivos, una conexión, etc. Los servicios son utilizados por programas denominados programas clientes que se ejecutan en equipos clientes. Por eso se utiliza el término "cliente" (cliente FTP, cliente de correo electrónico, etc.) cuando un programa que se ha diseñado para

⁸ (Internetlab, 2017)

ejecutarse en un equipo cliente, capaz de procesar los datos recibidos de un servidor (en el caso del cliente FTP se trata de archivos, mientras que para el cliente de correo electrónico se trata de correo electrónico). En la manera de describir la forma de trabajo entre los clientes y los ordenadores se define: Cliente: Es el ordenador que pide información a otro, mediante la aplicación de un programa llamado cliente. Este contacta con el servidor y da formato a la petición de la información y da formato a la respuesta.

- b) **Aplicaciones en el lado del cliente:** el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java "applets" o Javascript. El servidor proporciona el código de las aplicaciones al cliente y éste las ejecuta mediante el navegador Web. Por tanto, es necesario que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas Scripts). Generalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje Javascript y Java, aunque pueden añadirse más lenguajes mediante el uso de plugins.
- c) **Aplicaciones en el lado del servidor:** el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera código HTML; el servidor envía al cliente este código recién creado por medio del protocolo HTTP.

Las aplicaciones en el lado del servidor mayormente suelen ser la mejor opción para desarrollar aplicaciones Web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste último no necesita ninguna capacidad añadida para ejecutar la aplicación, como sí ocurre en el caso de querer ejecutar aplicaciones que incluyan scripts con

javascript o java. Así pues, cualquier cliente que disponga de un navegador Web básico puede utilizar este tipo de aplicaciones.⁹



Figura 2.5 Funcionamiento Servidor Web
(Fuente: gorkamu.com/2017/06/entendiendo-protocolo-http/)

8.5 HTML

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web

⁹ (Wikipedia, servidor Web, 2017)

(WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.¹⁰

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser Gedit en GNU/Linux, el Bloc de notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, TextPad, Vim, Notepad++, entre otros.

Existen, además, otros editores para la realización de sitios web con características WYSIWYG (What You See Is What You Get, o en español: «lo que ves es lo que obtienes»). Estos editores permiten ver el resultado de lo que se está editando en tiempo real, a medida que se va desarrollando el documento. Ahora bien, esto no significa una manera distinta de realizar sitios web, sino que una forma un tanto más simple, ya que estos programas, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML, la cual va generando todo el código a medida que se va trabajando. Algunos ejemplos de editores WYSIWYG son KompoZer, Microsoft FrontPage o Adobe Dreamweaver.

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que debe aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador.

Toda etiqueta se identifica porque está encerrada entre los signos menor que y mayor que (<>), y algunas tienen atributos que pueden tomar algún valor. En general las etiquetas se aplicarán de dos formas especiales:

¹⁰ (Wikipedia, HTML, 2017)

- Se abren y se cierran, como por ejemplo: `negrita`, que se vería en su navegador web como negrita.
- No pueden abrirse y cerrarse, como `<hr />`, que se vería en su navegador web como una línea horizontal.
- Otras que pueden abrirse y cerrarse, como por ejemplo `<p>`.¹¹

```

<!DOCTYPE HTML>
<html>
  <head>
    <title>Ejemplo1</title>
  </head>
  <body>
    <p>Párrafo de ejemplo</p>
  </body>
</html>

```

Figura 2.6 Etiquetas Básicas o Mínimas
(Fuente: es.wikipedia.org/wiki/HTML)

8.6 Android

Android es un Sistema Operativo de código abierto para dispositivos móviles por eso viene hacer uno de los mejores Sistemas Operativos dando a conocer sus mejores versiones como. Petit Four, Froyo, Jelly Bean, Kitkat, etc.

Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance

¹¹ (Wikipedia, HTML, 2017)

(un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Los dispositivos de Android venden más que las ventas combinadas de Windows Phone e IOS.

Android es una plataforma de código abierto para dispositivos móviles que está basada en Linux y desarrollada por Open Handset Alliance, se prevé que los primeros teléfonos con Android aparezcan en el segundo semestre de 2008 y compañías poderosas como LG, Motorola y HTC ya han diseñado alguno de los prototipos que incorporarán el Sistema Android.¹²

Es una stack de software para dispositivos móviles que incluye un Sistema Operativo, Middleware y aplicaciones de base. Los desarrolladores pueden crear aplicaciones para la plataforma usando el SDK de Android. Las solicitudes se han escrito utilizando el lenguaje de programación Java y se ejecutan en Dalvik, una máquina virtual personalizada que se ejecuta en la parte superior de un núcleo de Linux.

Android ha visto numerosas actualizaciones desde su liberación inicial. Estas actualizaciones al sistema operativo base típicamente arreglan fallos y agregan nuevas funciones. Generalmente cada actualización del sistema operativo Android es desarrollada bajo un nombre en código de un elemento relacionado con dulces en orden alfabético.¹³

- a) Ventajas de Android: Algunas ventajas de Android para instalarse prácticamente en todo tipo de dispositivos, sean móviles, portátiles e incluso microondas, hace que Android siempre esté presente en los terminales más potentes del mercado siendo una apuesta importante por fabricantes y operadoras por la posibilidad de que independientemente del

¹² (www.monografias.com)

¹³ (SistemaOperativoAndroid, 2017)

potencial, gama o prestaciones del dispositivo, Android podrá adaptarse a la perfección a todo tipo de necesidades. El hecho de que Android esté liberado con licencia Apache y código abierto lo convierte en un sistema operativo totalmente libre para que un desarrollador no solo pueda modificar su código sino también mejorarlo.

- b) Desventajas de Android: Las desventajas de Android es multitarea: esto es un arma de doble filo. Por un lado, tiene un gran aporte positivo, como he comentado más arriba, pero tiene dos grandes contras. Para empezar el hecho de tener varias aplicaciones abiertas hacen que el consumo de la batería se dispare y por otro lado Android no siempre cierra todas las aplicaciones así que hace falta tener una aplicación que cierre las aplicaciones abiertas.

8.6.1 Estructura de Android

Los componentes del Sistema Operativo de Android, cada sección se describe en detalle a continuación:¹⁴

- a) Aplicaciones: Las aplicaciones base incluyen un cliente de email, programa de SMS, calendario, mapas, navegador, contactos, y otros. Todas las aplicaciones son escritas en el lenguaje de programación Java.
- b) Framework de Aplicaciones: Los desarrolladores tienen acceso completo a los mismos Apis del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede hacer luego uso de esas capacidades (sujeto a reglas de seguridad del framework). Éste mismo mecanismo permite que los

¹⁴ (SistemaOperativoAndroid, 2017)

componentes sean reemplazados por el usuario. Una capa de servicios disponibles para las aplicaciones incluye:

- i. Librerías: Android incluye un conjunto de librerías C/C++ usadas por varios componentes del sistema Android. Estas capacidades se exponen a los desarrolladores a través del framework de aplicaciones de Android. Algunas son: System C library (implementación librería C standard), librerías de medios, librerías de gráficos, 3d, SQLite, entre otras.
- c) Runtime de Android: Android incluye un conjunto de librerías base que proveen la mayor parte de las funcionalidades disponibles en las librerías base del lenguaje de programación Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalkiv ha sido escrito de forma que un dispositivo puede correr en múltiples máquinas virtuales de forma eficiente. (Deliverius). Dalkiv ejecuta archivos en el formato Dalvik Ejecutable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros, y corre clases compiladas por el compilador de Java que han sido transformadas al formato. Dex por la herramienta incluida "dx".
- d) Núcleo - Linux: Android depende de un Linux versión 2.6 para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, stack de red, y modelo de drivers. El núcleo también actúa como una capa de abstracción entre el hardware y el resto del stack de software.



Figura 2.7 Plataforma Android
(Fuente: www.monografias.com)

8.7 App Inventor

App Inventor es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. Las aplicaciones creadas con App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

Con Google App Inventor, se espera un incremento importante en el número de aplicaciones para Android debido a dos grandes factores: la simplicidad de uso, que facilitará la aparición de un gran número de nuevas aplicaciones; y Google Play, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente.¹⁵

La plataforma se puso a disposición del público el 15 de diciembre de 2010 y está dirigida a personas que no están familiarizadas con la programación informática. En la creación

¹⁵ (AppInventor, 2017)

de App Inventor, Google se basó en investigaciones previas significativas en informática educativa.

8.7.1 Características y funciones

El editor de bloques de la plataforma App Inventor, utiliza la librería Open Blocks de Java para crear un lenguaje visual a partir de bloques. Estas librerías están distribuidas por Massachusetts Institute of Technology bajo su licencia libre. El compilador que traduce el lenguaje visual de los bloques para la aplicación en Android utiliza Kawa como lenguaje de programación, distribuido como parte del sistema operativo GNU de la Free Software Foundation.

App Inventor permite crear una aplicación en una hora o menos, y se pueden programar aplicaciones más complejas en mucho menos tiempo que con los lenguajes más tradicionales, basados en texto. Inicialmente desarrollado por el profesor Hal Abelson y un equipo de Google Educación, mientras que Hal pasaba un año sabático en Google, App Inventor se ejecuta como un servicio Web administrado por personal del Centro del MIT para el aprendizaje móvil una colaboración de MIT de Ciencia Computacional e Inteligencia Artificial de laboratorio (CSAIL) y el Laboratorio de Medios del MIT. El App Inventor contaba en 2015 con una comunidad mundial de casi dos millones de usuarios que representaban a 195 países en todo el mundo. Más de 85 mil usuarios semanales activos de la herramienta han construido más de 4,7 millones de aplicaciones de Android. Una herramienta de código abierto que pretende realizar la programación y la creación de aplicaciones accesibles.¹⁶

¹⁶ (AppInventor, 2017)

8.8 Tratamiento Avanzado de Strings

El tratamiento de strings es un parte muy importante en Arduino puesto que se usa muy frecuentemente y principalmente se usa en las comunicaciones, ya sea puerto serie, bluetooth, XBee, http, etc...

El uso de strings hace un uso intensivo de memoria lo que hace que se pueda tener comportamientos extraños en los sketches o incluso se tenga desbordamiento de memoria.

A la hora de usar strings en Arduino, se hace uso de la clase String, que nos ofrece unos métodos y es muy sencilla de usar, a cambio de ser poco eficiente a nivel de SRAM o usar los strings como arrays de chars que es más complejo de manejar pero más potente y tenemos más control del uso de memoria.

String (Objeto):

Arduino nos ofrece una clase llamada String que facilita el uso de las cadenas de caracteres con unos métodos muy sencillos de usar y poder los operadores que conocemos para los Strings.

Se trata de una clase que permite usar y manipular cadenas de texto de una forma más sencilla que los strings. Puedes concatenar, añadir, buscar, etc... usando los métodos/funciones que ofrece esta clase.

Los Strings tienen un uso intensivo de memoria, pero son muy útiles y se van a utilizar mucho en el apartado de comunicación.

Al no ser un tipo de dato propiamente dicho sino una clase, tiene unas funciones asociadas (métodos), operadores y unas propiedades.

8.9 Sistema de Control

Sistema de control es el conjunto de dispositivos que actúan juntos para lograr un objetivo de control.

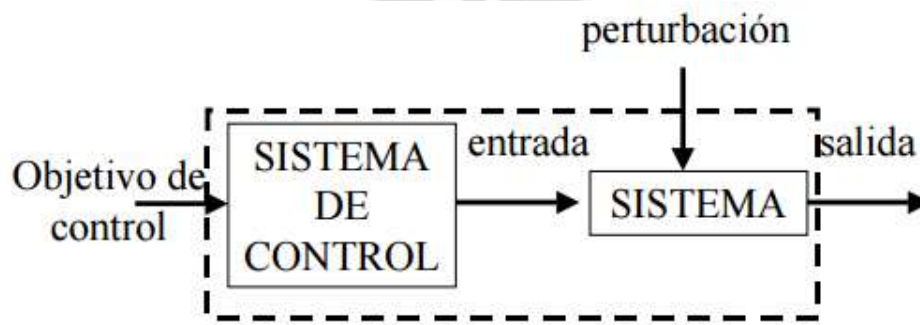


Figura 2.8 Sistema de control
(Fuente: sites.google.com/site/tecnoindus1/sistemas-de-control)

Tenemos los siguientes sistemas de control:

a) Control de lazo abierto:

El control en lazo abierto se caracteriza porque la información o variables que controlan el proceso circulan en una sola dirección.



Figura 2.9 Control de lazo abierto
(Fuente: sites.google.com/site/tecnoindus1/sistemas-de-control)

b) Control de lazo cerrado

El control en lazo cerrado se caracteriza porque existe una realimentación a través de los sensores desde el proceso hacia el sistema de control, que permite a éste último conocer si las acciones ordenadas a los actuadores se han realizado correctamente sobre el proceso.¹⁷

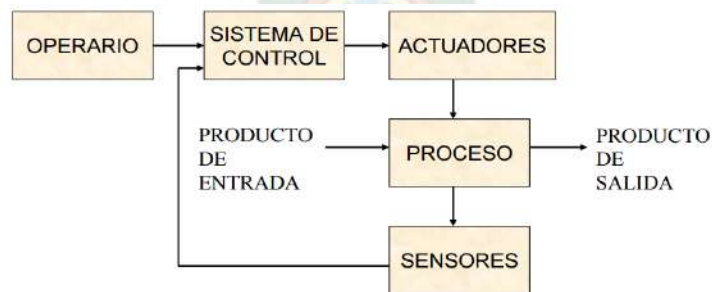


Figura 2.10 Control lazo cerrado
(Fuente: sites.google.com/site/tecnoindus1/sistemas-de-control)

¹⁷ (<http://es.slideshare.net>, s.f.)

9 Desarrollo Práctico/Experimental

9.1 Diagrama de Bloques del Funcionamiento del Sistema

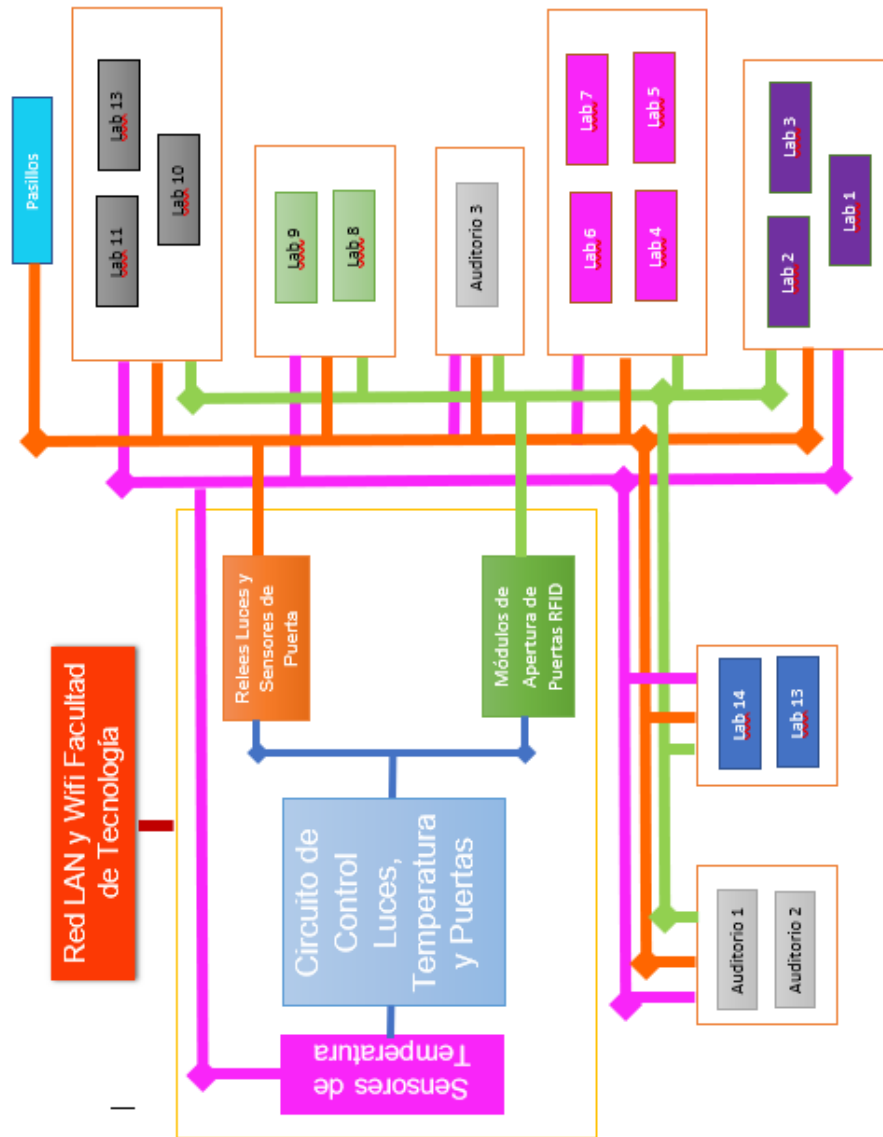


Figura 3.1 Diagrama Luces, Estados de las Puertas, Control de Acceso, y Control de Temperatura
(Fuente: Autor)

9.2 Relevamiento de los Sistemas

Para el diseño del “SISTEMA DE AUTOMATIZACION PARA LA FACULTAD DE TECNOLOGIA (EDIFICIO BLOQUE “B”) POR MEDIO DE DISPOSITIVOS ELECTRONICOS”, se utilizará una tecnología que permite controlar el encendido y apagado de luces en los diferentes laboratorios mediante una aplicación android, el control de acceso a laboratorios mediante la tarjeta RFID de cada docente o responsable, y el control automático de temperatura en los laboratorios de computación. Para este proceso se utiliza un software, hardware y la red LAN (o Wifi) de la Facultad de Tecnología.

9.2.1 Aplicabilidad del Sistema

El sistema fue diseñado para la Facultad de Tecnología Edificio Bloque B, ubicado en la Av. Arce Zona Sopocachi.

El área que fue diseñado el sistema cubre los ambientes interiores de la Facultad de Tecnología Edificio Bloque “B”.

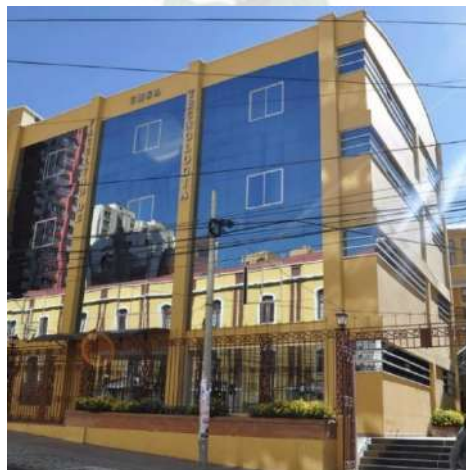


Figura 3.2 Facultad de Tecnología Edificio Bloque “B” (Av. 6 de Agosto)
(Fuente: Autor)



Figura 3.3 Facultad de Tecnología Edificio Bloque “B” (Av. Arce)
(Fuente: Autor)

Lugares de Instalación de control de Luces, control de temperatura y control de acceso a laboratorios (tarjeta RFID):

Tabla 3.1:

Lugares de Instalación Edificio Bloque “B” (Fuente: Autor)

Piso	Lugar	Control de Luces	Control de Temperatura	Control de Acceso a Laboratorios
Planta Baja	Laboratorio 13	Si	No	Si
	Laboratorio 14	Si	Si	Si
Primer Piso	Laboratorio 10	Si	No	Si
	Laboratorio 11	Si	No	Si

	Laboratorio 12	Si	No	Si
Segundo Piso	Laboratorio 8	Si	No	Si
	Laboratorio 9	Si	Si	Si
Tercer Piso	Auditórium 3	Si	Si	Si
Cuarto Piso	Laboratorio 4	Si	Si	Si
	Laboratorio 5	Si	No	Si
	Laboratorio 6	Si	No	Si
	Laboratorio 7	Si	No	Si
Quinto Piso	Laboratorio 1	Si	Si	Si
	Laboratorio 2	Si	No	Si
	Laboratorio 3	Si	No	Si
Sexto Piso	Auditórium 1	Si	No	Si
	Auditórium 2	Si	Si	Si
Pasillos	-----	Si	No	No

9.3 Hardware

Al ser Arduino una plataforma de hardware libre tanto su diseño como su distribución se utiliza libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. Por eso existen varios tipos de placa oficiales, las creadas por la comunidad Arduino o las no oficiales creadas por terceros, pero con características similares. En la placa Arduino es donde se conectan los sensores, actuadores y otros elementos necesarios para comunicarnos con el sistema.¹⁸

Para poder interconectar los diferentes elementos electrónicos a nuestra red, se utilizará microcontroladores. Se ha elegido la tarjeta Arduino Mega y Nano.

9.3.1 Fuente de Alimentación:

Para el funcionamiento del Arduino Mega, Nano y los diferentes módulos acoplables, se utiliza una fuente de alimentación ATX (computadora) de entrada 220VAC, salida de 5 VDC - 12VDC y el voltaje que proporciona Arduino en los dos modelos que se utiliza de 3.3VDC.

9.3.2 Arduino Mega

Las especificaciones técnicas de la tarjeta son:

¹⁸ (Arduino, 2017)

Tabla 3.2:

Especificaciones Arduino Mega (Fuente: arduino.cl/arduino-mega-2560)

CARACTERITICAS	ARDUINO MEGA
Microcontrolador	ATEMEGA 2560
Voltaje de Alimentación	5 voltios
Pines digitales E/S	54 donde 15 son PWD
Pines Analógicos	16
Memoria Flash	248KB
SRAM	8KB
EEPROM	4KB
Velocidad de Reloj	16MHz



Figura 3.4 Arduino Mega
(Fuente: arduino.cl/arduino-mega-2560/)

9.3.3 Arduino Nano

Las especificaciones técnicas de la tarjeta son:

Tabla 3.3:

Especificaciones Arduino Nano (Fuente: arduino.cl/arduino-nano)

CARACTERITICAS	ARDUINO NANO
Microcontrolador	Atmel ATmega328
Voltaje de operación	5 voltios
Voltaje de entrada (recomendado)	7-12 V
Pines E/S Digitales	14 (de los cuales 6 proveen de salida PWM)
Entradas Analógicas	8 corriente máx. por cada PIN de E/S: 40 mA
Memoria Flash	32 KB (ATmega328) de los cuales 2KB son usados por el bootloader
SRAM	2 KB
EEPROM	1 KB
Velocidad de Reloj	16MHz

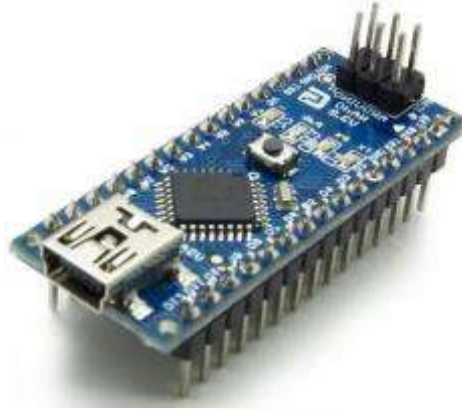


Fig.3.5 Arduino Nano
(Fuente: arduino.cl/arduino-nano/)

9.4 Diseño del Sistema

Se realiza el diseño de control de los relés para el encendido, apagado de luces y apertura de puertas con la polarización del transistor en configuración de interruptor o switch:

- Cuando se utiliza este tipo de configuración la corriente de base debe tener un valor para lograr que el transistor entre en corte y otro entre en saturación.
- Un transistor en corte tiene una corriente de colector (I_C) mínima (Prácticamente igual a cero) y un voltaje colector emisor (V_{CE}) máximo (casi igual al voltaje de alimentación).
- Un transistor en saturación tiene una corriente de colector (I_C) máxima y un voltaje colector emisor (V_{CE}) casi nulo (cero voltios).
- Para lograr que el transistor entre en corte, el valor de la corriente de base (I_B) debe ser bajo.
- Para lograr que el transistor entre en saturación, el valor de la corriente de base (I_B) debe calcularse dependiendo de la carga que se esté operando entre encendido y apagado (funcionamiento de interruptor o switch).

- Para que el transistor funcione en saturación, I_C debe ser máximo y V_{CE} mínimo y para que este en corte, I_C debe ser mínimo y V_{CE} el máximo.

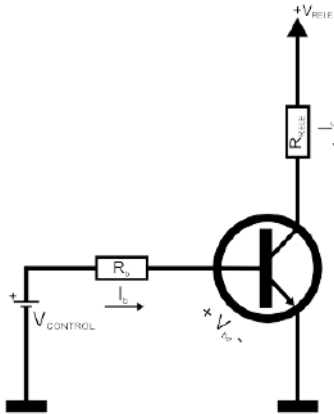


Figura 3.6 Circuito de activación de relé desde la salida de un microcontrolador (3.3V o 5V)
(Fuente: Autor)

Dónde:

- V_{RELE} : Tensión necesaria para activar el relé.
- R_{RELE} : Resistencia interna del circuito de control del relé. Este valor lo conseguimos del datasheet o lo podemos medir con un multímetro. Necesitaremos el valor de la resistencia para poder calcular la corriente que pasa por el circuito de control del relé.
- V_{CE} : Tensión entre colector y emisor.
- V_{BE} : Tensión entre base y emisor.
- R_b : Valor de la resistencia para controlar la corriente necesaria para activar la base del transistor.
- $V_{Control}$: Tensión que obtenemos del pin de salida del microcontrolador (3.3V/5V).

- Como se puede ver en el circuito hay dos mallas: la que incluye la tensión que activa el relé, el relé y la tensión que cae entre el colector y el emisor y la malla que incluye la tensión de control, la resistencia de base y la tensión que cae entre base y emisor.

i. Primera Malla:

$$-v_{RELE} + I_C R_{RELE} + v_{CE} \dots\dots\dots (1)$$

Cuando el transistor está saturado la caída de voltaje en V_{CE} es muy baja debido a que se comporta como interruptor cerrado. Por lo tanto, podemos reescribir la ecuación:

$$I_C = \frac{v_{RELE}}{R_{RELE}} \dots\dots\dots (2)$$

ii. Segunda Malla:

$$-v_{control} + I_B R_B + v_{BE} \dots\dots\dots (3)$$

Despejamos R_B y nos queda:

$$R_B = \frac{v_{control} - v_{BE}}{I_B} \dots\dots\dots (4)$$

Calculo final:

Relación Corriente BJT:

$$I_C = \beta I_B \dots\dots\dots (5)$$

Con esta relación de corrientes tenemos todo lo necesario para poder encontrar R_B :

$$R_B = \frac{(v_{control} - v_{BE})\beta}{I_C} \dots\dots\dots (6)$$

Donde:

- V_{BE} para los transistores de silicio es 0,6 V

- β se obtiene con un multímetro o en el datasheet del transistor.

Para el control de temperatura se utilizó el transistor de potencia de silicio TIP31c

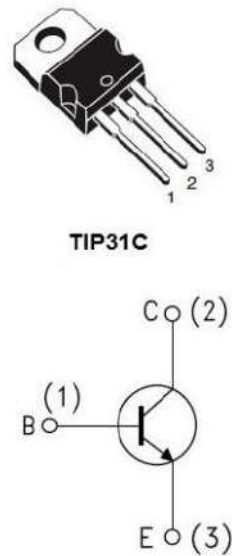


Figura 3.7 Transistor de Potencia TIP31C

(Fuente: www.electronicoscaldas.com/transistores-bjt/158-transistor-tip31c.html)

Se diseñó tres circuitos controladores de distintas funciones que se detalla a continuación:

- a) Control de Luces y Verificación del Estado de las Puertas

Para el control de Luces se hace la conexión a la Red LAN de la Facultad de Tecnología por el cual utilizaremos el módulo Ethernet ENC28J60.

Módulo ENC28J60

Este módulo permite conectar a internet cualquier microcontrolador con interfaz SPI y con la capacidad de memoria RAM y Flash para ejecutar un stack TCP/IP embebido (Microchip TCP/IP, uIP, etc.). Utiliza el Controlador Ethernet ENC28J60 de Microchip que posee las

características necesarias para manejar los requerimientos del protocolo de red. La tarjeta incluye todo lo necesario para el interfaz de red, incluido el conector Ethernet RJ45 con transformadores de aislamiento. Solamente hace falta conectarla a los pines SPI y a la alimentación.



Figura 3.8 Modulo Ethernet ENC28J60
(Fuente: www.nextiafenix.com/producto/modulo-ethernet-enc28jc60/)

Características:

- Voltaje de Operación: 3.3V - 5V
- Conector RJ45 HR911105A con transformadores de aislamiento incorporados
- LED de Encendido
- Interfaz Ethernet 802.3 10 Mbps Half o Full Duplex
- Interfaz con el microcontrolador host por SPI
- Soportado por el Stack TCP/IP de Microchip Technology
- Cristal de 25 Mhz¹⁹

¹⁹ (Internet de las cosas, 2017)

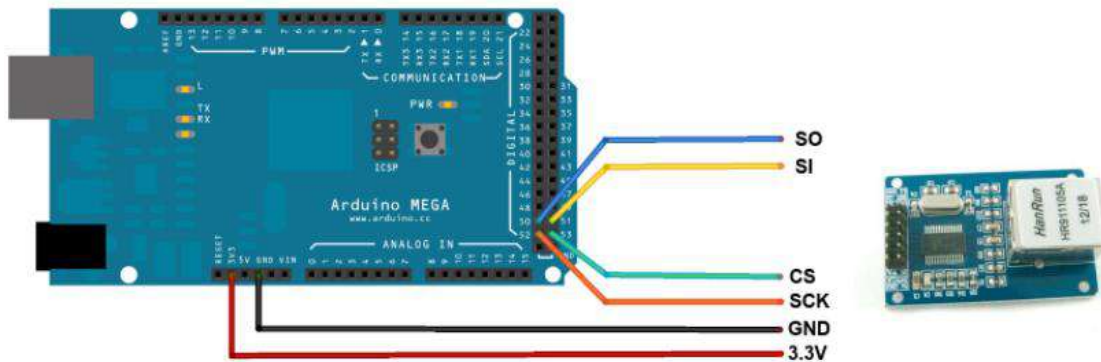


Figura 3.9 Conexión Modulo Ethernet ENC28J60
 (Fuente: en.code-bude.net/2013/06/22/how-to-use-enc28j60-ethernet-shield-with-arduino-mega-2560/)

Para la lectura el estado de las puertas (abierto o cerrado) se utiliza el sensor de final de carrera.

Sensor Final de Carrera

Estos sensores tienen dos tipos de funcionamiento: modo positivo y modo negativo.

En el modo positivo el sensor se activa cuando el elemento a controlar tiene una tarea que hace que el eje se eleve y se conecte con el objeto móvil con el contacto NC (normal cerrado). Cuando el muelle (resorte de presión) se rompe el sensor se queda desconectado.

El modo negativo es la inversa del modo anterior, cuando el objeto controlado tiene un saliente que empuje el eje hacia abajo, forzando el resorte de copa y haciendo que se cierre el circuito

Este sensor se conecta al Arduino Mega para controlar el estado de las puertas en los laboratorios de la Facultad de Tecnología Edificio Bloque “B”



Figura 3.10 Final de Carrera
(Fuente: tienda.bricogeek.com/interruptores/415-fin-de-carrera-con-palanca-5a.html)



b) Control de Temperatura

Para el control de temperatura se utiliza el sensor de temperatura DHT11. Este sensor se conecta en el Arduino Mega y activa un ventilador al pasar la temperatura de 20 °C.

Sensor de Temperatura DHT11

Algunos dispositivos son capaces de obtener varias mediciones en el mismo módulo. El módulo DHT11 capaz de representar digitalmente la humedad ambiental medida en % además de la temperatura en C°, tiene una precisión decimal y dispone de su propia librería que contiene los métodos para recoger sus mediciones.

Entre sus ventajas podemos mencionar el bajo costo y el despliegue de datos digitales. Esto supone una gran ventaja frente a los sensores del tipo análogo, como el LM335 por ejemplo, en los cuales las fluctuaciones en el voltaje alteran la lectura de datos.

Entre las desventajas, el DHT11 solo lee enteros, no podemos leer temperaturas con decimales. Para poder leer datos desde este sensor de una forma sencilla necesitamos descargar la librería DTH11.

Tabla 3.4:

Especificaciones Sensor de Temperatura (Fuente: www.picuin.com/es/uno-doc/dht11.html)

SENSOR DE TEMPERATURA (DHT11) ²⁰	
CARACTERISTICAS	
Temperatura	Resolución 16Bit
	Rango a 25°C ±2°C
	Tiempo de respuesta 1/e (63%) 10s
Características eléctricas	Fuente de alimentación DC 3.5 ~ 5.5V

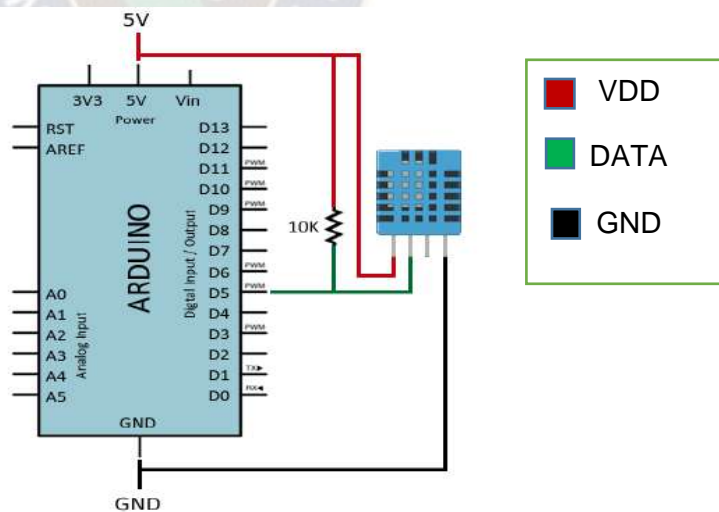


Figura 3.12 Conexión Sensor de Temperatura
(Fuente: www.luisllamas.es/arduino-dht11-dht22/)

²⁰ (Importadora Nova, DHT11, 2017)

Conexión del Display LCD 16x2 y módulo I2C

Se conecta al Arduino Nano, para mostrar los datos de la temperatura.

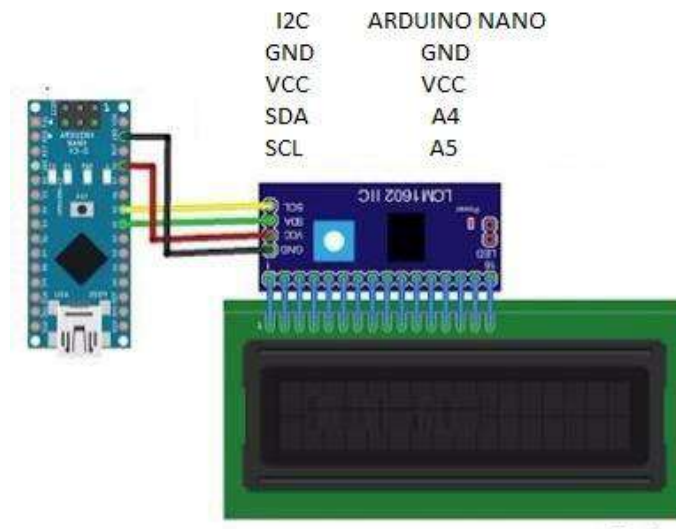


Figura 3.13 Conexión LCD Y Módulo I2C - Arduino Nano
(Fuente: articulo.mercadolibre.com.ve/MLV-489336282-pantalla-lcd1602-hd44780-16x2-con-interface-i2c-fondo-azul-_JM?quantity=1)

En la Fig. 3.14 se muestra la conexión de los sensores de temperatura DHT11 que activan los ventiladores al pasar la temperatura de 20 °C de los laboratorios de computación de la Facultad de Tecnología Edificio Bloque “B”, fue realizado en el simulador PROTEUS.

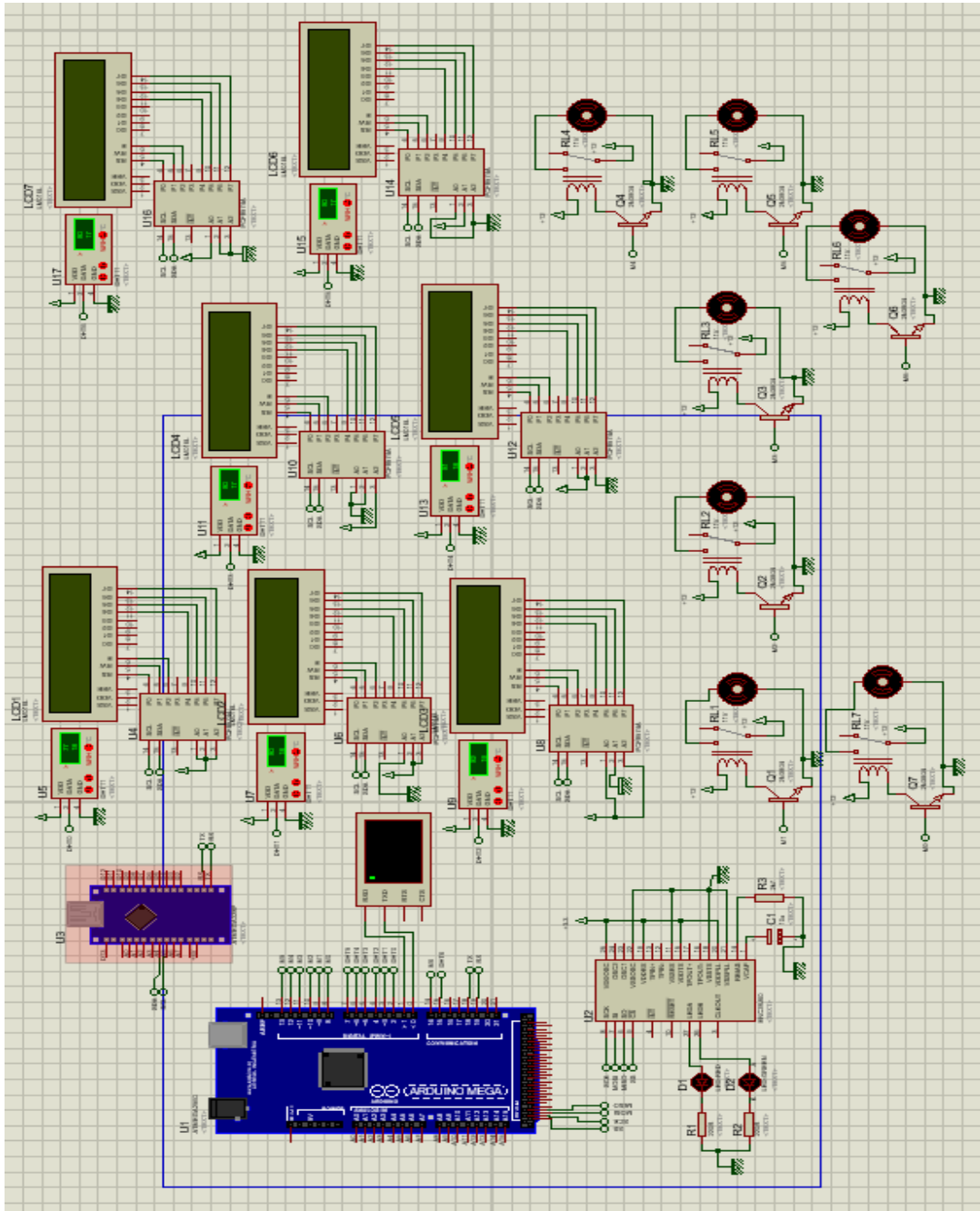


Figura 3.14 Circuito Control de Temperatura – Arduino Mega y Nano
(Fuente: Autor)

c) Control de Acceso

Para el control de acceso se utiliza el módulo RFID – RC522.

Modulo Lector RFID-RC522 RF

El Módulo Lector RFID-RC522 RF utiliza 3.3V como voltaje de alimentación y se controla a través del protocolo SPI, así como el protocolo UART, por lo que es compatible con casi cualquier micro controlador, Arduino o tarjeta de desarrollo. El RC522 también utiliza un sistema avanzado de modulación y demodulación para todo tipo de dispositivos pasivos de 13.56Mhz. Puesto que se hará una lectura y escritura de la tarjeta, es necesario conocer las características de los bloques de memoria una tarjeta: La tarjeta que viene con el módulo RFID cuenta con 64 bloques de memoria (0-63) donde se hace lectura y/o escritura. Cada bloque de memoria tiene la capacidad de almacenar sobre todo hasta 16 Bytes. El número de serie consiste de 5 valores hexadecimales, se podría utilizar esto para hacer una operación dependiendo del número de serie.

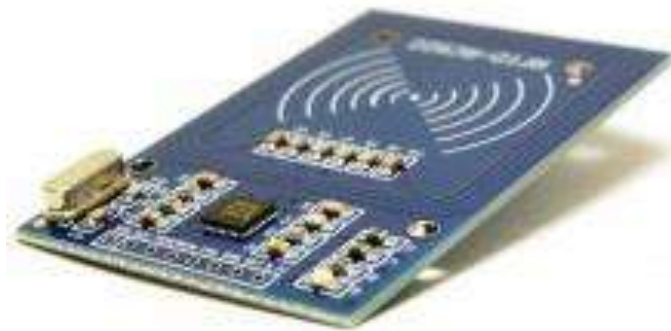


Figura 3.15 Modulo RFID-RC522

(Fuente: www.nova.com.bo/modulo-rfid-rc522-1356mhz-rc522-s50-mifare.html)

Características del módulo lector RFID-RC522:

- Modelo: MF522-ED
- Corriente de operación: 13-26mA a 3.3V
- ISB de stand by: 10-13mA a 3.3V
- ISM de sleep-mode: <80uA
- IM máxima: 30mA
- Frecuencia de operación: 13.56Mhz
- Distancia de lectura: 0 a 60mm
- Protocolo de comunicación: SPI
- Velocidad de datos máxima: 10Mbit/s
- Dimensiones: 40 x 60 mm
- Temperatura de operación: -20 a 80°C
- Humedad de operación: 5%-95%
- Máxima velocidad de SPI: 10Mbit/s²¹

²¹ (Importadora Nova, 2017)

Tabla 3.5:

Conexión Módulo RFID (Fuente: Autor)

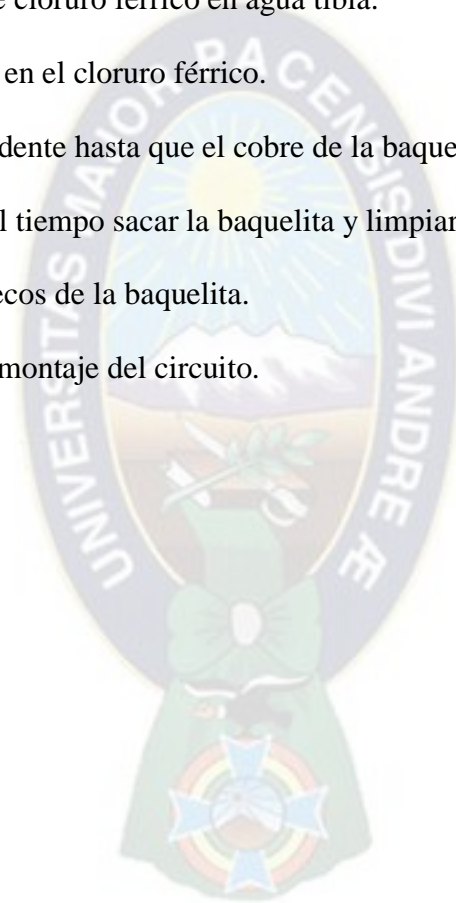
Módulo RC522	Arduino Mega (Pines)
SDA (SS)	53
SCK	52
MOSI	51
MISO	50
IRQ	No conectado
GND	GND
RST	9
3.3V	3.3V

9.5 Construcción de las Placas Electrónicas

Placa electrónica es la superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora. El circuito impreso se utiliza para conectar eléctricamente a través de las pistas conductoras, y sostener mecánicamente, por medio de la base, un conjunto de componentes electrónicos, las pistas es generalmente de cobre mientras que la base se fabrica de resinas de fibra de vidrio reforzada, Pertinax, pero también cerámica, plástico, teflón o polímeros como la baquelita.

Se detalla a continuación los pasos a seguir para la construcción de la placa electrónica para el control de luces y estados de puertas, control de temperatura y apertura de puertas mediante módulo RFID:

- Se crea el plano del circuito impreso en el programa Proteus (ARES).
- Traspasar el plano impreso en papel termotransferible a la baquelita.
- Preparar la solución de cloruro férrico en agua tibia.
- Introducir la baquelita en el cloruro férrico.
- Esperar un tiempo prudente hasta que el cobre de la baquelita se quite.
- Cuando haya pasado el tiempo sacar la baquelita y limpiar con alcohol.
- Luego perforar los huecos de la baquelita.
- Por último, se hace el montaje del circuito.



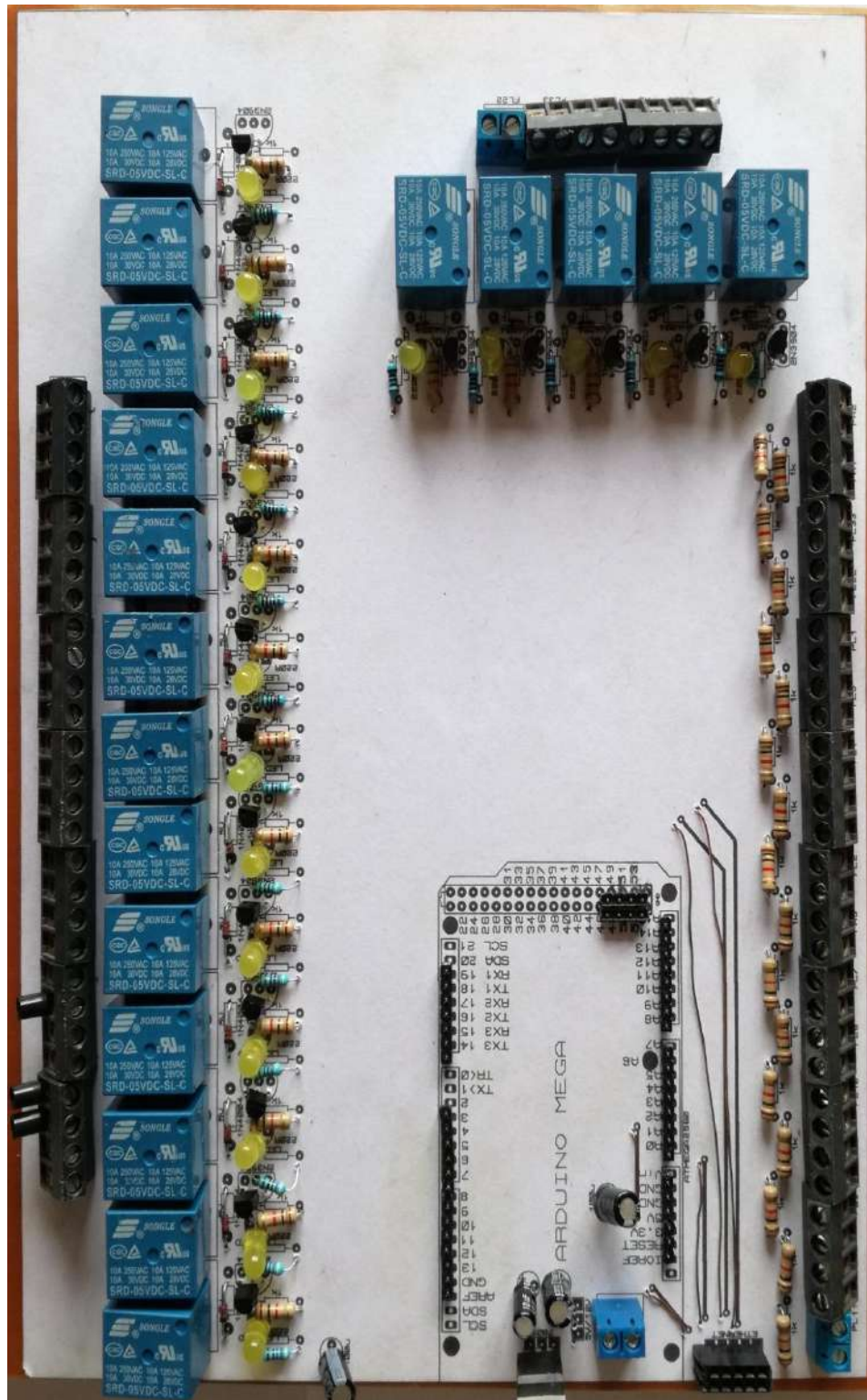


Figura 3.16b Placa Electrónica Control de Luces y Estado de las Puertas (Lado A)
(Fuente: Autor)

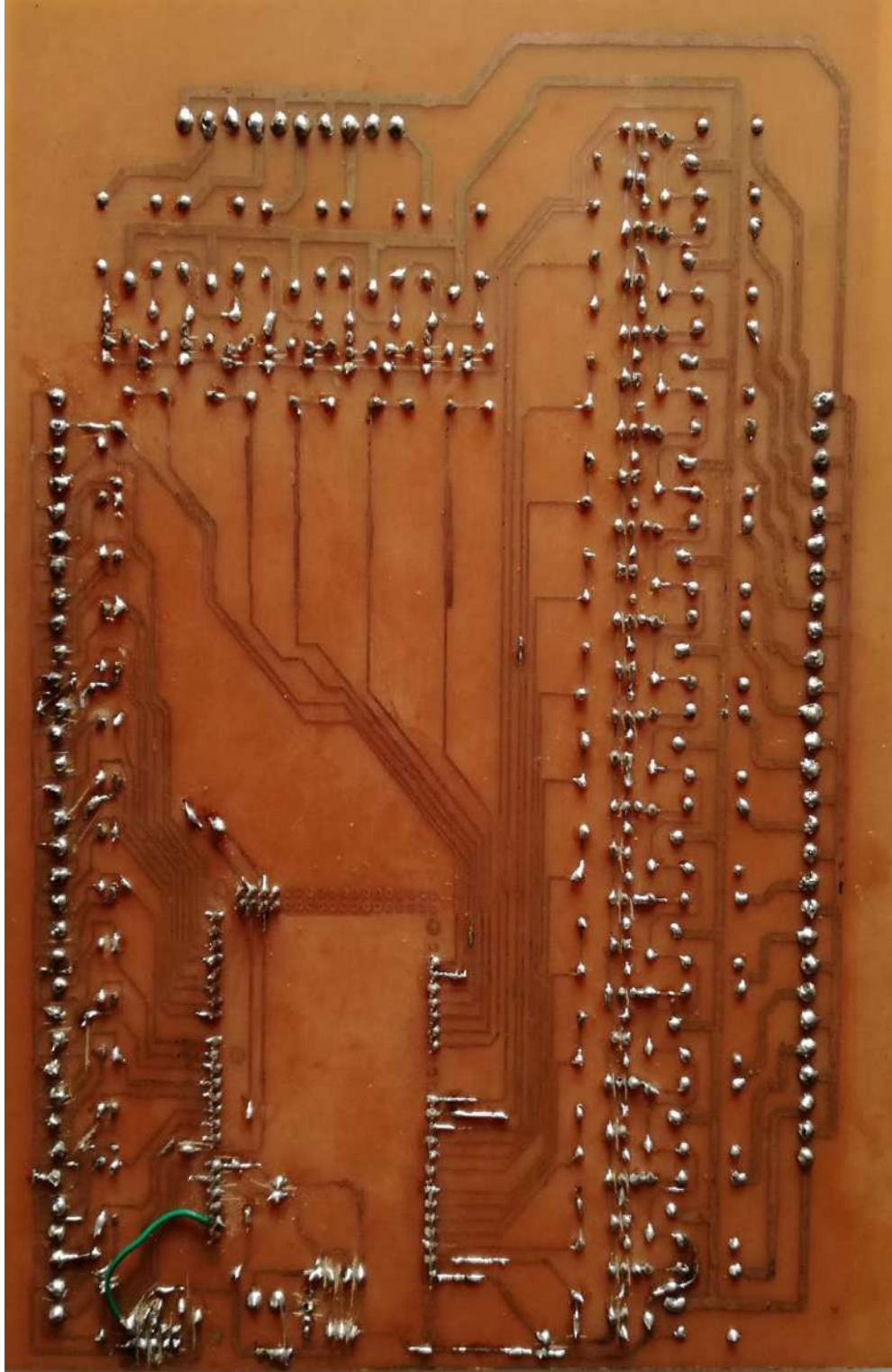


Figura 3.16c Placa Electrónica Control de Luces y Estado de las Puertas (Lado B)
(Fuente: Autor)

b) Diseño de la construcción de la placa de control de temperatura:

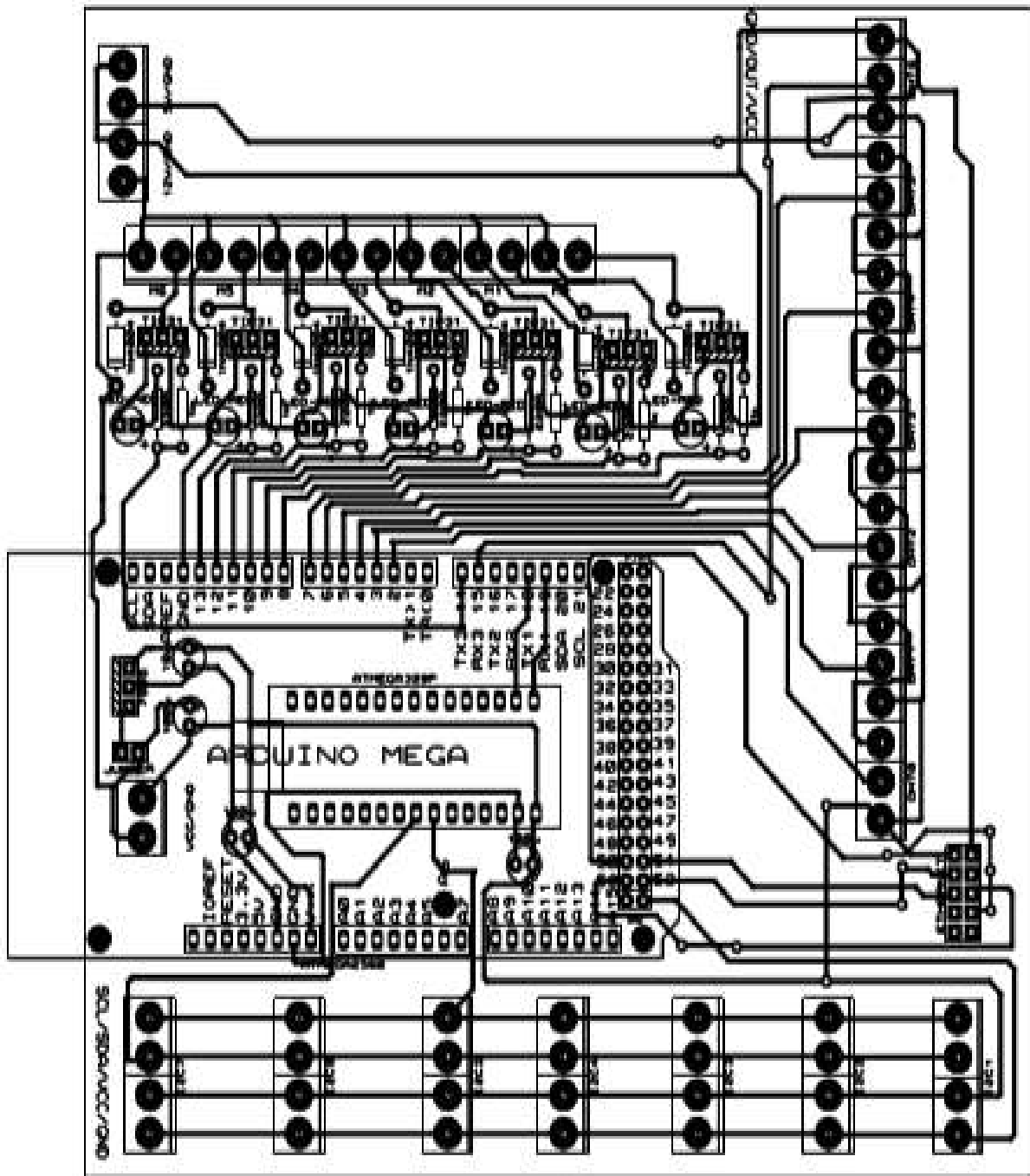


Figura 3.17a Plano Impreso – Control de Temperatura
(Fuente: Autor)

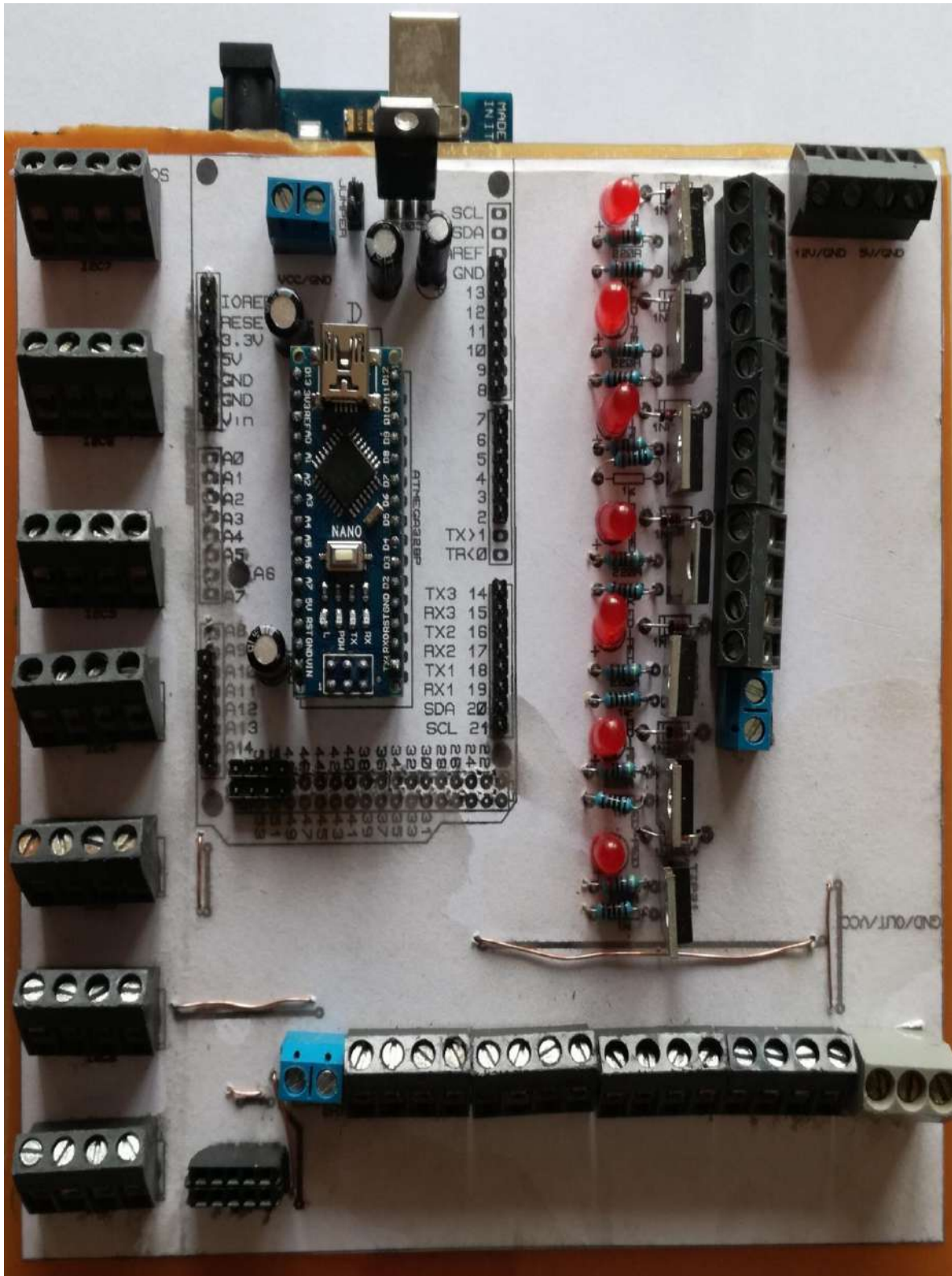


Figura 3.17b Placa Electrónica Control de Temperatura (Lado A)
(Fuente: Autor)

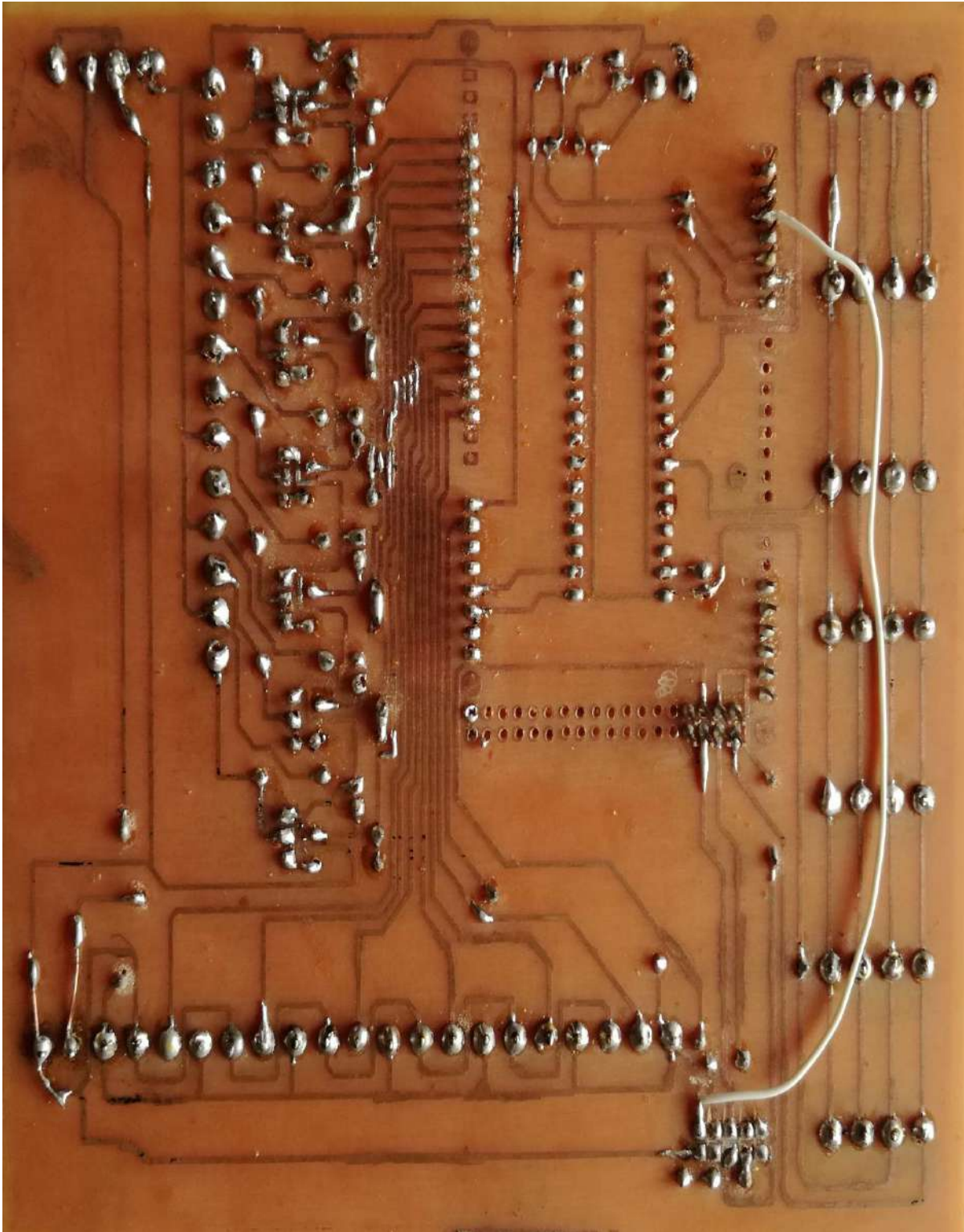


Figura 3.17c Placa Electrónica Control de Temperatura (Lado B)
(Fuente: Autor)

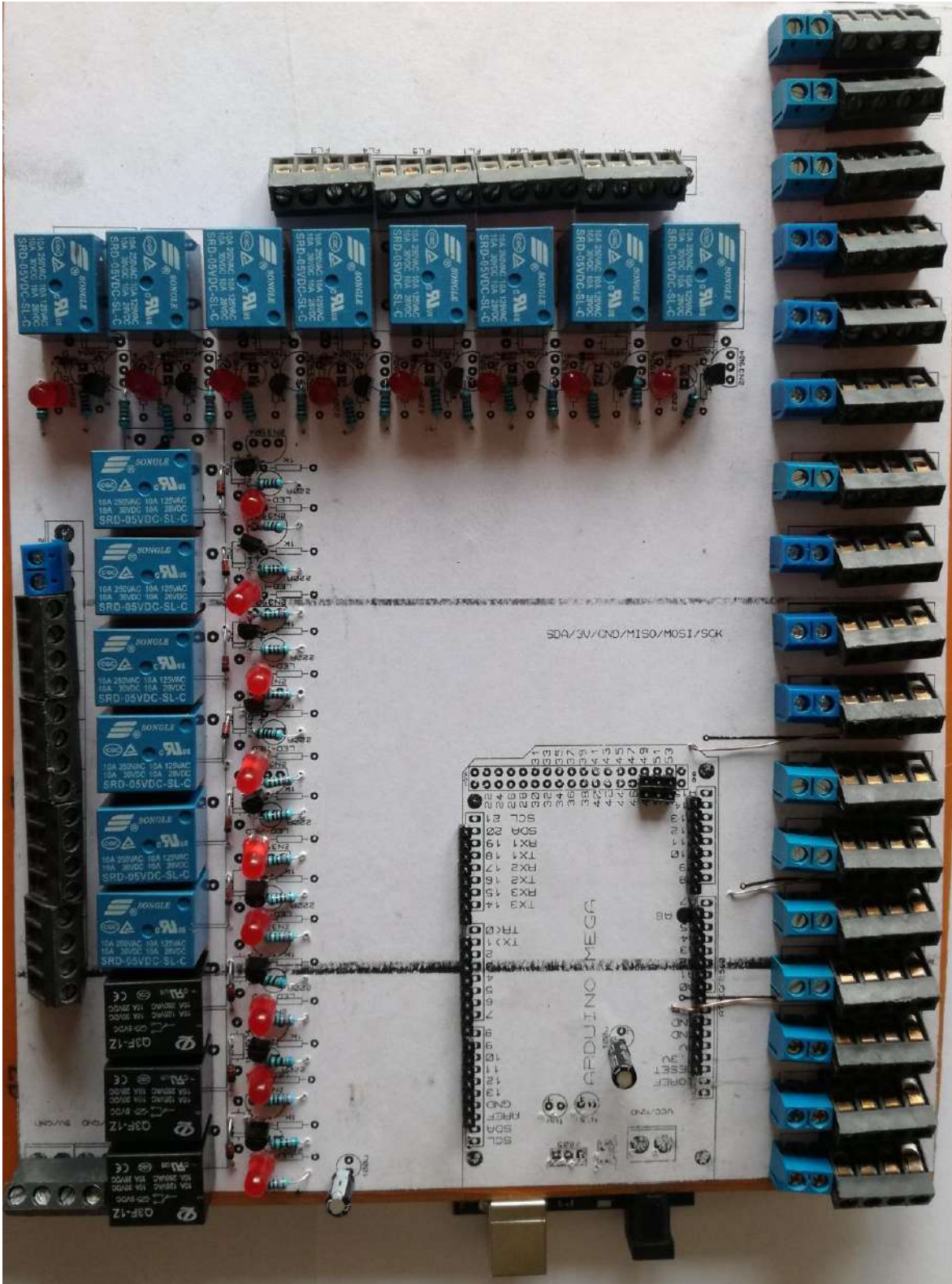


Figura 3.18b Placa Electrónica Control de Acceso Modulo RFID (Lado A)
(Fuente: Autor)

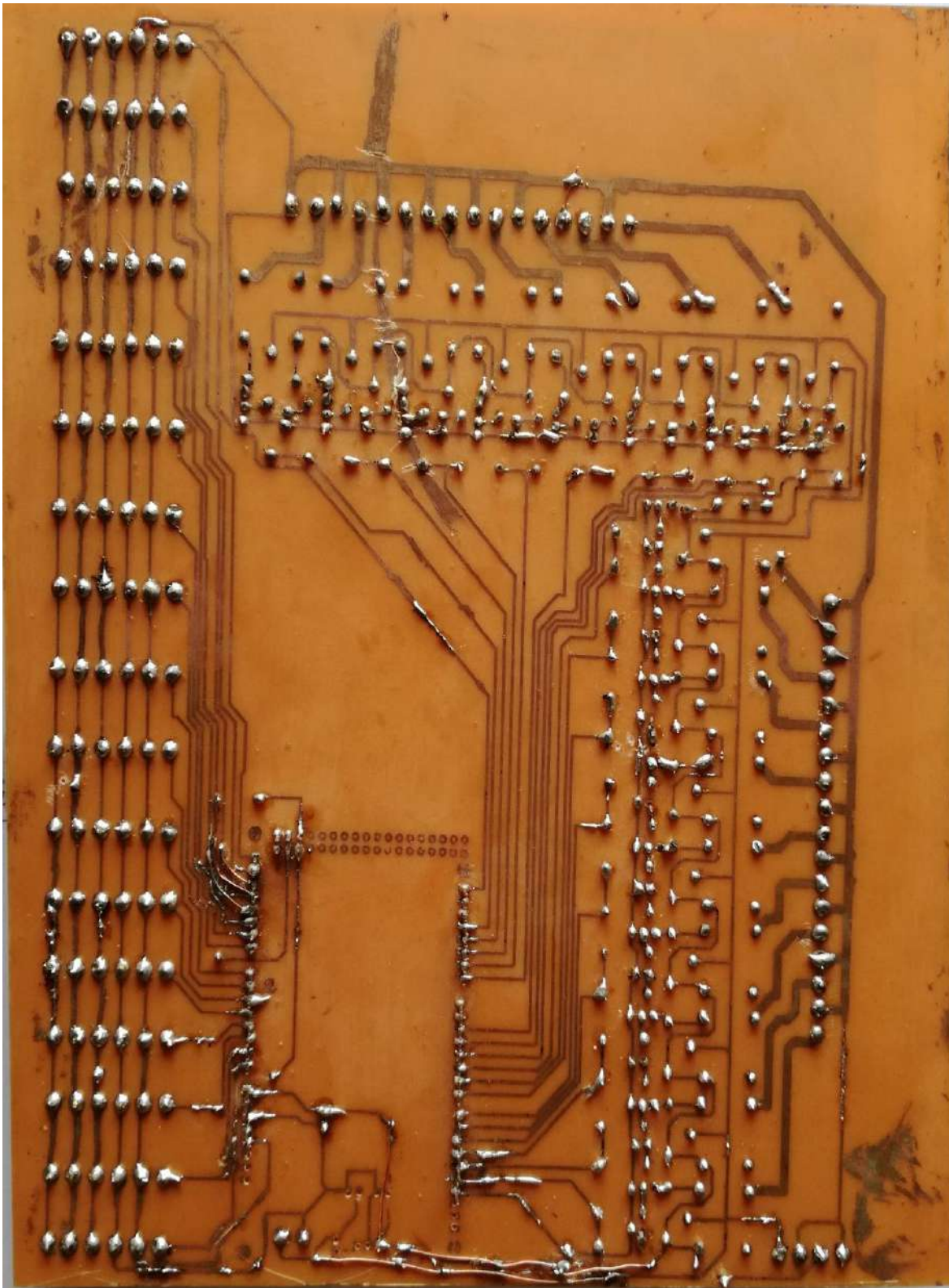


Figura 3.18c Placa Electrónica Control de Acceso Modulo RFID (Lado B)
(Fuente: Autor)

9.6 Software

Para realizar el proyecto de grado “SISTEMA DE AUTOMATIZACION PARA LA FACULTAD DE TECNOLOGIA (EDIFICIO BLOQUE “B”) POR MEDIO DE DISPOSITIVOS ELECTRONICOS”, se realizó la programación de la aplicación Android y la programación de control en el Arduino mega.

9.6.1 IDE Arduino

El entorno de desarrollo integrado también llamado IDE (sigla en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios lenguajes.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además, en el caso de Arduino incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware.

El IDE de Arduino va a ser la herramienta de trabajo durante el desarrollo del proyecto de grado.

Los programas de Arduino están compuestos por un solo fichero con extensión “.ino”, aunque es posible organizarlo en varios ficheros. El fichero principal siempre debe estar en una carpeta con el mismo nombre que el fichero.²²

²² (Arduino, 2017)

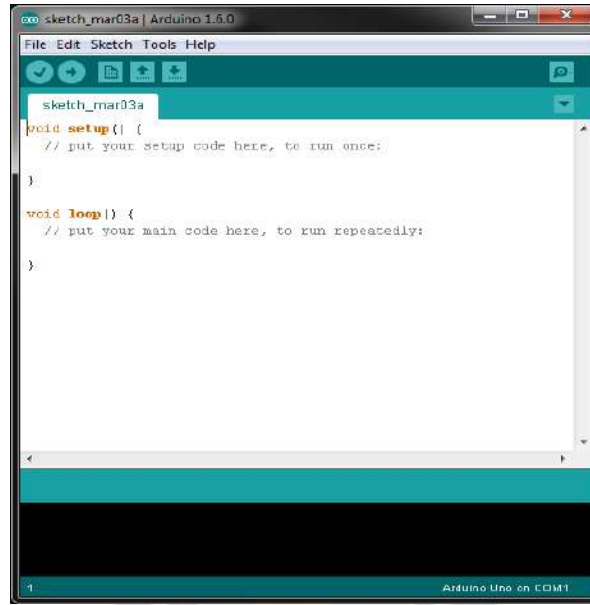


Figura 3.19 Entorno de Programación IDE Arduino
(Fuente: aprendiendoarduino.wordpress.com/2016/03/30/instalacion-del-ide-arduino/)

9.6.1.1 Estructura de un Programa

Un programa de Arduino o proyecto tiene la extensión `.ino`, no es importante que el programa este en único fichero, pero si es imprescindible que todos los ficheros estén dentro del mismo directorio que el fichero principal.

La estructura básica de un programa de Arduino es bastante simple y se compone de al menos dos partes. Estas son obligatorias y encierran bloques que contienen declaraciones, estamentos o instrucciones:

- `Setup ()`, es a parte encargada de recoger la configuración
- `Loop ()`, es la que contiene el programa que se ejecuta cíclicamente

9.6.1.2 Lenguaje de programación Arduino

Las herramientas necesarias para programar los microcontroladores AVR de Atmel son avr-binutils, avr-libc y ya están incluidas en el IDE de Arduino, pero cuando compilamos y cargamos un programa estamos usando estas herramientas.

Arduino ofrece una api o core que facilitan la programación de los pines de entrada y salida de los puertos de comunicación, así como otras librerías para operaciones específicas. El propio IDE ya incluye estas librerías y no es necesario declararlas expresamente.

9.6.2 Aplicación Android

Para la realización de la aplicación Android se utiliza el programa App Inventor 2 que es un entorno de desarrollo de aplicaciones para dispositivos Android.

Al construir las aplicaciones para Android se trabaja con dos herramientas: App Inventor Designer y App Inventor Blocks Editor.

- a) En Designer se construye el Interfaz de Usuario, eligiendo y situando los elementos con los que interactuará el usuario y los componentes que utilizará la aplicación.
- b) En el Blocks Editor se define el comportamiento de los componentes de la aplicación.

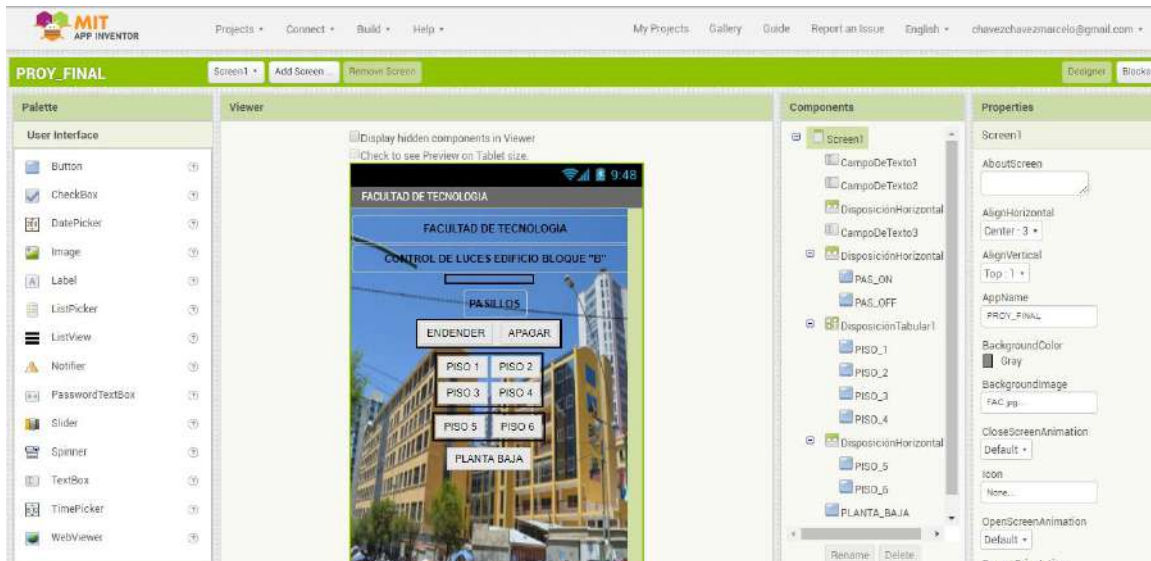


Figura 3.20 Entorno Diseño App Inventor 2
(Fuente: Autor)



Figura 3.21 Entorno Bloques App Inventor 2
(Fuente: Autor)

9.6.3 Diseño de la Aplicación en App Inventor

En el proyecto es incorporado el modulo ethernet con la finalidad de poder conectar el celular Android con el Sistema de Automatización para la Facultad De Tecnología (Edificio Bloque “B”) por Medio de Dispositivos Electrónicos, una vez conectados es posible realizar el control de encendido – apagado de luces, verificación del estado de puertas (abierto o cerrado) y la temperatura de los ambientes de laboratorios.

Se diseñó la aplicación mediante el programa App Inventor2, con una pantalla principal donde se podrá acceder al menú de acceso a sub menús de control de luces, verificación de estado de puertas y temperatura de los ambientes de laboratorios, en las figuras 3.4, 3.5, 3.6, 3.7a, 3.7b, 3.7c y 3.7d se muestran el diseño de la pantalla principal, el sub menú y la programación utilizada en cada piso.

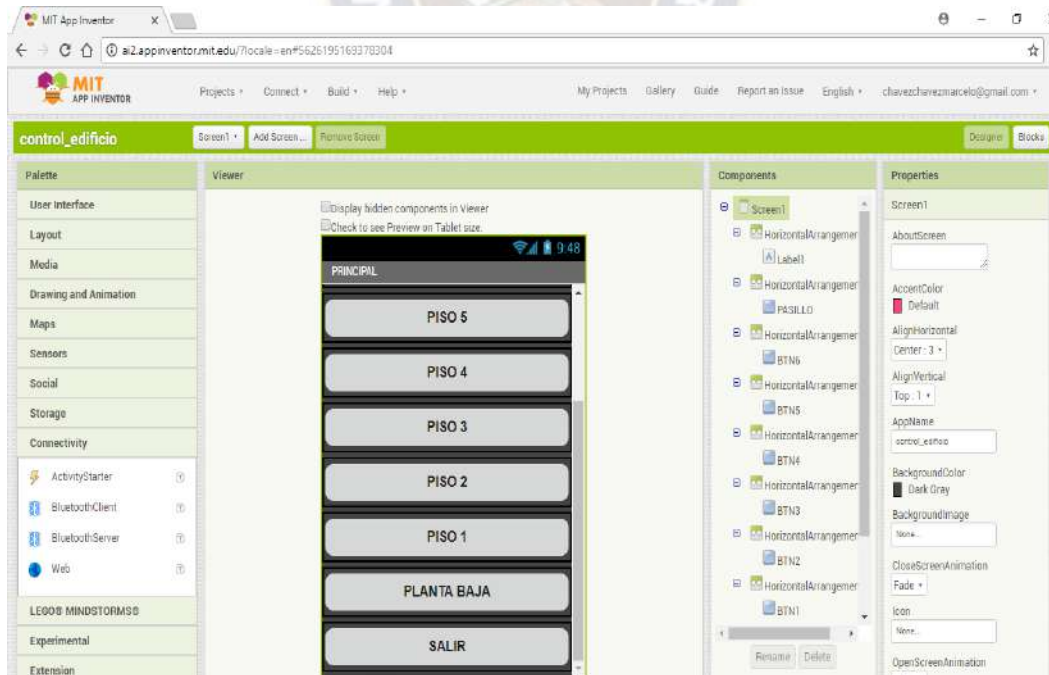


Figura 3.22 Diseño de la pantalla principal
(Fuente: Autor)



Figura 3.23 Código de pantalla principal
(Fuente: Autor)

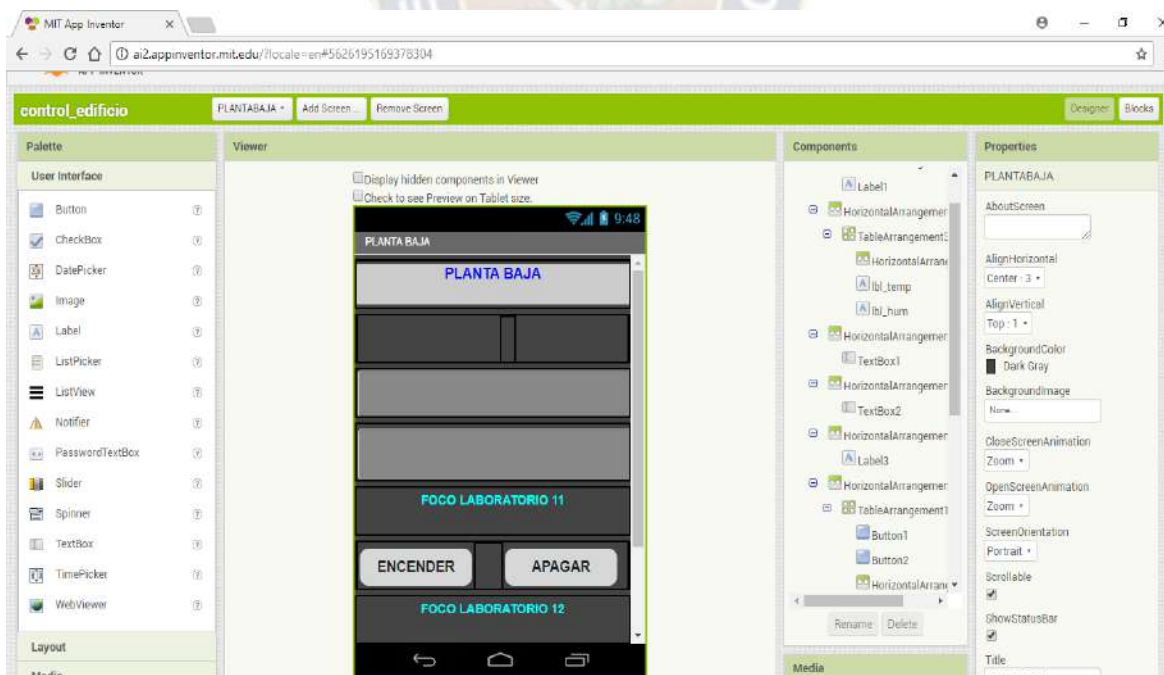


Figura 3.24 Diseño de submenú utilizado en cada piso
(Fuente: Autor)

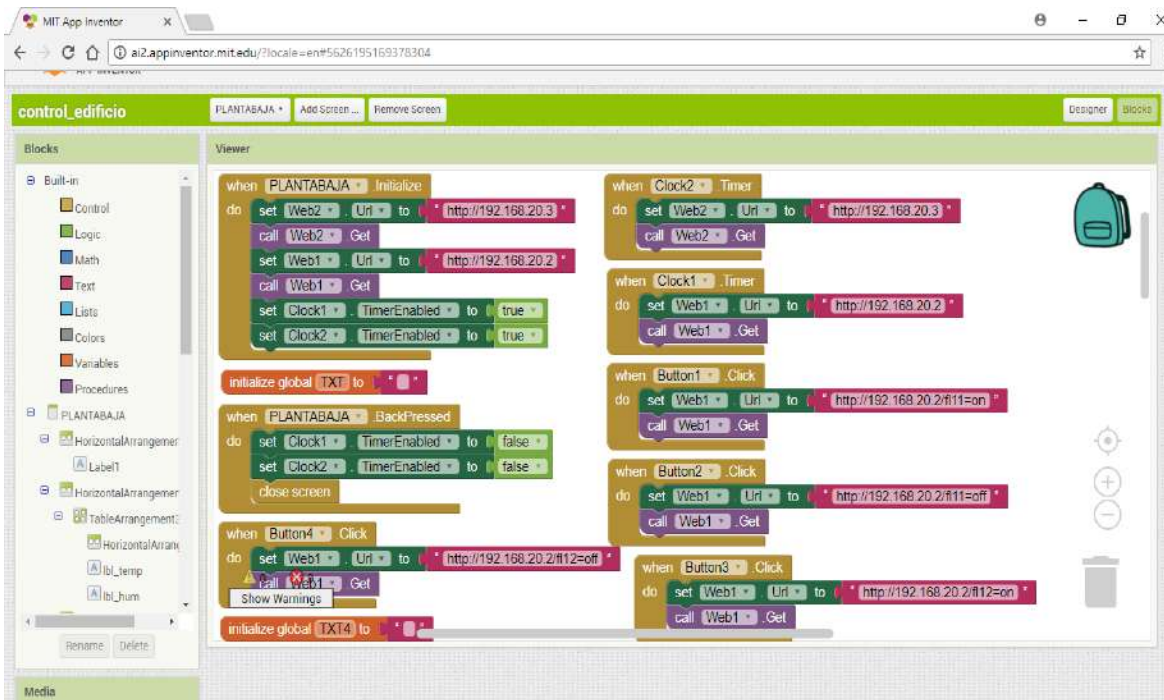


Figura 3.25a Código submenú utilizado en cada piso
(Fuente: Autor)

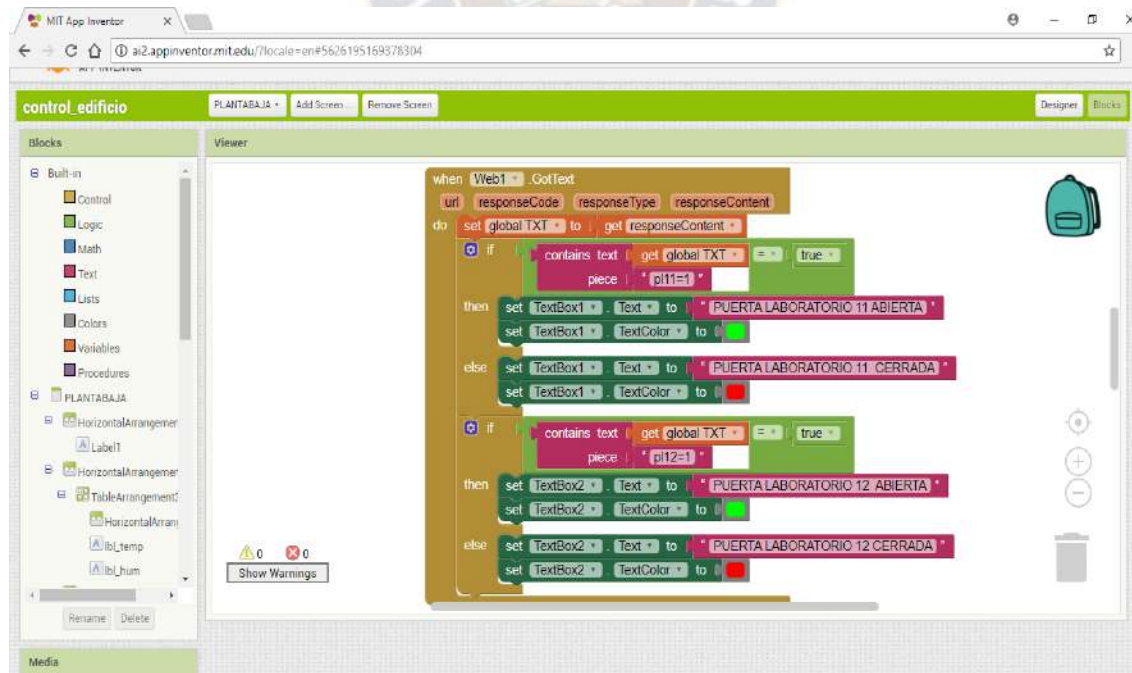


Figura 3.25b Código submenú utilizado en cada piso
(Fuente: Autor)

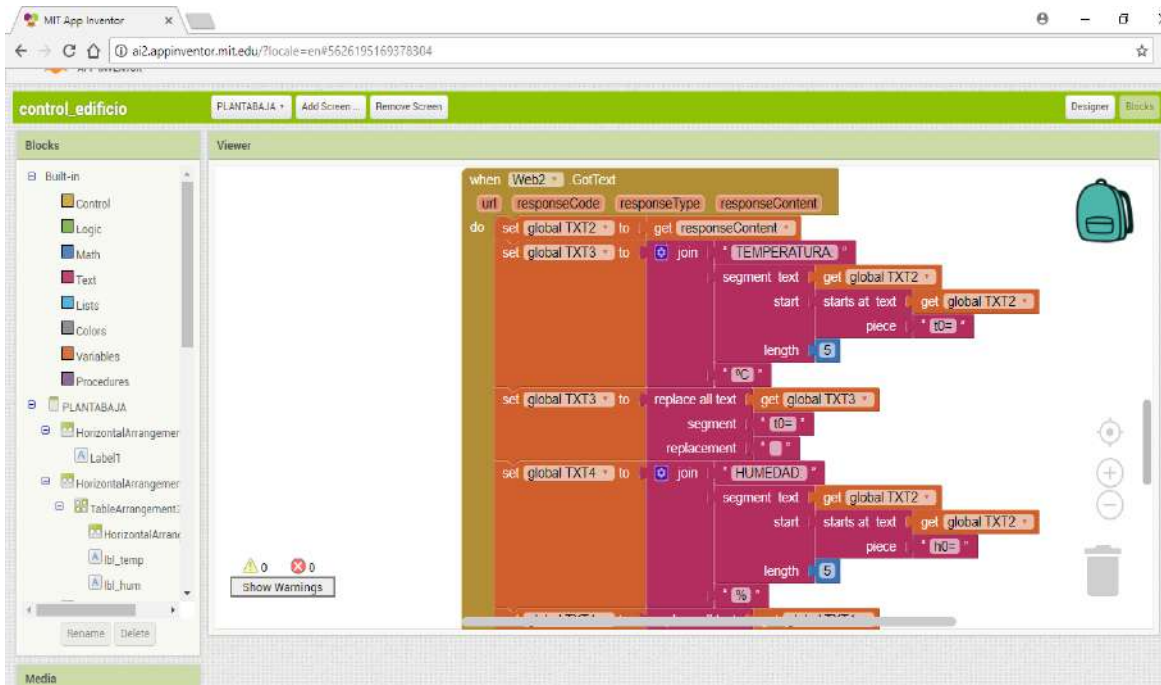


Figura 3.25c Código submenú utilizado en cada piso
(Fuente: Autor)

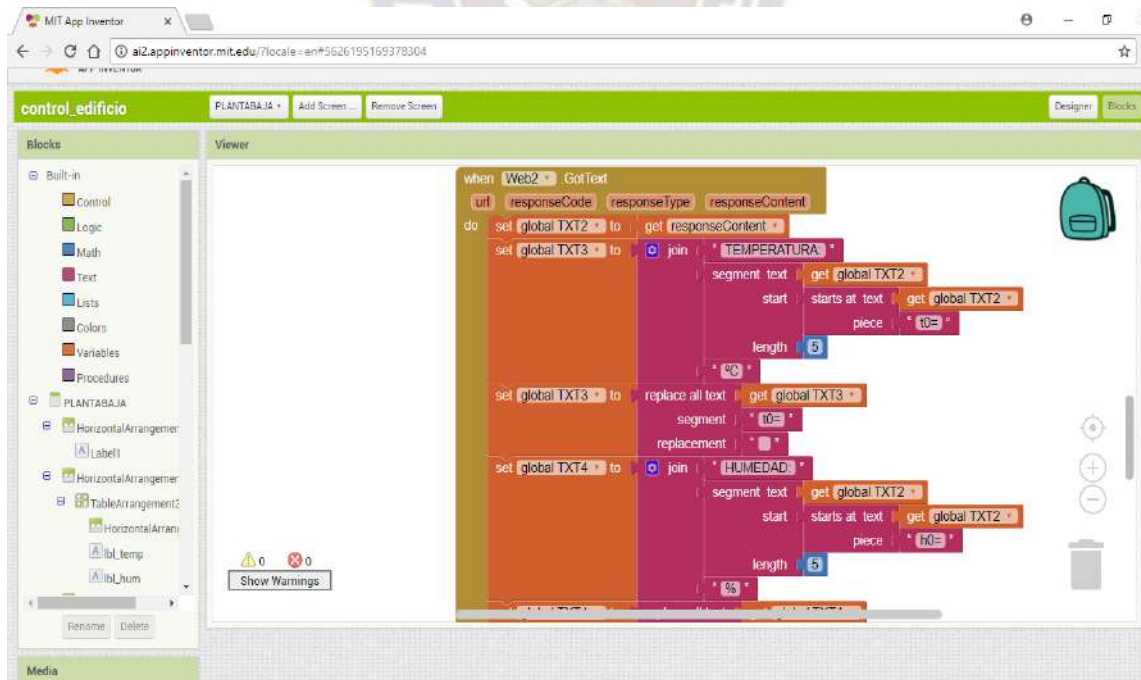


Figura 3.25d Código submenú utilizado en cada piso
(Fuente: Autor)

9.6.4 Envió de Datos

El envío de datos se realiza a través de la aplicación Android, el modulo procesa el servicio si está disponible.

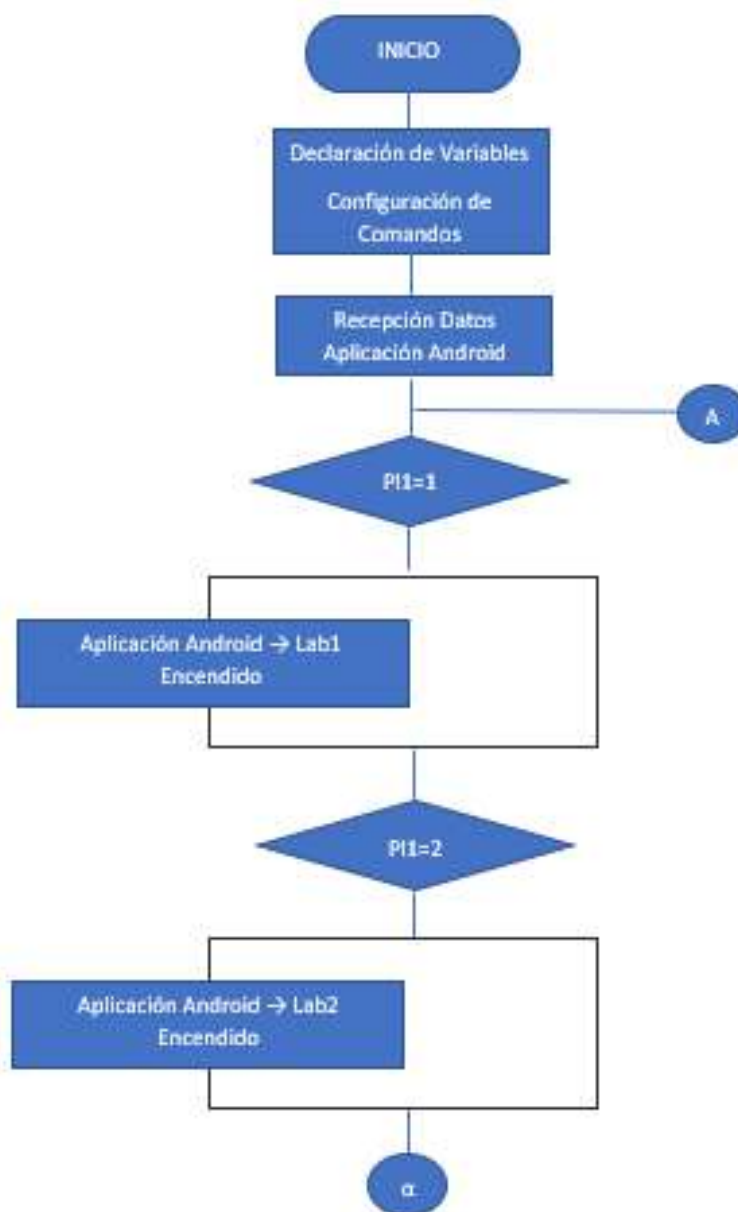


Figura 3.26a Diagrama de Flujo del programa principal Luces parte 1 (Fuente: Autor)

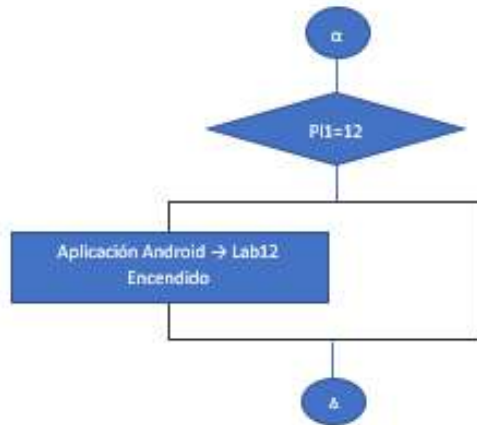


Figura 3.26b Diagrama de Flujo del programa principal Luces parte 2 (Fuente: Autor)

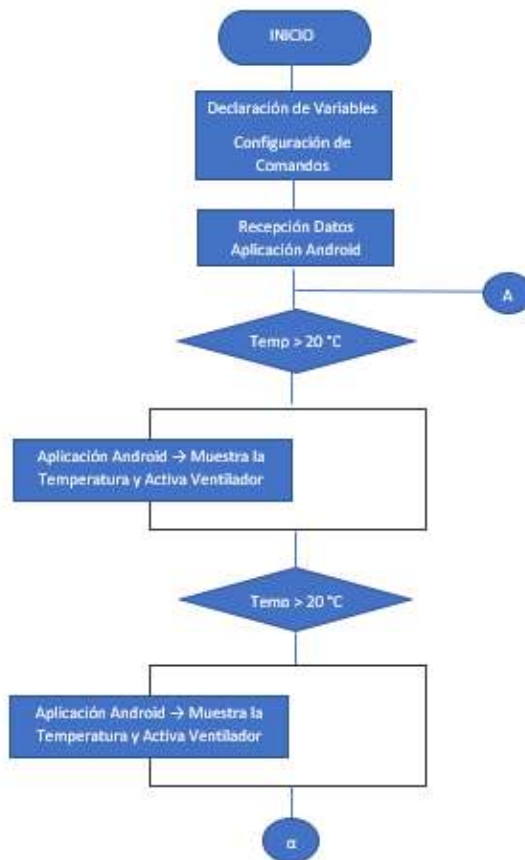


Figura 3.27a Diagrama de Flujo del programa principal Temperatura parte 1 (Fuente: Autor)

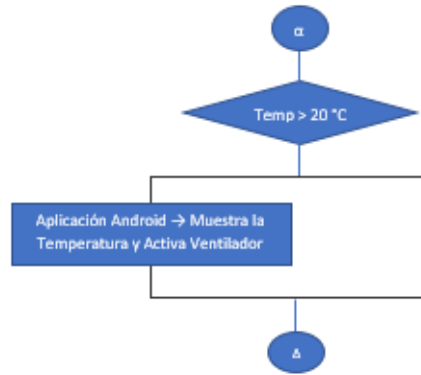


Figura 3.27b Diagrama de Flujo del programa principal Temperatura parte 2
(Fuente: Autor)

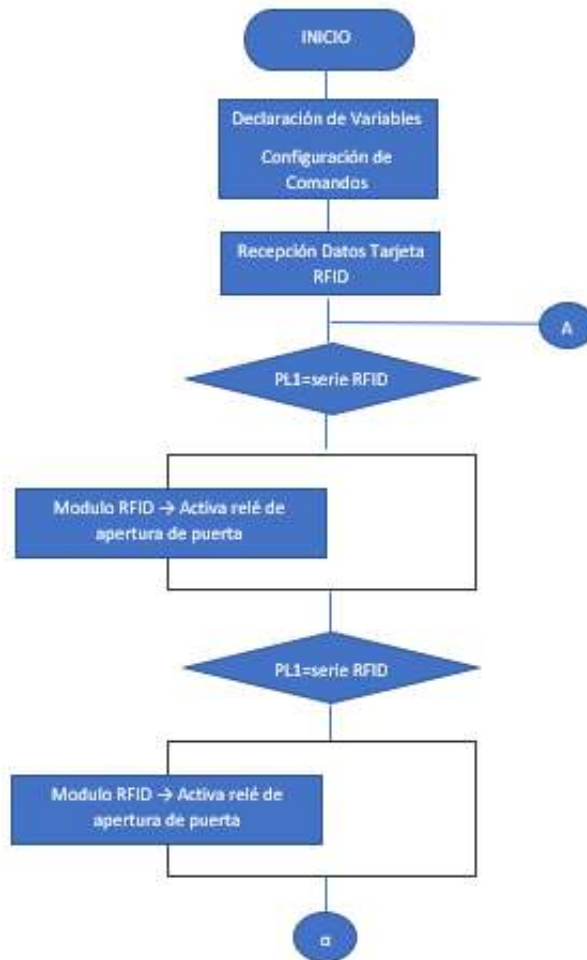


Figura 3.28a Diagrama de Flujo del programa principal Control de Acceso parte 1
(Fuente: Autor)

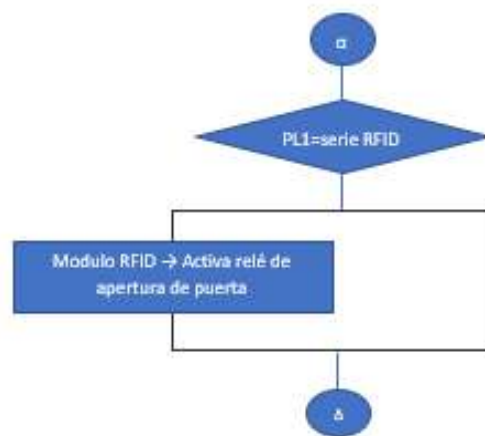


Figura 3.28b Diagrama de Flujo del programa principal Control de Acceso parte 2
(Fuente: Autor)

9.7 Pruebas y Resultados de Funcionamiento del Sistema

Para verificar el funcionamiento del proyecto se implementa un ambiente de prueba de acuerdo a los objetivos y requerimientos establecidos. En el montaje de este ambiente se ha contemplado el uso de un hardware como una Tablet normal sobre el cual se instalará la APK diseñada para el control de luces y verificación del estado de las puertas (abierto o cerrado).

Una vez instalado los diferentes tipos de sensores, se realiza las pruebas de acuerdo a las funcionalidades que se han establecido para el sistema como las posibilidades de elevación de temperatura en laboratorios de computación, el acceso mediante la tarjeta magnética y el control de encendido y/o apagado de luces.



Figura 3.29a Tablero Demostrativo
(Fuente: Autor)



Figura 3.29b Ventana Principal Aplicación Android
(Fuente: Autor)

a) Prueba control de encendido y apagado de luces



Figura 3.30a Ventana Planta Baja Aplicación Android
(Fuente: Autor)

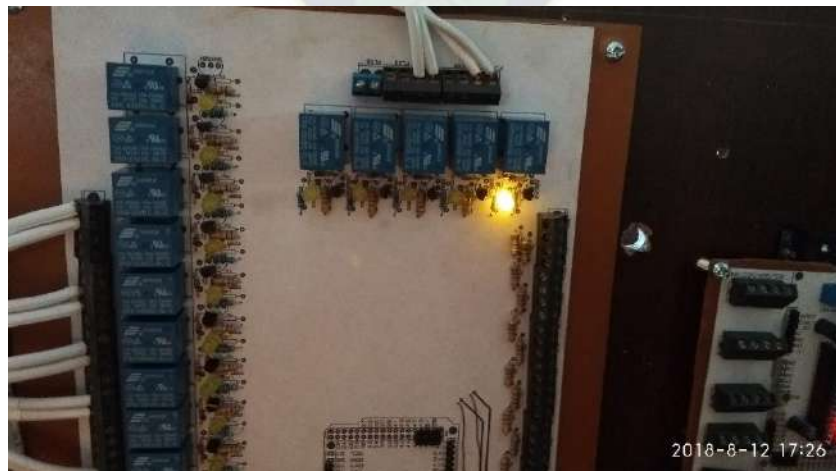


Figura 3.30b Indicador Led Encendido de Luz
(Fuente: Autor)



Figura 3.30c Encendido de Luz Mediante la Aplicación Android
(Fuente: Autor)

b) Control de temperatura



Figura 3.31a Indicador de Temperatura Aplicación Android
(Fuente: Autor)



Figura 3.31b Indicador Pantalla LCD
(Fuente: Autor)

c) Apertura de Puertas Modulo RFID

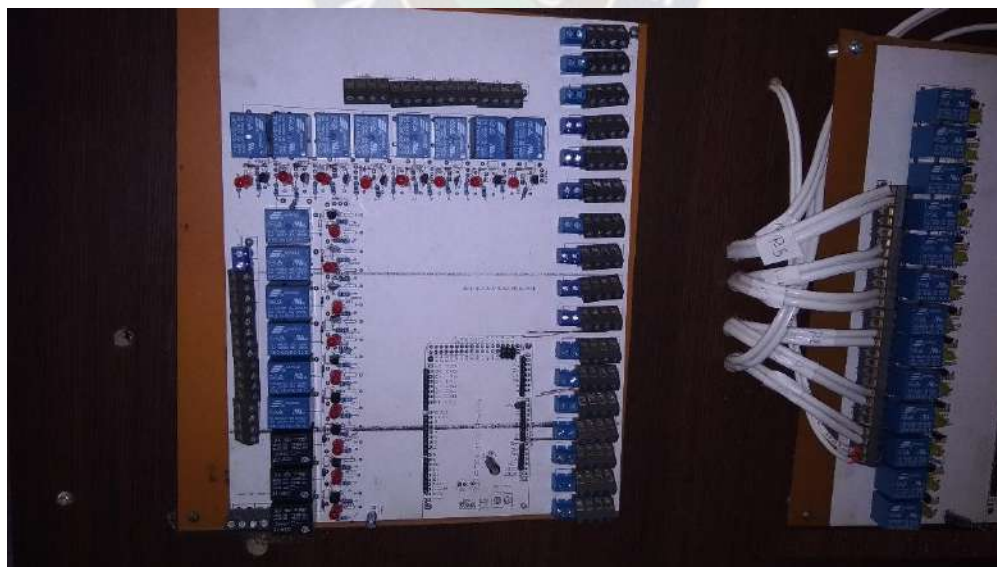
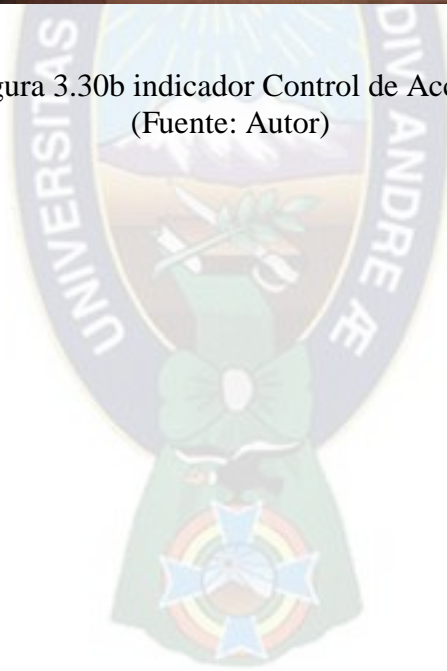


Figura 3.30a Circuito Control de Acceso
(Fuente: Autor)



Figura 3.30b indicador Control de Acceso
(Fuente: Autor)



10 Costos

Para el costo del presente proyecto de grado y la instalación de Sistema de Automatización para la Facultad de Tecnología (edificio bloque “b”) por medio de dispositivos electrónicos se indican en la siguiente tabla:

Tabla 4.1:

Costos (Fuente: Autor)

Componente	Cantidad	Costo Unitario (Bolivianos)	Costo (Bolivianos)
Tarjeta Arduino Mega 2560	3	300.-	900.-
Tarjeta Arduino Nano	1	150.-	150.-
Modulo Ethernet	2	100.-	200.-
Módulo 4 de relés	34	5.-	170.-
Sensor de Temperatura	7	50.-	350.-
Display 16x2	7	25.-	175.-
Módulo I2C	7	35.-	245.-
Módulo RFID	34	100.-	3400.-
Sensor final de carrera	34	2.-	68.-
Chapa electrónica	34	600.-	20400.-
TOTAL			26058.-

11 Cronograma de Actividades

N°	TIEMPO ACTIVIDAD	PERIODO DE EJECUCION (2018)				
		JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE
1	Planificación del software a utilizar					
2	Instalación, configuración de Arduino y Android					
3	Planificación de los sistemas de control automático de luz, temperatura y apertura de puertas					
4	Diseño de los circuitos a utilizar					
5	Instalación y pruebas de los sistemas					

12 Conclusiones

- Se diseñó el Sistema de Automatización para la Facultad de Tecnología (Edificio bloque “B”) por medio de dispositivos electrónicos con las condiciones adecuadas y necesarias, en cuanto al diseño de temperatura, apagado, encendido de luces y el sistema de control de apertura de puertas se tuvo un resultado óptimo en el funcionamiento.
- Al hacer las pruebas se logró hacer uso de los distintos sensores electrónicos, demostrando una mayor eficacia en la disponibilidad del sistema, con el fin de incrementar el ahorro de la energía y recursos.
- Realizando las pruebas del diseño de control de luces y estado de puertas se demostró que el diseño es estable y un funcionamiento óptimo.
- Efectuando las pruebas del diseño de control de temperatura se tuvo un resultado de un diseño estable y un rendimiento adecuado a las necesidades de los laboratorios.
- En las pruebas del diseño de apertura de puertas se obtuvo un circuito estable y un funcionamiento correcto para el ingreso a los laboratorios por medio de los módulos RFID

- Realizando las pruebas del diseño de la aplicación que controla los distintos sensores para los laboratorios de la Facultad de Tecnología Edificio Bloque “B”, se tuvo un funcionamiento correcto en cuanto al manejo y funcionamiento del sistema.
- Se midió la capacidad real del sistema realizando pruebas, de acuerdo a la plataforma Arduino, todas las pruebas fueron satisfactorias.

13 Recomendaciones

- Verificar el diseño de las placas electrónicas aspectos importantes: Comprobar la continuidad, posibles cortos, soldaduras frías, etc. Y realizar varias pruebas para un buen funcionamiento.
- Instalación: Como prioridad se sugiere que el sistema de automatización debe ser instalado en el centro de cómputo de la Facultad de Tecnología, debido a que el sistema necesitará un ambiente con ventilación visto que estará funcionando las 24 Horas del día.
- Para un buen desempeño de este sistema el voltaje debe estar no menos de los 12v., la alimentación del sistema debe estar adecuadamente instalado para evitar corto circuitos que puedan afectar a las placas electrónicas.
- Administración: Se recomienda que el uso del Sistema de Automatización sea por el administrador de la Facultad de Tecnología, así se podrá tener un mejor uso y rendimiento del sistema.

14 Bibliografía

- Aosong*. (23 de Agosto de 2017). Obtenido de <https://akizukidenshi.com/download/ds/aosong/DHT11.pdf>
- AppInventor*. (2017). Obtenido de https://es.wikipedia.org/wiki/App_Inventor
- Arduino*. (23 de Agosto de 2017). Obtenido de <https://translate.google.com/translate?hl=es&sl=en&u=https://www.arduino.cc/en/Main/arduinoBoardMega&prev=search>
- Arduino Nano*. (23 de Agosto de 2017). Obtenido de <https://store.arduino.cc/usa/arduino-nano>
- Condori, O. C. (2014). *Proyecto de Grado: Sistema de Control Domotico Via GSM para el Instituto de Rehabilitacion Infantil*.
<http://es.slideshare.net>. (s.f.). Recuperado el 9 de Junio de 2016, de <http://es.slideshare.net>
- <http://es.wikipedia.org/wiki/Android>. (s.f.). Recuperado el 9 de Junio de 2016, de <http://es.wikipedia.org/wiki/Android>
- <http://es.wikipedia.org/wiki/Arduino>. (s.f.). Recuperado el 9 de Junio de 2016, de <http://es.wikipedia.org/wiki/Arduino>
- <http://es.wikipedia.org/wiki/Sensor>. (s.f.). Recuperado el 9 de Junio de 2016, de <http://es.wikipedia.org/wiki/Sensor>
- <http://isa.uniovi.es>. (s.f.). Recuperado el 9 de Junio de 2016, de <http://isa.uniovi.es>
- <http://metodologiadelainvestigaciinsiis.blogspot.com>. (s.f.). Recuperado el 9 de Junio de 2016, de <http://metodologiadelainvestigaciinsiis.blogspot.com>
- <http://www.academia.edu>. (s.f.). Recuperado el 10 de Junio de 2016, de <http://www.academia.edu>
- <http://www.monografias.com>. (s.f.). Recuperado el 9 de julio de 2016, de <http://www.monografias.com>
- <http://www.rincondelvago.com>. (s.f.). Recuperado el 9 de Junio de 2016, de <http://www.rincondelvago.com>
- Importadora Nova*. (24 de Agosto de 2017). Obtenido de <http://www.importadoranova.com/electronica/40-modulo-rfid-rc522-1356mhz-rc522-s50-mifare.html>
- Importadora Nova, DHT11*. (23 de Agosto de 2017). Obtenido de <http://www.importadoranova.com/electronica/38-sensor-de-temperatura-y-humedad-dht11.html>

Importadora Nova, Mega. (23 de Agosto de 2017). Obtenido de <http://www.importadoranova.com/home/23-arduino-mega-2560-r3.html>

Importadora Nova, Nano. (23 de Agosto de 2017). Obtenido de <http://www.importadoranova.com/electronica/24-arduino-nano.html>

Ingventor (Blogspot). (24 de Año de 2017). Obtenido de <http://ingventor.blogspot.com/2016/01/tutorial-arduino-conectar-lcd-162-por.html>

Internet de las cosas. (23 de Agosto de 2017). Obtenido de <http://www.internetdelascosas.cl/2013/06/02/conexion-ethernet-para-arduino-con-enc28j60/>

Internetlab. (24 de Agosto de 2017). Obtenido de <https://www.internetlab.es/post/628/como-funciona-exactamente-un-servidor-web/>

JavaScript. (26 de Agosto de 2017). Obtenido de <https://www.javascript.com/>

Masa adelante. (24 de agosto de 2017). Obtenido de <http://www.masadelante.com/faqs/servidor>

Monografias, Arduino. (2017). Obtenido de www.monografias.com/arduino

Mysql. (27 de Agosto de 2017). Obtenido de <https://dev.mysql.com/doc/>

Notepad. (25 de Agosto de 2017). Obtenido de <https://notepad-plus-plus.org/>

PHP NET. (25 de Agosto de 2017). Obtenido de <http://php.net/manual/es/language.constants.syntax.php>

Saber Satagoniatec. (23 de Agosto de 2017). Obtenido de <http://saber.patagoniatec.com/modulo-ethernet-enc28j60-arduino-argentina-ptec/>

SistemaOperativoAndroid. (2017). Obtenido de <http://www.monografias.com/trabajos101/sistema-operativo-android/sistema-operativo-android2.shtml>

Wikipedia, Appserv. (26 de Agosto de 2017). Obtenido de <https://es.wikipedia.org/wiki/Appserv>

Wikipedia, final de carrea. (s.f.). Obtenido de https://es.wikipedia.org/wiki/Sensor_final_de_carrera

Wikipedia, HTML. (25 de Agosto de 2017). Obtenido de <https://es.wikipedia.org/wiki/HTML>

Wikipedia, Navegador Web. (24 de Agosto de 2017). Obtenido de https://es.wikipedia.org/wiki/Navegador_web

Wikipedia, PHP. (25 de Agosto de 2017). Obtenido de <https://es.wikipedia.org/wiki/PHP>

Wikipedia, servidor Web. (24 de Agosto de 2017). Obtenido de https://es.wikipedia.org/wiki/Servidor_web

Wikipedia, Java. (26 de Agosto de 2017). Obtenido de <https://es.wikipedia.org/wiki/JavaScript>
www.monografias.com. (s.f.).

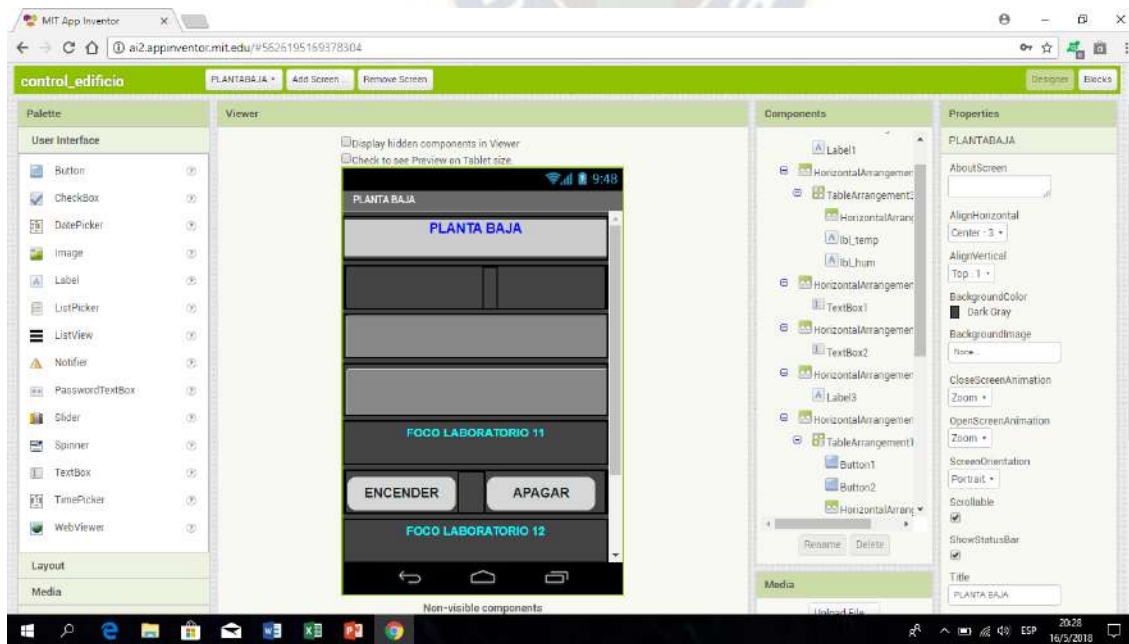
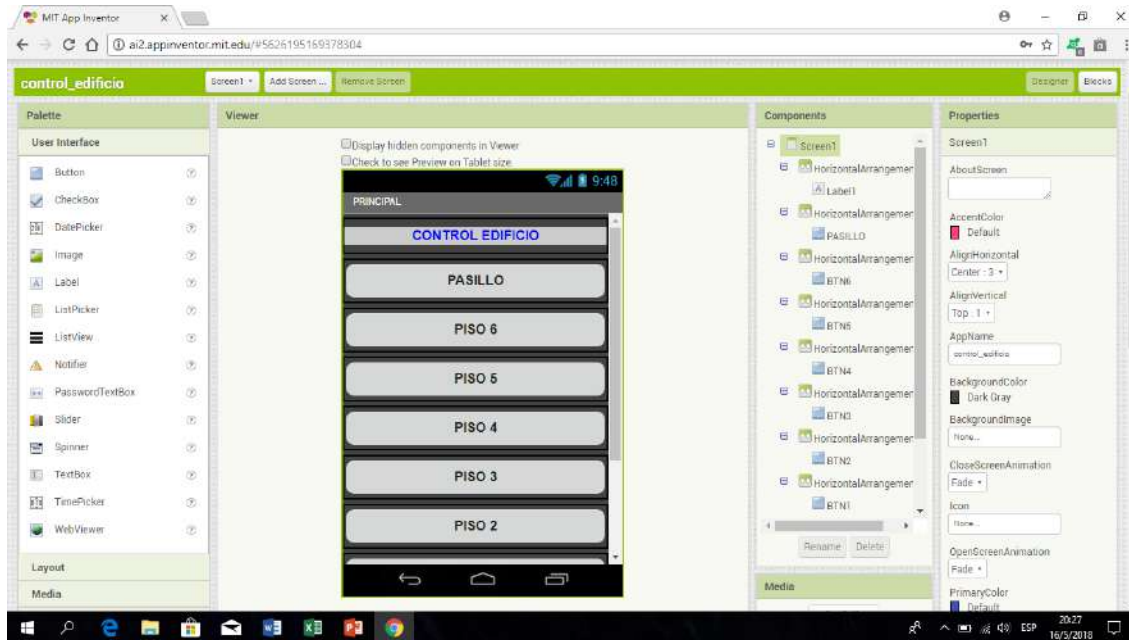


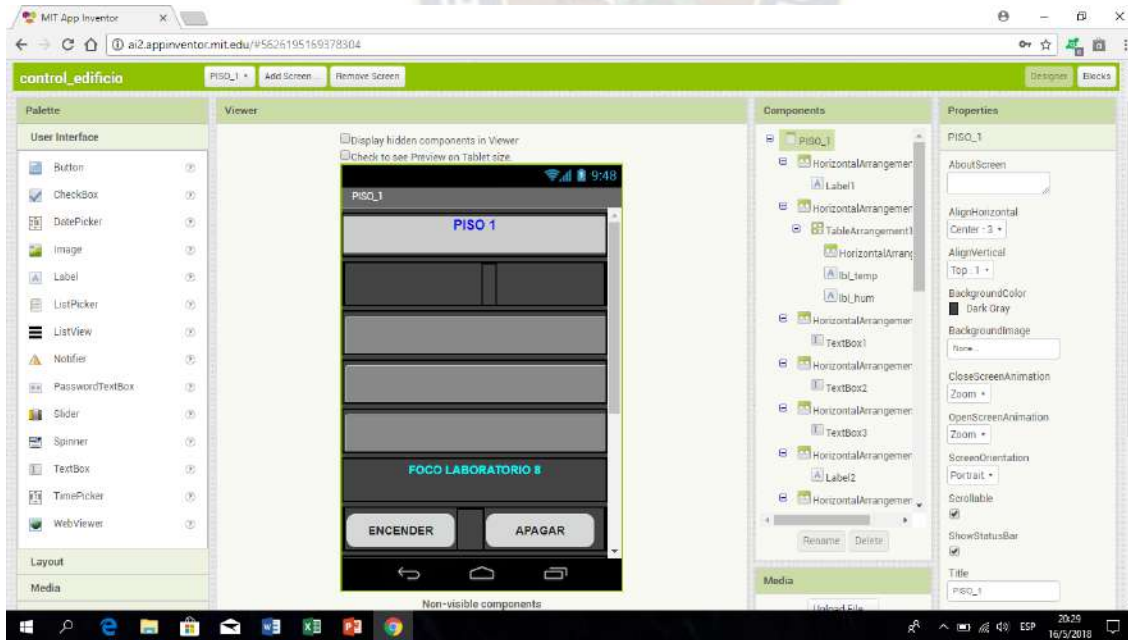
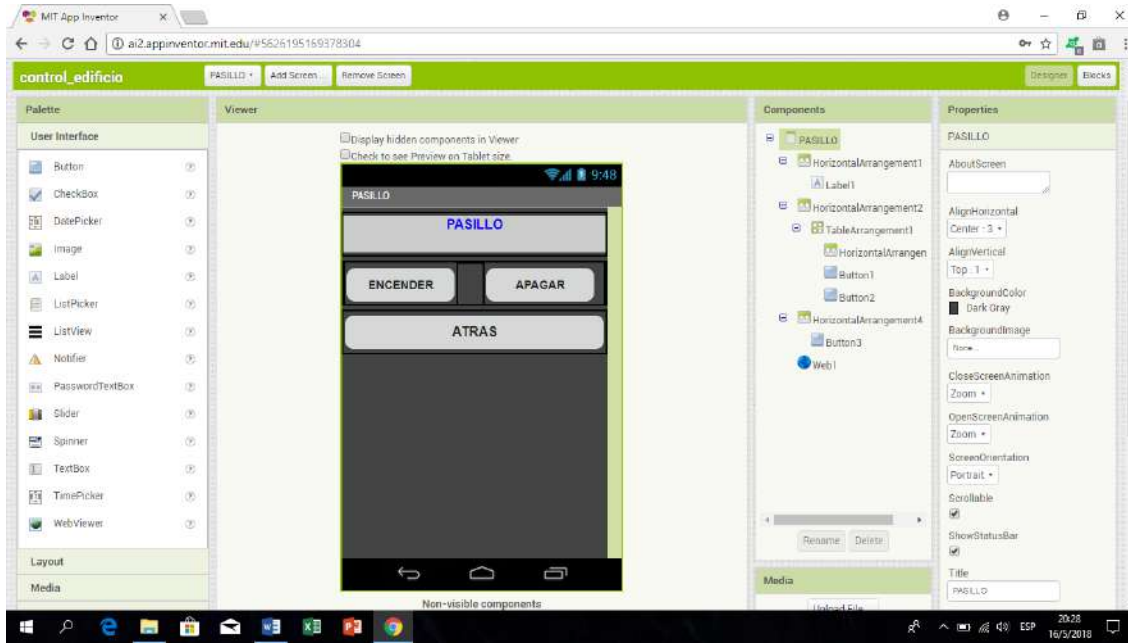
ANEXO 1

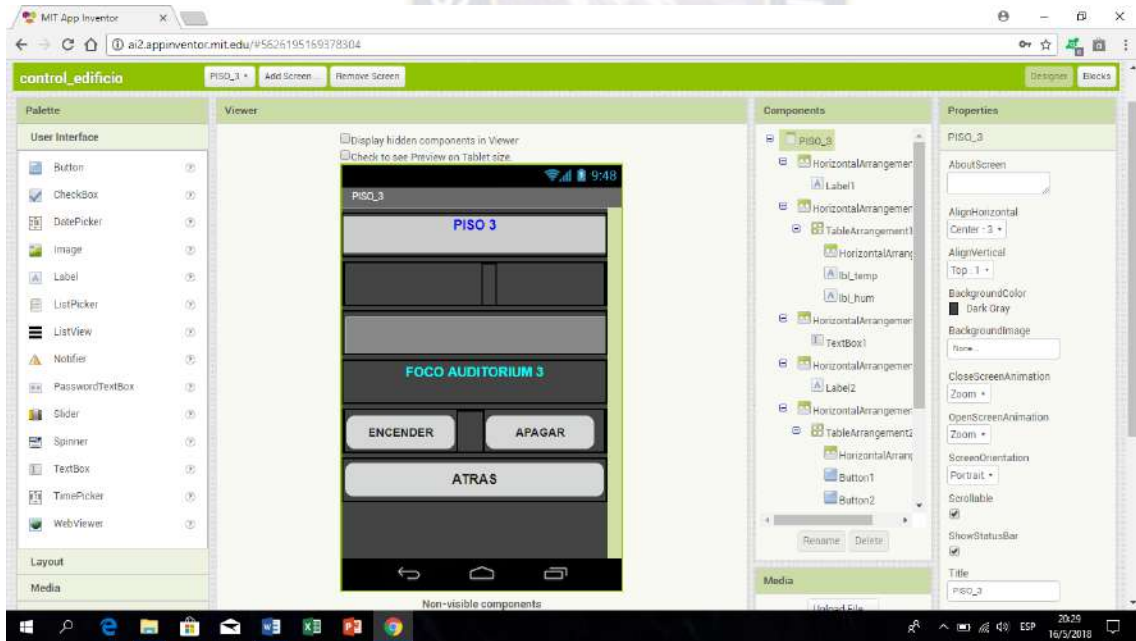
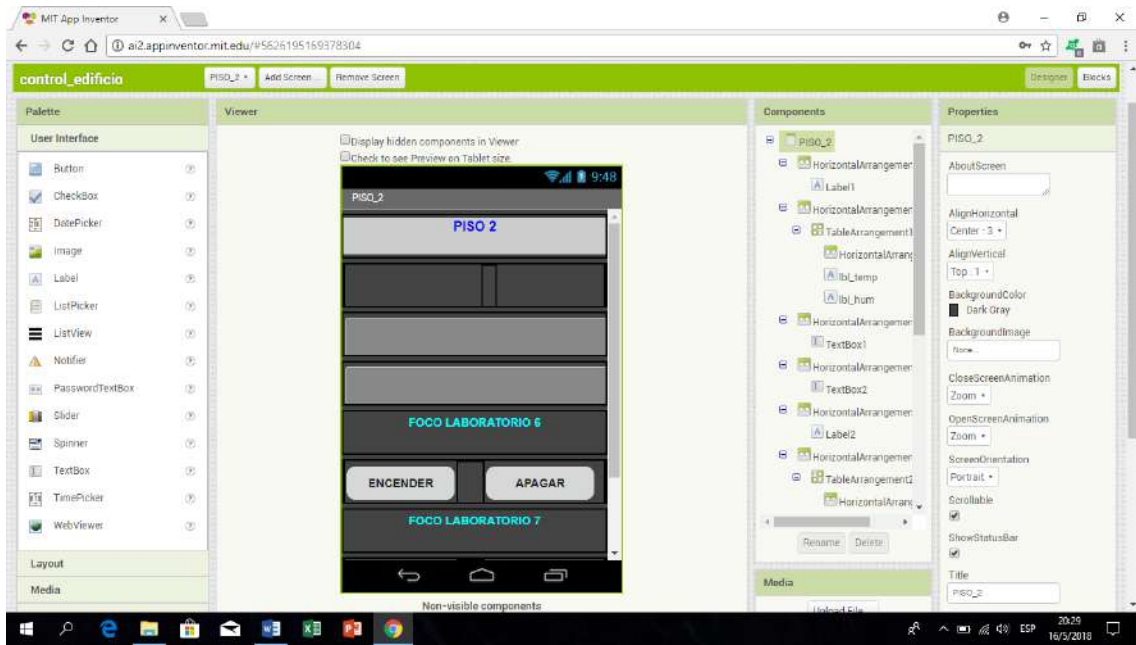


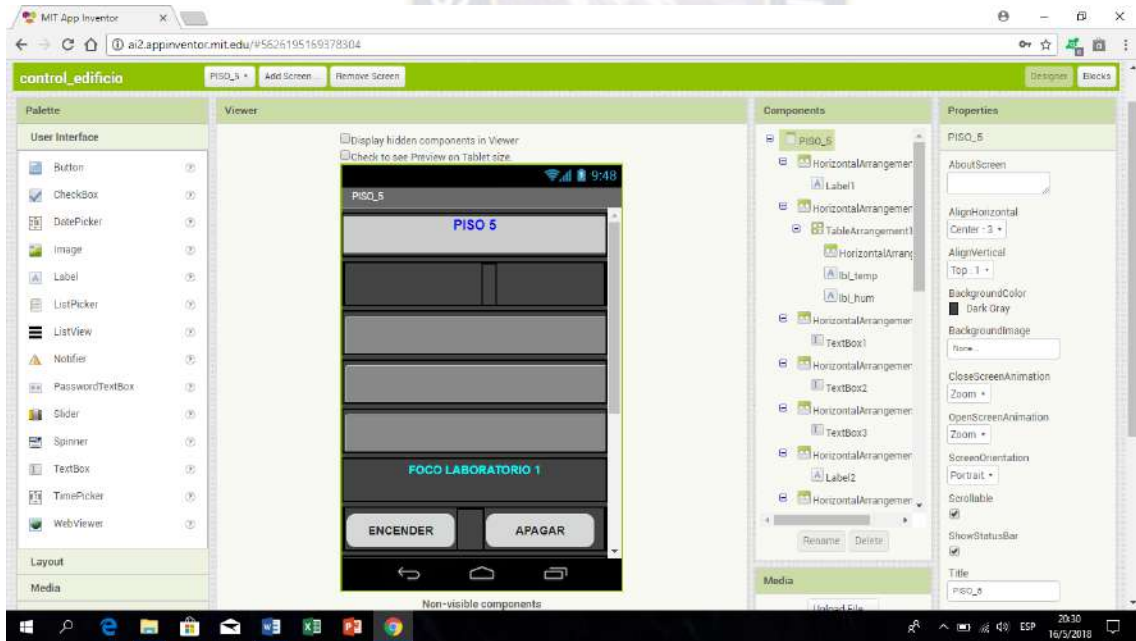
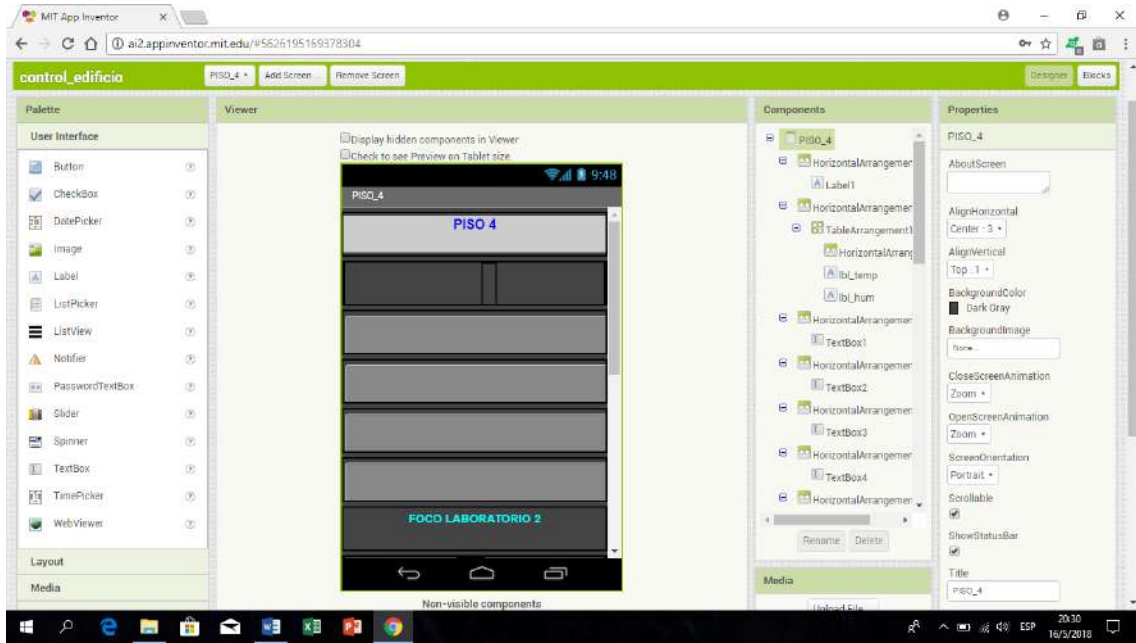
Programación Aplicación Android App Inventor

a. Entorno del diseño:

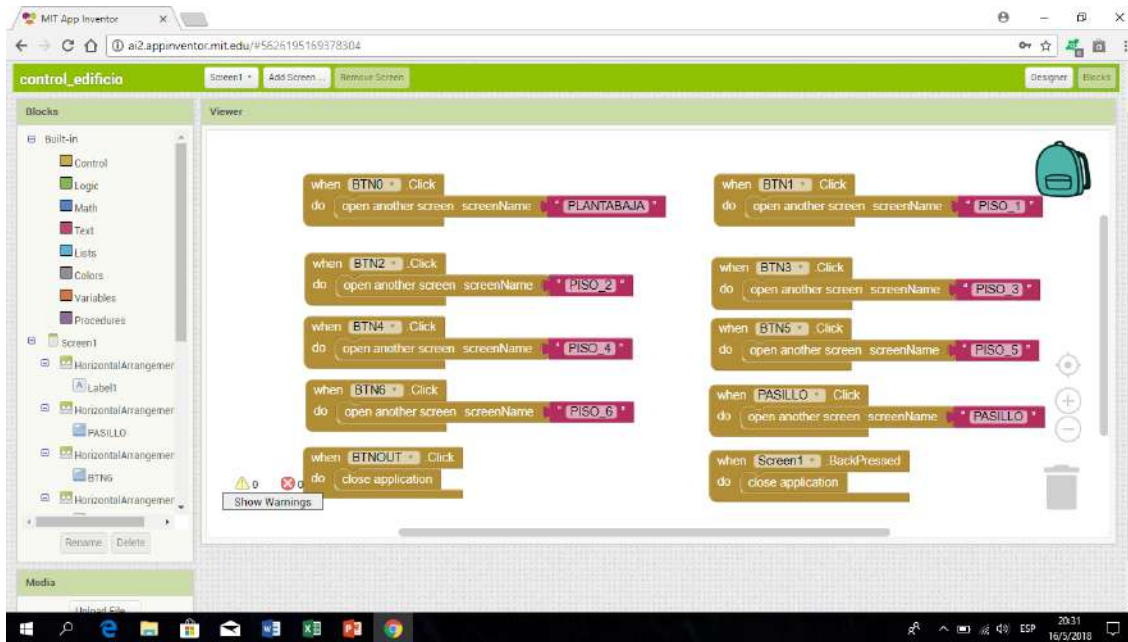






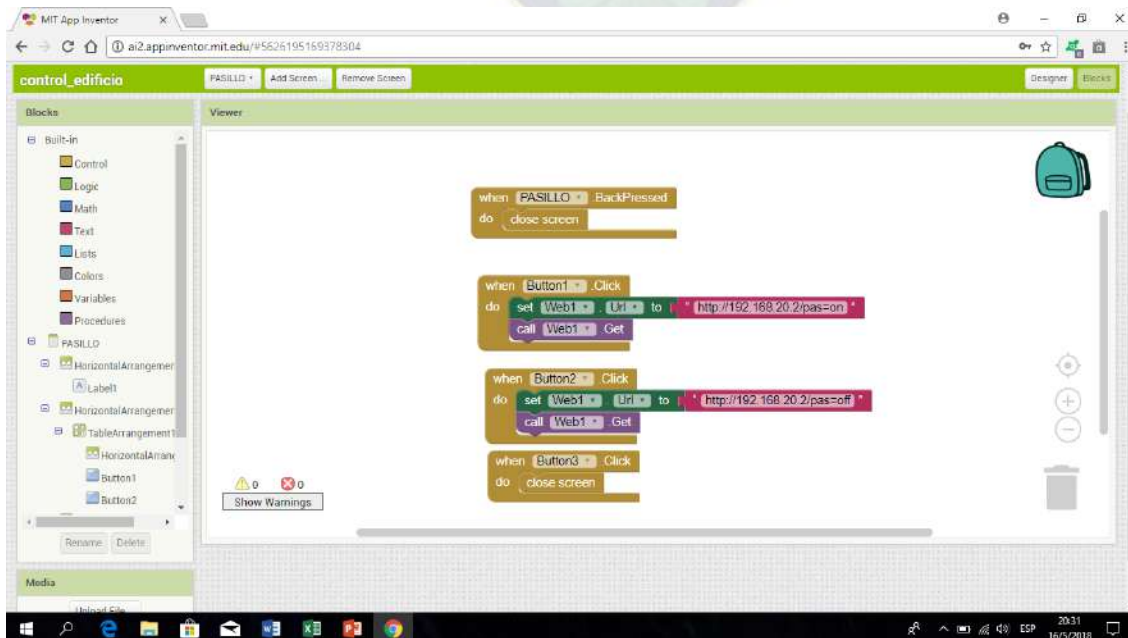


- b. Entorno de diseño de la aplicación:
I. Programa de llamado de ventanas:

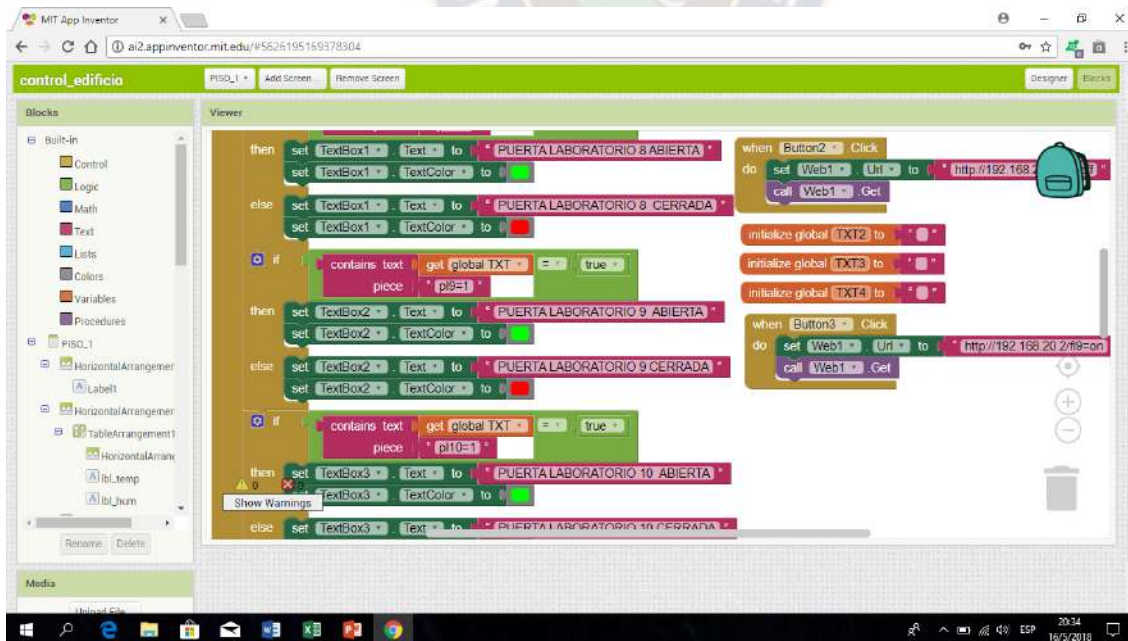
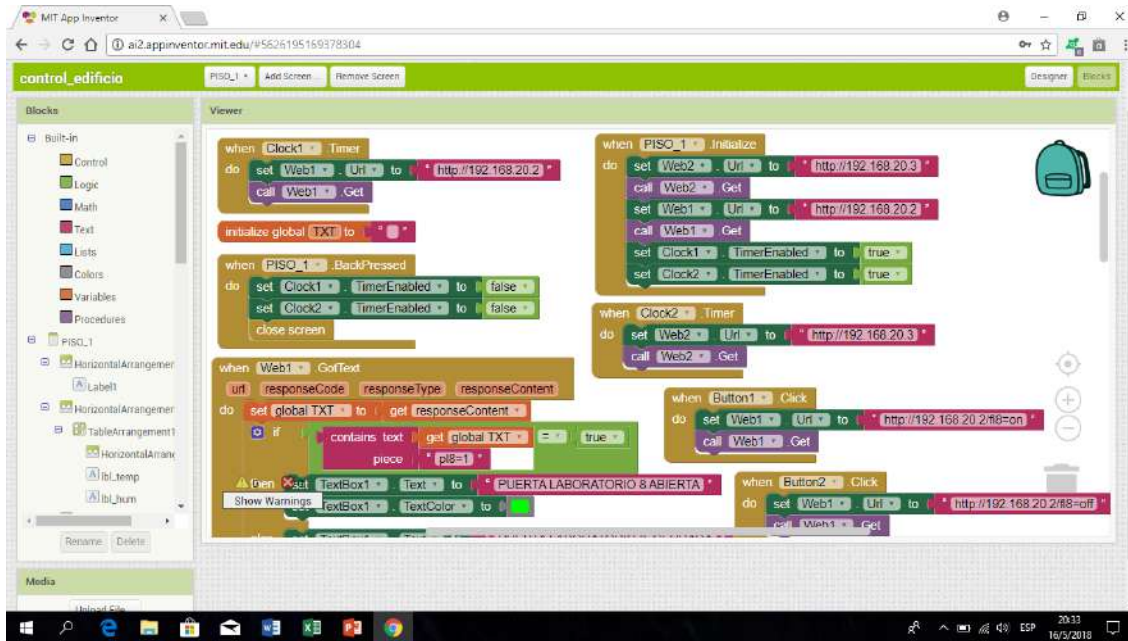


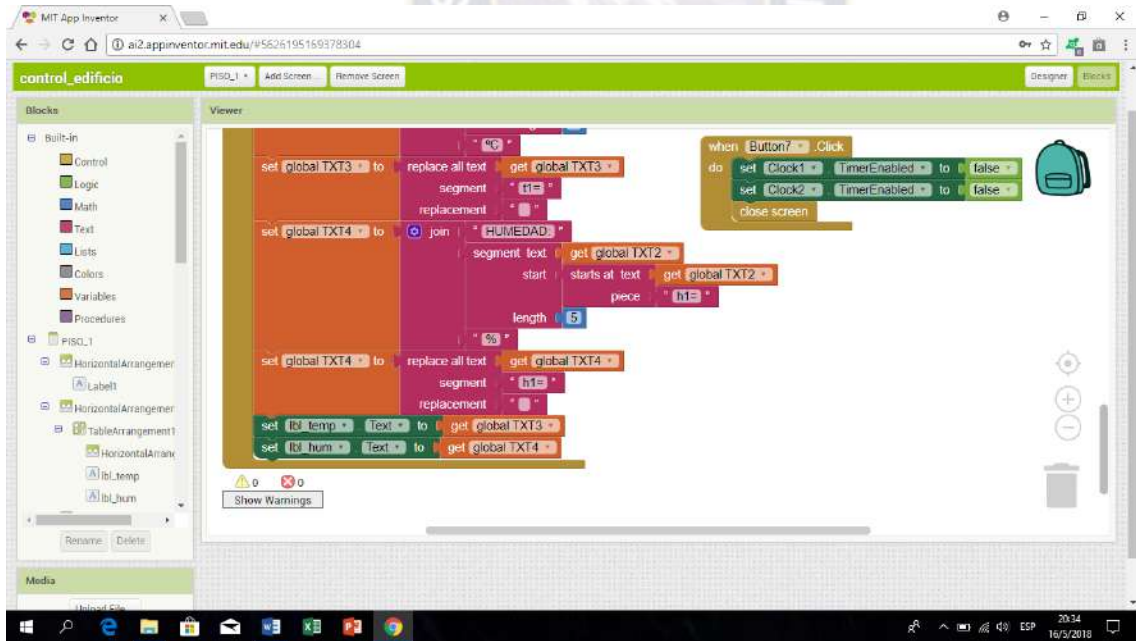
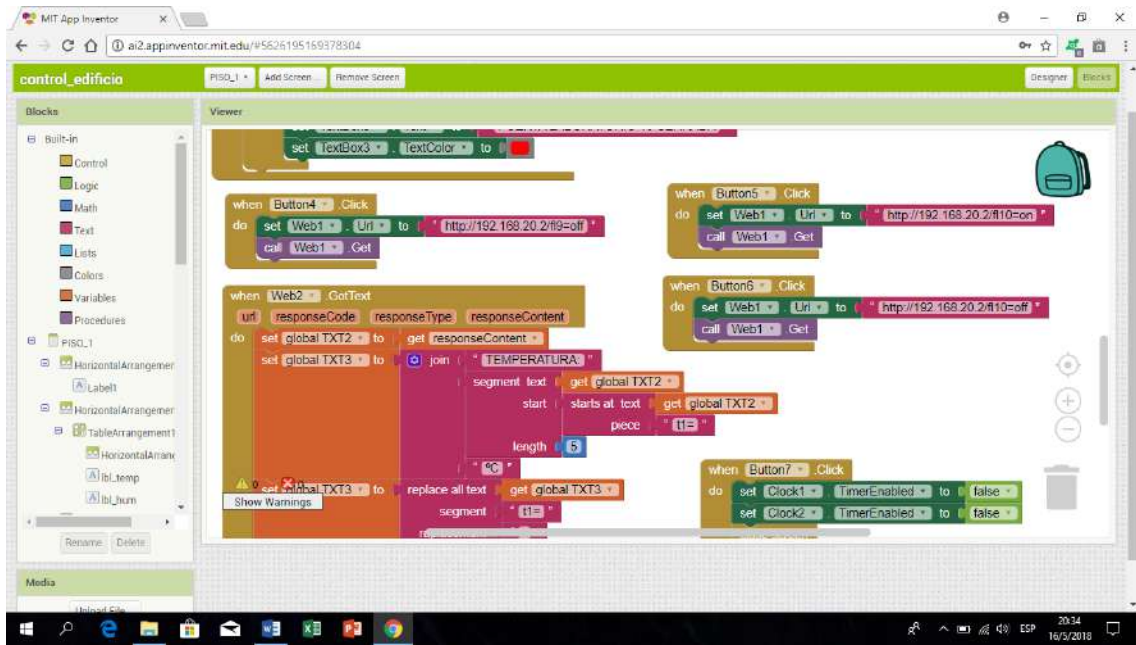
- II. Programa de control

Pasillo:

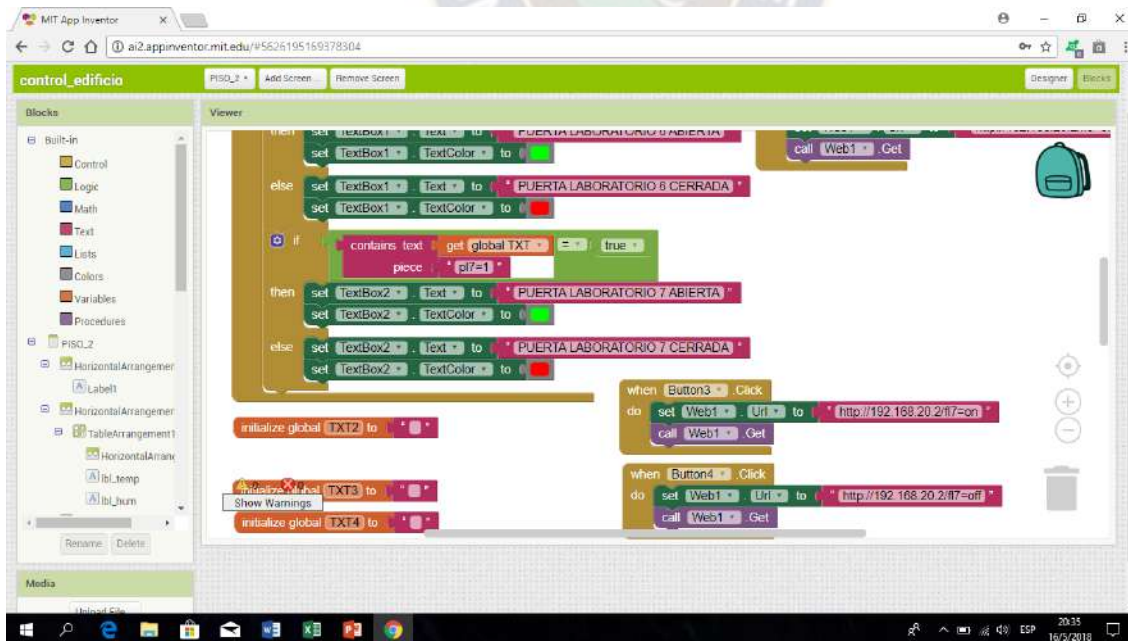
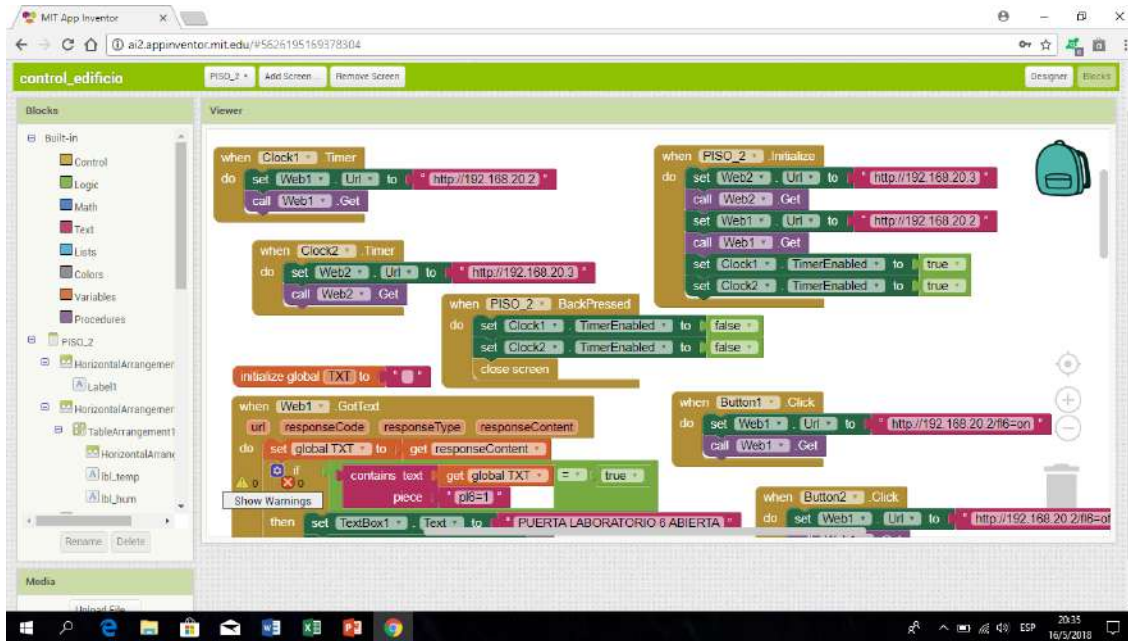


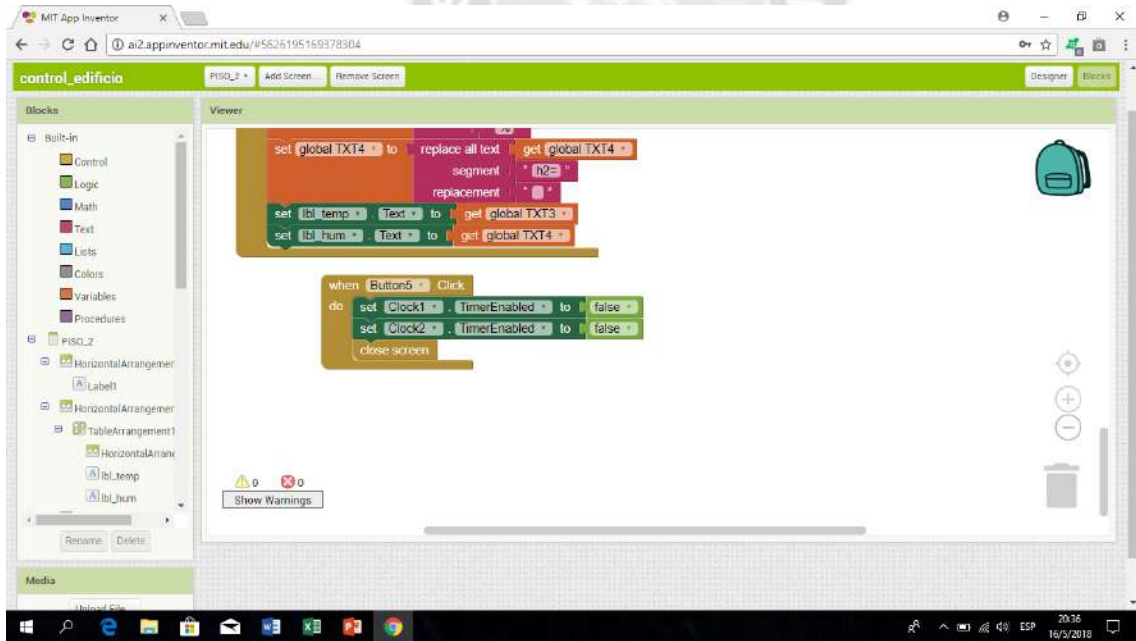
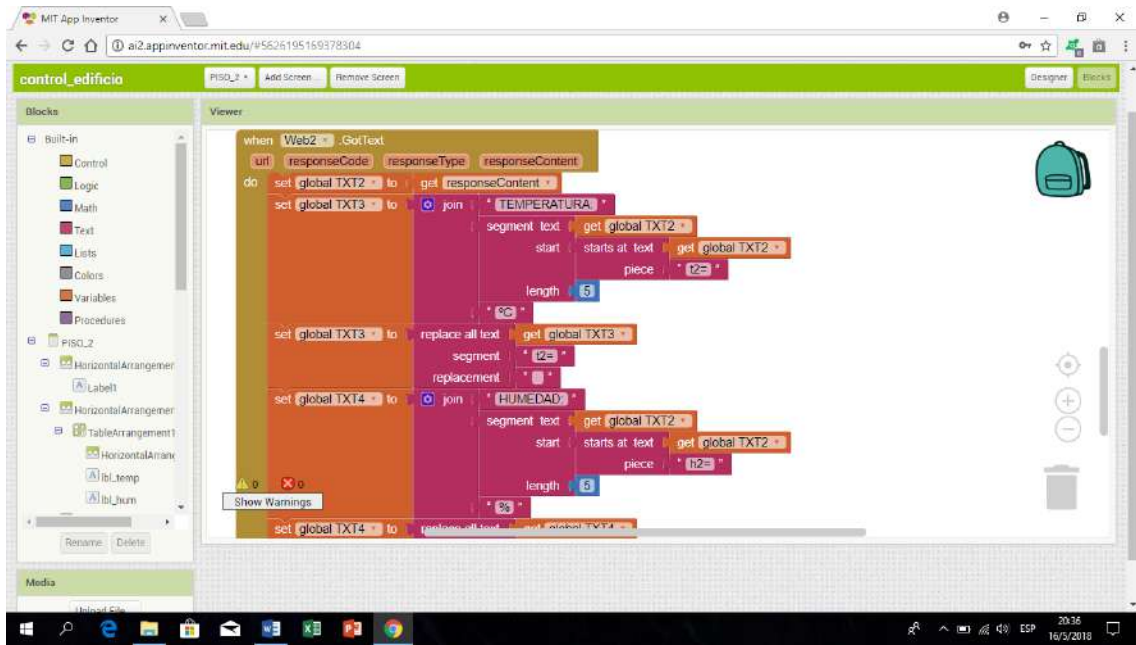
Piso 1



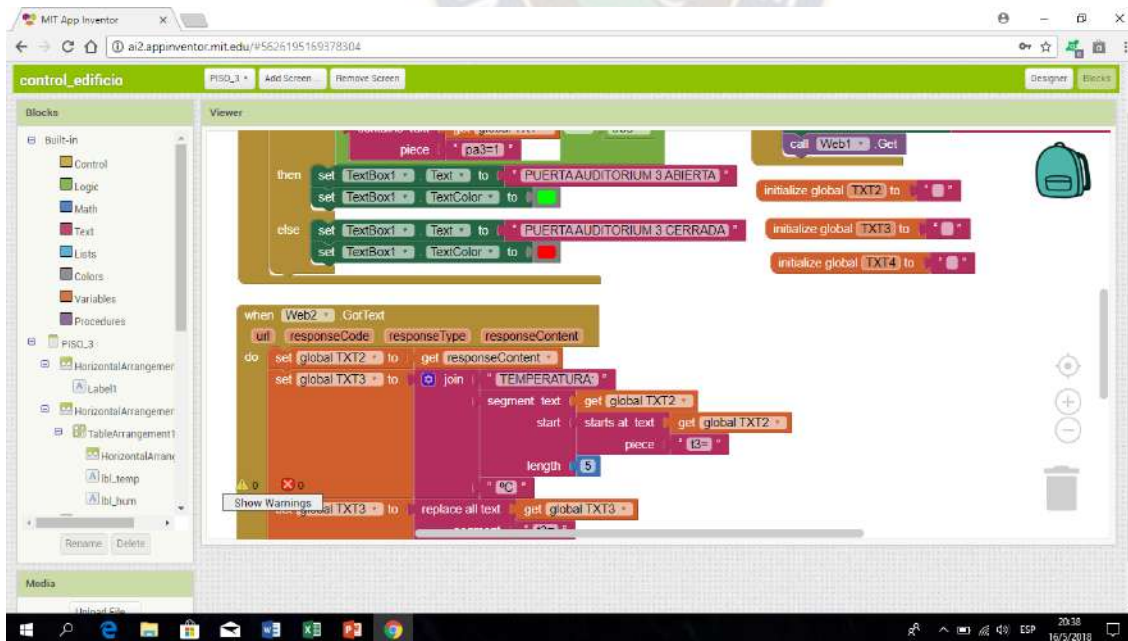
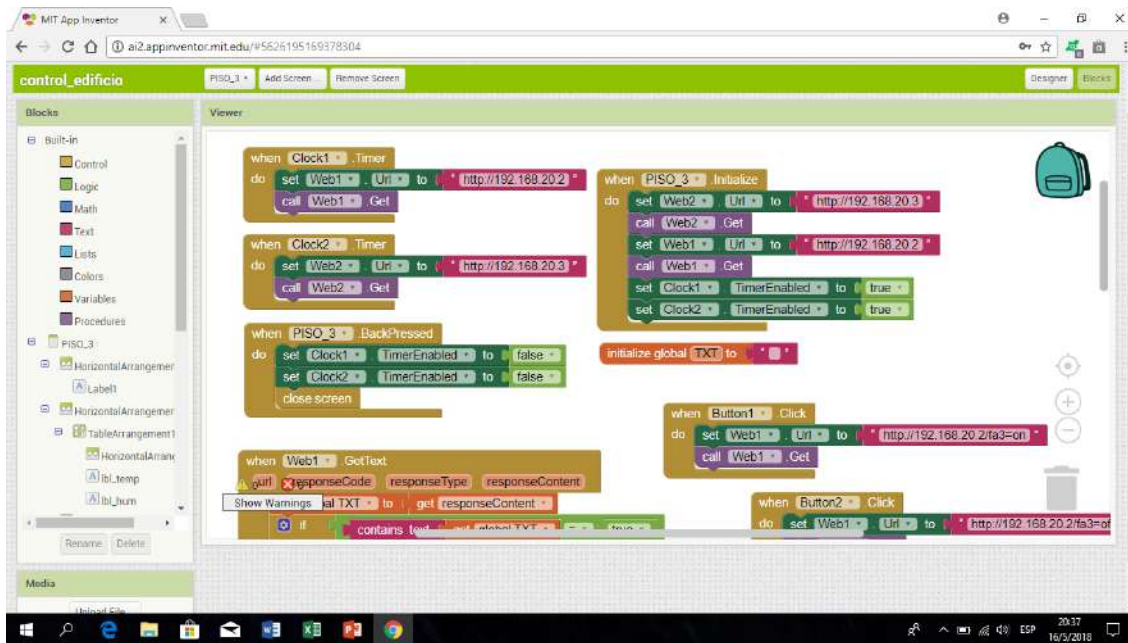


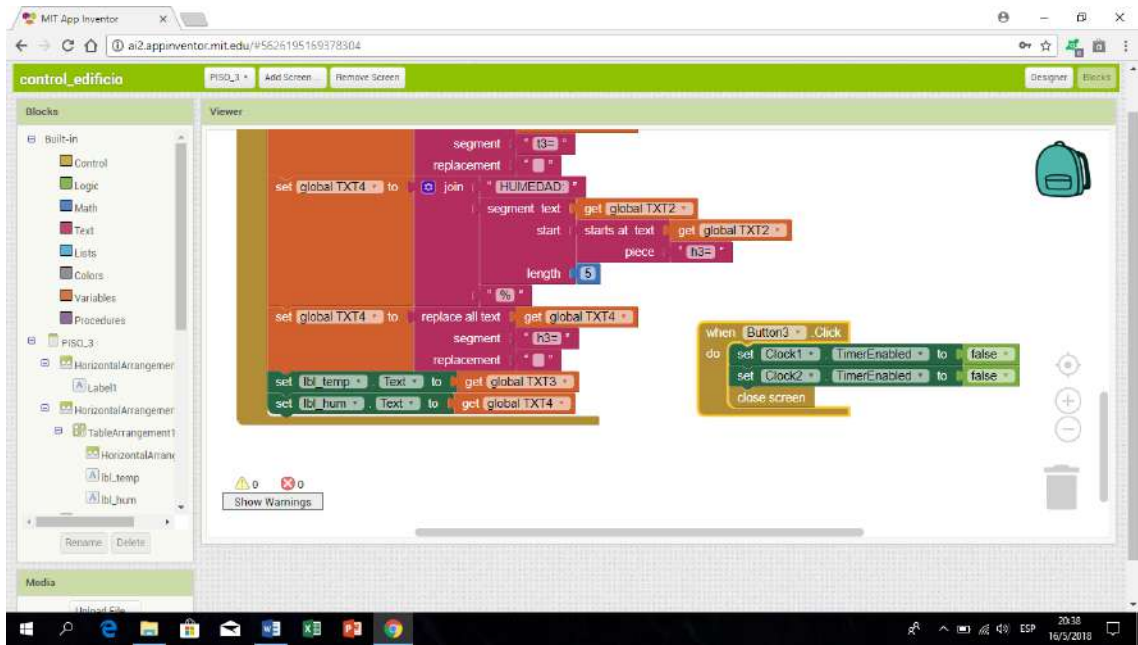
Piso 2:



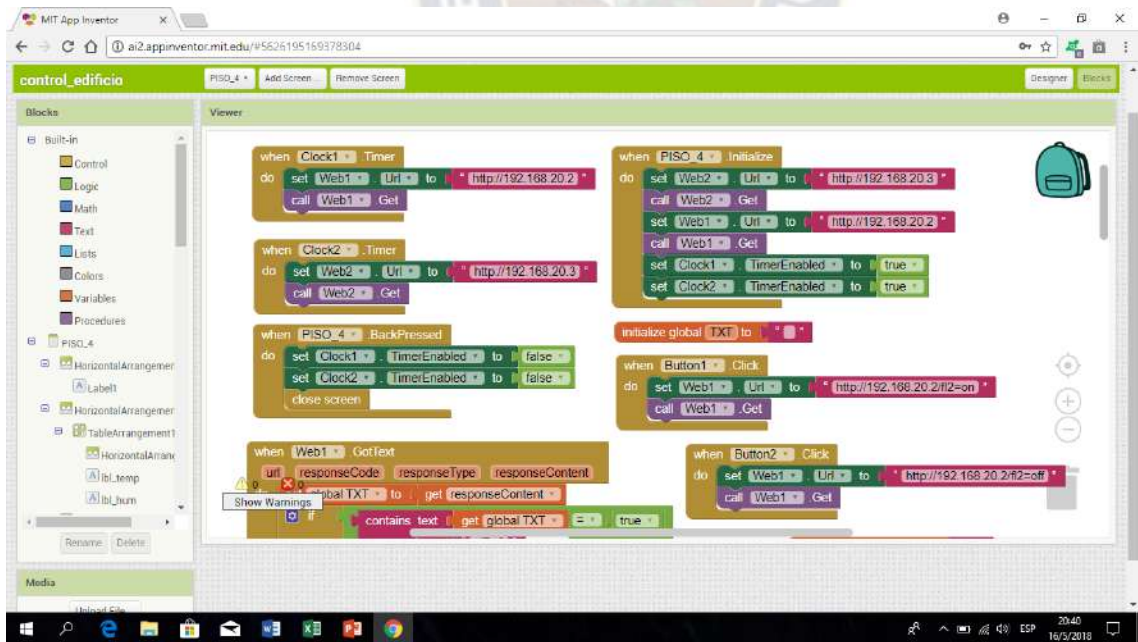


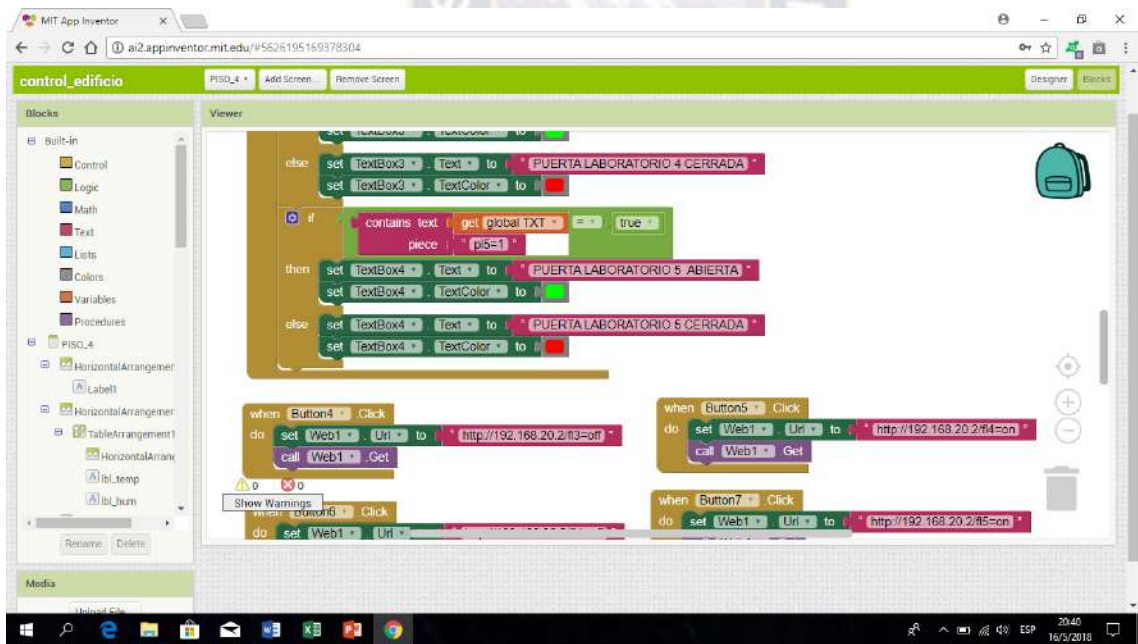
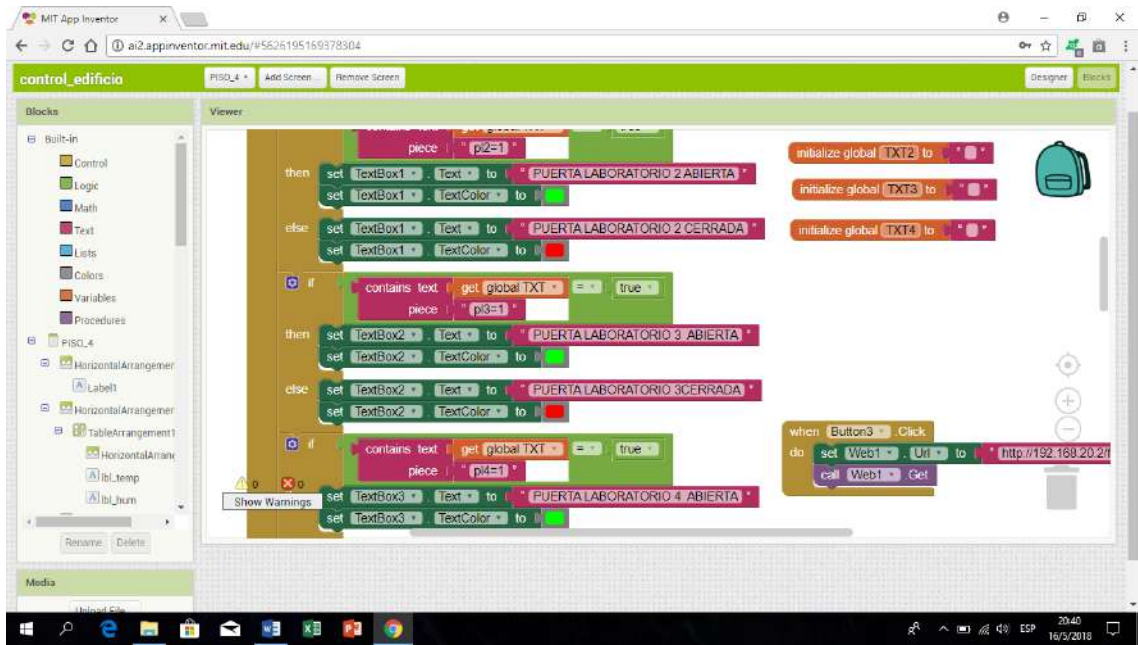
Piso 3:

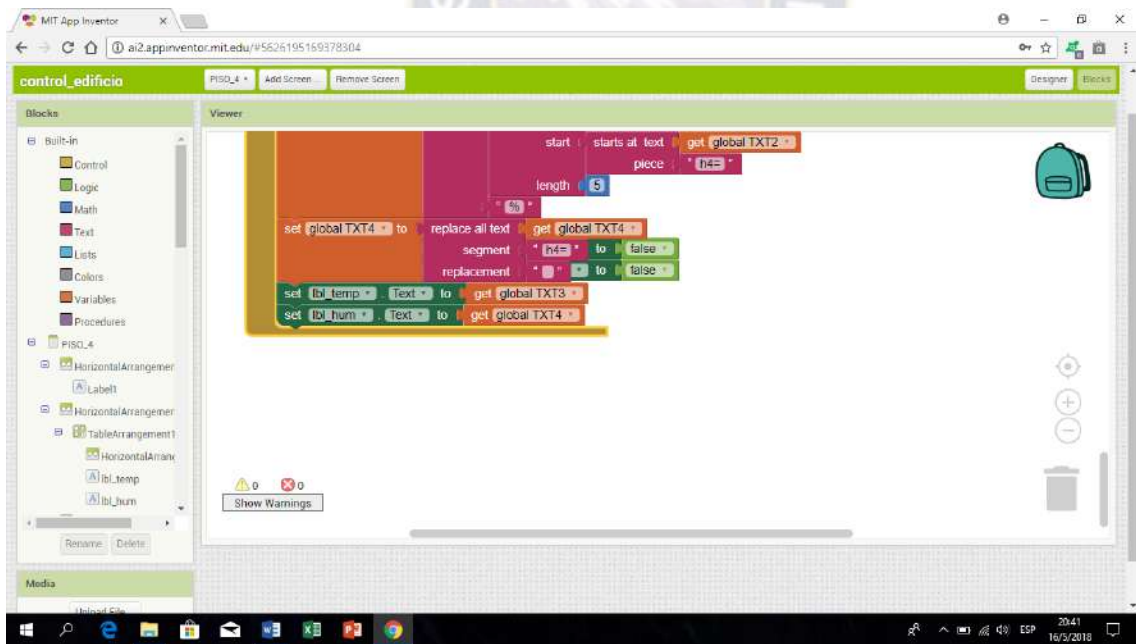
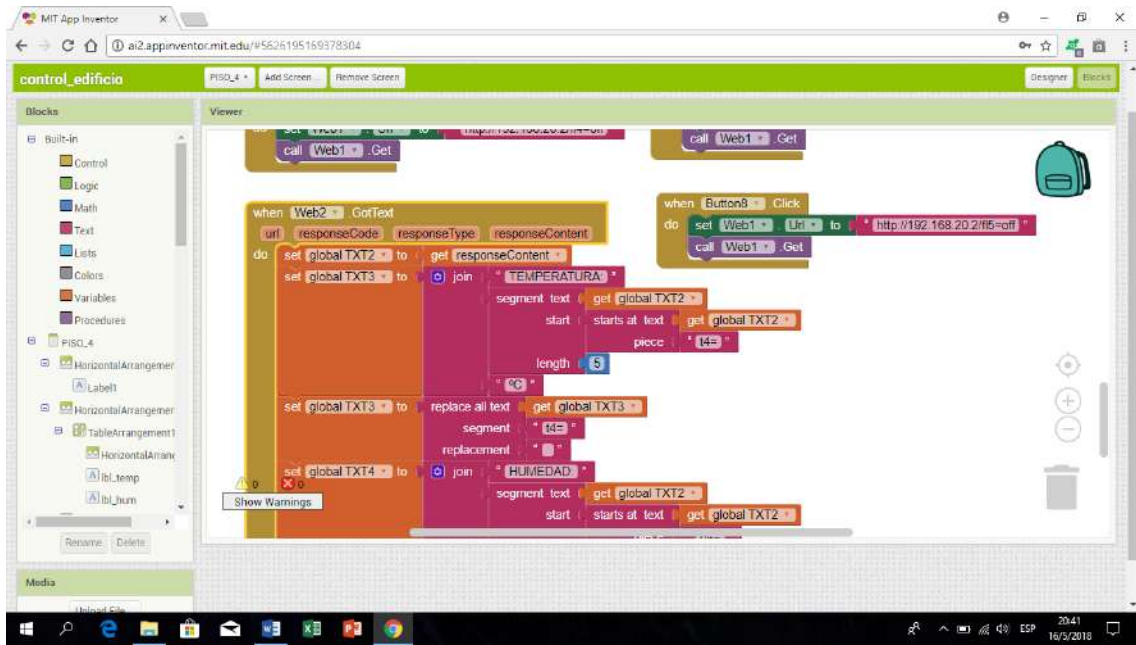




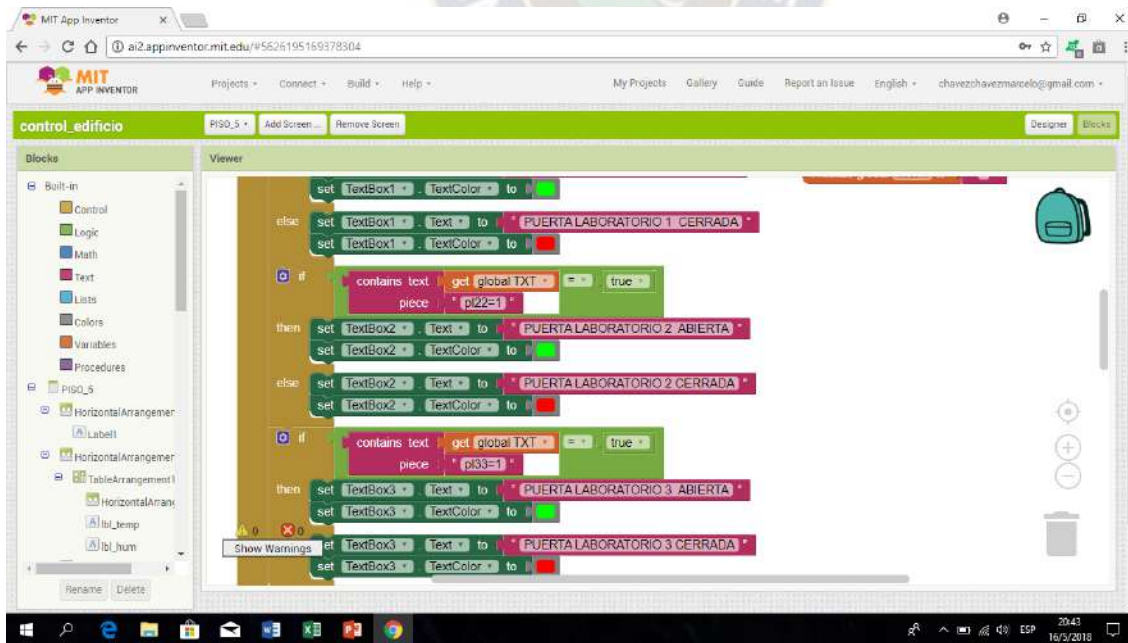
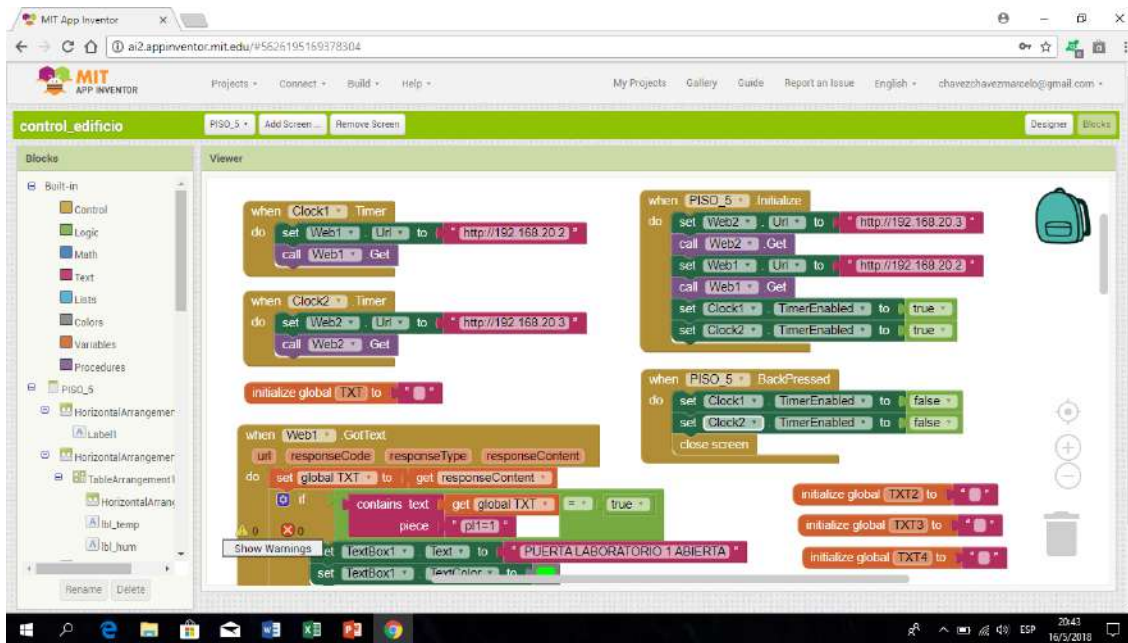
Piso 4:

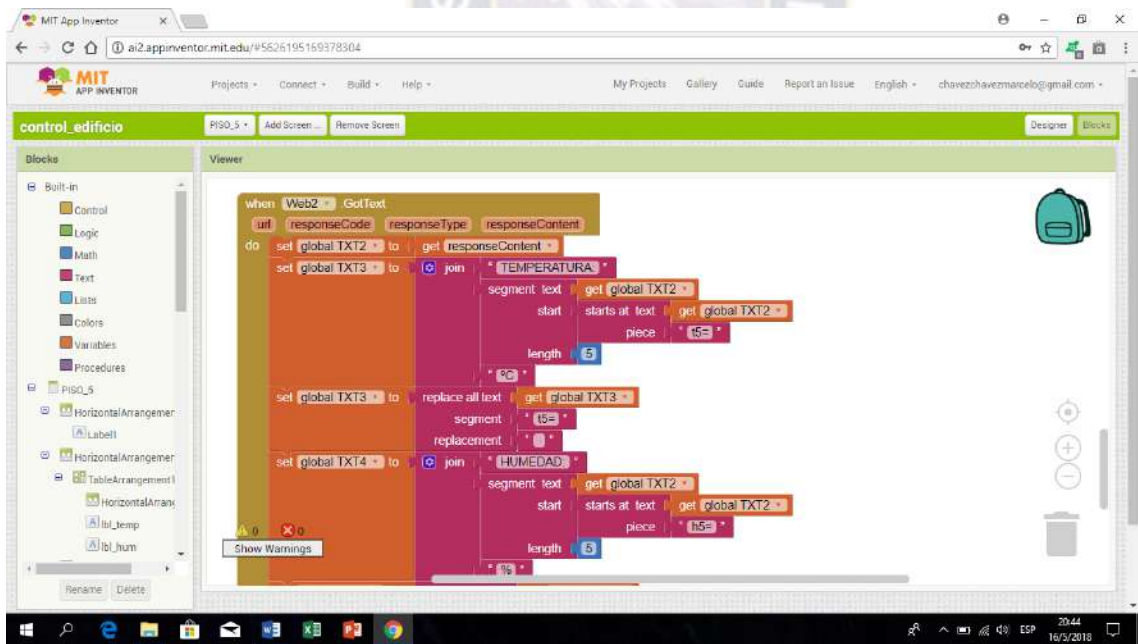
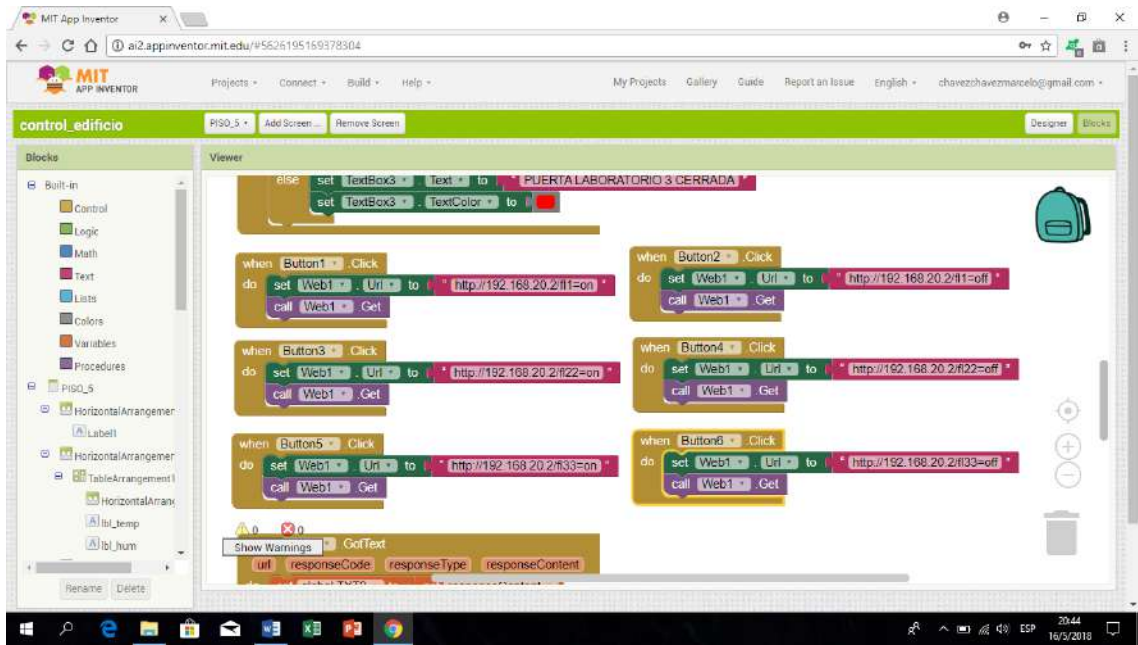


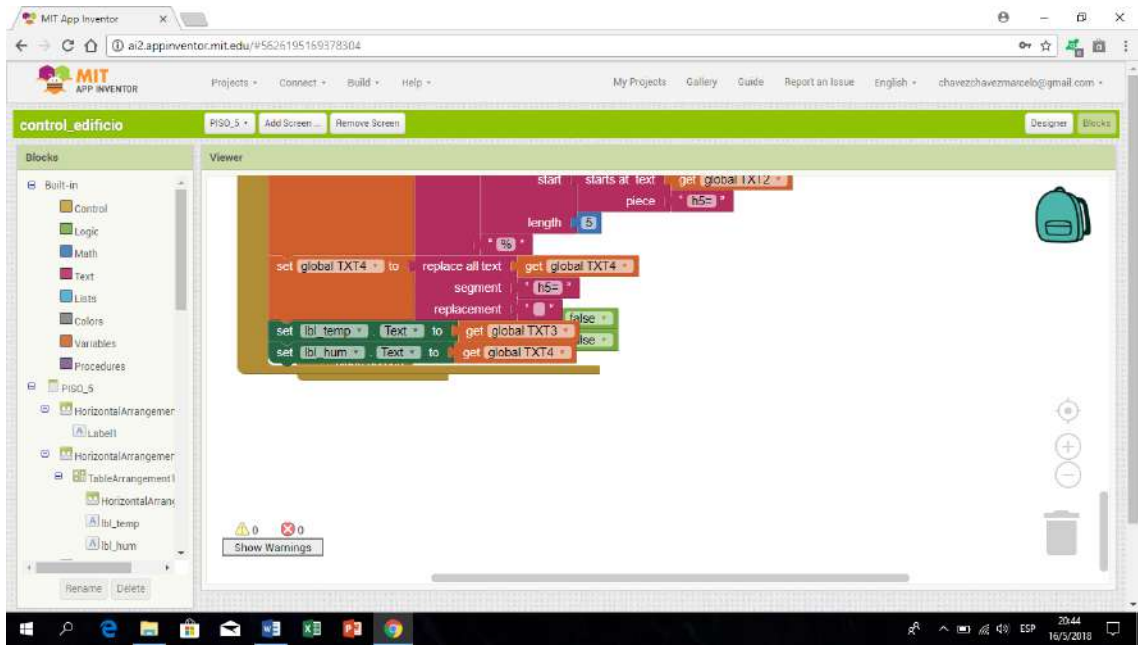




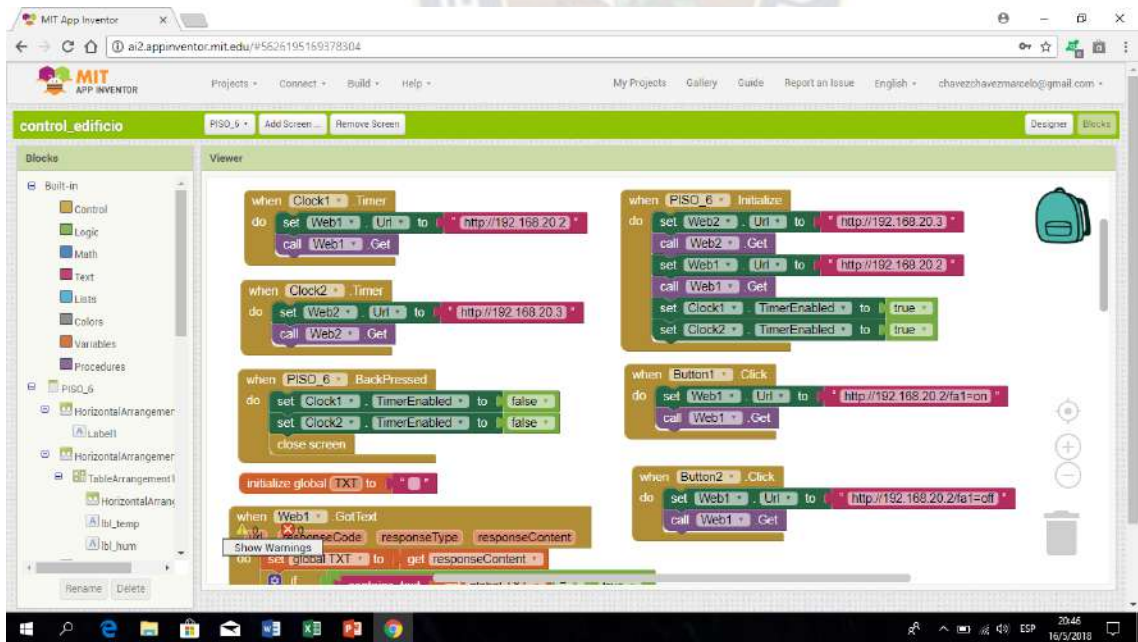
Piso 5:

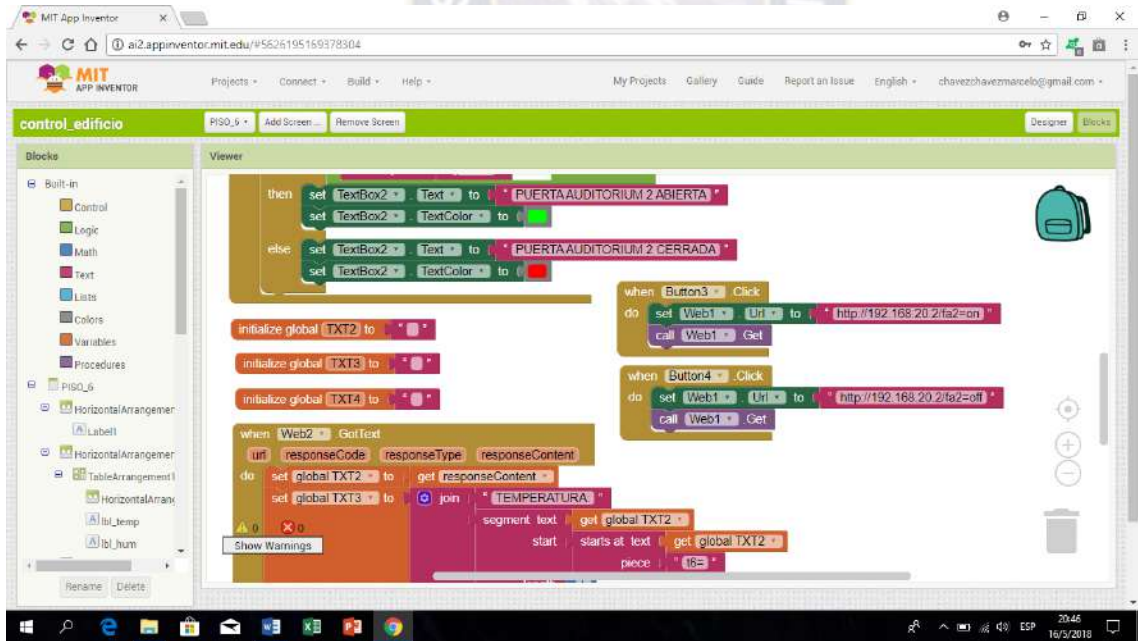
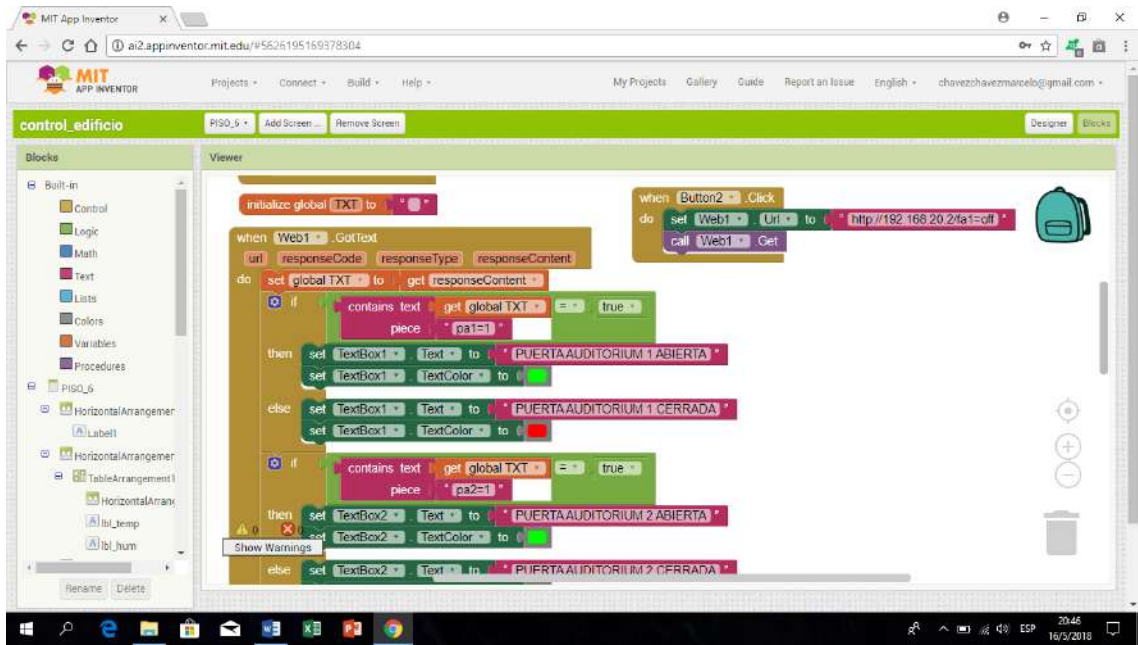


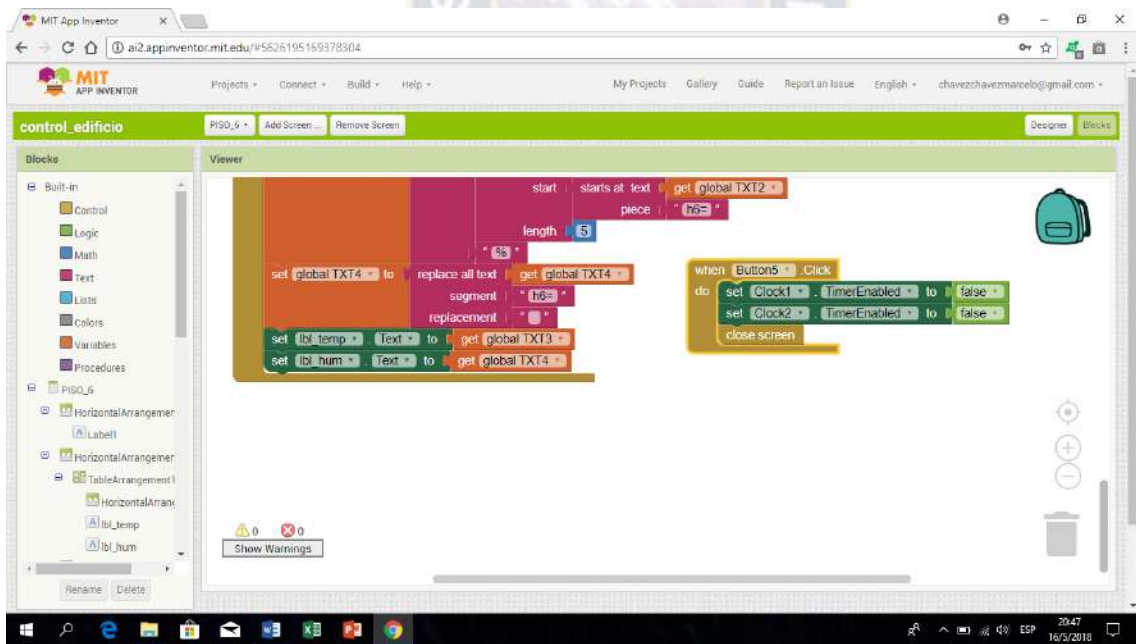
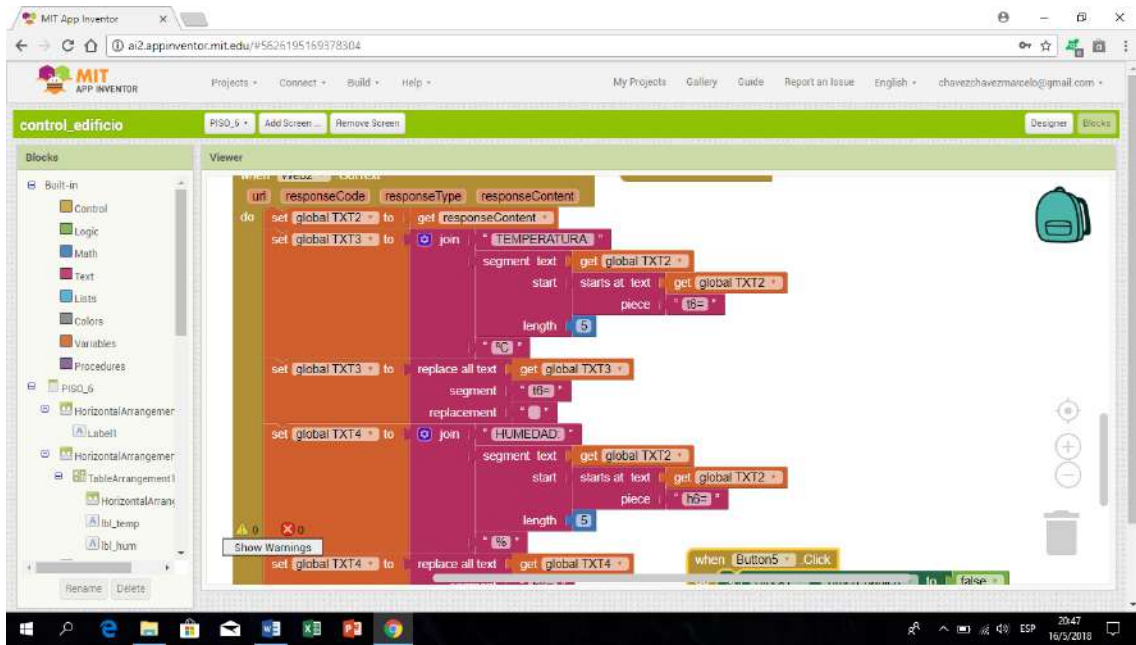




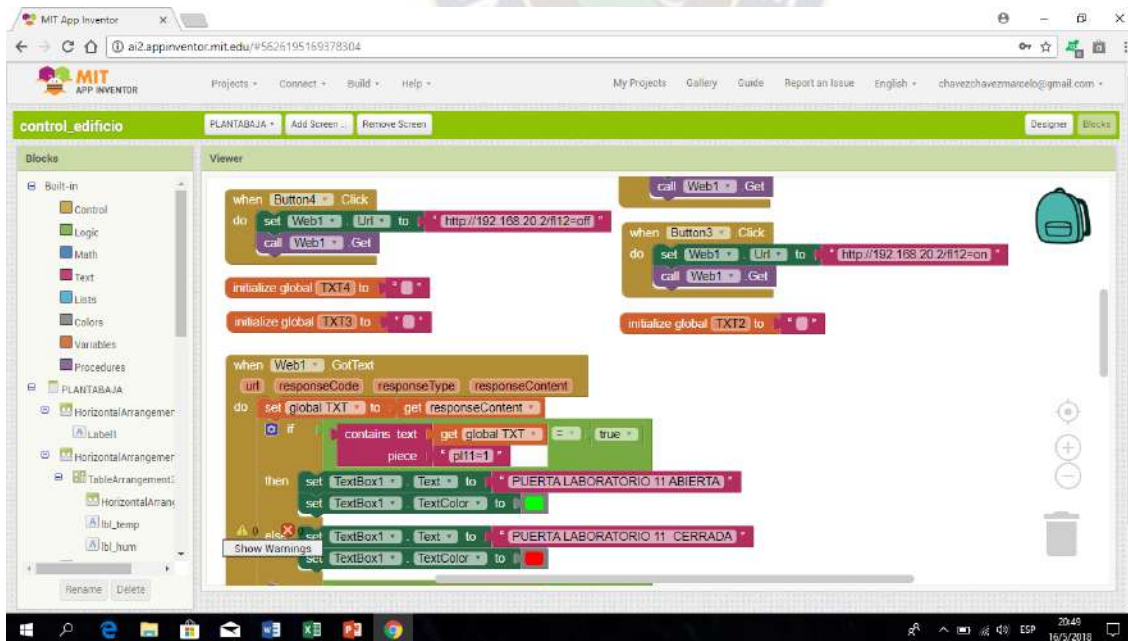
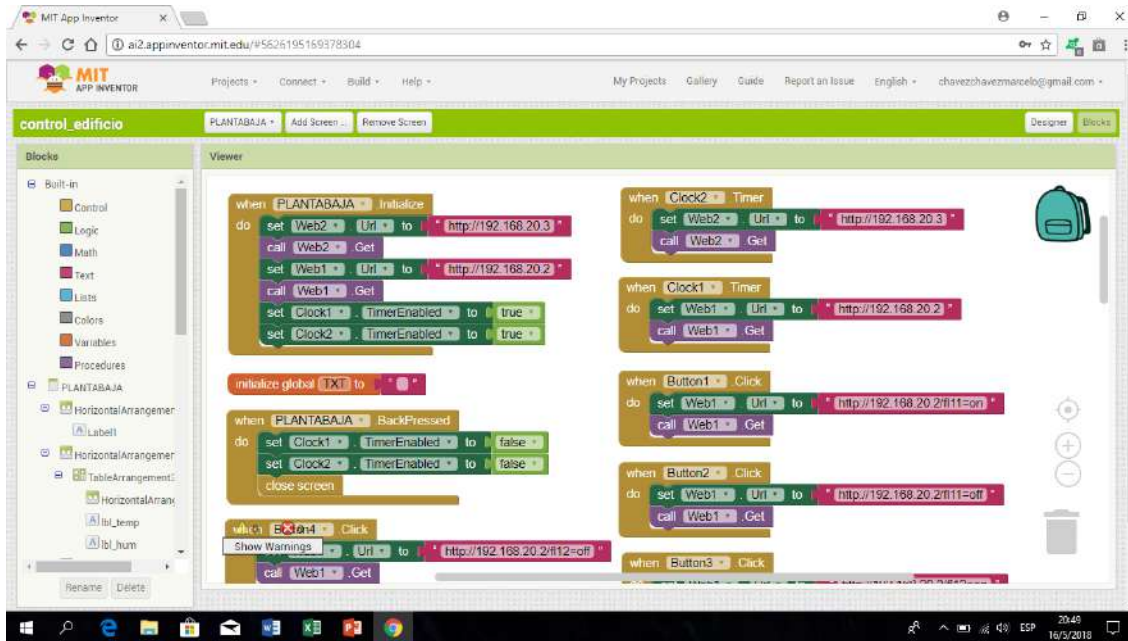
Piso 6:

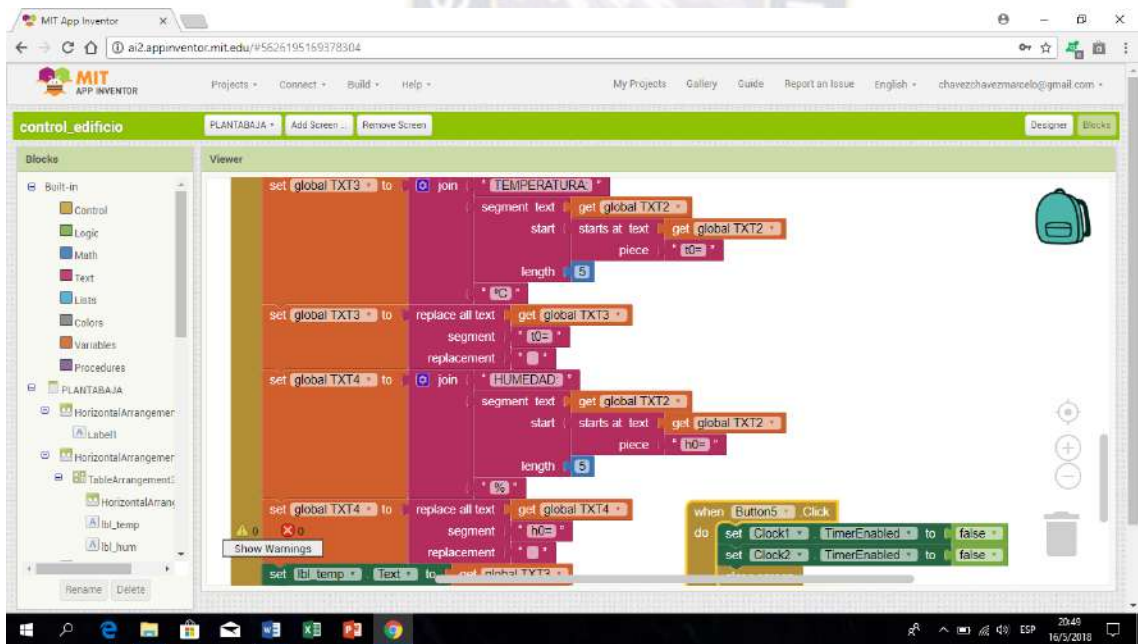
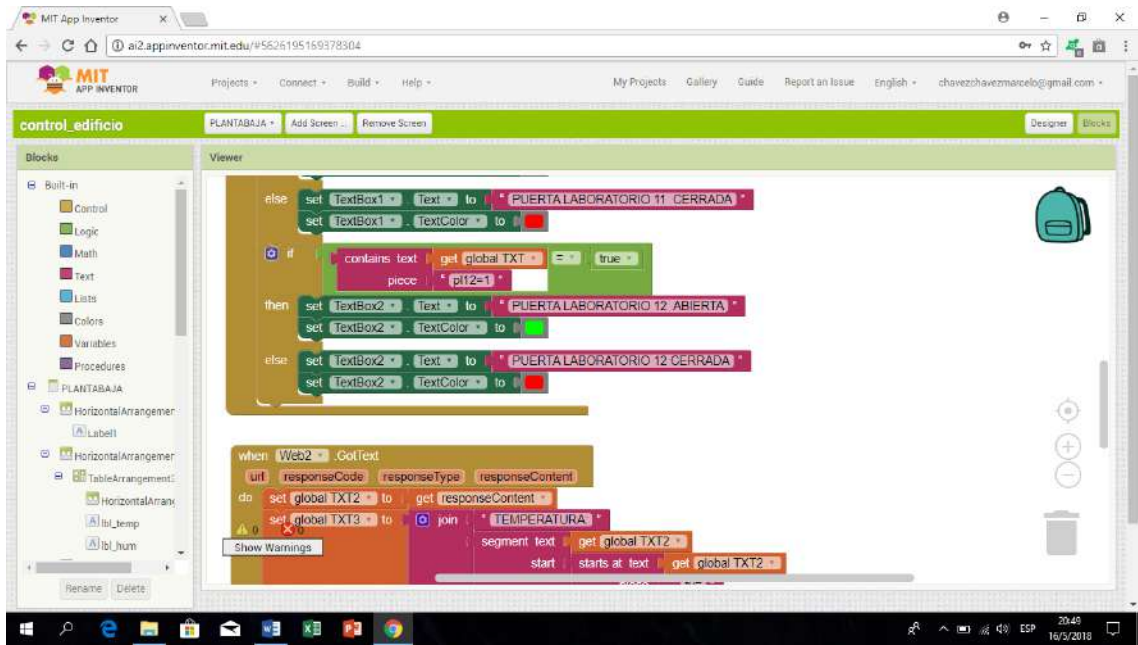


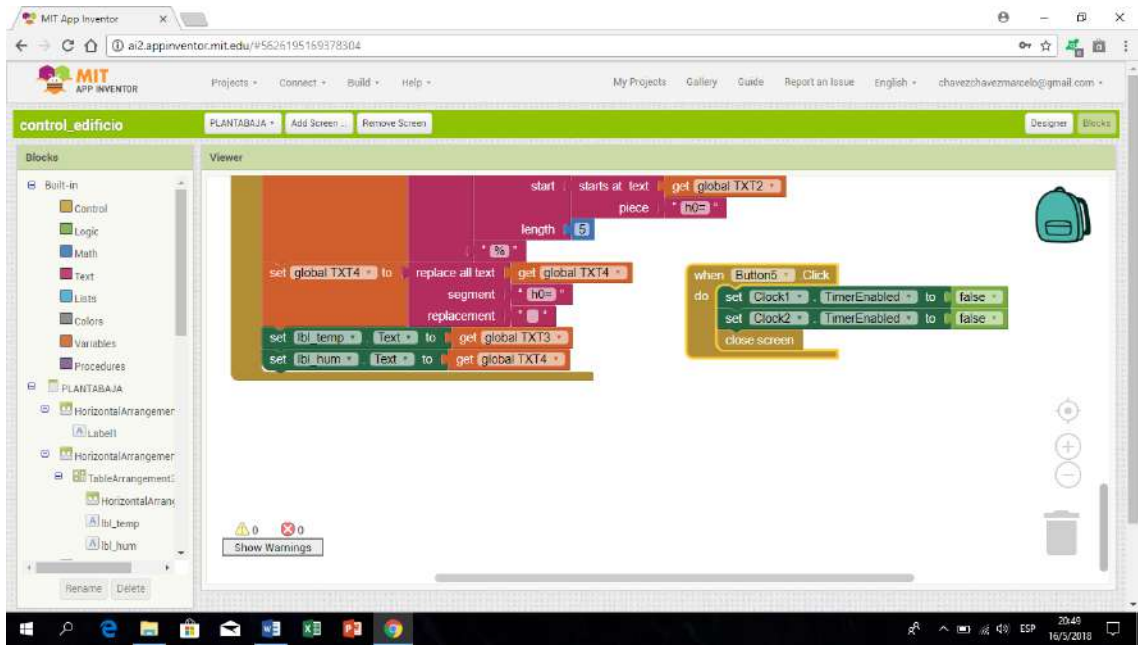




Planta Baja:







ANEXO 2



CÓDIGO CONTROL DE LUCES

```
#include "etherShield.h"

#include "ETHER_28J60.h"

int k;

int fl11=13,fl12=12,pl11=54,pl12=55;//planta baja

int fl8=11,fl9=10,fl10=9,pl8=56,pl9=57,pl10=58;//piso 1

int fl6=8,fl7=7,pl6=59,pl7=60;//piso 2

int fa3=6,pa3=61;//piso 3

int fl2=5,fl3=4,fl4=3,fl5=2,pl2=62,pl3=63,pl4=64,pl5=65;//piso 4

int fl1=14,fl22=15,fl33=16,pl1=66,pl22=67,pl33=68;//piso 5

int fa1=17,fa2=18,pa1=69,pa2=20;//piso 6

int pasillo=19;//pasillo

static uint8_t mac[6] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

static uint8_t ip[4] = {192,168,20,2};

static uint16_t port = 80;

ETHER_28J60 e;

void setup()

{

    delay(2000);

    Serial.begin(9600);

    for(k=2;k<=19;k++)

        pinMode(k,1);

    for(k=54;k<=69;k++)
```

```
        pinMode(k,0);
pinMode(20,0);
e.setup(mac, ip, port);
Serial.println("ETHERNET INICIADO");
}
void loop()
{
    delay(100);
    char* params;
    if (params = e.serviceRequest())
    {
        e.print("<h3>p11=");
        e.print(digitalRead(pl11));
        e.print("</h3><br /><h3>p12=");
        e.print(digitalRead(pl12));
        e.print("</h3><br /><h3>p18=");
        e.print(digitalRead(pl18));
        e.print("</h3><br /><h3>p19=");
        e.print(digitalRead(pl19));
        e.print("</h3><br /><h3>p110=");
        e.print(digitalRead(pl110));
        e.print("</h3><br /><h3>p16=");
        e.print(digitalRead(pl16));
        e.print("</h3><br /><h3>p17=");
        e.print(digitalRead(pl17));
    }
}
```

```
e.print("</h3><br /><h3>pa3=");
e.print(digitalRead(pa3));
e.print("</h3><br /><h3>pl2=");
e.print(digitalRead(pl2));
e.print("</h3><br /><h3>pl3=");
e.print(digitalRead(pl3));
e.print("</h3><br /><h3>pl4=");
e.print(digitalRead(pl4));
e.print("</h3><br /><h3>pl5=");
e.print(digitalRead(pl5));
e.print("</h3><br /><h3>pl1=");
e.print(digitalRead(pl1));
e.print("</h3><br /><h3>pl22=");
e.print(digitalRead(pl22));
e.print("</h3><br /><h3>pl33=");
e.print(digitalRead(pl33));
e.print("</h3><br /><h3>pa1=");
e.print(digitalRead(pa1));
e.print("</h3><br /><h3>pa2=");
e.print(digitalRead(pa2));
e.print("</h3><br />");
if (strcmp(params, "fl11=on") == 0)
{
    digitalWrite(fl11, 1);
    Serial.println("FOCO LAB 11 ON");
```

```
}
else if (strcmp(params, "fl11=off") == 0)
{
    digitalWrite(fl11, 0);
    Serial.println("FOCO LAB 11 OFF");
}
else if (strcmp(params, "fl12=on") == 0)
{
    digitalWrite(fl12, 1);
    Serial.println("FOCO LAB 12 ON");
}
else if (strcmp(params, "fl12=off") == 0)
{
    digitalWrite(fl12, 0);
    Serial.println("FOCO LAB 12 OFF");
}
else if (strcmp(params, "fl8=on") == 0)
{
    digitalWrite(fl8, 1);
    Serial.println("FOCO LAB 8 ON");
}
else if (strcmp(params, "fl8=off") == 0)
{
    digitalWrite(fl8, 0);
    Serial.println("FOCO LAB 8 OFF");
}
```

```
}  
else if (strcmp(params, "fl9=on") == 0)  
{  
    digitalWrite(fl9, 1);  
    Serial.println("FOCO LAB 9 ON");  
}  
else if (strcmp(params, "fl9=off") == 0)  
{  
    digitalWrite(fl9, 0);  
    Serial.println("FOCO LAB 9 OFF");  
}  
else if (strcmp(params, "fl10=on") == 0)  
{  
    digitalWrite(fl10, 1);  
    Serial.println("FOCO LAB 10 ON");  
}  
else if (strcmp(params, "fl10=off") == 0)  
{  
    digitalWrite(fl10, 0);  
    Serial.println("FOCO LAB 10 OFF");  
}  
else if (strcmp(params, "fl6=on") == 0)  
{  
    digitalWrite(fl6, 1);  
    Serial.println("FOCO LAB 6 ON");  
}
```

```
}  
else if (strcmp(params, "fl6=off") == 0)  
{  
    digitalWrite(fl6, 0);  
    Serial.println("FOCO LAB 6 OFF");  
}  
else if (strcmp(params, "fl7=on") == 0)  
{  
    digitalWrite(fl7, 1);  
    Serial.println("FOCO LAB 7 ON");  
}  
else if (strcmp(params, "fl7=off") == 0)  
{  
    digitalWrite(fl7, 0);  
    Serial.println("FOCO LAB 7 OFF");  
}  
else if (strcmp(params, "fa3=on") == 0)  
{  
    digitalWrite(fa3, 1);  
    Serial.println("FOCO AUD 3 ON");  
}  
else if (strcmp(params, "fa3=off") == 0)  
{  
    digitalWrite(fa3, 0);  
    Serial.println("FOCO AUD 3 OFF");  
}
```

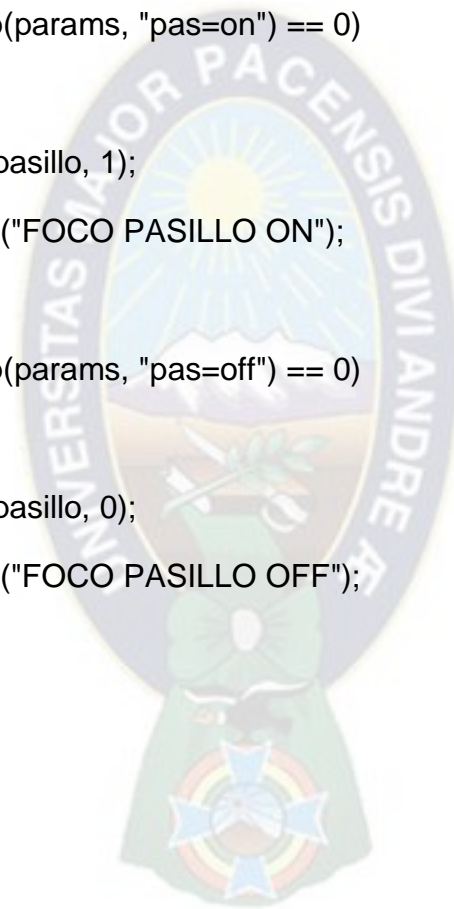


```
}  
else if (strcmp(params, "fl2=on") == 0)  
{  
    digitalWrite(fl2, 1);  
    Serial.println("FOCO LAB 2 ON");  
}  
else if (strcmp(params, "fl2=off") == 0)  
{  
    digitalWrite(fl2, 0);  
    Serial.println("FOCO LAB 2 OFF");  
}  
else if (strcmp(params, "fl3=on") == 0)  
{  
    digitalWrite(fl3, 1);  
    Serial.println("FOCO LAB 3 ON");  
}  
else if (strcmp(params, "fl3=off") == 0)  
{  
    digitalWrite(fl3, 0);  
    Serial.println("FOCO LAB 3 OFF");  
}  
else if (strcmp(params, "fl4=on") == 0)  
{  
    digitalWrite(fl4, 1);  
    Serial.println("FOCO LAB 4 ON");  
}
```

```
}  
else if (strcmp(params, "fl4=off") == 0)  
{  
    digitalWrite(fl4, 0);  
    Serial.println("FOCO LAB 4 OFF");  
}  
else if (strcmp(params, "fl5=on") == 0)  
{  
    digitalWrite(fl5, 1);  
    Serial.println("FOCO LAB 5 ON");  
}  
else if (strcmp(params, "fl5=off") == 0)  
{  
    digitalWrite(fl5, 0);  
    Serial.println("FOCO LAB 5 OFF");  
}  
else if (strcmp(params, "fl1=on") == 0)  
{  
    digitalWrite(fl1, 1);  
    Serial.println("FOCO LAB 1 ON");  
}  
else if (strcmp(params, "fl1=off") == 0)  
{  
    digitalWrite(fl1, 0);  
    Serial.println("FOCO LAB 1 OFF");  
}
```

```
}  
else if (strcmp(params, "fl22=on") == 0)  
{  
    digitalWrite(fl22, 1);  
    Serial.println("FOCO LAB 22 ON");  
}  
else if (strcmp(params, "fl33=off") == 0)  
{  
    digitalWrite(fl33, 0);  
    Serial.println("FOCO LAB 33 OFF");  
}  
else if (strcmp(params, "fa1=on") == 0)  
{  
    digitalWrite(fa1, 1);  
    Serial.println("FOCO AUD 1 ON");  
}  
else if (strcmp(params, "fa1=off") == 0)  
{  
    digitalWrite(fa1, 0);  
    Serial.println("FOCO AUD 1 OFF");  
}  
else if (strcmp(params, "fa2=on") == 0)  
{  
    digitalWrite(fa2, 1);  
    Serial.println("FOCO AUD 2 ON");  
}
```

```
}  
else if (strcmp(params, "fa2=off") == 0)  
{  
    digitalWrite(fa2, 0);  
    Serial.println("FOCO AUD 2 OFF");  
}  
else if (strcmp(params, "pas=on") == 0)  
{  
    digitalWrite(pasillo, 1);  
    Serial.println("FOCO PASILLO ON");  
}  
else if (strcmp(params, "pas=off") == 0)  
{  
    digitalWrite(pasillo, 0);  
    Serial.println("FOCO PASILLO OFF");  
}  
e.respond();  
}  
}
```



ANEXO 3



CÓDIGO CONTROL DE TEMPERATURA

```
#include <Wire.h>

#include "LCD.h"

#include "LiquidCrystal_I2C.h"

LiquidCrystal_I2C lcd0(0x3F,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd1(0x27,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd2(0x3D,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd3(0x3C,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd4(0x3B,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd5(0x3A,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd6(0x39,2,1,0,4,5,6,7,3,POSITIVE);

/*LiquidCrystal_I2C lcd0(0x27,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd1(0x26,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd2(0x25,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd3(0x24,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd4(0x23,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd5(0x22,2,1,0,4,5,6,7,3,POSITIVE);

LiquidCrystal_I2C lcd6(0x21,2,1,0,4,5,6,7,3,POSITIVE);*/

String txt;

char x,sw1;

void setup()

{

    lcd0.begin(16,2);

    lcd1.begin(16,2);
```



```

    lcd2.begin(16,2);
    lcd3.begin(16,2);
    lcd4.begin(16,2);
    lcd5.begin(16,2);
    lcd6.begin(16,2);
    Serial.begin(9600);
}

void loop()
{
    while(sw1==0)
    {
        while(Serial.available())
        {
            x=Serial.read();
            txt=txt+x;
            delay(5);
        }
        if(txt.length()>0)
        {
            lcd0.setCursor(0,0);
            lcd0.print("TEMP=");
            lcd0.print(txt);
            lcd0.write(0b11011111);
            lcd0.print("C ");

```

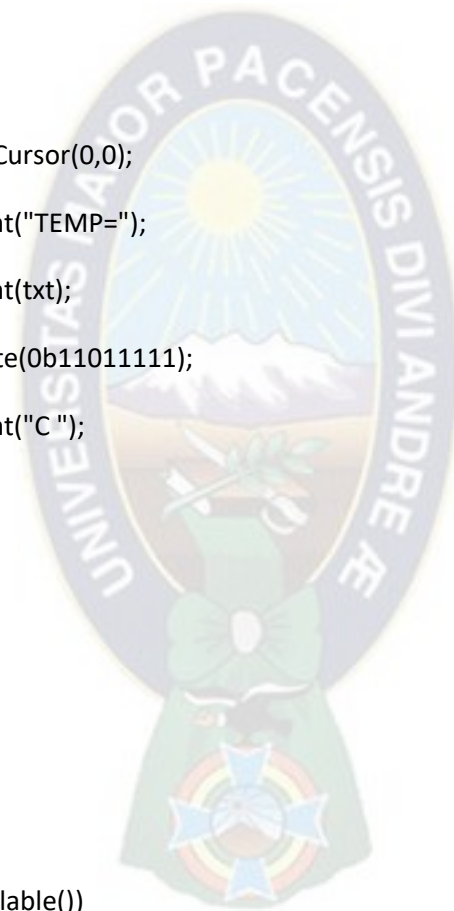


```
        txt="";
        sw1=1;
    }
}
while(sw1==1)
{
    while(Serial.available())
    {
        x=Serial.read();
        txt=txt+x;
        delay(5);
    }
    if(txt.length()>0)
    {
        lcd0.setCursor(0,1);
        lcd0.print("HUMEDAD=");
        lcd0.print(txt);
        lcd0.print("% ");
        txt="";
        sw1=2;
    }
}

while(sw1==2)
{
```



```
while(Serial.available())
{
    x=Serial.read();
    txt=txt+x;
    delay(5);
}
if(txt.length()>0)
{
    lcd1.setCursor(0,0);
    lcd1.print("TEMP=");
    lcd1.print(txt);
    lcd1.write(0b11011111);
    lcd1.print("C ");
    txt="";
    sw1=3;
}
}
while(sw1==3)
{
    while(Serial.available())
    {
        x=Serial.read();
        txt=txt+x;
        delay(5);
    }
}
```



```
if(txt.length()>0)
{
    lcd1.setCursor(0,1);
    lcd1.print("HUMEDAD=");
    lcd1.print(txt);
    lcd1.print("% ");
    txt="";
    sw1=4;
}
}

while(sw1==4)
{
    while(Serial.available())
    {
        x=Serial.read();
        txt=txt+x;
        delay(5);
    }
    if(txt.length()>0)
    {
        lcd2.setCursor(0,0);
        lcd2.print("TEMP=");
        lcd2.print(txt);
        lcd2.write(0b11011111);
    }
}
```



```

        lcd2.print("C ");

        txt="";

        sw1=5;

    }

}

while(sw1==5)

{

    while(Serial.available())

    {

        x=Serial.read();

        txt=txt+x;

        delay(5);

    }

    if(txt.length()>0)

    {

        lcd2.setCursor(0,1);

        lcd2.print("HUMEDAD=");

        lcd2.print(txt);

        lcd2.print("% ");

        txt="";

        sw1=6;

    }

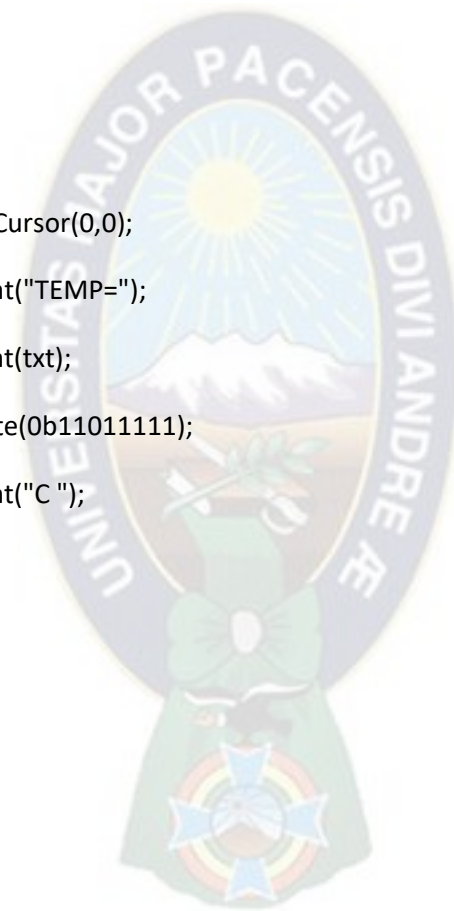
}

while(sw1==6)

```



```
{  
  while(Serial.available())  
  {  
    x=Serial.read();  
    txt=txt+x;  
    delay(5);  
  }  
  if(txt.length()>0)  
  {  
    lcd3.setCursor(0,0);  
    lcd3.print("TEMP=");  
    lcd3.print(txt);  
    lcd3.write(0b11011111);  
    lcd3.print("C ");  
    txt="";  
    sw1=7;  
  }  
}  
while(sw1==7)  
{  
  while(Serial.available())  
  {  
    x=Serial.read();  
    txt=txt+x;  
    delay(5);
```




```

}
if(txt.length()>0)
{
    lcd3.setCursor(0,1);
    lcd3.print("HUMEDAD=");
    lcd3.print(txt);
    lcd3.print("% ");
    txt="";
    sw1=8;
}
}

while(sw1==8)
{
    while(Serial.available())
    {
        x=Serial.read();
        txt=txt+x;
        delay(5);
    }
    if(txt.length()>0)
    {
        lcd4.setCursor(0,0);
        lcd4.print("TEMP=");
        lcd4.print(txt);
    }
}

```



```

        lcd4.write(Ob11011111);

        lcd4.print("C ");

        txt="";

        sw1=9;

    }

}

while(sw1==9)

{

    while(Serial.available())

    {

        x=Serial.read();

        txt=txt+x;

        delay(5);

    }

    if(txt.length()>0)

    {

        lcd4.setCursor(0,1);

        lcd4.print("HUMEDAD=");

        lcd4.print(txt);

        lcd4.print("% ");

        txt="";

        sw1=10;

    }

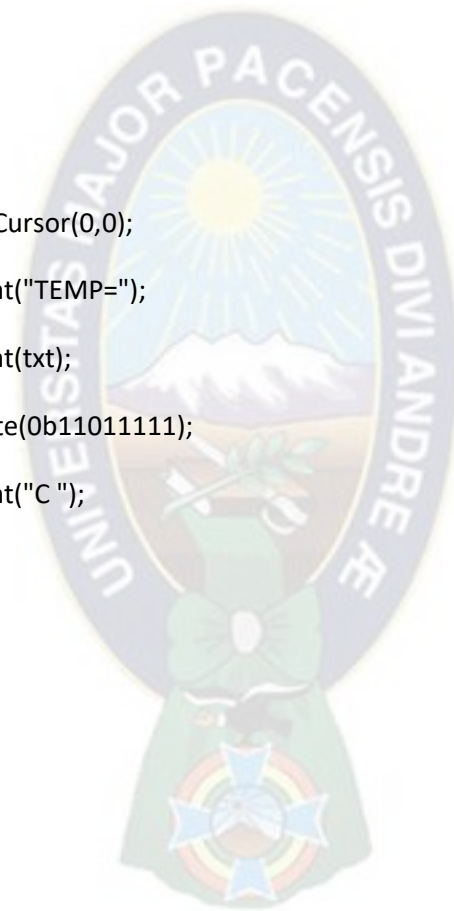
}

while(sw1==10)

```



```
{  
  while(Serial.available())  
  {  
    x=Serial.read();  
    txt=txt+x;  
    delay(5);  
  }  
  if(txt.length()>0)  
  {  
    lcd5.setCursor(0,0);  
    lcd5.print("TEMP=");  
    lcd5.print(txt);  
    lcd5.write(0b11011111);  
    lcd5.print("C ");  
    txt="";  
    sw1=11;  
  }  
}  
while(sw1==11)  
{  
  while(Serial.available())  
  {  
    x=Serial.read();  
    txt=txt+x;  
    delay(5);
```



```
}
if(txt.length()>0)
{
    lcd5.setCursor(0,1);
    lcd5.print("HUMEDAD=");
    lcd5.print(txt);
    lcd5.print("% ");
    txt="";
    sw1=12;
}
}

while(sw1==12)
{
    while(Serial.available())
    {
        x=Serial.read();
        txt=txt+x;
        delay(5);
    }
    if(txt.length()>0)
    {
        lcd6.setCursor(0,0);
        lcd6.print("TEMP=");
        lcd6.print(txt);
    }
}
```



```

        lcd6.write(Ob11011111);

        lcd6.print("C ");

        txt="";

        sw1=13;

    }

}

while(sw1==13)

{

    while(Serial.available())

    {

        x=Serial.read();

        txt=txt+x;

        delay(5);

    }

    if(txt.length()>0)

    {

        lcd6.setCursor(0,1);

        lcd6.print("HUMEDAD=");

        lcd6.print(txt);

        lcd6.print("% ");

        txt="";

        sw1=0;

    }

}

}

```



ANEXO 4



CÓDIGO CONTROL DE ACCESO A LABORATORIOS

```
#include <SPI.h>

#include <MFRC522.h>

#define tarjetas 2

MFRC522 ss0(A0, 20);

MFRC522 ss1(A1, 20);

MFRC522 ss2(A2, 20);

MFRC522 ss3(A3, 20);

MFRC522 ss4(A4, 20);

MFRC522 ss5(A5, 20);

MFRC522 ss6(A6, 20);

MFRC522 ss7(A7, 20);

MFRC522 ss8(A8, 20);

MFRC522 ss9(A9, 20);

MFRC522 ss10(A10, 20);

MFRC522 ss11(A11, 20);

MFRC522 ss12(A12, 20);

MFRC522 ss13(A13, 20);

MFRC522 ss14(A14, 20);

MFRC522 ss15(A15, 20);

MFRC522 ss16(21, 20);

int k;

int fl11=13,fl12=12;//planta baja

int fl8=11,fl9=10,fl10=9;//piso 1
```



```

int fl6=8,fl7=7;//piso 2

int fa3=6;//piso 3

int fl2=5,fl3=4,fl4=3,fl5=2;//piso 4

int fl1=14,fl22=15,fl33=16;//piso 5

int fa1=17,fa2=18;//piso 6

byte autorizados[tarjetas][6] = {

    {0x31, 0x16, 0x16, 0xDB },

    {0xFA, 0x9A, 0xAD, 0x03 },

    //,{0x10, 0x14, 0x39, 0x2E,} ejemplo de como autorizar más tarjetas 0x83,.....

};

void setup()

{

    Serial.begin(9600);

    for(k=2;k<=18;k++)

        pinMode(k,1);

    for(k=2;k<=18;k++)

        digitalWrite(k,0);

    SPI.begin();

    ss0.PCD_Init();

    ss1.PCD_Init();

    ss2.PCD_Init();

    ss3.PCD_Init();

    ss4.PCD_Init();

    ss5.PCD_Init();

    ss6.PCD_Init();

```



```
ss7.PCD_Init();
ss8.PCD_Init();
ss9.PCD_Init();
ss10.PCD_Init();
ss11.PCD_Init();
ss12.PCD_Init();
ss13.PCD_Init();
ss14.PCD_Init();
ss15.PCD_Init();
ss16.PCD_Init();
}

void loop()
{
    delay(100);
    //planta baja
    if (ss0.PICC_IsNewCardPresent())
    {
        if (ss0.PICC_ReadCardSerial())
        {
            if (autorizado(ss0.uid.uidByte))
            {
                imprimir(ss0.uid.uidByte, ss0.uid.size);
                Serial.println();
                Serial.println("rfid0 tarjeta valida");
            }
        }
    }
}
```



```
Serial.println("abriendo laboratorio 11");  
  
digitalWrite(fl11,1);  
  
delay(3000);  
  
Serial.println("cerrando laboratorio 11");  
  
digitalWrite(fl11,0);  
  
}  
  
else  
  
{  
  
    imprimir(ss0.uid.uidByte, ss0.uid.size);  
    Serial.println();  
    Serial.println("rfid0 tarjeta invalida");  
}  
    ss0.PICC_HaltA();  
}  
}  
  
if (ss1.PICC_IsNewCardPresent())  
{  
    if (ss1.PICC_ReadCardSerial())  
    {  
        if (autorizado(ss1.uid.uidByte))  
        {  
  
            imprimir(ss1.uid.uidByte, ss1.uid.size);  
  
            Serial.println();  
  
            Serial.println("rfid1 tarjeta valida");  
  
            Serial.println("abriendo laboratorio 12");
```

```
        digitalWrite(fl12,1);

        delay(3000);

        Serial.println("cerrando laboratorio 12");

        digitalWrite(fl12,0);

    }

    else

    {

        imprimir(ss1.uid.uidByte, ss1.uid.size);

        Serial.println();

        Serial.println("rfid1 tarjeta invalida");

    }

    ss1.PICC_HaltA();

}

}

//piso 1

if (ss2.PICC_IsNewCardPresent())

{

    if (ss2.PICC_ReadCardSerial())

    {

        if (autorizado(ss2.uid.uidByte))

        {

            imprimir(ss2.uid.uidByte, ss2.uid.size);

            Serial.println();

            Serial.println("rfid2 tarjeta valida");

            Serial.println("abriendo laboratorio 8");
```

```
        digitalWrite(fl8,1);
        delay(3000);
        Serial.println("cerrando laboratorio 8");
        digitalWrite(fl8,0);
    }
    else
    {
        imprimir(ss2.uid.uidByte, ss2.uid.size);
        Serial.println();
        Serial.println("rfid2 tarjeta invalida");
    }
    ss2.PICC_HaltA();
}
}
if (ss3.PICC_IsNewCardPresent())
{
    if (ss3.PICC_ReadCardSerial())
    {
        if (autorizado(ss3.uid.uidByte))
        {
            imprimir(ss3.uid.uidByte, ss3.uid.size);
            Serial.println();
            Serial.println("rfid3 tarjeta valida");
            Serial.println("abriendo laboratorio 9");
            digitalWrite(fl9,1);
```



```
        delay(3000);

        Serial.println("cerrando laboratorio 9");

        digitalWrite(fl9,0);

    }

    else

    {

        imprimir(ss3.uid.uidByte, ss3.uid.size);

        Serial.println();

        Serial.println("rfid3 tarjeta invalida");

    }

    ss3.PICC_HaltA();

}

}

if (ss4.PICC_IsNewCardPresent())

{

    if (ss4.PICC_ReadCardSerial())

    {

        if (autorizado(ss4.uid.uidByte))

        {

            imprimir(ss4.uid.uidByte, ss4.uid.size);

            Serial.println();

            Serial.println("rfid4 tarjeta valida");

            Serial.println("abriendo laboratorio 10");

            digitalWrite(fl10,1);

            delay(3000);
```

```
Serial.println("cerrando laboratorio 10");
digitalWrite(fl10,0);
}
else
{
    imprimir(ss4.uid.uidByte, ss4.uid.size);
    Serial.println();
    Serial.println("rfid4 tarjeta invalida");
}
ss4.PICC_HaltA();
}
}
//piso 2
if (ss5.PICC_IsNewCardPresent())
{
    if (ss5.PICC_ReadCardSerial())
    {
        if (autorizado(ss5.uid.uidByte))
        {
            imprimir(ss5.uid.uidByte, ss5.uid.size);
            Serial.println();
            Serial.println("rfid5 tarjeta valida");
            Serial.println("abriendo laboratorio 6");
            digitalWrite(fl6,1);
            delay(3000);
```

```
        Serial.println("cerrando laboratorio 6");
        digitalWrite(fl6,0);
    }
    else
    {
        imprimir(ss5.uid.uidByte, ss5.uid.size);
        Serial.println();
        Serial.println("rfid5 tarjeta invalida");
    }
    ss5.PICC_HaltA();
}
}
if (ss6.PICC_IsNewCardPresent())
{
    if (ss6.PICC_ReadCardSerial())
    {
        if (autorizado(ss6.uid.uidByte))
        {
            imprimir(ss6.uid.uidByte, ss6.uid.size);
            Serial.println();
            Serial.println("rfid6 tarjeta valida");
            Serial.println("abriendo laboratorio 7");
            digitalWrite(fl7,1);
            delay(3000);
            Serial.println("cerrando laboratorio 7");
```

```
        digitalWrite(fl7,0);
    }
    else
    {
        imprimir(ss6.uid.uidByte, ss6.uid.size);
        Serial.println();
        Serial.println("rfid6 tarjeta invalida");
    }
    ss6.PICC_HaltA();
}
}
//piso 3
if (ss7.PICC_IsNewCardPresent())
{
    if (ss7.PICC_ReadCardSerial())
    {
        if (autorizado(ss7.uid.uidByte))
        {
            imprimir(ss7.uid.uidByte, ss7.uid.size);
            Serial.println();
            Serial.println("rfid7 tarjeta valida");
            Serial.println("abriendo auditorium 3");
            digitalWrite(fa3,1);
            delay(3000);
            Serial.println("cerrando auditorium 3");
```

```
        digitalWrite(fa3,0);
    }
    else
    {
        imprimir(ss7.uid.uidByte, ss7.uid.size);
        Serial.println();
        Serial.println("rfid7 tarjeta invalida");
    }
    ss7.PICC_HaltA();
}
}
//piso 4
if (ss8.PICC_IsNewCardPresent())
{
    if (ss8.PICC_ReadCardSerial())
    {
        if (autorizado(ss8.uid.uidByte))
        {
            imprimir(ss8.uid.uidByte, ss8.uid.size);
            Serial.println();
            Serial.println("rfid8 tarjeta valida");
            Serial.println("abriendo laboratorio 2");
            digitalWrite(fl2,1);
            delay(3000);
            Serial.println("cerrando laboratorio 2");
```

```
        digitalWrite(fl2,0);
    }
    else
    {
        imprimir(ss8.uid.uidByte, ss8.uid.size);
        Serial.println();
        Serial.println("rfid8 tarjeta invalida");
    }
    ss8.PICC_HaltA();
}
}
if (ss9.PICC_IsNewCardPresent())
{
    if (ss9.PICC_ReadCardSerial())
    {
        if (autorizado(ss9.uid.uidByte))
        {
            imprimir(ss9.uid.uidByte, ss9.uid.size);
            Serial.println();
            Serial.println("rfid9 tarjeta valida");
            Serial.println("abriendo laboratorio 3");
            digitalWrite(fl3,1);
            delay(3000);
            Serial.println("cerrando laboratorio 3");
            digitalWrite(fl3,0);
```



```
    }
    else
    {
        imprimir(ss9.uid.uidByte, ss9.uid.size);
        Serial.println();
        Serial.println("rfid9 tarjeta invalida");
    }
    ss9.PICC_HaltA();
}
}
if (ss10.PICC_IsNewCardPresent())
{
    if (ss10.PICC_ReadCardSerial())
    {
        if (autorizado(ss10.uid.uidByte))
        {
            imprimir(ss10.uid.uidByte, ss10.uid.size);
            Serial.println();
            Serial.println("rfid10 tarjeta valida");
            Serial.println("abriendo laboratorio 4");
            digitalWrite(fl4,1);
            delay(3000);
            Serial.println("cerrando laboratorio 4");
            digitalWrite(fl4,0);
        }
    }
}
```

```
else
{
    imprimir(ss10.uid.uidByte, ss10.uid.size);
    Serial.println();
    Serial.println("rfid10 tarjeta invalida");
}
ss10.PICC_HaltA();
}
}
if (ss11.PICC_IsNewCardPresent())
{
    if (ss11.PICC_ReadCardSerial())
    {
        if (autorizado(ss11.uid.uidByte))
        {
            imprimir(ss11.uid.uidByte, ss11.uid.size);
            Serial.println();
            Serial.println("rfid11 tarjeta valida");
            Serial.println("abriendo laboratorio 5");
            digitalWrite(fl5,1);
            delay(3000);
            Serial.println("cerrando laboratorio 5");
            digitalWrite(fl5,0);
        }
    }
}
else
```

```
    {
        imprimir(ss11.uid.uidByte, ss11.uid.size);
        Serial.println();
        Serial.println("rfid11 tarjeta invalida");
    }
    ss11.PICC_HaltA();
}
}
//piso 5
if (ss12.PICC_IsNewCardPresent())
{
    if (ss12.PICC_ReadCardSerial())
    {
        if (autorizado(ss12.uid.uidByte))
        {
            imprimir(ss12.uid.uidByte, ss12.uid.size);
            Serial.println();
            Serial.println("rfid12 tarjeta valida");
            Serial.println("abriendo laboratorio 1");
            digitalWrite(fl1,1);
            delay(3000);
            Serial.println("cerrando laboratorio 1");
            digitalWrite(fl1,0);
        }
        else
```

```
    {
        imprimir(ss12.uid.uidByte, ss12.uid.size);
        Serial.println();
        Serial.println("rfid12 tarjeta invalida");
    }
    ss12.PICC_HaltA();
}
}
if (ss13.PICC_IsNewCardPresent())
{
    if (ss13.PICC_ReadCardSerial())
    {
        if (autorizado(ss13.uid.uidByte))
        {
            imprimir(ss13.uid.uidByte, ss13.uid.size);
            Serial.println();
            Serial.println("rfid13 tarjeta valida");
            Serial.println("abriendo laboratorio 22");
            digitalWrite(fl22,1);
            delay(3000);
            Serial.println("cerrando laboratorio 22");
            digitalWrite(fl22,0);
        }
        else
        {
```

```
        imprimir(ss13.uid.uidByte, ss13.uid.size);
        Serial.println();
        Serial.println("rfid913 tarjeta invalida");
    }
    ss13.PICC_HaltA();
}
}
if (ss14.PICC_IsNewCardPresent())
{
    if (ss14.PICC_ReadCardSerial())
    {
        if (autorizado(ss14.uid.uidByte))
        {
            imprimir(ss14.uid.uidByte, ss14.uid.size);
            Serial.println();
            Serial.println("rfid14 tarjeta valida");
            Serial.println("abriendo laboratorio 33");
            digitalWrite(fl33,1);
            delay(3000);
            Serial.println("cerrando laboratorio 3");
            digitalWrite(fl33,0);
        }
        else
        {
            imprimir(ss14.uid.uidByte, ss14.uid.size);
```

```
        Serial.println();

        Serial.println("rfid14 tarjeta invalida");
    }

    ss14.PICC_HaltA();
}

}

//piso 6
if (ss15.PICC_IsNewCardPresent())
{
    if (ss15.PICC_ReadCardSerial())
    {
        if (autorizado(ss15.uid.uidByte))
        {
            imprimir(ss15.uid.uidByte, ss15.uid.size);
            Serial.println();
            Serial.println("rfid15 tarjeta valida");
            Serial.println("abriendo auditorium 1");
            digitalWrite(fa1,1);
            delay(3000);
            Serial.println("cerrando auditorium 1");
            digitalWrite(fa1,0);
        }
        else
        {
            imprimir(ss15.uid.uidByte, ss15.uid.size);
```



```
        Serial.println();
        Serial.println("rfid15 tarjeta invalida");
    }
    ss15.PICC_HaltA();
}
}
if (ss16.PICC_IsNewCardPresent())
{
    if (ss16.PICC_ReadCardSerial())
    {
        if (autorizado(ss16.uid.uidByte))
        {
            imprimir(ss16.uid.uidByte, ss16.uid.size);
            Serial.println();
            Serial.println("rfid16 tarjeta valida");
            Serial.println("abriendo auditorium 2");
            digitalWrite(fa2,1);
            delay(3000);
            Serial.println("cerrando auditorium 2");
            digitalWrite(fa2,0);
        }
        else
        {
            imprimir(ss16.uid.uidByte, ss16.uid.size);
            Serial.println();
        }
    }
}
```

```

        Serial.println("rfid16 tarjeta invalida");
    }
    ss16.PICC_HaltA();
}
}
}

boolean autorizado(byte *rfid)
{
    for(int i = 0; i<tarjetas; i++)
    {
        if(esigual(rfid, autorizados[i]))
            return true;
    }
    return false;
}

boolean igual(byte *key, byte *rfid)
{
    for (int j = 0; j < 4; j++)
        {
            if (key[j] != rfid[j])
                {
                    return false;
                }
        }
    return true;
}

```



```
}  
void imprimir(byte *buffer, byte bufferSize)  
{  
    for (byte i = 0; i < bufferSize; i++)  
    {  
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");  
        Serial.print(buffer[i], HEX);  
    }  
}
```

