



UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO,
MONITOREADO Y CONTROLADO REMOTAMENTE EN TIEMPO REAL
MEDIANTE UNA APLICACIÓN ANDROID PARA SMARTPHONE”**

**Tesis de grado previa a la
obtención del título de Ingeniero en
Electrónica y Telecomunicaciones**

AUTOR

Gabriel Ricardo Charry Villamagua

DIRECTOR

Ing. Luis Enrique Chuquimarca Jiménez, M.I.

LOJA-ECUADOR

2016

CERTIFICACIÓN

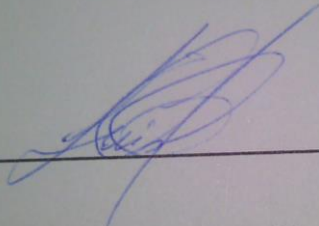
Ing. Luis Enrique Chuquimarca Jiménez, M.I.

DIRECTOR DE TESIS

CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación, cuyo tema versa en **“Diseño e implementación de un sistema domótico, monitoreado y controlado remotamente en tiempo real mediante una aplicación Android para smartphone”**, previa a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones, realizado por el señor egresado: **Gabriel Ricardo Charry Villamagua**, la misma que cumple con la reglamentación y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, 13 de enero del 2016

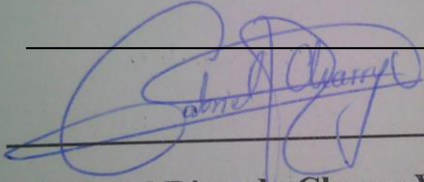


Ing. Luis Enrique Chuquimarca Jiménez, M.I.
DIRECTOR DE TESIS

AUTORÍA

Yo, GABRIEL RICARDO CHARRY VILLAMAGUA, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional - Biblioteca Virtual.



Gabriel Ricardo Charry Villamagua
110386373-2

Loja, 13 de enero del 2016

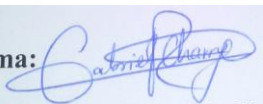
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRONICA DEL TEXTO COMPLETO.

Yo, **Gabriel Ricardo Charry Villamagua** declaro ser autor de la tesis titulada **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO, MONITOREADO Y CONTROLADO REMOTAMENTE EN TIEMPO REAL MEDIANTE UNA APLICACIÓN ANDROID PARA SMARTPHONE”**, como requisito para obtener el grado de **Ingeniero en Electrónica y Telecomunicaciones**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja a los trece días del mes de enero del dos mil dieciséis.

Firma: 

Autor: Gabriel Ricardo Charry Villamagua

Cédula: 1103863732

Dirección: Loja (Víctor M. Encalada 26-58 entre Agustín Aguirre R. y Eduardo Mora)

Correo electrónico: gabrichav@yahoo.es

Teléfono: 072616528

Celular: 0990022022

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Luis Enrique Chuquimarca Jiménez, M.I

Tribunal de Grado: Ing. Juan Gabriel Ochoa Aldeán, Mg. Sc.

Ing. Andy Fabricio Vega León, Mg. Sc.

Ing. Mario Alberto Espinoza Tituana, M.Sc.

DEDICATORIA

Esta tesis se la dedico a mis padres y hermanos, a mis amigos, maestros, y a todas las personas que siempre estuvieron ayudándome de una u otra manera durante mis años de estudio.

AGRADECIMIENTO

A Dios, por la vida que nos ha concedido a mi familia y a mí, y por todas las oportunidades que nos brinda.

A mis padres, por guiarme por el buen camino, por hacer un trabajo casi perfecto y maravilloso conmigo, y por sacrificarse como nadie más por mis hermanos y por mí.

A mis hermanos, porque siempre me han apoyado, pese a no haber estado mucho con ellos ni brindarles la mejor parte de mí, y también por su preocupación y amor incondicional.

A Karen, una persona muy especial, y al igual que mis padres y hermanos, la razón de mi vida, ya que sin su ayuda no hubiera llegado, como lo hice, hasta estas instancias de mi vida.

Finalmente, a todas las personas, especialmente a mis profesores y maestros, y quienes me han apoyado, no solamente en los estudios, sino también en cosas de la vida para crecer como persona y que me han ayudado a llegar hasta este punto, y convertirme en un profesional.

TABLA DE CONTENIDOS

CERTIFICACIÓN	II
AUTORÍA	III
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR	IV
DEDICATORIA	V
AGRADECIMIENTO.....	VI
TABLA DE CONTENIDOS	VII
ÍNDICE DE FIGURAS.....	IX
ÍNDICE DE TABLAS	X
1.- TÍTULO	1
2.- RESUMEN.....	2
2.1.- ABSTRACT	3
3.- INTRODUCCIÓN	4
4.- REVISIÓN DE LITERATURA	6
4.1 CAPÍTULO I: Domótica	6
4.1.1 Dispositivos de las Viviendas Inteligentes	7
4.1.1.1 Pasarela Residencial.....	7
4.1.1.2 Tipos de sistemas domóticos	7
4.1.1.3 Sensores.....	10
4.1.1.4 Actuadores	12
4.1.1.5 Electrodomésticos Inteligentes	12
4.1.2 Tecnologías de cableado para la Red de Datos	13
4.1.2.1 Ethernet	13
4.1.2.2 HomePlug.....	13
4.1.2.3 HomePNA	14
4.1.2.4 HomeRF	14
4.1.2.5 Bluetooth	14
4.1.2.6 Wi-Fi	14
4.1.3 Red de Control	15
4.1.4 Redes de Acceso Remoto a la Vivienda Inteligente	15
4.1.5 Principales Estándares para la Automatización de Viviendas.....	15
4.1.5.1 X-10	16
4.1.5.2 KNX.....	16
4.1.5.3 EIB	16
4.1.5.4 BatiBUS	16
4.1.5.5 SCP	17
4.1.5.6 BACnet.....	17
4.1.6 Normas para el Cableado Estructurado.....	17

4.2 CAPÍTULO II: Manipulación y control de variables a través de Arduino	18
4.2.1 Arduino	18
4.2.1.1 Descripción de la placa Arduino	19
4.2.2 Arduino Mega 2560 R3	21
4.2.3 Arduino Ethernet Shield	22
4.3 CAPÍTULO III: Desarrollo de la aplicación en Android.....	23
4.3.1 ¿Qué es un Smartphone?.....	23
4.3.2 Sistemas operativos para Smartphones	24
4.3.3 Android	26
4.3.4 Diseño de la aplicación en el MIT App Inventor de Android	27
5.- MATERIALES Y MÉTODOS.....	29
5.1 Materiales	29
5.2 Métodos	30
6.- RESULTADOS	45
6.1.- Sistema Implementado	45
6.2.- Comunicación	48
6.3.- Aplicación Android.....	48
7.- DISCUSIÓN.....	50
8.- CONCLUSIONES	53
9.- RECOMENDACIONES	55
10.- BIBLIOGRAFÍA	57
11.- ANEXOS.....	61

ÍNDICE DE FIGURAS

Figura 1. Topología Estrella	8
Figura 2. Sensor DHT11	11
Figura 3. Sensor PIR DYP-ME 003.....	11
Figura 4. Arduino UNO	20
Figura 5. Arduino Mega.	21
Figura 6. Caso extremo y no recomendable de Shields colocadas sobre la placa Arduino	22
Figura 7. Arduino Ethernet Shield.....	23
Figura 8. Smartphone.....	24
Figura 9. Arquitectura de Android.....	27
Figura 10. Topología de la red domótica.	31
Figura 11. Rango de detección del sensor PIR DYP-ME003.....	33
Figura 12. Conexiones entre el Arduino y los sensores DHT11 y DYP-ME003	34
Figura 13. Valor de temperatura instantáneo.....	35
Figura 14. Últimas mediciones de temperatura	35
Figura 15. Tweet informando presencia de intruso.....	36
Figura 16. Diagrama de flujo para programación de los sensores	37
Figura 17. Cámara IP	38
Figura 18. Diagrama de flujo para la programación respecto al control del sistema	39
Figura 19. Diagrama circuital usado para controlar el ventilador.....	40
Figura 20. Diagrama de conexión de los elementos electrónicos para el control del ventilador	42
Figura 21. Diagrama de flujo para programar la aplicación Android	43
Figura 22. Conectividad en MIT App Inventor	44
Figura 23. Sistema de Arduinos Mega	45
Figura 24. Ventilador, Calefactor y Sensor DHT11.	46
Figura 25. Sensor de presencia DYP-ME003.....	47
Figura 26. Interruptor para activar/desactivar el sensor de presencia.	47
Figura 27. Casilla para ingresar contraseña.....	48
Figura 28. Botones para controlar el sistema desde dentro o fuera de la casa.....	49
Figura 29. Botones para acceder al control, monitoreo o video vigilancia del sistema	49
Figura 30. Datasheet sensor DHT11	77
Figura 31. Parámetros eléctricos del sensor DYP-ME003.....	78
Figura 32. Ubicación de canaletas y cables dentro de ellas.....	79
Figura 33. Sellado de canaletas	79
Figura 34. Interruptores de los focos a modificar	80
Figura 35. Caja central y motor empotrados a la pared.....	80
Figura 36. Sensor DHT11 ubicado en punto estratégico.....	81
Figura 37. Ubicación del circuito controlador de las persianas en la caja central	81
Figura 38. Transformador, motor y circuito controlador dentro de la caja central	82
Figura 39. Caja central finalizada	82
Figura 40. Cámara IP ubicada	83
Figura 41. Conexiones realizadas para alimentar cámara IP	83

ÍNDICE DE TABLAS

Tabla 1. Características Eléctricas del sensor DHT11	11
Tabla 2. Características técnicas del sensor PIR DYP-ME003	12
Tabla 3. Características de operación del Arduino Mega.....	21
Tabla 4. Lista de materiales utilizados	29
Tabla 5. Lista de equipos utilizados.....	29

1.- TÍTULO

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO,
MONITOREADO Y CONTROLADO REMOTAMENTE EN TIEMPO REAL
MEDIANTE UNA APLICACIÓN ANDROID PARA SMARTPHONE”**

2.- RESUMEN

El presente proyecto de tesis tiene como objetivo diseñar e implementar un sistema domótico, el mismo que es controlado remotamente utilizando una aplicación desarrollada en el sistema operativo Android, el cual es muy común en los actuales teléfonos inteligentes.

Además, se centra en la creación de un sistema de tipo domótico, de fácil utilización para los usuarios, teniendo como punto principal el control de la iluminación, calefacción, ventilación y seguridad en una vivienda, con la capacidad de monitoreo constante y remoto.

El sistema desarrollado en este proyecto tiene la característica de utilizar una arquitectura de comunicación en forma de estrella, teniendo como nodo central un sistema de Arduinos Mega R3 2560 junto con su respectiva Shield Ethernet Mega, los cuales se encargan del control de todo el sistema a través de la aplicación desarrollada para Android; así mismo, los nodos de calefacción y ventilación son monitoreados a través del sensor de temperatura y humedad DHT11, y finalmente el sensor de presencia PIR DYP–ME003 informa si algún intruso ha ingresado en el hogar.

La interfaz de usuario es sencilla de entender, amigable hacia el usuario, y se puede instalar en cualquier dispositivo Android (smartphone, Tablet). Consta de pocos botones, los mismos que permiten elegir al usuario si va a realizar el control y monitoreo desde adentro o fuera de la red LAN, abrir visores web para observar el estado de los sensores y poder acceder a la cámara IP y al control del hogar. Sólo basta con tener una conexión a internet en nuestro equipo móvil.

2.1.- ABSTRACT

This thesis aims to design and implement an automated system, the same that is controlled remotely using an application developed on Android's operating system, which is very common in today's smartphones.

In addition, it focuses on the creation of a home automation system type, easy to use for users, with the main point control lighting, heating, ventilation and safety in housing, with the ability to constantly and remote monitoring.

The system developed in this project has the characteristic of a communication architecture using star-shaped, with the central node system Arduino Mega R3 2560 together with their respective Ethernet Mega Shield, which are responsible for control of the entire system through the application developed for Android; likewise, heating and ventilation nodes are monitored through temperature and humidity sensor DHT11, and finally the presence sensor PIR DYP - ME 003 reports if an intruder has entered the home.

The user interface is simple to understand, friendly to the user, and can be installed on any Android device (smartphone, tablet). It consists of a few buttons, the same that allow the user to choose whether to perform control and monitoring from inside or outside the LAN, open web viewers to observe the state of the sensors and to access the IP camera and control home. Just enough to have an internet connection to our mobile equipment.

3.- INTRODUCCIÓN

La demanda por el confort, la gestión y uso eficiente de la energía e iluminación, junto con la seguridad del hogar, cada vez toma más importancia, por lo que en la actualidad la domótica está siendo considerada para la automatización de edificios, centros comerciales y viviendas familiares, generando un fuerte impacto en la sociedad gracias a las ventajas que ofrece y sobre todo por la facultad de acoplarse a dispositivos remotos, tales como smartphones, Tablets y otros dispositivos inteligentes.

Gran cantidad de viviendas han empezado a incorporar una serie de automatismos, la mayoría de ellos basados en el control de iluminación, puertas o seguridad, todos estos instalados de forma individual y no conectados a un solo dispositivo central, por lo que se estaría hablando de una casa automatizada, más no de una casa domotizada. El presente proyecto propone una alternativa en el control de los sistemas implementados en una red domótica, con la ayuda de la plataforma Arduino Mega R3 2560 y una aplicación desarrollada en Android, los mismos que permiten al usuario tener el control total desde el exterior de la casa, ya sea desde la oficina, un parque, otra ciudad o cualquier sitio que tenga acceso al Internet, para poder manejar el sistema de forma remota.

Por tanto, el hardware y software desarrollados permitirán la creación de un sistema domótico de bajo costo y fácil de utilizar, mejorando de esta manera la calidad de vida de los usuarios, y a su vez, abriéndose paso en un mercado que se encuentra completamente disponible en nuestro país; además, los sistemas domóticos instalados por empresas comerciales tienen costos muy elevados para ser implementados en los hogares.

Objetivo General

Diseñar e implementar un sistema domótico que pueda ser monitoreado y controlado remotamente a través de una aplicación Android para Smartphone

Objetivos Específicos

- Realizar el diseño de un sistema domótico para una vivienda, que se ajuste a las condiciones o parámetros de control establecidos con anterioridad.

- Utilizar el diseño elaborado del sistema domótico, para implementarlo en una vivienda.
- Crear una aplicación Android para el control y monitoreo remoto del sistema domótico.

4.- REVISIÓN DE LITERATURA

4.1 CAPÍTULO I: Domótica

El término domótica es utilizado para referirse a la automatización de una vivienda que permite el control local o remoto de la misma mediante la implementación de equipos y tecnologías adecuados, promoviéndole al usuario comodidad, seguridad y ahorro energético [1].

El control domótico de la vivienda es una evolución de la vivienda tradicional donde además de poseer puertas y ventanas automatizadas, es necesario disponer al menos de un sistema de detección de intrusos, climatización y acondicionamiento; es por eso que los servicios que ofrece un sistema domótico están inmersos en cuatro áreas, las cuales son:

- Seguridad y vigilancia: Contempla tanto la seguridad personal como la seguridad del patrimonio, consiste en resguardar el hogar contra intrusos, además de brindar seguridad ante accidentes y prestaciones de salud.
- Automatización y control: Facilita al usuario la obtención de un mayor nivel de comodidad en las actividades que desarrolle dentro de la vivienda; es así que en este aspecto se incluyen servicios que incorporen algún tipo de automatismo como el control y accionamiento de persianas, accionamiento automático de electrodomésticos, de sistemas de riego, sistemas de gestión y control de alarmas, de iluminación, etc.
- Comunicaciones: Ofrecen la posibilidad de intercomunicación entre las diferentes redes de la vivienda, (por ejemplo al encender un foco desde internet, están interactuando la red eléctrica y la red de datos), además de lograr comunicarse con redes externas como Internet.
- Tele servicios y ocio: Consiste en realizar todas las tareas sin la necesidad de salir del hogar, como lo son telebanca, aparte de tener servicios tales como video bajo demanda, juegos en línea, etc [2].

4.1.1 Dispositivos de las Viviendas Inteligentes

4.1.1.1 Pasarela Residencial

Conocida también como *residential gateway*, es el equipo que se encuentra en el límite entre las redes de acceso externas y las redes internas de la casa inteligente. Este dispositivo es instalado en un hogar por medio del Proveedor de Servicios de Internet (ISP), debido a que por medio de este aparato, el operador podrá ofrecer sus servicios al usuario. Básicamente se trataría de un router inteligente que podrá ofrecer interconexiones, y servicios de valor añadido, como seguridad, entretenimiento, comunicaciones, etc.

En una residencia, dichas pasarelas ayudan en las necesidades de convergencia, que cada vez son mayores en la actualidad, y se deben prácticamente a la aparición de nuevas tecnologías de comunicación en la casa como el incremento del número de computadoras en el hogar, aparición de nuevos dispositivos y electrodomésticos que necesitan de una conexión a internet, entre otros.

Una pasarela posee las siguientes características:

- Debe ser sencilla la instalación, por lo general *Plug and Play* (enchufar y usar).
- Debe ser muy segura debido a que si ladrones informáticos logran penetrarla, podrán acceder a todos los artefactos de nuestra red.
- Debe permitir que se le realice funciones de mantenimiento para detectar errores en ella o algún otro componente dentro de la red.
- Debe ser escalable, es decir, debe permitir la expansión y ser adaptable.
- Debe ser capaz de soportar múltiples servicios tales como capacidad de procesamiento, memoria y poseer un sistema operativo robusto [3].

4.1.1.2 Tipos de sistemas domóticos

Sistemas descentralizados: Son aquellos que no se rigen por una plataforma central, es decir, todos los elementos del sistema domótico descentralizado tienen inteligencia por tanto son totalmente independientes unos de otros, por lo que no es necesario disponer de un dispositivo central que esté controlando todos los dispositivos conectados a la red. Se utilizan en sistemas domóticos de alta gama, es decir, sistemas domóticos más sofisticados, dentro de los cuales cada dispositivo final tiene un microprocesador que le

permite “pensar por sí mismo”, que acción tomar. Por ejemplo, una lavadora podría elegir el tipo de ropa que va a lavar, sin necesidad que un dispositivo central le ordene que acción ejecutar [3].

Sistemas centralizados: Son sistemas que poseen un dispositivo central o unidad central inteligente, en este caso, un Arduino Mega (ver capítulo II, sección d.2.2), que se encarga del procesamiento de toda la información del sistema y controla los dispositivos actuadores, tales como: relés, contactores y motores; además de recibir información de diferentes sensores, siendo capaz de monitorearlos, a través de una programación previamente desarrollada [1], [3].

En resumen, el dispositivo principal se asemeja al cerebro, puesto que si deja de funcionar, todo el sistema en la vivienda se vendría abajo, siendo ésta su principal desventaja [3]. La función del Arduino, como unidad central, es recibir la información de las variables de control (sensores, información que le llega desde el smartphone), y ordena que las variables a controlar (motores, relés, etc.), ejecuten las acciones que les corresponden realizar [1].

En el sistema centralizado se utiliza una topología en estrella, debido a que depende de un nodo principal (nodo central), y de nodos periféricos (ver Figura 1); cabe señalar, además, que los nodos periféricos se pueden conectar con sus similares únicamente a través del nodo central. En este tipo de topología, el nodo central se conecta con los nodos periféricos a través de enlaces punto a punto [4], [5].

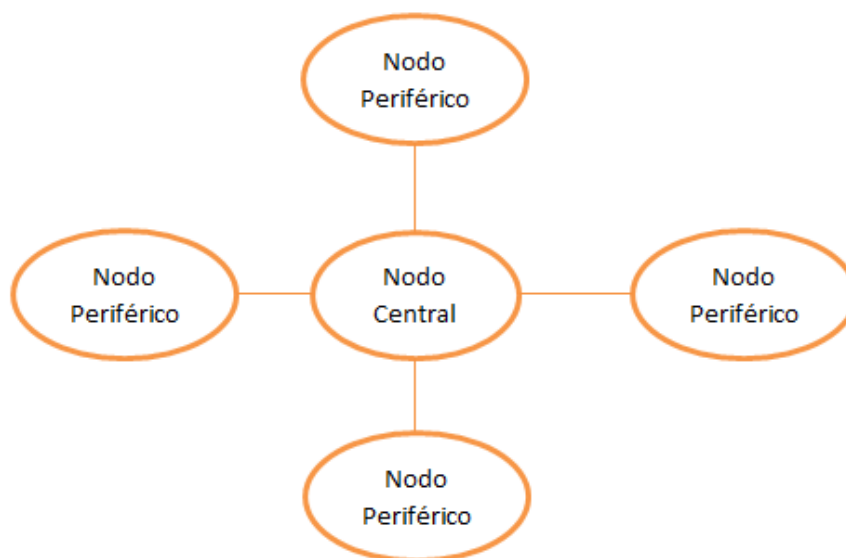


Figura 1. Topología Estrella [El autor].

Se encarga de recolectar la información brindada por los distintos sensores del hogar, para luego procesarla y producir las distintas órdenes para que los actuadores puedan ejecutar las acciones correspondientes.

Posee la ventaja de reducir el número de equipos necesarios para controlar toda la vivienda, puesto que todos los sistemas se conectarían directamente a dicho artefacto, aunque dependiendo del fabricante es posible conectar un número limitado de dispositivos a controlarse.

La comunicación entre los distintos elementos del sistema y el dispositivo central de control se realiza a través de protocolos domóticos, como lo son X-10, EIB, EHS, etc., de los cuales se hablará más adelante. Actualmente se están utilizando tecnologías de comunicación inalámbrica, ya que éstas permiten escalabilidad, flexibilidad y facilidad en la instalación del sistema, y en caso de que algún intruso pretenda penetrar dicho sistema se están proporcionando distintos códigos para encriptar la información [3].

Un usuario, para realizar el control respectivo de los elementos del hogar, debe conectarse, de manera local o remota, al aparato central a través de una interfaz. A continuación se puede apreciar algunos tipos de interfaces:

- Interfaz local: la centralilla cuenta con una pantalla y un teclado para realizar configuraciones de manera local. Posee una interfaz sencilla y básica debido al tamaño y resolución de la pantalla. Se conecta a una pc a través de un cable usb para configurar dicho equipo. Se utilizan pulsadores para activar o desactivar la centralilla.
- Interfaz de voz: debe poseer un módulo GMS en su interior o estar conectado a una toma telefónica (a través de RJ11 o RJ45 si se trata de RTC o RDSI respectivamente). El usuario podrá conocer el estado de la red tan sólo con marcar un número telefónico desde un teléfono fijo o teléfono celular. La centralita informará al usuario a través de palabras o frases pregrabadas y almacenadas en el software del equipo central. Se puede introducir una contraseña al sistema para mayor seguridad.
- Interfaz web: posee un servidor web integrado, ofrece mayor facilidad para el uso de las personas de la casa, tiene una interfaz muy potente y rápida que otras interfaces (como la de voz).

- Interfaz de mensajes móviles: utilizado más para sistemas de seguridad, puesto que en caso de robo o intrusión de un extraño, el sistema puede enviar un mensaje al usuario debido a que posee conexión con la red GSM (a través de una tarjeta SIM por ejemplo) [3].

4.1.1.3 Sensores

Un sistema domótico debe ser sencillo, eficiente y robusto; lo cual depende de los parámetros que se quiere controlar y de los sensores a monitorear, necesarios para conocer el estado en que se encuentran los diversos factores del hogar. En efecto, la domotización supone incorporar nuevos elementos en la vivienda, entre ellos un elemento de control centralizado, los sensores, los actuadores y los aparatos electrónicos dotados de capacidad de intercomunicación [1].

Los sensores son dispositivos que ayudan a medir factores físicos o químicos que suceden a su alrededor, indicando luego la información de cómo se encuentran las variables medidas o si existe algún tipo de peligro en torno al hogar. Amplifican los sentidos de las personas, captan mejor las señales que no se podrían recibir, apreciar y notar con total exactitud y la transforman en una medida que todos pueden captar, de tal forma uno se da cuenta de la variación que ha ocurrido con dicha energía o variable medida en el medio [6].

Existe una amplia variedad de sensores, tal es el caso de sensores de temperatura, de humedad, de presencia (para detectar personas), para determinar si hay fugas de gas o agua, de intensidad lumínica, de distancia, entre otros [6]–[8]. De toda esta gama de sensores, en la elaboración del presente trabajo de titulación se utilizaron los sensores detallados a continuación:

- Sensor de temperatura y humedad (DHT 11): Permite medir la temperatura de un ambiente o lugar, y la humedad que éste presenta; otorga datos digitales, por lo que sólo permitirá medir datos enteros o sin decimales (ver Anexo 5-1) [9], [10].

Se alimenta con 5 voltios, posee una salida digital, y la distancia máxima a la que puede conectarse a través de un cable es de 20 metros. Las características eléctricas del sensor se muestran a continuación [11], [12]:

Tabla 1. Características Eléctricas del sensor DHT11 [11], [12].

	Condición	Mínimo	Típico	Máximo
Alimentación	DC	3V	5V	5.5V
Suministro de Corriente	En medición	0.5mA		2.5mA
	Promedio	0.2mA		1mA
	En espera	100uA		150uA
Rango de Humedad	Normal	20% RH		90% RH
Rango de Temperatura	Normal	0°C		50°C

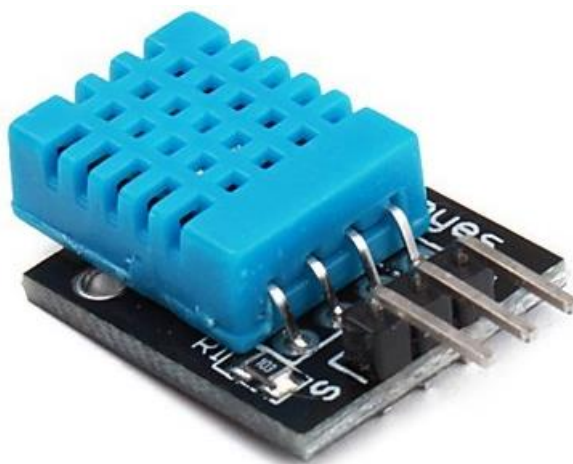


Figura 2. Sensor DHT11 [10].

- Sensor de presencia (PIR DYP – ME 003): Detecta los niveles de radiación infrarroja en un campo visual de 100°, mismo que le permite divisar, con gran exactitud, si algún objeto se cruza delante de él (ver Anexo 5-2) [13], [14].



Figura 3. Sensor PIR DYP-ME 003 [14].

Se lo puede alimentar con un voltaje de 4.5 a 20 voltios, otorga una señal digital a su salida, y se lo puede ubicar a una distancia de 3 metros (predeterminado) a 7 metros (valor máximo) de la fuente de alimentación y recepción de datos [14].

Posee las siguientes características eléctricas:

Tabla 2. Características técnicas del sensor PIR DYP-ME003 [14].

	Condición	Mínimo	Típico	Máximo
Alimentación	DC	4.5V	5V	20V
Suministro de Corriente	En espera		< 50uA	
Nivel de Salida	Alto		3.3V	
	Bajo		0V	

4.1.1.4 Actuadores

Son elementos utilizados por el sistema para modificar el estado de ciertas instalaciones (como la eléctrica) y ciertos equipos. Los más utilizados son los contactores de carril o relés de actuación, los contactores para base de enchufe, las electroválvulas de corte de suministro (por ejemplo para gas y agua) y las sirenas o alarmas [3].

Son usados por ejemplo en conjunto con los sensores de presencia, de manera que cuando el sensor detecte movimiento, pueda encender mediante un relé un foco o una lámpara.

Los actuadores vienen de la mano con los parámetros de control, que son variables que se pueden modificar en dependencia de las circunstancias o necesidades que se deseen. Se puede controlar modificar la iluminación, la temperatura ambiente, aparatos de riego, el movimiento de puertas, ventanas, cortinas y persianas, el encendido y apagado de electrodomésticos, cámaras de vigilancia, y algunos otros elementos más [1].

4.1.1.5 Electrodomésticos Inteligentes

Los electrodomésticos (frigoríficos, lavadoras, hornos, etc.) de nueva generación, que muy pronto van a formar parte de instalaciones domóticas modernas, son llamados también electrodomésticos domóticos ya que pueden conectarse a la red y ser controlados y monitoreados por teléfono o por internet, pero aún no están disponibles en el mercado para consumidores de nivel económico medio, debido a que poseen un precio extremadamente caro.

Un ejemplo de electrodoméstico inteligente es el frigorífico adaptado con una pantalla táctil y ordenador empotrados en la puerta, desde el cual se puede realizar el control de todos los otros electrodomésticos de la vivienda. Desde este frigorífico se puede

navegar por internet y realizar compras sin salir de la casa, consultar recetas, hacer un repaso a las provisiones existentes y su fecha de caducidad.

Este y otros electrodomésticos tienen diseños novedosos y compactos, muy atractivos para los usuarios, pudiendo adaptarse a todo tipo de habitaciones (cocinas por ejemplo), siendo sencillos de manejar ya que poseen interfaces amigables al usuario [3].

4.1.2 Tecnologías de cableado para la Red de Datos

Las redes para las residencias se clasifican en tres categorías:

- Cables nuevos: obligan a instalar, en la casa, una infraestructura cableada y son las que necesitan menos inversión en cuanto a equipamiento y accesorios se refiere. Sobresale Ethernet.
- Sin nuevos cables: utilizan la infraestructura actual en la vivienda para crear una red LAN. Sobresalen HomePNA y HomePlug.
- Inalámbricas o vía radio: proporcionan movilidad dentro de la vivienda pero con un ancho de banda limitado. Sobresalen Bluetooth, HomeRF y Wi-Fi [3].

4.1.2.1 Ethernet

Es la tecnología más usada en residencias y oficinas a través del cableado estructurado con cable UTP de bajo costo. Cubre alrededor del 95% de las redes empresariales. Se pueden ubicar tomas para computadoras en cualquier lugar de la vivienda, otorgando conectividad a los dispositivos que se necesite, a través del cable UTP categoría 5e (como mínimo). Utiliza el protocolo de acceso al medio CSMA/CD (Acceso Múltiple por Detección de Portadora/Detección de Colisiones). Para que algunas aplicaciones puedan acceder remotamente a equipos, se usa protocolos de red, como TCP/IP. Esta red, desde su nacimiento, ha ido evolucionando gracias a la demanda de los usuarios, primero a Fast Ethernet, a Gigabit Ethernet, y a 10 Gigabit Ethernet. [3].

4.1.2.2 HomePlug

Utiliza la red eléctrica de baja tensión para enviar datos (a través de la modulación OFDM), evitando así los costos de instalación de nuevos cables. Otorga mucha flexibilidad a la red y movilidad, debido a que los usuarios pueden conectarse a la red desde cualquier lugar de la casa donde haya una toma eléctrica. Es el competidor directo

de Wi-Fi, aunque los chips (PLC) para trabajar con HomePlug son más caros ya que se tiene que eliminar el ruido provocado por la red de electricidad [3].

4.1.2.3 HomePNA

Aprovecha el cableado eléctrico existente en una vivienda para enviar datos y voz. Se debe tener una toma en cada habitación para que sea posible utilizar mencionada tecnología, puesto que si se posee sólo una toma telefónica en la vivienda, no se podrán conectar los artefactos necesarios para tener la red domótica. Utilizada especialmente para el entorno SOHO (Small Office, Home Office), es decir para personas que poseen la oficina en su propia casa [3].

4.1.2.4 HomeRF

Es un tipo de tecnología de comunicación inalámbrica. Permite distribuir en audio y video en equipos cuyos recursos de hardware son escasos. Fue creado con la idea de que en una pc se descargue videos, canciones, etc. por medio de internet, y a través de la tecnología HomeRF poder reproducir de dichos archivos en los dispositivos mencionados. Además posee la cualidad de proveer el acceso simultáneo por voz de hasta 8 usuarios a la centralita, a diferencia de Bluetooth y Wi-Fi, los cuales no poseen dicha capacidad de comunicación [3].

4.1.2.5 Bluetooth

Enlace inalámbrico de corto alcance (hasta 10 metros) usado en las redes inalámbricas de área personal (WPAN) más para transferencia de archivos y datos. Es una red de bajo costo cuya función principal es que todos los dispositivos de la red puedan interconectarse e interoperar entre sí. Permite la transmisión de voz, datos, e incluso video. Se considera una tecnología complementaria con Wi-Fi, debido a que mencionada tecnología es clave en redes WLAN, mientras que la tecnología Bluetooth es importante para las redes WPAN [3].

4.1.2.6 Wi-Fi

Nombre comercial de IEEE 802.11b. Funciona en la frecuencia de 2.4 GHz (banda ISM). El alcance que posee es mucho mayor al del Bluetooth y va desde los 25 metros hasta los 300 metros, dependiendo de los factores y equipos de transmisión y recepción. Sus ondas electromagnéticas son capaces de atravesar muros y otros obstáculos,

necesario para cubrir a la perfección una vivienda. Permite la conexión de dos o más dispositivos entre sí. Posee una gran aceptación en locales con puntos de acceso públicos, como lo son los hoteles, parques, aeropuertos, campus universitarios, etc. [3].

4.1.3 Red de Control

Es utilizada para aplicaciones destinadas a la automatización del hogar, por lo que queda reservada únicamente para el manejo de los distintos sensores y actuadores de la red. Esta red de control suele ser centralizada porque trata de reducir la complejidad y también el tamaño de los sensores y actuadores, de manera que el equipo central posea toda la inteligencia de la red y permita al usuario interactuar con todos los elementos a través de él [3].

4.1.4 Redes de Acceso Remoto a la Vivienda Inteligente

Resulta interesante y seguro para los usuarios de las redes domóticas el poder ingresar al sistema desde otros lugares, por ejemplo cuando se encuentren de vacaciones y tengan que realizar un viaje, o si pretenden que el sistema les envíe periódicamente mensajes al celular de cómo se encuentra la red o la vivienda en caso de que no se encuentren en ella momentáneamente, por lo que algunas maneras de acceder a la red remotamente se pueden dar a través de:

- Internet.
- Red Telefónica.
- Cable.
- LMDS (*Local Multipoint Distribution Service*).
- PLC (*Power Line Communications*).
- Satélite.
- Comunicaciones móviles [3].

4.1.5 Principales Estándares para la Automatización de Viviendas

Actualmente existen protocolos de control para dispositivos domóticos, que están diseñados para cubrir necesidades específicas, y se los expone a continuación [3].

4.1.5.1 X-10

Protocolo más antiguo usado en aplicaciones de domótica. Puede transmitir datos a través de la red eléctrica de baja tensión, lo que permite que no se tiendan nuevos cables. Es un protocolo no propietario, y los circuitos integrados implementados por este protocolo tienen un costo muy bajo. Es el protocolo más usado en los Estados Unidos y en Europa a nivel residencial y de pequeñas empresas. Los productos X-10 comprados en USA son totalmente incompatibles con la red eléctrica europea [3], [8].

4.1.5.2 KNX

Protocolo para las viviendas y las oficinas. Evolución y unión de los protocolos EIB, EHS y BatiBUS para crear un solo protocolo europeo. Puede trabajar sobre cuatro medios físicos distintos permitiendo a los instaladores adaptar la red a las condiciones de una vivienda, aunque lo más habitual es que el protocolo KNX utilice cableado propio de par trenzado [3].

4.1.5.3 EIB

EIB (*European Installation Bus*) es un protocolo europeo creado para contrarrestar el mercado japonés y americano. Se basa en el modelo OSI y es un sistema descentralizado. Una gran variedad de electrodomésticos se pueden conectar vía EIB al dispositivo central, pudiéndose programar y ajustar desde cualquier lugar de la casa. Trabaja sobre par trenzado, tiene una topología bus y utiliza el protocolo CSMA/CA (Acceso Múltiple por Detección de Portadora/Evita Colisiones) [3], [8].

4.1.5.4 BatiBUS

El estándar BatiBUS es un bus simple que utiliza como medio físico el par trenzado, aunque también puede utilizar el cable telefónico o el cable eléctrico. Soporta cualquier topología de red, como la topología en anillo, árbol, estrella, etc. El cable permite la intercomunicación entre todos los sistemas de control del edificio o vivienda, en este caso, como: el sistema de aire acondicionado, control de luces, aperturas y cierres, sistema de calefacción, etc. además, el bus proporciona energía a los distintos sensores, por lo que no se necesita de una fuente externa para alimentarlos. Utiliza, a nivel de acceso, la técnica de CSMA/CA, por lo que cualquier elemento podrá comunicar un mensaje siempre y cuando la línea esté libre o disponible [3], [8].

4.1.5.5 SCP

SCP (*Simple Control Protocol*) es un protocolo creado por Microsoft y General Electric para la automatización de edificios y viviendas. Permite una comunicación segura y robusta en redes de comunicación de baja velocidad y mucho ruido entre dispositivos con memoria y capacidad de proceso limitadas. En la actualidad, algunas compañías como Domsys o Mitsubishi Electric Corporation fabrican circuitos integrados de bajo costo y que ocupan poco espacio, permitiendo su uso en una amplia variedad de electrodomésticos y equipos electrónicos usados en las casas [3], [8].

4.1.5.6 BACnet

Protocolo norteamericano abierto, utilizado para la automatización del hogar, que es patrocinado por fabricantes de equipos de calefacción y aire acondicionado. Posee un conjunto de reglas de hardware y software que permiten la comunicación entre dos dispositivos de la red, independientemente de si éstos están usando protocolos como EIB, Batibus, EHS, TCP/IP, etc. Soporta hasta cinco opciones distintas en cuanto a tecnologías de red se refiere [3], [8].

4.1.6 Normas para el Cableado Estructurado

La Asociación de Industrias Electrónicas (EIA) y la Asociación de la Industria de las Telecomunicaciones (TIA), desarrollan y publican estándares y normas para el cableado estructurado para las redes LAN. A continuación se presenta cuáles son los estándares utilizados con mayor frecuencia en las instalaciones de cableado estructurado:

- TIA/EIA 568A: es un estándar desarrollado para el cableado de telecomunicaciones dentro de edificios comerciales. Dentro de éste se especifican las topologías de red recomendadas, límites de distancia, requisitos mínimos para el cableado, conectores, etc. ha sido reemplazado actualmente por TIA/EIA 568B.
- TIA/EIA 568B: es también un estándar de cableado que especifica los requisitos de los componentes y de la transmisión según los medios que se disponga o elija. Provee un sistema de cableado genérico, específicamente para telecomunicaciones, que soporta un entorno lleno de varios productos y proveedores. En sus respectivas enmiendas se puede conocer cuál curvatura maneja para los cables UTP y par trenzado apantallado, además de los componentes de cableado, modelos de sistemas, entre otras cosas.

- TIA/EIA 569A: se usa en edificios comerciales para recorridos y Espacios de Telecomunicaciones, donde se especifica los diseños y prácticas de construcción dentro de cada edificio. Específicamente se dan estándares para trabajar en salas o áreas en los que se instalan equipos y medios de telecomunicaciones.
- TIA/EIA 570A: es el estándar usado para comerciales menores y para residencias, donde se incluye soporte para seguridad, audio, alarmas, sensores, intercomunicadores y televisión. Este estándar debe implementarse en viviendas en construcción, en extensiones del hogar o remodelaciones de edificios donde vivan uno o más inquilinos.
- TIA/EIA 606: es el estándar de Administración para la Infraestructura de las Telecomunicaciones dentro de edificios comerciales, además se incluye estándares para la rotulación del cableado. Cada identificación debe ser exclusiva a cada unidad de conexión y debe estar marcado tanto en el hardware como en el cableado. Todas las etiquetas deberán cumplir los requisitos propios de legibilidad, adhesión y protección.
- TIA/EIA 607: es el estándar de Requisitos de Conexión a Tierra y conexión de Telecomunicaciones para Edificios Comerciales. Admite un entorno de trabajo de varios proveedores y equipos, así como las distintas prácticas, para diversos sistemas, de conexión a tierra. en él se especifican los puntos de la interfaz entre los sistemas de puesta a tierra de los edificios y la configuración de las conexiones a tierra de los equipos de telecomunicaciones, así como también las configuraciones de la conexión a tierra ideales para admitir mencionados equipos [15], [16].

En el presenta trabajo de titulación se trabajará con la norma TIA/EIA 570A, puesto que es una normativa para uso residencial, y se acoplará a las necesidades que se disponga. En ella se mencionan aspectos importantes para la red, tales como la topología, que debe ser en estrella, además del cableado, que debe ser de máximo 150 metros desde el Punto de Acceso del ISP hasta el resto de equipos finales del usuario. Además se escogerán algunos puntos de otros estándares para que el cableado del hogar esté lo más correcto posible en cuanto a estandarización se refiere [17].

4.2 CAPÍTULO II: Manipulación y control de variables a través de Arduino

4.2.1 Arduino

La plataforma Arduino es una plataforma *Open Source* tanto para *software* como para *hardware*, está basado en una placa de circuito impreso que contiene un microcontrolador Atmega propio de la marca Atmel. Cuenta con entradas y salidas analógicas y digitales útiles en el momento de programar y controlar dispositivos tales como sensores, alarmas, motores, actuadores, etc. [18], [19].

Ofrece un entorno de desarrollo intuitivo simplificando el proceso de trabajar con microcontroladores, además brinda varios tipos de placas con diferentes características según las necesidades del programador, las mismas que son accesibles y de bajo costo. Al ser una herramienta *Open Source*, tanto el software como el hardware de Arduino están publicados bajo licencia libre, permitiendo a los programadores y desarrolladores experimentados ampliar dicha plataforma. Así mismo posee una página web en la que se puede encontrar todo lo necesario para la programación de Arduino, desde la opción de descarga del entorno de desarrollo IDE, su instalación, librerías y unas pequeñas guías de programación [19].

Se programa en una computadora, en el programa específico de Arduino (entorno de desarrollo IDE), y se transfiere dicha programación por medio de un cable USB propio para dicha placa. Se puede alimentar directamente con el cable USB, pero para aplicaciones experimentales es necesario alimentar con una fuente externa de 7 a 12V, debido a la corriente que entrega el puerto USB de la computadora.

En el IDE de Arduino, la estructura de programación es sencilla. Primero se añaden las librerías propias de Arduino, y en este caso, de los componentes relaciones directamente con Arduino, tal es el caso de los sensores. Luego se coloca el *setup* y *loop*. El “*void setup*” se ejecuta una única vez, es la primera acción que realiza el programa y tiene que ver con la preparación del mismo (declaración de las distintas variables, nombrar pines como entradas o salidas, etc.). El “*void loop*” va a ser leído constantemente pues es la ejecución en sí del programa, contiene las órdenes (código) para controlar cada variable establecida en el “*void setup*” [20].

4.2.1.1 Descripción de la placa Arduino

Es una placa de libre utilización, que uno mismo puede construir teniendo los componentes necesarios y las pistas correctas del circuito impreso [20]. La tarjeta Arduino contiene trece entradas/salidas digitales, seis entradas analógicas y un puerto

serial que permite realizar la comunicación con periféricos (ver Figura 4). Además incluye un conector USB y un pulsador para resetear la placa en caso que exista fallos en los procesos que se vayan a realizar con la tarjeta Arduino [19], [21].

Las tarjetas Arduino son placas que contienen un microcontrolador Atmega perteneciente a la familia AVR 8-bits RISC (*Reduce Instruction Set Computer*), cuyo diseño está basado en la arquitectura Harvard, es decir, la unidad de proceso tiene por separado la memoria de programa y la memoria de datos. La característica más importante del AVR es que el núcleo cuenta con memoria de programa tipo flash lo que permite almacenar los códigos de programa que se realicen. Esta memoria de programa puede contener entre 4 Kbytes y 256 Kbytes de información [19].

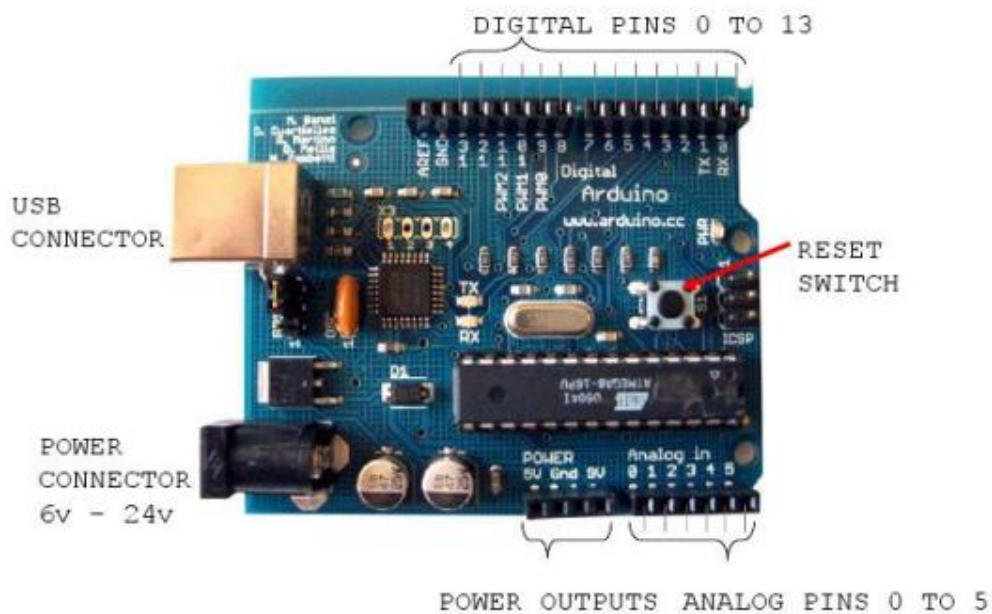


Figura 4. Arduino UNO [21].

La memoria de datos tiene por función principal contener un espacio de almacenamiento temporal. En esta memoria se tiene tanto la memoria RAM interna y externa como también la memoria EEPROM.

Dentro de la comunidad Arduino se han ido desarrollando gran variedad de tarjetas Arduino, las mismas que responden a las diferentes extensiones que se han ido realizando desde las primeras placas, según la demanda y aceptación de los usuarios. Las placas Arduino más utilizadas y que varían según las funciones a realizar tenemos: Arduino UNO, Duemilanove, Arduino BT, Lilypad, Leonardo y el Arduino Mega 2560 R3. De toda esta clasificación, en la siguiente sección se detalla sobre el modelo Mega

2560 R3, ya que este es el modelo de la placa Arduino que se utiliza en el desarrollo de este trabajo de investigación [19].

4.2.2 Arduino Mega 2560 R3

La placa Arduino Mega 2560 R3 es una plataforma potente y la más grande de todas sus antecesoras, está basado en el microcontrolador Atmega 2560 con una capacidad de memoria flash de 256 KB para poder almacenar información, de los cuales 8 KB son utilizados para el gestor de arranque; 8 KB de memoria SRAM y 4 KB de memoria EEPROM. Tiene 54 pines digitales de entrada y salida, de los cuales 15 pueden ser utilizados como salida PWM. Posee 16 entradas analógicas, 4 puertos seriales, un oscilador de 16 MHz, una conexión USB, cabecera ICSP, un botón de reinicio y un conector para su alimentación, la misma que puede ser brindada externamente mediante una batería o adaptador AC/DC, entre del rango de 6 a 20 voltios como límite [19], [21].



Figura 5. Arduino Mega [El autor].

En la siguiente tabla se resumen las características de operación necesarias para trabajar con una placa Arduino Mega 2560 R3 [19], [22]:

Tabla 3. Características de operación del Arduino Mega [19], [22].

Voltaje de operación	7 a 12 V
Voltaje de entrada (límite)	6 a 20 V
Pines digitales E/S	54 (15 para PWM)
Pines de entrada analógica	16
Puertos serie	4
Corriente DC	40 mA
Memoria Flash	256 KB (8 KB para bootloader)
SRAM	8 KB

EEPROM	4 KB
Velocidad de reloj	16 MHz

4.2.3 Arduino Ethernet Shield

En primer lugar una Shield es un circuito impreso especial, puesto que puede colocarse sobre el Arduino, ampliando las capacidades que éste posee. Se pueden colocar varias Shields una sobre otra a manera de torre, aunque no se recomienda realizar dicha práctica, bastaría con utilizar una o máximo dos Shields para obtener el máximo beneficio del Arduino [23].

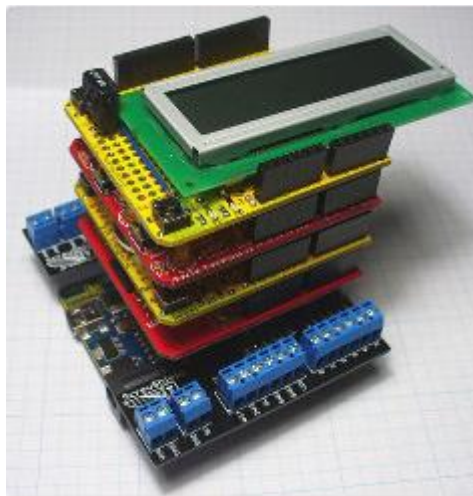


Figura 6. Caso extremo y no recomendable de Shields colocadas sobre la placa Arduino [23].

La placa Arduino Ethernet permite a una placa Arduino Mega conectarse a internet. Dicha Shield está basada en el chip Wiznet W5100, el cual proporciona un enlace de red IP que permite usar tanto el protocolo TCP como UDP.

La Shield Ethernet tiene un estándar de conexión RJ-45 lo que permite realizar una conexión física con la red Ethernet, además incorpora una ranura para poder insertar una tarjeta micro SD, la misma que puede ser utilizada para almacenar archivos para enviar a través de la red [24].

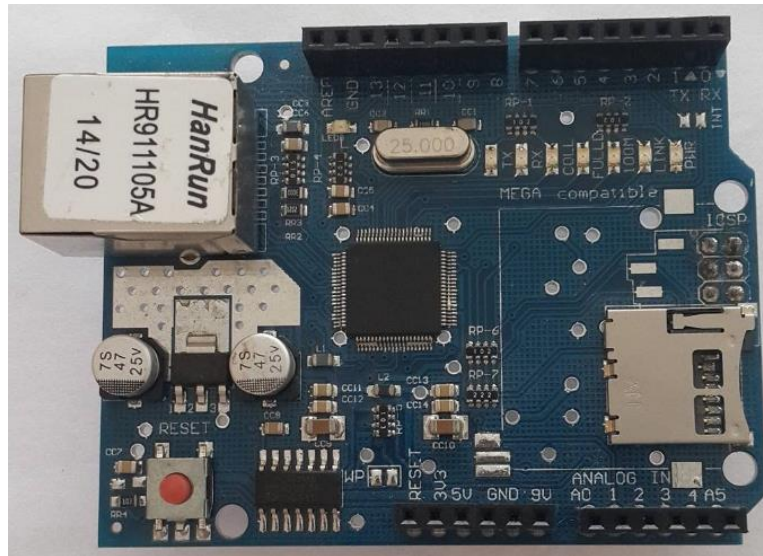


Figura 7. Arduino Ethernet Shield [El autor].

Para la utilización de esta Shield, se puede utilizar librerías propias de Arduino que permiten realizar la lectura de la tarjeta Ethernet y su respectiva programación.

La placa tiene los siguientes Leds de información:

- **PWR**: Indica que Arduino y la Shield están correctamente alimentadas.
- **LINK**: Indica la conexión a una red, cuando parpadea significa que la Shield está transmitiendo o recibiendo datos.
- **FULLD**: Se enciende cuando la conexión de la red es full dúplex.
- **100M**: Indica que la conexión de red es de 100 Mbps.
- **RX**: Este Led parpadeará cuando la Shield recibe datos.
- **TX**: Este Led parpadeará cuando la Shield transmite datos.
- **COLL**: Este Led se accionará cuando la red detecta colisiones [24].

4.3 CAPÍTULO III: Desarrollo de la aplicación en Android

4.3.1 ¿Qué es un Smartphone?



Figura 8. Smartphone [El autor].

Un *Smartphone* o celular inteligente es un dispositivo móvil que, además de permitir llamar y enviar mensajes, posee la capacidad de conectarse con internet, siendo una mezcla entre computadora y celular, por lo que también se le conoce como “pc de bolsillo”. Un *Smartphone* debe contar con un sistema operativo el cual le permita organizar la información personal, instalación de aplicaciones y el intercambio de información con otros equipos. Además, en ellos se puede tener redes sociales, correo, GPS, y cientos de aplicaciones de todo tipo [25], [26].

Los teléfonos inteligentes poseen la característica principal de tener pantallas táctiles así como la conectividad a Internet, pero lo más interesante es que permiten al usuario instalar programas adicionales y de terceros, lo que facilita incorporar nuevas funcionalidades a estos dispositivos [26].

4.3.2 Sistemas operativos para Smartphones

Cada *Smartphone* tiene un sistema operativo (SO) específico dependiendo de la marca que sea. Las marcas más conocidas en el mercado son: Apple, Motorola BlackBerry,

Nokia, Samsung, etc.; y entre los sistemas operativos más utilizados están: Android, iOS, BBOS, siendo Android el líder indiscutible [26]–[28]. Para saber qué sistema operativo se utiliza en los dispositivos inteligentes (*tablets*, *smarthphones*) tenemos la siguiente explicación: iOS es usado para los dispositivos de Apple; Android es utilizado por Samsung, LG y algunos otros; BlackBerryOS para BlackBerry, tal y como su nombre lo indica; y, Windows Phone para Nokia [27], [28].

Un sistema operativo es un *software* que permite la interacción entre el usuario y las aplicaciones, establece la base sobre la cual se escribirán los programas que maneja el smartphone, además controla y administra los recursos de *hardware* [29]. Por lo tanto, éste es el programa más importante de un teléfono inteligente, ya que sin el sistema operativo el celular Smartphone no poseería las condiciones excepcionales que ofrece [29], [30].

El sistema operativo realiza tareas elementales, por ejemplo: si un programa necesita enviar información a dispositivos periféricos, no es necesario comunicarse directamente con ellos, sino hacer llegar la petición al sistema operativo y éste se encargará de comunicar dicha petición a los dispositivos periféricos respectivos. Otras tareas simples pueden ser enviar información a la pantalla, no perder directorios en el disco, no perder archivos, entre otros [29], [30].

Dentro de las funciones que desempeña un sistema operativo se encuentran:

- Gestionar la lectura y escritura de archivos: Mediante la autorización o no al acceso de archivos y aplicaciones.
- Administrar el procesador y las autorizaciones: Encargándose de la seguridad de la ejecución de programas, es decir garantiza la confidencialidad de la información.
- Gestionar las entradas y salidas a través de drivers: Permitiendo unificar y controlar el acceso de los programas a los recursos del dispositivo.
- Gestionar la memoria de acceso aleatorio y ejecución de aplicaciones: Encargándose de gestionar el espacio de memoria asignada para las aplicaciones, y en caso de que la memoria física es insuficiente, el sistema operativo crea una memoria virtual garantizando que se ejecuten aplicaciones que consuman más recursos de lo que posee la memoria RAM [29], [30].

4.3.3 Android

Es un sistema operativo basado en Linux, cuyo origen está dado por la *Open Handset Alliance* formada por 48 empresas repartidas en todo el planeta. Android fue comprado en 2007 por Google, aunque fue algunos años después que se dio a conocer debido a las ventajas que poseía y a las facilidades que comenzó a prestar. Actualmente está posicionado como uno de los sistemas operativos más vendidos en el mundo, [31].

Posee una arquitectura fundamentada en cuatro niveles (ver Figura 9) con la característica de ser de código abierto, es decir permite a los usuarios realizar la modificación del sistema mediante la programación en Java. Presenta innumerables beneficios tal es el caso de las aplicaciones, ya que se pueden diseñar sin la necesidad de poseer programas avanzados. Además existe la tienda virtual *Google Play* de Android, la cual posee aplicaciones de todo tipo. Hay que recalcar que dicha tienda es segura, puesto que también existe la posibilidad de encontrar aplicaciones en otras páginas no oficiales, y éstas pueden contener virus o algún problema por el estilo [28].

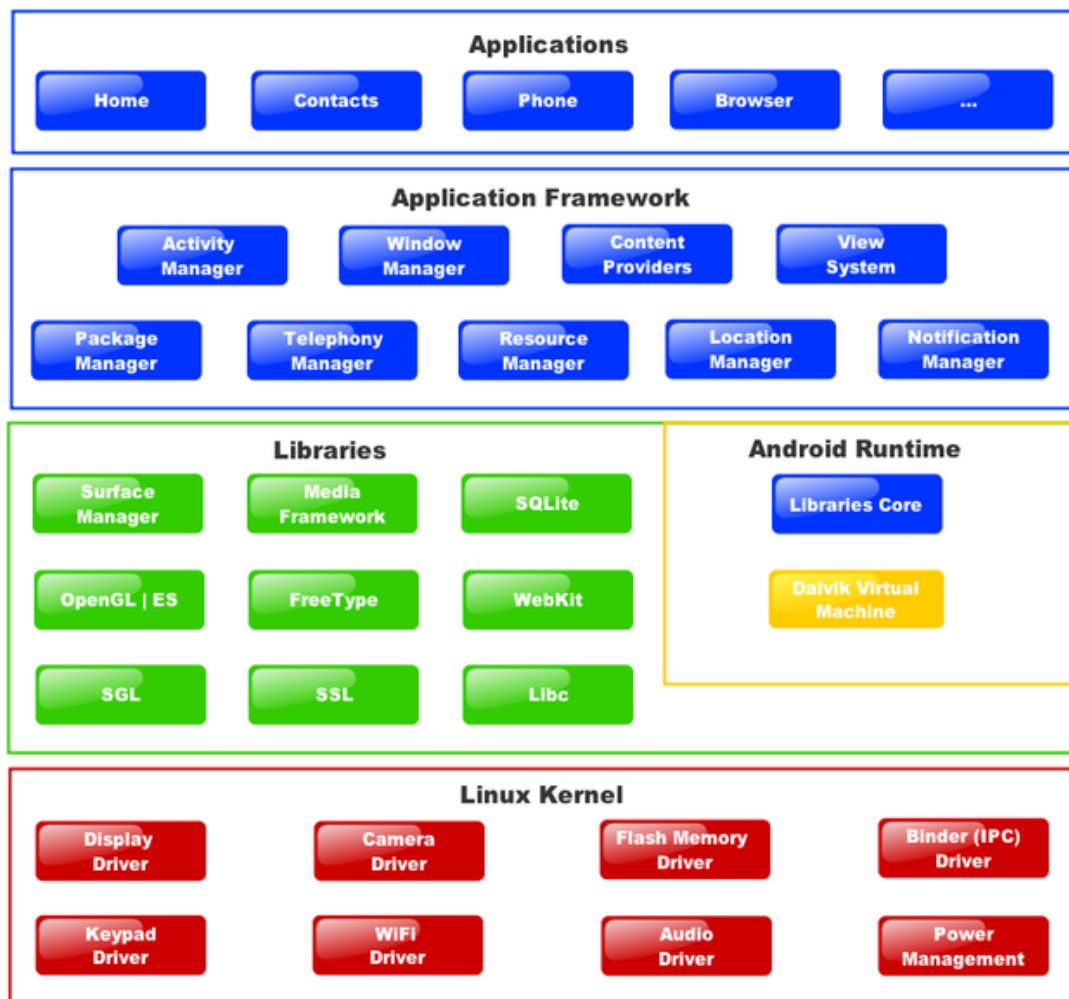


Figura 9. Arquitectura de Android [31].

Se aconseja usar solamente las aplicaciones que se encuentran ofertadas en *Google Play* por razones de seguridad, e incluso para aprovechar las funcionalidades que ofrece esta tienda ya que en ésta se puede encontrar libros muy útiles tanto para entretenimiento, ciencia ficción e investigación, músicas, películas, etc. motivo por el cual se considera a *Google Play* como la pionera en ventas en el mercado a nivel mundial [27], [28], [31].

4.3.4 Diseño de la aplicación en el MIT App Inventor de Android

Existen muchas maneras de hacer una aplicación para Android, y muchos programas para realizarlos, por ejemplo el AndroidCreator, que es un programa online y gratuito que permite diseñar aplicaciones para *tablets* y *Smartphones*, siempre y cuando éstos tenga como sistema operativo a Android [32].

Existe también otro programa online llamado GoodBarber, el cual tiene la desventaja de ser un programa pagado, pero ofrece un periodo de prueba de 30 días para diseñar gratuitamente aplicaciones para teléfono móviles y *tablets* [33].

Otro programa pionero en la programación para Android es el MIT App Inventor, el cual tiene la ventaja de ser un programa online fácil y gratuito, lo que facilita el trabajo de programación si se tiene pensado hacer una aplicación no muy complicada. Además maneja un lenguaje de programación basado en bloques, es decir el entorno de programación es gráfico e intuitivo, solamente se deben mover las “fichas” o bloques e ir colocándolas en un orden determinado de manera que la sintaxis sea la correcta. Por lo tanto no es necesario ser un experto programador para diseñar una aplicación en MIT App Inventor.

En el caso de elaborar aplicaciones desarrolladas con cualquiera de los programas mencionados, se las puede comercializar; para esto se lo puede hacer desde el *Android Market*, que permite distribuir o publicar aplicaciones hechas por programadores para cualquier necesidad. En un principio todas las aplicaciones eran gratuitas, pero en vista de que retrasaba su desarrollo y iPhone ganó dinero con aplicaciones pagadas, Android decidió ofrecer algunas aplicaciones (más avanzadas) a un cierto precio [34].

5.- MATERIALES Y MÉTODOS

5.1 Materiales

Durante el desarrollo de la presente investigación se utilizaron los siguientes materiales:

Tabla 4. Lista de materiales utilizados.

MATERIALES	CANTIDAD
Arduino Mega	2 unidades
Shield Ethernet	2 unidades
Sensor DHT 11	1 unidad
Sensor PIR DYP-ME 003	1 unidad
Batería LiPo	1 unidad
Oscilador de cuarzo	1 unidad
Cable UTP CAT 5e	48 metros
Motor Síncrono	1 unidad
Interruptor de tope	2 unidades
Resistencias	27 unidades
Diodos	6 unidades
Transistores	12 unidades
Capacitores	8 unidades
Puente rectificador	1 unidad
Transformador	1 unidad
Reguladores	2 unidades
Relés	6 unidades
Borneras	13 unidades
Circuitos Impresos	2 unidades
Cautín	1 unidad
Estaño	2 metros
Canaletas 2m	13 unidades
Cable sólido 14	1 metro
Poleas	1 unidad
Cordel	3 metros
Persianas	2 unidades
Focos	2 unidades

Tabla 5. Lista de equipos utilizados.

EQUIPOS	CANTIDAD
Smartphone	1 unidad
Cámara de video-vigilancia IP	1 unidad
Router	1 unidad
Calefactor	1 unidad
Ventilador	1 unidad

5.2 Métodos

Los métodos utilizados fueron:

Método Hipotético – Deductivo, porque primero se indagó para entender el problema de investigación; después se planteó una hipótesis que fue evaluada; y al final se presentó el resultado esperado, el cual determinó que la hipótesis, y por ende el proyecto, se cumplió.

Método Sistémico, porque se partió de varios conocimientos, ya sean investigados o adquiridos durante el estudio de la carrera, para poder realizar el diseño de la red domótica.

Método Empírico: Lógica Experimental, porque a través del pensamiento lógico, se pudo diseñar el software apropiado para el Arduino; obviamente se fueron realizando varias pruebas o experimentos con tal de obtener la programación que generó los resultados deseados.

Procedimiento

En primer lugar se definió todo lo que se iba a utilizar, en qué lugar de la vivienda iban a estar los diferentes elementos del sistema domótico y las características del sistema. Posteriormente se definió de manera específica las series de los dispositivos a emplear.

Para el desarrollo del presente trabajo se consideró el control de las siguientes variables:

- Sistema de iluminación (encendido y apagado de las luces del hogar).
- Movimiento de persianas o cortinas.
- Encendido y apagado de aparatos eléctricos (para el control de temperatura y humedad del hogar).
- Gestión de la seguridad con el monitoreo remoto de una cámara de seguridad y vigilancia.

Luego de investigar e indagar se obtuvo la información acerca de los sistemas domóticos, por lo que se decidió que esta investigación estaría basada en los sistemas centralizados y manejaría una topología en estrella.

Al finalizar con la instalación del sistema domótico, éste debería quedar conectado como se puede apreciar en la siguiente figura. Dentro del sistema domótico se tienen 2 Arduinos Mega, cada uno de los cuales será el nodo central (topología estrella) y manejará los sensores o las etapas de potencia y control respectivamente. El “FOCO1” está representando a los focos que se piensa controlar, además del calefactor y ventilador, puesto que dichos elementos se conectarán exactamente de la misma manera al Arduino central, pero por cuestiones de estética no se los ubicó a todos en la figura. Ambos Arduinos, además de la Cámara IP, irán conectados al Router, y a través de él podrán enviar la información a la web, para poder visualizar los datos desde cualquier lugar con internet.

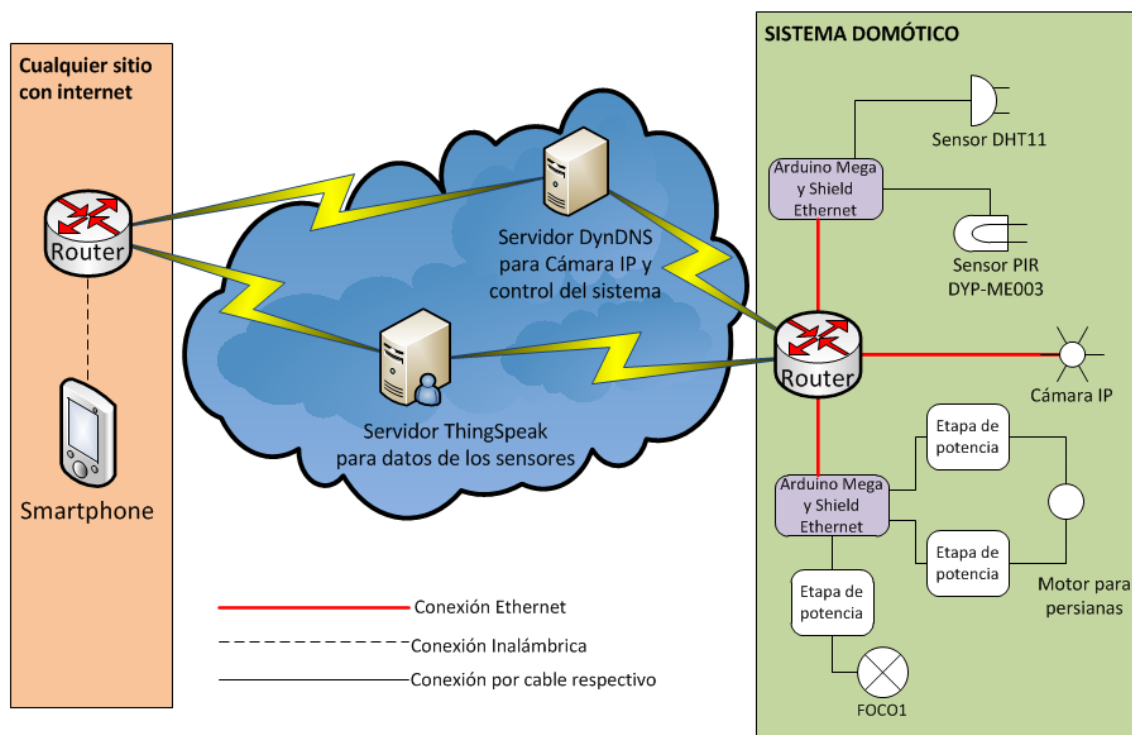


Figura 10. Topología de la red domótica [El autor].

El estándar de automatización utilizado fue el BatiBUS, debido a las características que ofrece dicho estándar, indicadas en el capítulo 1 (ver sección 4.1.5.4); además la norma de cableado estructurado que se siguió es la norma TIA/EIA 568B (ver capítulo 1, sección 4.1.5.4), debido a las especificaciones técnicas que ofrece: utiliza una topología en estrella, la distancia máxima para cableado horizontal es de 70 a 90 metros para pares de cobre (UTP), define el conector a utilizar (conector RJ45) y la asignación de pares de cables (4 pares de hilos de distinto color dentro del cable UTP), la categoría del cable UTP (5 o 6).

El desarrollo y elaboración del trabajo se dividió en 6 instancias fundamentales, las cuales son:

1. Diseño del sistema en AutoCAD.
2. Obtención de los datos de los sensores y exportación de ellos fuera de la red.
3. Exportar video de la cámara IP.
4. Control del sistema domótico desde el navegador.
5. Montaje de equipos y materiales.
6. Desarrollo de la aplicación para Android.

➤ **Primera instancia**

El diseño del sistema se lo realizó sobre el plano de la vivienda obtenido previamente, en el cual se dispuso el lugar donde irían situados los dispositivos y las canaletas. Cabe recalcar que desde el inicio se tuvo en cuenta que éste no sería el plano definitivo, puesto que cualquier contratiempo o mejora que surgiera luego en la implementación, debía ser tomada en cuenta para modificar nuevamente el plano (ver Anexo 4).

➤ **Segunda instancia**

El sensor DHT11 posee 4 pines, pero soldado a un pequeño circuito que contiene una resistencia de pull-up, otorga los 3 pines (5V, salida, GND) mostrados en la Figura 2. Se puede alimentar con una tensión de entre 3 a 5 voltios, su salida es una señal digital, entrega un rango de temperatura de 0°C a 50°C con una precisión de $\pm 2^\circ\text{C}$ y de 20% a 90% de humedad relativa (RH), con un error de 4% de RH [35].

El proceso de comunicación entre el Arduino y el sensor de temperatura y humedad es de aproximadamente 4ms. La transmisión completa es de 40bits de donde se obtiene la temperatura y humedad. Los 40 bits están distribuidos de la siguiente manera [35]:

$$\text{Alta hum (8b)} + \text{baja hum (8b)} + \text{alta temp (8b)} + \text{baja temp (8b)} = \text{paridad (8b)}$$

Por ejemplo si se recibe la trama 0011 0101 0000 0000 0001 1000 0000 0000 0100 1101, calculando se tendría [35]:

$$0011\ 0101 + 0000\ 0000 + 0001\ 1000 + 0000\ 0000 = 0100\ 1101,$$

Por lo que la humedad relativa, de acuerdo a los bits 0011 0101 (35 en hexadecimal), sería de 53%, y la temperatura, de acuerdo a los bits 0001 1000 (18 en hexadecimal) sería de 24°C [35].

Pero, para evitarse problemas y complicarse con la programación, varios programadores expertos han desarrollado las librerías de Arduino propias para este sensor, de manera que el procedimiento para obtener los datos procesados de temperatura y humedad es mucho más sencillo [36].

En el caso del sensor de movimiento PIR DYP-ME003 resultó más fácil el proceso de obtención de datos, puesto que mencionado sensor entrega únicamente una 1 lógico (5V) en caso de detectar presencia dentro de un rango específico (ver Figura 11), caso contrario en su salida tendremos un 0 lógico (GND).

Las conexiones efectuadas entre el Arduino Mega y los sensores utilizados, se muestran en la Figura 12.

Luego se comenzó a buscar la manera de cómo enviar los datos obtenidos por los sensores de presencia y humedad-temperatura (ver capítulo 1, sección 4.1.1.3) a través de la web, encontrando así la plataforma ThingSpeak para el sensor de temperatura y humedad, la cual puede obtener los datos proporcionados por dichos sensores y presentarlos de diversas maneras.

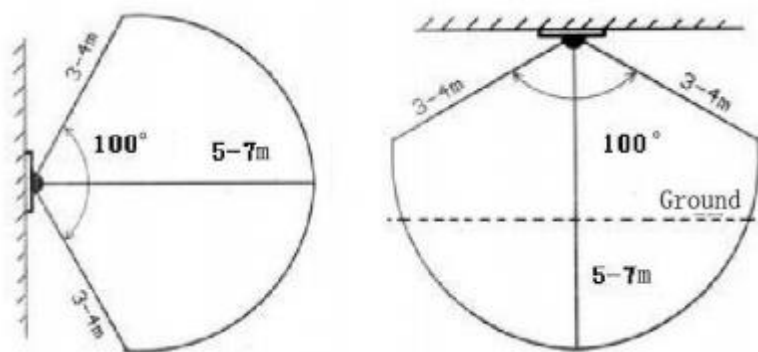


Figura 11. Rango de detección del sensor PIR DYP-ME003 [14].

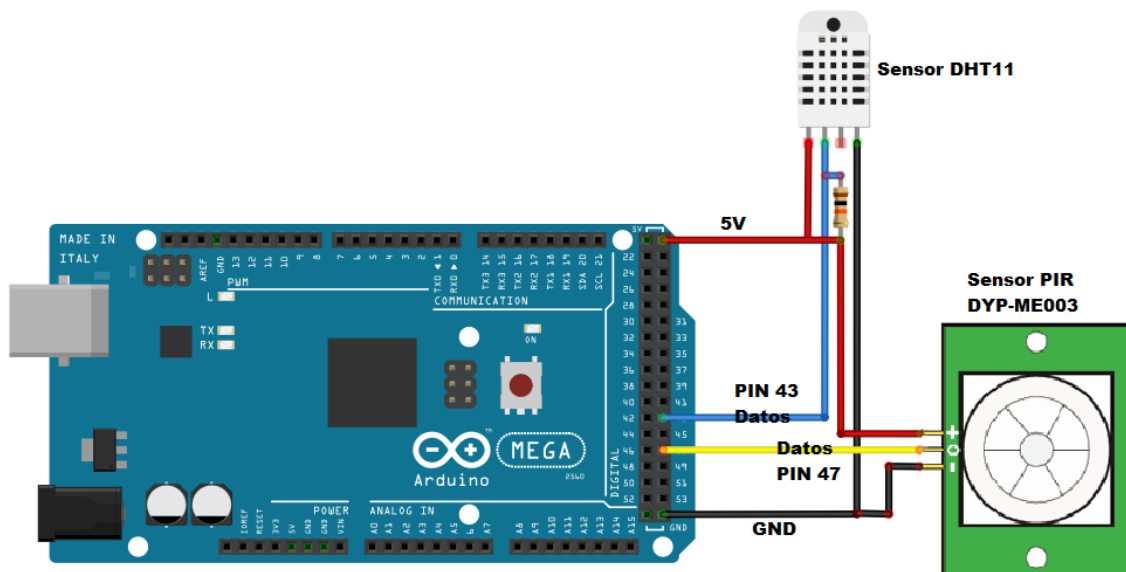


Figura 12. Conexiones entre el Arduino y los sensores DHT11 y DYP-ME003 [El autor].

ThingSpeak tiene la ventaja de ser una página gratuita, por lo que todas las personas pueden acceder a ella, de igual manera posee mucha información, variada y bien explicada, acerca de cómo utilizar cada herramienta que nos ofrece y cómo hacer para que interactúen entre ellas, además proporciona una manera sencilla de enlazarse con la plataforma Arduino. Esta página posee ciertas características tales como [37]:

- Crear canales y plugins.
- Analizar y visualizar gráficos de datos en MATLAB.
- Realizar acciones derivadas de ciertos cambios en sensores.
- Enviar alertas por Twitter, etc.

Para enviar los datos a la nube, se utilizaron los siguientes pasos:

1. Se programó el Arduino Mega (ver capítulo 2, sección 4.2.2) para recibir los datos del sensor DHT11.
2. Para poder usar la plataforma mencionada, se descargó la librería de ThingSpeak para Arduino.
3. Se creó un “canal” en ThingSpeak, (permite mostrar gráficas de los sensores, entre otras cosas), que posee una “api key” única.
4. Dicha “api key” es ingresada manualmente en la respectiva línea de comando en Arduino para poder publicar los datos en la plataforma.

Los datos se los puede observar como un solo valor obtenido en ese instante (ver Figura 13) o como una serie (predefinida) de mediciones obtenidas en un lapso de tiempo determinado (ver Figura 14).

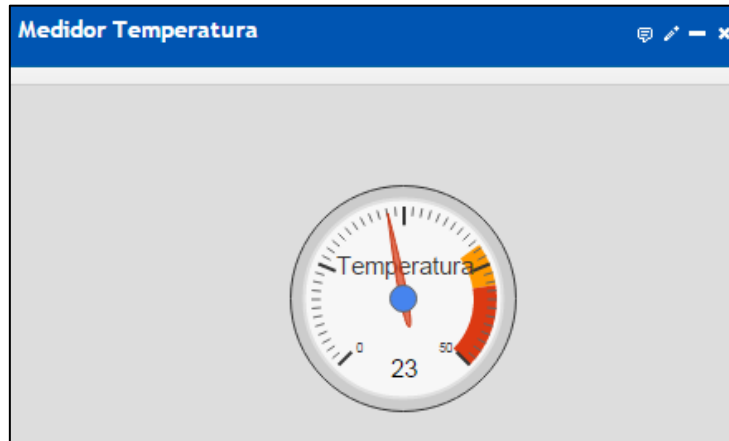


Figura 13. Valor de temperatura instantáneo [El autor].

Para el caso del sensor de presencia se modificó el algoritmo anterior, considerando el enlace entre Arduino y la página social Twitter. De igual manera, Twitter posee una especie de clave denominada “token” para poder publicar tweets a través del Arduino, por lo que al momento de ingresar un intruso a la casa y ser detectado mediante el sensor de movimiento, se enviará instantáneamente un tweet a una cuenta específica informando lo sucedido (ver Figura 15).

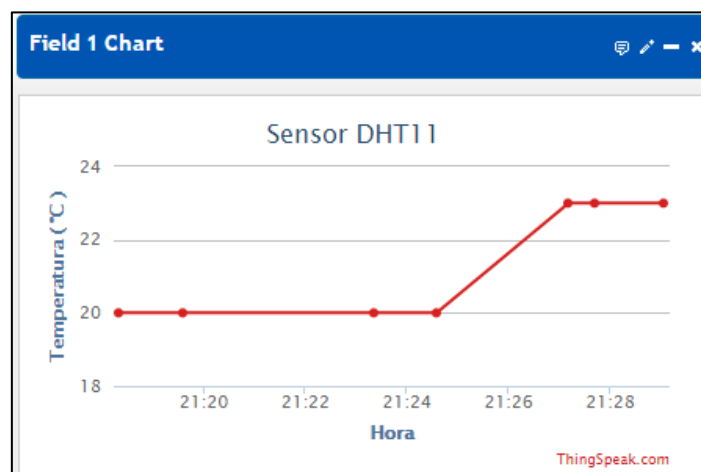


Figura 14. Últimas mediciones de temperatura [El autor].

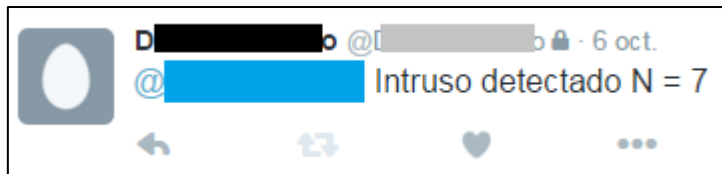


Figura 15. Tweet informando presencia de intruso [El autor].

La lógica seguida para realizar la programación del Arduino, respecto a los sensores, se muestra en la Figura 16.

➤ Tercera instancia

En primer lugar una cámara IP es una videocámara que puede enviar video y audio (en algunos casos) a través de un router o switch a una Red de Área Local (LAN), o a través de la web [38].

Las cámaras IP brindan algunos servicios, por ejemplo activar la aplicación para detectar presencia, tal y como lo haría el sensor DYP-ME003 (ver capítulo 1, sección 4.1.1.3), y en caso de que esto ocurra, se enviaría un correo o grabaría una secuencia de imágenes demostrando la presencia de un intruso. Además se pueden conectar sensores externos a la cámara, como de presencia, humo, fuego, gas, etc., y acoplar alarmas [38].

Para trabajar con la cámara de seguridad, primero se procedió a instalar la videocámara IP, dándole luego una IP fija (de preferencia) de acuerdo a los parámetros de nuestra red LAN. A continuación se puede observar el video y efectuar el movimiento de dicha cámara desde el navegador web a través de una computadora, pero sólo dentro de la misma LAN.

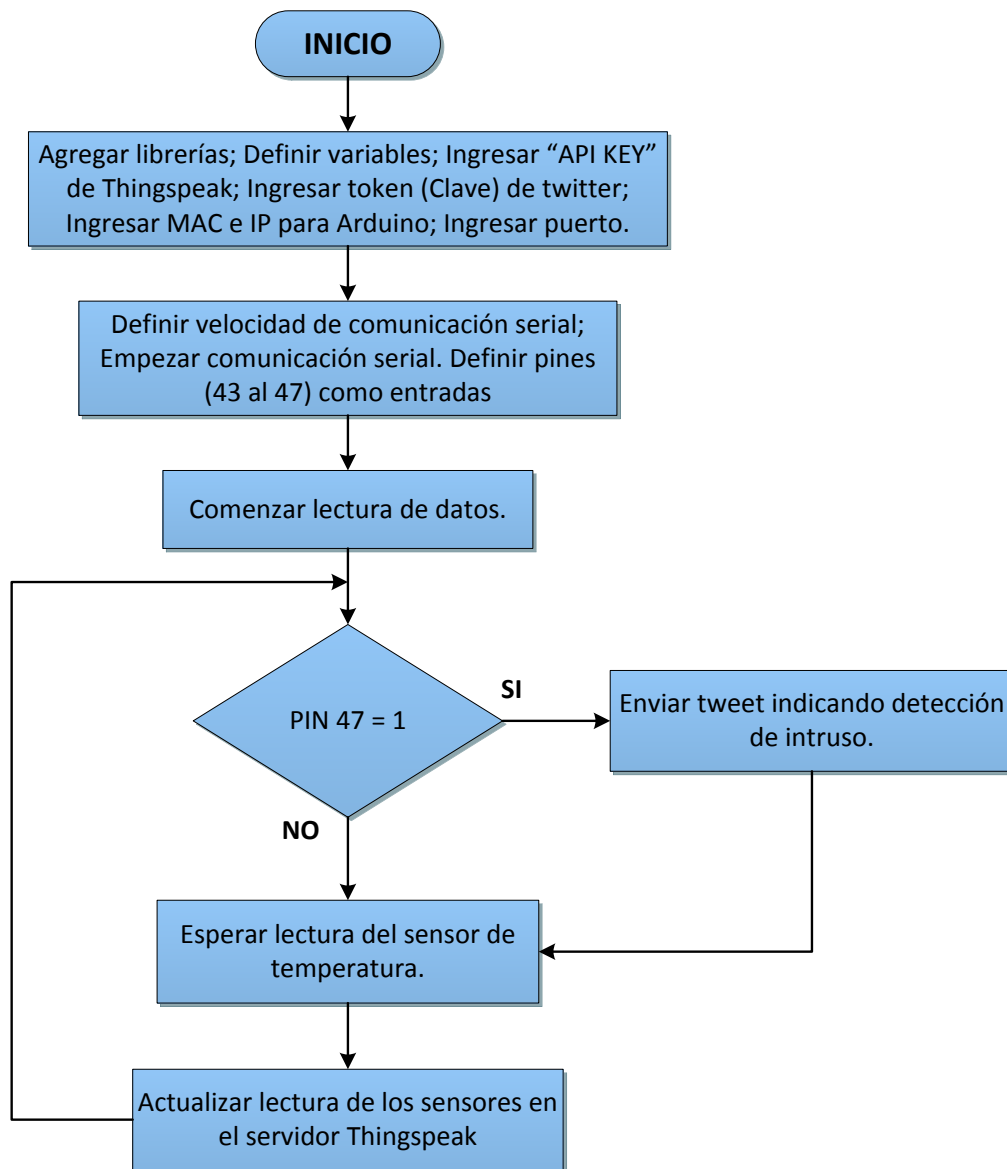


Figura 16. Diagrama de flujo para programación de los sensores [El autor].

Para enviar el video de una cámara IP al internet se tiene varios métodos: tener una IP pública, o tener un servicio DDNS (Dynamic DNS). Para el método del servicio DDNS, se puede acceder a algunos servidores, como lo son DynDNS, TZO, No-IP, entre otros, que tienen compatibilidad con algunos routers, dependiendo del servidor [39].

DynDNS es un servidor pagado y, como tal, es uno de los que mayores alternativas presta y más disponibilidad posee en los routers. TZO es también un servidor pagado, de la misma empresa que DynDNS, y con características similares a éste. No-IP es un servidor gratuito, presentado como alternativa a servidores pagados, pero no posee total disponibilidad en los routers [39].



Figura 17. Cámara IP [El autor].

Para poder enviar el video y ser observado fuera de la LAN, se cambió el puerto que viene por defecto en la cámara (80) a un puerto libre cualquiera, por ejemplo el 81. Después se configuró el router de manera que el nuevo puerto configurado esté activo (abierto) para la IP de la cámara. Estos pasos son necesarios debido a que se trabajó con el servidor DynDNS, el cual provee nombres de dominio, para poder acceder desde cualquier sitio a nuestra cámara IP.

Una vez obtenida la cuenta pagada en este servidor, se procedió a crear un nombre de dominio propio, por ejemplo “pruebatesis.dyndns.org”, el mismo que va a redirigir o asociar ese nombre con una IP pública específica, en este caso la IP pública asociada a la LAN del sistema domótico. Luego de realizar dichos pasos, ya se podrá acceder a la cámara IP desde cualquier parte de la red, ingresando a la dirección “pruebatesis.dyndns.org:81”.

➤ **Cuarta instancia**

Para realizar el programa para controlar los diferentes dispositivos del sistema con una interfaz web, se usó como base ejemplos otorgados por otros programadores. Se le dio una MAC e IP al Arduino, y se cambió el puerto por defecto (80) a otro puerto libre, 82 para el caso del control.

Luego se creó algunos botones en código HTML (lenguaje propio de páginas web) puesto que lo que se pretende es acceder desde el navegador a los botones, y en principio poder controlar todo desde la web.

A continuación se estableció la lógica de control (ver Figura 18) y se la plasmó en la programación del Sketch de Arduino, teniendo en cuenta todos los dispositivos a usarse.

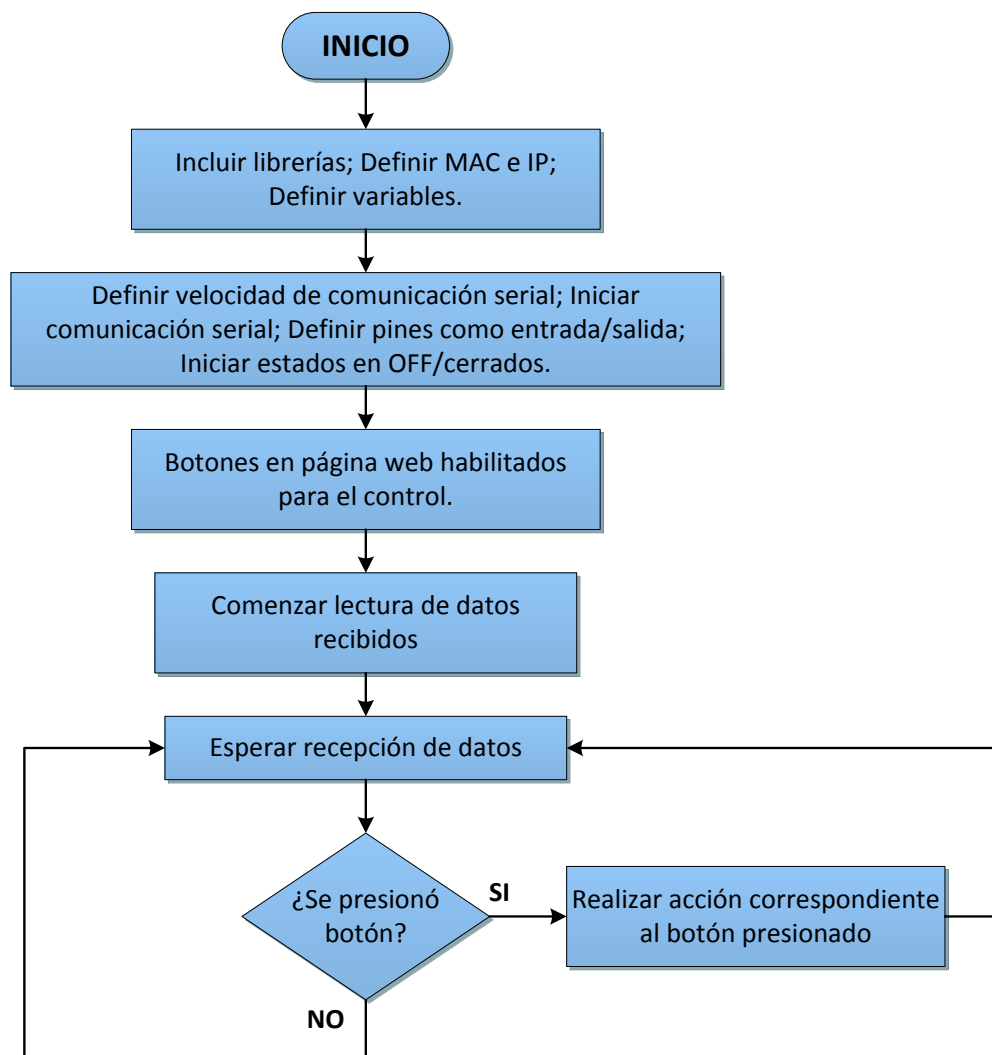


Figura 18. Diagrama de flujo para la programación respecto al control del sistema [El autor].

Para poder controlar el sistema domótico desde cualquier parte de la web, se realizaron los mismos pasos que se siguieron con la cámara IP, es decir, como la cuenta en DynDNS ya estaba configurada para la cámara, únicamente se configuró el router para que la IP del Arduino trabaje en el puerto 82.

A partir de ese momento, ya fue posible controlar el sistema desde fuera de la red, únicamente había que entrar con el nombre de dominio y el puerto, por ejemplo “pruebatesis.dyndns.org:82”.

➤ Quinta instancia

Luego de poder visualizar la cámara y acceder al control del sistema desde fuera de la red LAN, se montó toda la red en la vivienda, siguiendo el modelo del plano realizado previamente en AutoCAD.

Lo que se ubicó primero fue todo lo que tenía que ver con la cámara IP, es decir las canaletas que van desde la cámara hacia el router y el punto de corriente para alimentar la dicha cámara.

Después se hicieron todos los arreglos en cuanto a las cortinas (ubicar motor, cordeles, etc.), ventilador y calefactor; a continuación, se efectuaron las conexiones que iban a permitir la conmutación de ambos focos a controlar, permitiendo de esta manera encenderlos manual y remotamente. Durante la ubicación del circuito que permite la conmutación de los focos, también se ubicó el sensor de presencia en un punto cercano y directo a la puerta de acceso a la vivienda.

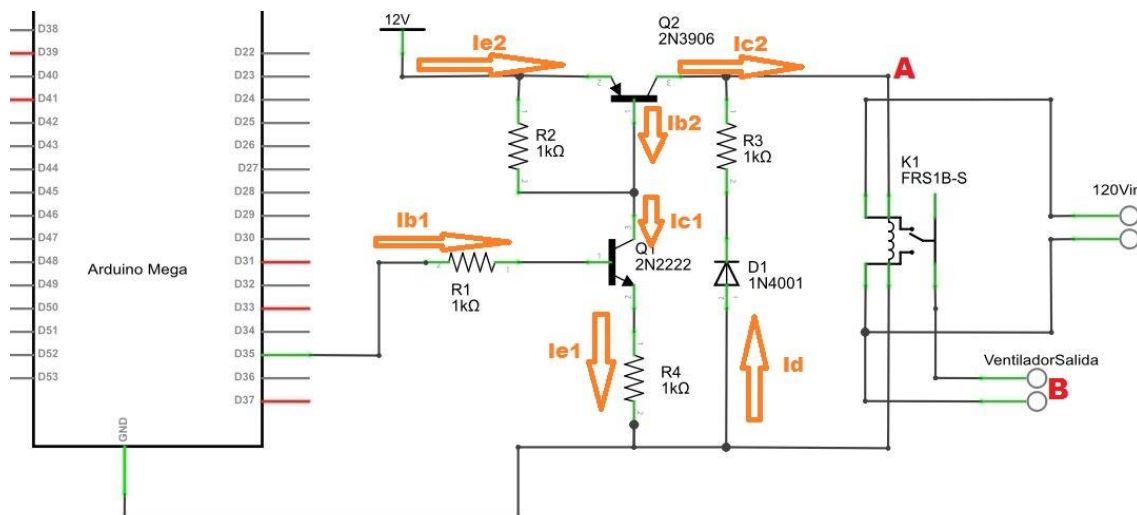


Figura 19. Diagrama circuital usado para controlar el ventilador [El autor].

En la Figura 19 se puede apreciar parte del diagrama circuital utilizado para controlar cada uno de los elementos del sistema domótico. Consta básicamente de un transistor NPN y un transistor PNP, que trabajan en la región de corte y saturación. Trabajarán en corte cuando la salida de los pines del Arduino esté en 0 lógico ($I_{b1} = 0$) haciendo que

la salida del transistor PNP sea de 0 voltios. Esto se puede constatar de la siguiente manera:

$$\begin{aligned}
 I_{b1} &= 0 \\
 \therefore I_{e1} = 0 &\quad \rightarrow \quad I_{b2} = I_{c1} = 0 \quad \rightarrow \quad I_{c2} = 0 \\
 \therefore V_{A-GND} &= 0V \quad \rightarrow \quad V_B = 0V
 \end{aligned}$$

Como se puede apreciar, ambos transistores, al no tener corriente de base, estarían en la región de corte, trabajando como circuitos abiertos, por lo que el voltaje entre el punto A y GND sería de 0V, impidiendo de esta manera que el relé se active, dejando en la salida B un voltaje de 0V.

Pero ambos transistores trabajarán en saturación cuando las salidas del Arduino estén en 1 lógico ($I_{b1} = 40mA$), permitiendo que en la salida del transistor PNP hayan 12 voltios, activando de esta manera el relé y ejecutando la acción seleccionada.

La corriente que entrega el Arduino por cada PIN es de 40mA, corriente suficiente para saturar el transistor NPN, haciendo que el voltaje entre el colector y emisor (de cada transistor) sea de 0V (valor que indica saturación del transistor), de manera que cada transistor trabajaría como circuito cerrado. Entonces tendríamos lo siguiente:

$$\begin{aligned}
 I_{b1} &= 40mA \quad \therefore \text{transistor1 en saturación} \\
 \therefore I_{b2} &= I_{c1} = I_{e1} = I_{c1}^{m\acute{a}x} \\
 \therefore V_{A-GND} &= 12V \\
 \therefore V_B &= 120V
 \end{aligned}$$

Debido a la saturación de los transistores, el voltaje entre el punto A y GND sería de 12V, ideal para hacer que el relé conmute, permitiendo de esta manera que la salida en el punto B sea de 120V, requeridos para activar el ventilador.

La corriente I_d es simplemente una corriente que puede quedar libre luego de desactivar el relé, debido a la bobina que éste posee, y para contrarrestar se ubica el diodo y resistencias de libre circulación, impidiendo que dicha corriente dañe algún dispositivo electrónico.

En la siguiente figura se puede observar la manera en que van conectados los distintos elementos electrónicos al Arduino para realizar el debido control de los elementos del sistema.

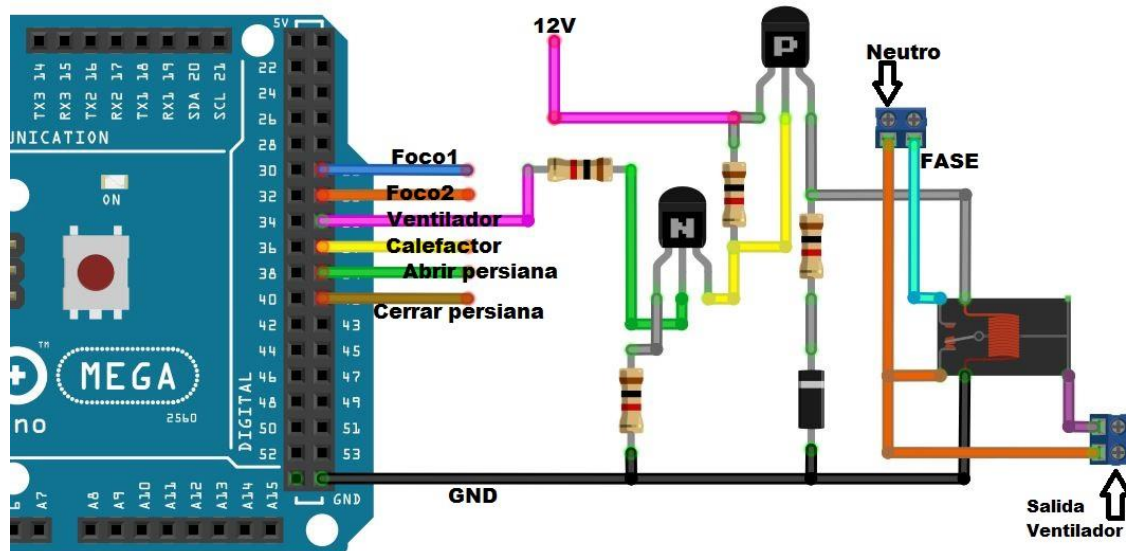


Figura 20. Diagrama de conexión de los elementos electrónicos para el control del ventilador [El autor].

Luego de tener los equipos en su lugar, y haber modificado algunos aspectos como ubicación del calefactor y ventilador, se colocaron las canaletas restantes y el cableado respectivo dirigido hacia una toma de corriente, el router o el Arduino.

Finalmente se ubicó el sensor DHT 11 cerca al calefactor y ventilador, de tal manera que le sea fácil captar la temperatura y humedad del ambiente, las cuales pueden ser modificadas por los artefactos antes mencionados. Además se ubicaron los Arduinos centrales, transformador, etc. en la caja de paso primaria, ubicada cerca a la cortina, cubriendo además el motor encargado de abrir/cerrar dichas cortinas.

➤ Sexta instancia

En última instancia se desarrolló la aplicación en Android y como se explicó en el capítulo 3 (ver sección 4.3.4), se puede programar online de una manera sencilla en el MIT App Inventor, para conseguir la aplicación que se desee y necesite.

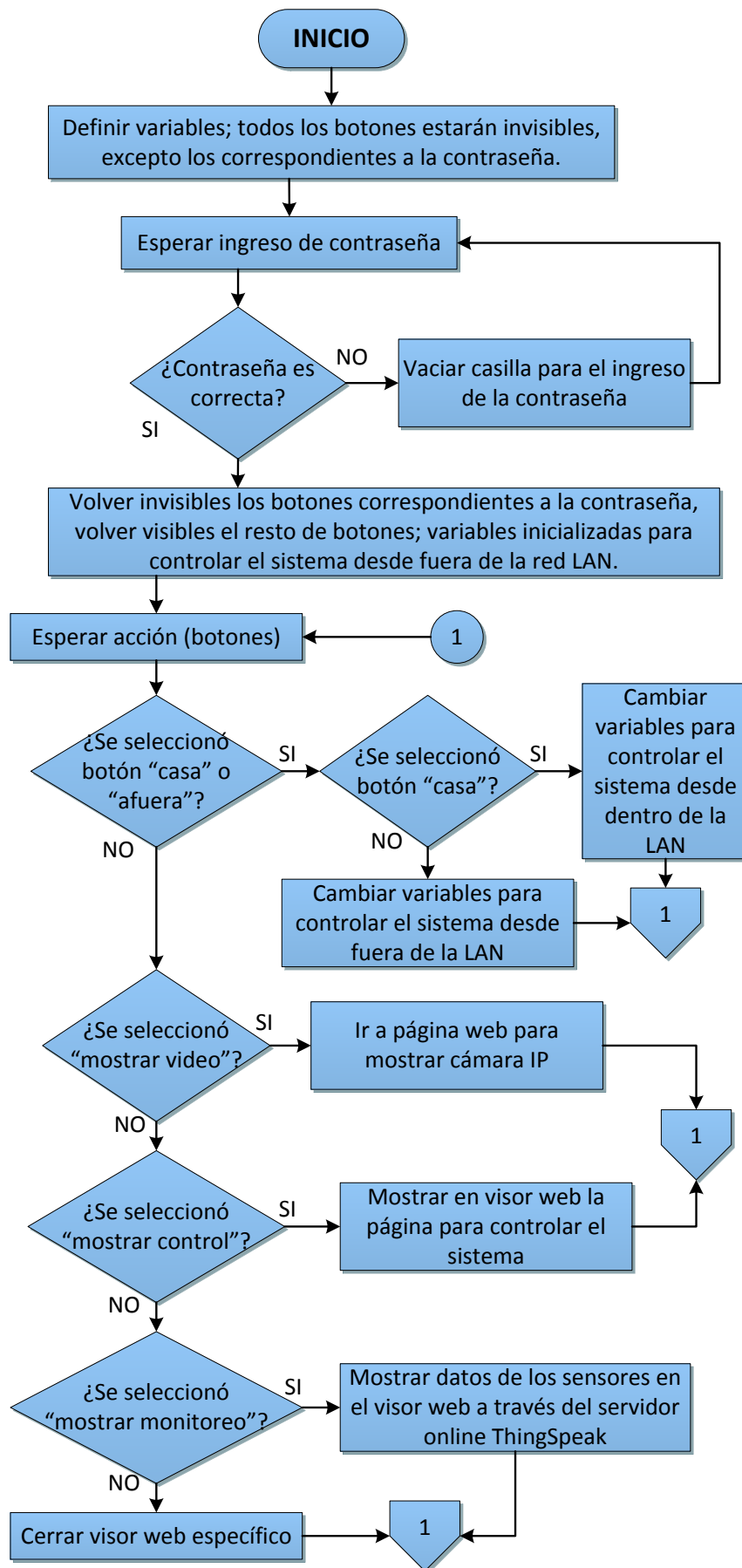


Figura 21. Diagrama de flujo para programar la aplicación Android [El autor].

Una vez que se tiene los datos recolectados en internet, proceder a realizar la aplicación (en App Inventor) es sencillo, ya que lo primero que se hace es adecuar cómo van a ir los botones, cuadros de diálogo, etc. en la pantalla, es decir cómo quiere que se vea su aplicación, y luego se programa cada paso correctamente y con la sintaxis adecuada. La programación de dicha aplicación, al igual que la programación de los sensores y sistema de control, siguió su lógica respectiva, basándose en el diagrama de flujo que se muestra en la Figura 21.

Hay que destacar que este programador en línea posee un elemento que hace referencia a la conectividad (ver Figura 22), de donde se debe seleccionar el elemento “web” puesto que los datos están ahí.

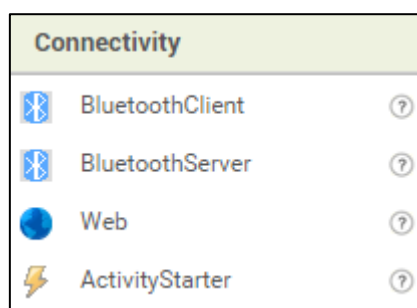


Figura 22. Conectividad en MIT App Inventor [El autor].

Una vez colocados todos los elementos visuales en el lugar deseado, y seleccionado “web” en el la parte de conectividad, se procede finalmente a programar e invocar la página web que se desea observar en el celular o Tablet, siempre y cuando tengan sistema operativo Android.

6.- RESULTADOS

6.1.- Sistema Implementado

El sistema domótico, al estar basado en la topología estrella, consta de un nodo central y los nodos periféricos. El nodo central consta de dos Arduino Mega, cada uno de ellos desempeñando una función específica: un Arduino recoge los datos transmitidos por los sensores y los envía a la plataforma ThingSpeak y a Twitter respectivamente, y el otro Arduino crea la página web para controlar los elementos seleccionados del hogar.



Figura 23. Sistema de Arduinos Mega [El autor].

Como se puede observar en la figura anterior, dichos Arduinos están ubicados en una caja blanca elevada (ver Anexo 6, Figura 39), la misma que contiene también a la fuente de alimentación, transformador, circuito controlador y motor.

El calefactor y ventilador se los dejó a cierta altura del suelo únicamente para poderlos observar a través del video transmitido en vivo por la cámara IP. Debido a esto, el sensor DHT 11 se lo colocó cerca a estos aparatos para captar más rápidamente los cambios en la temperatura y humedad del ambiente.



Figura 24. Ventilador, Calefactor y Sensor DHT11 [El autor].

El motor abre y cierra las cortinas con la ayuda de un cordel a una velocidad normal, ya que se evitó, en lo posible, que las abriera de una manera estrepitosa y violenta, o a una velocidad demasiado lenta. La programación está fijada para que las cortinas se abran hasta cierto punto y no se puedan abrir más, ya sea que se vuelva a presionar el botón de abrir. Lo mismo sucede al momento de cerrar las cortinas.

Para el caso del control de dos focos, se construyó un circuito pequeño que, realizando las respectivas conexiones con los cables eléctricos de dichos focos, permite la conmutación ON/OFF de los mismos, es decir, se puede encender un foco a través de la aplicación Android y puedo apagarlo manualmente a través de su respectivo interruptor, y viceversa. Dicho circuito está ubicado dentro de una caja blanca secundaria cercana a la puerta de ingreso a la vivienda, y aprovechando dicha cercanía se implantó en la caja el sensor de presencia (ver Figura 25), y un interruptor pequeño para encender o apagar mencionado sensor (ver Figura 26) según la situación lo amerite.



Figura 25. Sensor de presencia DYP-ME003 [El autor].



Figura 26. Interruptor para activar/desactivar el sensor de presencia [El autor].

6.2.- Comunicación

La comunicación que se da entre el envío y recepción de datos es de la siguiente forma:

- Respecto al monitoreo de los sensores: se envía la información directo a la plataforma online ThingSpeak, que permite visualizar los datos desde cualquier parte a través de un link específico. Para recibir los datos en la aplicación sólo se accede al link, donde ya se ha ordenado previamente los gráficos a mostrarse.
- Respecto al control: el Arduino crea una página web con botones que cumplen diversas funciones, y al dar clic o seleccionar cualquier botón en dicha página se envía una letra que, al ser interpretada correctamente por el Arduino Mega, ejecutará la acción correspondiente con dicha letra. La aplicación únicamente accederá a la página web de control a través de un link especial.

6.3.- Aplicación Android

La aplicación Android para Smartphone consta básicamente de dos partes: ingreso y control. Al ingresar en la aplicación, tendremos que ingresar la contraseña correcta si queremos monitorear el sistema y controlarlo, si la contraseña no es válida se reproducirá un mensaje diciendo “clave incorrecta”, y si es válida la contraseña ingresada entonces el mensaje será “clave correcta”.

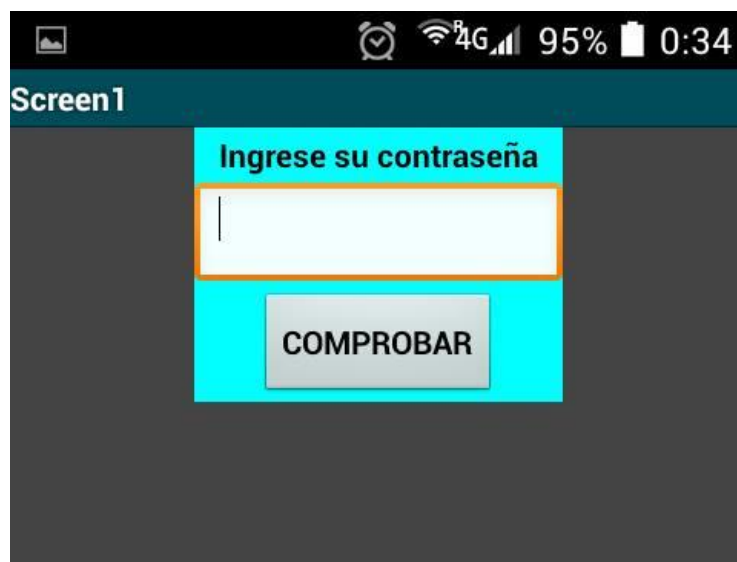


Figura 27. Casilla para ingresar contraseña [El autor].

Una vez ingresada la contraseña correcta, se habilitarán los botones que nos permitirán controlar y monitorear el sistema. Hay dos botones principales: “casa” y “afuera”. El

programa se inicia por defecto como si se estuviera fuera de la red LAN para realizar el control, mas si uno se encuentra dentro de la LAN, debe seleccionar “casa” para poder controlar el sistema desde dentro de la red, caso contrario solo podrá visualizar los datos entregados por los sensores, mas no podrá acceder ni al control del sistema ni al control de la cámara IP.

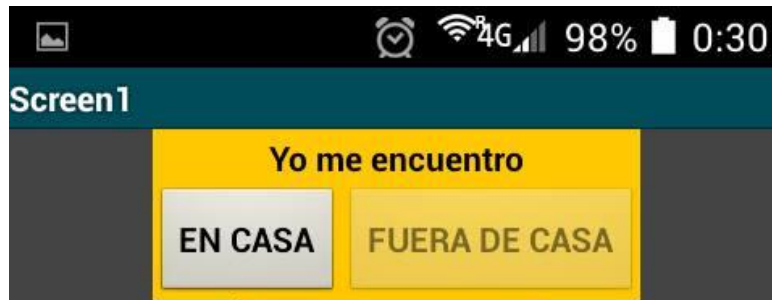


Figura 28. Botones para controlar el sistema desde dentro o fuera de la casa [El autor].

El resto de botones está distribuido de manera sencilla, de modo que el usuario pueda chequear la actividad de los sensores, pueda acceder remotamente a la cámara IP, y pueda desplegar u ocultar una ventana para controlar el sistema, dependiendo de lo que se desee realizar.



Figura 29. Botones para acceder al control, monitoreo o video vigilancia del sistema [El autor].

7.- DISCUSIÓN

Se considera que la arquitectura centralizada ya no se utiliza mucho debido a los altos costos que supone el cableado y la poca flexibilidad que ofrece, mientras que la arquitectura distribuida tiene el problema de que todos los dispositivos deben estar conectados individualmente a un equipo de inteligencia propio, lo que aumenta el tamaño y precio del dispositivo, y la complejidad e instalación del sistema. Pese a las características mostradas, se eligió utilizar un sistema centralizado, debido que eran pocos y cercanos los elementos que se iban a controlar y monitorear, de modo que utilizar dicha arquitectura resultaba mejor, más sencillo y barato que utilizar una arquitectura descentralizada.

En inicios se tenía pensado ubicar el sensor de presencia junto a la cámara, pero luego se decidió cambiar el lugar debido a varios factores, entre ellos la distancia desde el sensor hasta el Arduino Mega, para evitar la atenuación y por tanto la pérdida de la fidelidad de los datos entregados por el sensor de movimiento. Se lo ubicó, junto con su respectivo interruptor, en un punto cercano a la puerta de ingreso a la vivienda para aprovechar la caja blanca secundaria (ver Figura 25) y no tener que dañar la fachada del hogar agregando más cajas en las instalaciones. Además, dicho arreglo brinda comodidad al usuario del sistema domótico que se encuentra en casa, puesto que tiene la posibilidad de desactivar únicamente el sensor de movimiento para impedir el envío masivo de “alerta de intruso”, evitando de esta manera desactivar todo el sistema domótico.

Se trabajó con sensores alámbricos debido a que la casa donde se implementó el sistema es pequeña y no hubo inconvenientes ya que se ubicó la caja principal cerca a los sensores respecto a la máxima distancia permitida por cada sensor para la entrega fiable de sus datos. Además, y pese a algunas ventajas que ofrecen los sensores inalámbricos como que se necesita de menos canalización, dichos sensores poseen el inconveniente de ser más grandes de lo habitual, puesto que tendrían un tamaño similar al de un router, ya que deben tener dentro de ellos una fuente de alimentación, de igual manera deben estar colocados cerca a enchufes o puntos eléctricos para que puedan funcionar.

Un Arduino Mega posee 54 pines digitales y 16 pines analógicos, por lo que teóricamente podría albergar a una cantidad inmensa de sensores, pero en la práctica no

se tiende a saturarlo con demasiados sensores. Además, en caso de que se tenga una batería alimentando al Arduino, y muchos sensores conectados a él, la batería nos va a durar muy poco tiempo y va a ser ineficaz en los proyectos que se tenga pensado realizar. Si el diseño del sistema crece en cuanto a sensores se refiere, y si se dispone de una batería, se puede optar por monitorear sensores inalámbricos, puesto que van a tener su propia alimentación, y por lo tanto, el Arduino no tendrá que alimentar a tal cantidad de sensores, evitando también que se ubique una excesiva cantidad de canaletas para cubrir el cableado de dichos sensores.

El nodo central del sistema domótico, que en principio iba a utilizar solo un Arduino Mega, se tuvo que agrandar ubicando otro Arduino, de tal manera que uno iba a ser el encargado del monitoreo y publicación de los datos de los sensores, y el otro estaba encargado del control del sistema. Se hizo esta modificación debido a que, trabajando con todo el algoritmo en una solo sketch, surgieron problemas con las librerías de Twitter, ThingSpeak y Ethernet (servidor).

Existen diversos tipos de cámaras IP, pero para salir al internet de una manera más sencilla que utilizando un proveedor DDNS, se puede adquirir cámaras que cuentan con servidor privado, el cual permite observar el video desde cualquier parte de la web y sin la necesidad de adquirir programas o servicios adicionales.

Es necesario explorar el servicio que brindan los diferentes ISP (Proveedores del Servicio de Internet), puesto que se trabajó con 2 proveedores distintos, y se pudo comprobar que mientras uno de los ISP permitía abrir los puertos y controlar el sistema desde fuera de la LAN con facilidad, el otro ISP tenía todo bloqueado, por lo que tocó contratar una IP pública para poder acceder a los servicios que ofrecía otro ISP, pudiendo luego de eso desbloquear los puertos y lograr controlar el sistema desde fuera de la red.

Un problema bastante interesante que surgió con el sensor de movimiento fue cuando se detectaba un intruso y se publicaba un tweet. La red social Twitter no permite que la plataforma Arduino publique el mismo estado dos veces seguidas, ya que lo detecta como spam y deniega la publicación de dicho tweet. Se intentó mejorar este inconveniente de algunas maneras, como cambiar el usuario que realizaba las publicaciones, pero no se tuvo éxito alguno. Lo que se hizo entonces para evitar esa

restricción fue agregar al mensaje “intruso detectado N°” y adicionalmente un número, que iba a ir creciendo en orden y de uno en uno, y de esta manera se pudo publicar tweets seguidos sin que Twitter los tome como spam.

La aplicación creada posee una interfaz sencilla para todas las personas, tiene una clave de ingreso por razones de seguridad, y luego opciones bastante claras que puede elegir el usuario. Cabe destacar que la aplicación puede instalarse en cualquier dispositivo Android, sea Tablet o smartphone, pero si se tiene un dispositivo con un sistema operativo diferente, entonces dicha aplicación no servirá y no habrá como instalarla por razones de compatibilidad de sistemas operativos.

Se escogió el MIT App Inventor por las razones expuestas anteriormente, y además por las ventajas claras que posee sobre otros programas, como por ejemplo el GoodBarber y el Mono para Android (para el cual se debe instalar el SDK para Android), puesto que estos programas son pagados (se dice la versión económica del Mono es extremadamente cara para un programador con salario promedio) al contrario del MIT, que es online y gratuito; también la programación en el App Inventor está basada en bloques, contrario a lo que ocurre con los otros programas, en donde se tienen líneas de código para ejecutar una sentencia.

8.- CONCLUSIONES

- Se pudo realizar el diseño de un sistema domótico para una vivienda, pudiéndose adaptar a los parámetros planteados como el control de luces, persianas, ventilador y calefactor.
- Se utilizó el servidor ThingSpeak debido a que es una herramienta sencilla de utilizar y gratuita, que puede enlazarse directamente con Arduino, otorgando así mayor velocidad en la recepción y publicación de los datos en la web, por lo que mantiene informado al usuario sobre el estado en que se encuentra la red domótica.
- Se logró implementar un sistema domótico funcional en una vivienda, permitiendo el monitoreo y control remoto a través de una aplicación Android para Smartphone, obteniendo una mayor seguridad y accesibilidad a las variables de una manera sencilla desde cualquier sitio con internet.
- La implementación de un sistema domótico en un hogar se torna un poco complicado debido a que el diseño se tiene que acoplar a las medidas de la vivienda y se corre el riesgo de dañar la fachada, cosa que no ocurriría en una casa en construcción, pero a pesar de los contratiempos, se pudo implementar el sistema sin causar mayores daños en la fachada de la casa.
- Se creó una aplicación Android para controlar y monitorear remotamente un sistema domótico, utilizando la conexión con internet. Dicha aplicación brinda una interfaz amigable al usuario, que no necesita conocer direcciones ni puertos para acceder al sistema, únicamente selecciona botones que le permitirán conocer el estado de los sensores y ejecutar acciones en la vivienda.

- Se utilizó el sistema operativo Android debido a que está presente en la mayor cantidad de dispositivos electrónicos en el mundo, ocupando en 2014 un 81.5% del mercado, seguido de iOS con un 14.8%, según la consultora Canalys, la firma de investigación de mercados Gartner, y Strategy Analytics; y es además el que mayor cantidad de información posee respecto a programación y aplicaciones.

9.- RECOMENDACIONES

- Para el diseño e implementación de un sistema domótico se recomienda detallar bien qué elementos son los que se pretende controlar y monitorear, averiguar las ventajas de los distintos sensores, y buscar soluciones acorde con la economía del hogar.
- Para el diseño del sistema domótico se recomienda tener el plano de la casa, puesto que es mejor saber dónde quedan las columnas, por dónde pasan las instalaciones eléctricas, etc. Además, al diseñar en el plano real y actual de una vivienda se pueden considerar diversos factores los cuales permitirán que la implementación del sistema sea más rápida y no surjan mayores contratiempos.
- Es recomendable bajar los breakers eléctricos al momento de realizar extensiones que irán conectadas directo a los tomacorriente y cuando se trabaje con los focos de la casa, puesto que si se los quiere controlar manual y remotamente se debe agregar un pequeño circuito que permitirá la conmutación entre ON/OFF y conectarlo directo a la fase de los focos.
- Al crear una aplicación Android se recomienda en primer lugar leer libros como por ejemplo “El Gran Libro de Android”, para familiarizarse con dicho sistema operativo y ver tutoriales para aprender el entorno de programación del MIT App Inventor. Luego, es aconsejable realizar un diseño sencillo, el cual servirá para realizar distintas pruebas en cuanto a programación se refiere.

- Es aconsejable, para volver la aplicación más entretenida y amigable hacia el usuario, agregar en última instancia colores, figuras, imágenes y sonidos, de manera que cualquier persona pueda controlar el sistema y entender la aplicación.

- El programador online MIT App Inventor da una opción para traducir el lenguaje a español, inglés, francés, italiano, entre otros, pero se recomienda dejar el idioma en inglés, puesto que al momento de realizar la programación en bloques pueden surgir confusiones, debido a que la mayoría de códigos para programar (por ejemplo en Arduino) están dados en inglés.

10.- BIBLIOGRAFÍA

- [1] M. Fernández, «Instalación eléctrica y domótica para una vivienda unifamiliar», 12-jun-2012. [En línea]. Disponible en: http://biblioteca.unirioja.es/tfe_e/R000001357.pdf. [Accedido: 23-ago-2015].
- [2] J. Muñoz, J. Fons, V. Pelechano, y O. Pastor, «Hacia el Modelado Conceptual de Sistemas Domóticos», ene-2003. [En línea]. Disponible en: http://www.researchgate.net/profile/Oscar_Pastor2/publication/221595286_Hacia_el_Modelado_Conceptual_de_Sistemas_Domotics/links/0deec5200b854d13e8000000.pdf. [Accedido: 23-ago-2015].
- [3] J. Huidobro y R. Millán, *Domótica: Edificios Inteligentes*, 1ra ed. España: Creaciones Copyright, S.L., 2009.
- [4] J. Kurose y K. Ross, *Redes de computadoras: un enfoque descendente.*, 5.^a ed. Madrid, España: Pearson Educacion, S.A., 2010.
- [5] Cisco, *CCNA Exploration 4.0: Aspectos Básicos de Interworking.* .
- [6] R. Pallás Areny, *Sensores y Acondicionadores de Señal*, 4ta Edición. México D.F: Alfaomega Grupo Editor, S.A., 2007.
- [7] M. Rodríguez, «Hardware para Domótica: Sensores y Actuadores». [En línea]. Disponible en: https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&sqi=2&ved=0CDwQFjAF&url=http%3A%2F%2Fdocencia.etsit.urjc.es%2Fmoodle%2Fmod%2Fresource%2Fview.php%3Fid%3D6016&ei=uyCoVJq1DoXegwSrjYKgAw&usg=AFQjCNErmqd8T3__ICDcyuEj1ZBZ8tpRFw&bvm=bv.8200133. [Accedido: 23-ago-2015].
- [8] C. Romero, F. Vázquez, y C. de Castro, *Domótica e Inmótica. Viviendas y Edificios Inteligentes.*, 2da Edición. México D.F: Alfaomega Grupo Editor, S.A. de C.V., 2007.
- [9] A. García y K. Navarro, «DHT11: Sensor de humedad/temperatura para Arduino», *Panama Hitek*, 14-feb-2013. [En línea]. Disponible en: <http://panamahitek.com/dht11-sensor-de-humedadtemperatura-para-arduino/>. [Accedido: 22-ago-2015].
- [10] «Sensor de humedad y temperatura DHT11», *Microcontroladores y Electrónica*. [En línea]. Disponible en: <http://mikroe.es/sensor-de-humedad-y-temperatura-dht11/>. [Accedido: 27-ago-2015].

- [11] Sunrom Technologies, «Sensor de Temperatura y Humedad - DHT11». [En línea]. Disponible en: <http://robocraft.ru/files/datasheet/DHT11.pdf>.
- [12] «DHT11 Humidity and Temperatura Sensor». [En línea]. Disponible en: <http://www.micropik.com/PDF/dht11.pdf>. [Accedido: 24-nov-2015].
- [13] «Sensor de presença com módulo PIR DYP-ME003», *Arduino e Cia*, 04-jun-2014. [En línea]. Disponible en: <http://www.arduinoecia.com.br/2014/06/sensor-presenca-modulo-pir-dyp-me003.html>. [Accedido: 25-oct-2015].
- [14] «Specification of DYP-ME003», *Elec Freaks*, 03-jun-2011. [En línea]. Disponible en: <http://elec Freaks.com/store/download/datasheet/sensor/DYP-ME003/Specification.pdf>. [Accedido: 27-ago-2015].
- [15] «Guía para alicar la Norma TIA/EIA 568 para cableado estructurado», *Gobierno del Estado de Tabasco: Dirección General de las Tecnologías de la Información y Comunicaciones (DGTIC)*. .
- [16] J. Joskowicz, *Cableado Estructurado*. Montevideo, Uruguay, 2013.
- [17] «Residential Premise Cabling». [En línea]. Disponible en: https://www.bicsi.org/pdf/conferences/india/09_Presentations/Digi-Link.pdf.
- [18] A. Miranda, «Diseño e implementación de un sistema de comunicación inalámbrico dedicado a la domótica, controlado a través de una tablet, utilizando tarjetas de comunicación inalámbrica y la programación de los mismos.», dic-2014. [En línea]. Disponible en: <http://repositorio.usfq.edu.ec/bitstream/23000/3881/1/112333.pdf>. [Accedido: 25-oct-2015].
- [19] C. Tapia y H. Manzano, «Evaluación de la plataforma arduino e implementación de un sistema de control de posición horizontal», oct-2013. [En línea]. Disponible en: <http://dspace.ups.edu.ec/bitstream/123456789/5522/1/UPS-GT000511.pdf>. [Accedido: 25-oct-2015].
- [20] J. Pomares, «Manual de Arduino», 2009. [En línea]. Disponible en: <http://rua.ua.es/dspace/bitstream/10045/11833/1/arduino.pdf>. [Accedido: 09-sep-2015].
- [21] «Plataforma Arduino (Hardware)». [En línea]. Disponible en: http://platea.pntic.mec.es/~mhidalgo/documentos/02_PlataformaArduino.pdf. [Accedido: 27-oct-2015].

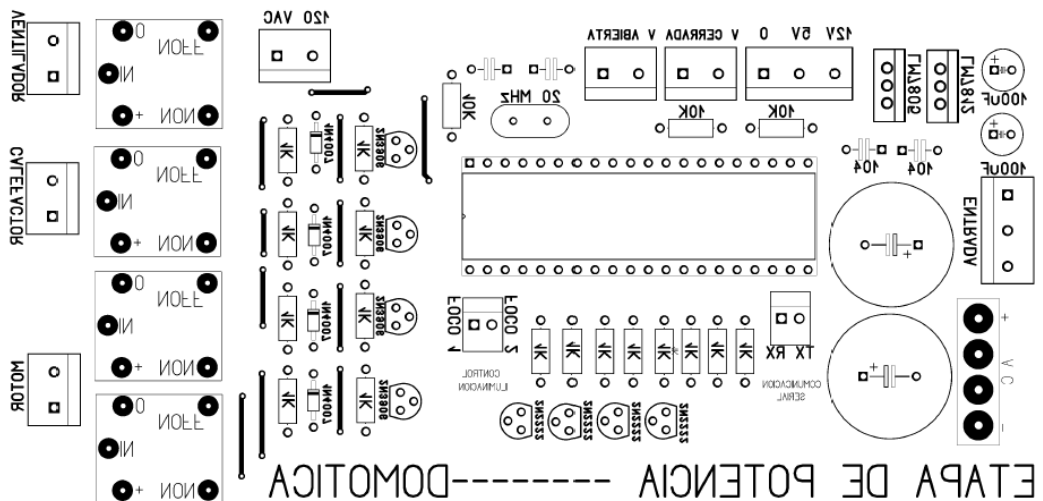
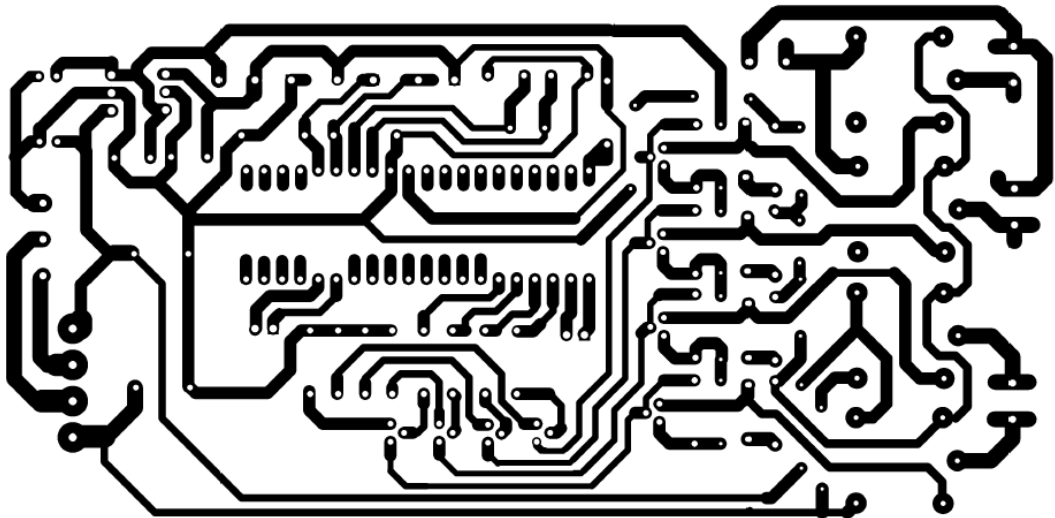
- [22] L. Gómez, «Control de pantallas uOED por puerto serie usando un microcontrolador Arduino Atmega2560 para representar animaciones.», jul-2012. [En línea]. Disponible en: http://e-archivo.uc3m.es/bitstream/handle/10016/17032/memoriaPFC_Lorena_Gomez_Fernandez%20final.pdf?sequence=1. [Accedido: 25-oct-2015].
- [23] Cesar, «¿Que son los Arduino Shields?», *Ayuda Electronica / Circuitos Electronicos, Ingenieria Electronica*, 07-dic-2010. [En línea]. Disponible en: <http://ayudaelectronica.com/%c2%bfque-son-los-arduino-shields/>. [Accedido: 09-sep-2015].
- [24] J. Ruiz, «Arduino + Ethernet Shield», *SlideShare*, 12-ene-2013. [En línea]. Disponible en: <http://es.slideshare.net/maxllontop123/arduino-ethernet-shield-1>. [Accedido: 01-nov-2015].
- [25] J. Romero, «Smartphones: The Pocketable PC», *IEEE Spectrum*, 30-dic-2010. [En línea]. Disponible en: <http://spectrum.ieee.org/telecom/wireless/smartphones-the-pocketable-pc>. [Accedido: 10-ene-2016].
- [26] F. Giraldo, «¿Qué es un smartphone? ¿Para qué sirve? ¿Cómo comprar uno?», *Informaticahoy*, 31-mar-2014. [En línea]. Disponible en: <http://www.informaticahoy.com.ar/telefonos-celulares/Que-es-un-smartphone.php>. [Accedido: 23-ago-2015].
- [27] C. Vialfa, «Análisis de los sistemas operativos para smartphones», *CCM*, jul-2015. [En línea]. Disponible en: <http://es.ccm.net/faq/8470-analisis-de-los-sistemas-operativos-para-smartphones>. [Accedido: 29-ago-2015].
- [22] «Sistemas operativos para Smartphones ¿Cuál nos conviene?», *El Baúl del Consumidor*, 16-ene-2013. Disponible en: <http://elbauldelconsumidor.blogspot.com/2013/01/sistemas-operativos-para-smartphone.html>. [Accedido: 28-ago-2015].
- [29] W. Stallings, *Sistemas Operativos: Aspectos internos y principios de diseño.*, 5ta ed. Madrid, España: Pearson Educacion, S.A., 2005.
- [30] «Sistema operativo», *CCM*. [En línea]. Disponible en: <http://es.ccm.net/contents/651-sistema-operativo>. [Accedido: 29-ago-2015].
- [31] A. Nieto, «¿Qué es Android?», *Xataka Android*, 08-feb-2011. [En línea]. Disponible en: <http://www.xatakandroid.com/sistema-operativo/que-es-android>. [Accedido: 28-ago-2015].

- [32] «Android Creator | Crear aplicaciones Android». [En línea]. Disponible en: <http://www.androidcreator.com/es/>. [Accedido: 25-oct-2015].
- [33] «GoodBarber - Crea tu aplicación para iPhone y Android», *GoodBarber - Crea tu aplicación para iPhone y Android*. [En línea]. Disponible en: <http://es.goodbarber.com>. [Accedido: 25-oct-2015].
- [34] P. Blanco, J. Camarero, A. Fumero, A. Warterski, y P. Rodríguez, «Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y el iPhone.», 2009. [En línea]. Disponible en: http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf. [Accedido: 07-sep-2015].
- [35] «Sensor de temperatura y humedad DHT11 - DHT22», *Omniblug*, 16-jul-2014. [En línea]. Disponible en: <http://www.omniblug.com/index.php/blog/115-sensor-dht11>. [Accedido: 04-dic-2015].
- [36] A. García y K. Navarro, «DHT11: Sensor de humedad/temperatura para Arduino», *Panama Hitek*, 14-feb-2013. [En línea]. Disponible en: <http://panamahitek.com/dht11-sensor-de-humedadtemperatura-para-arduino/>. [Accedido: 04-dic-2015].
- [37] «ThingSpeak». [En línea]. Disponible en: <https://thingspeak.com/>.
- [38] Valeriano, «Qué es y cómo Funciona una Cámara IP», *www.Dudaelectronica.com*. [En línea]. Disponible en: <http://valetron.eresmas.net/CamarasIP.htm>. [Accedido: 01-nov-2015].
- [39] «Arduino Ethernet Shield - Controla Tu Casa Por Internet», *EducaChip*, 16-nov-2014. [En línea]. Disponible en: <http://www.educachip.com/arduino-ethernet-shield/>. [Accedido: 01-nov-2015].

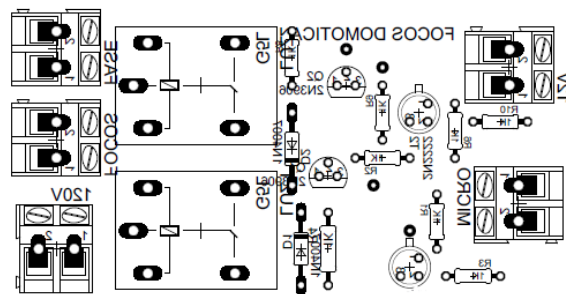
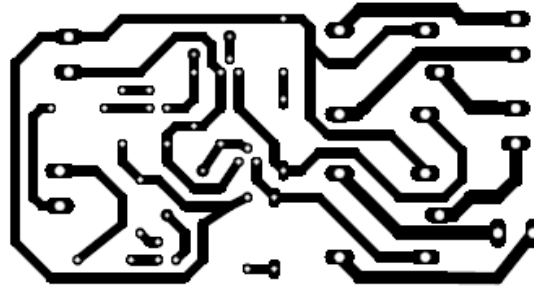
11.- ANEXOS

ANEXO 1. CIRCUITOS IMPRESOS Y DIAGRAMAS CIRCUITALES IMPLEMENTADOS

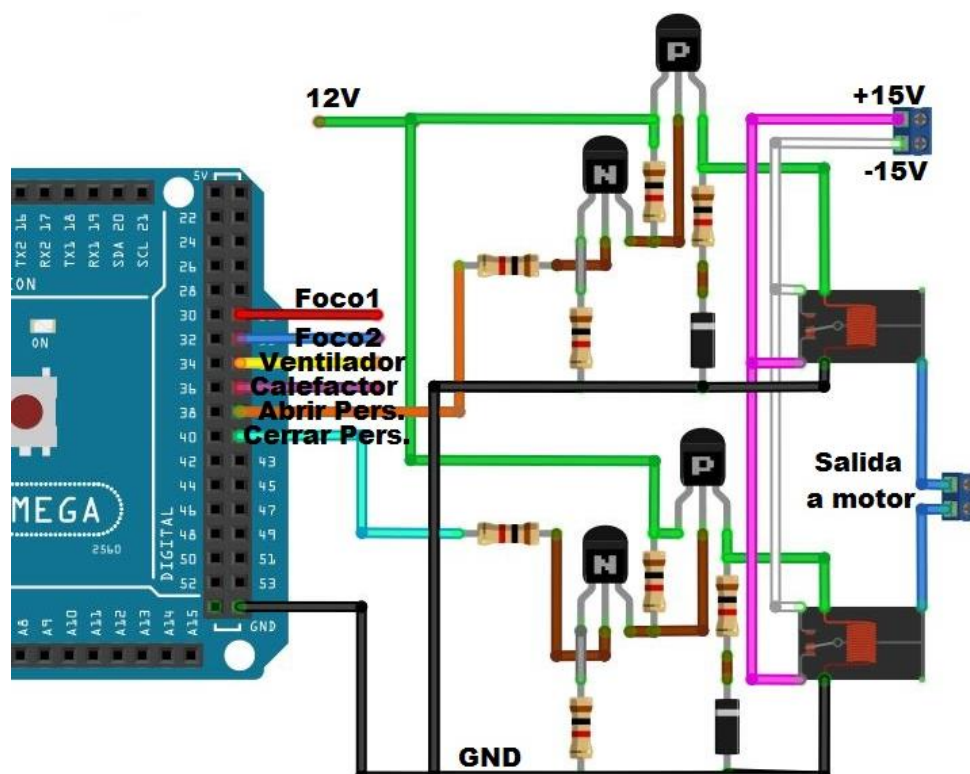
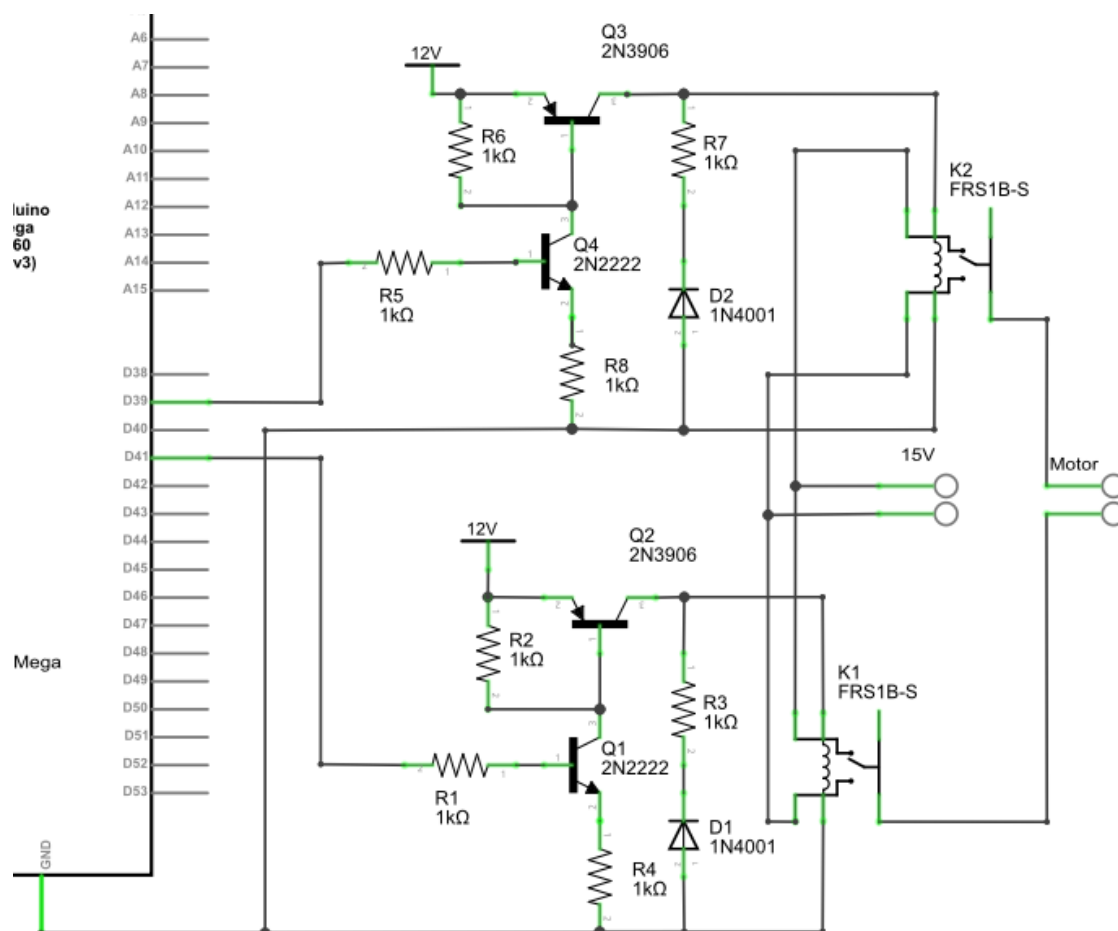
ANEXO 1.1: CIRCUITO IMPRESO DEL CONTROLADOR DE PERSIANAS, CALEFACTOR Y VENTILADOR.



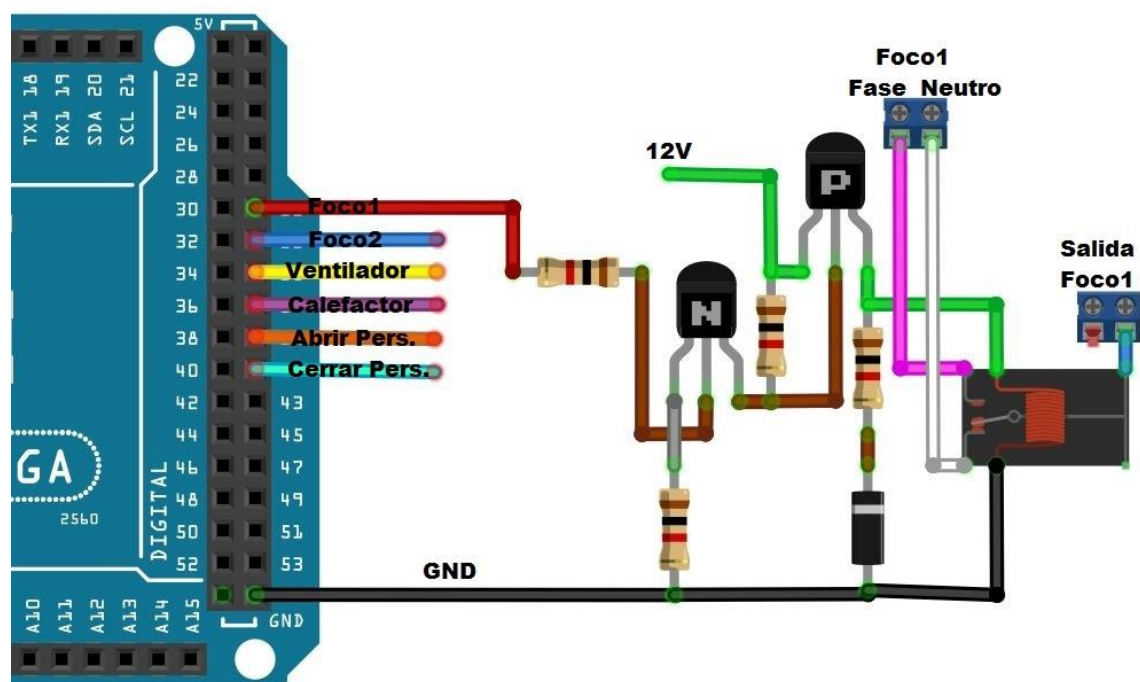
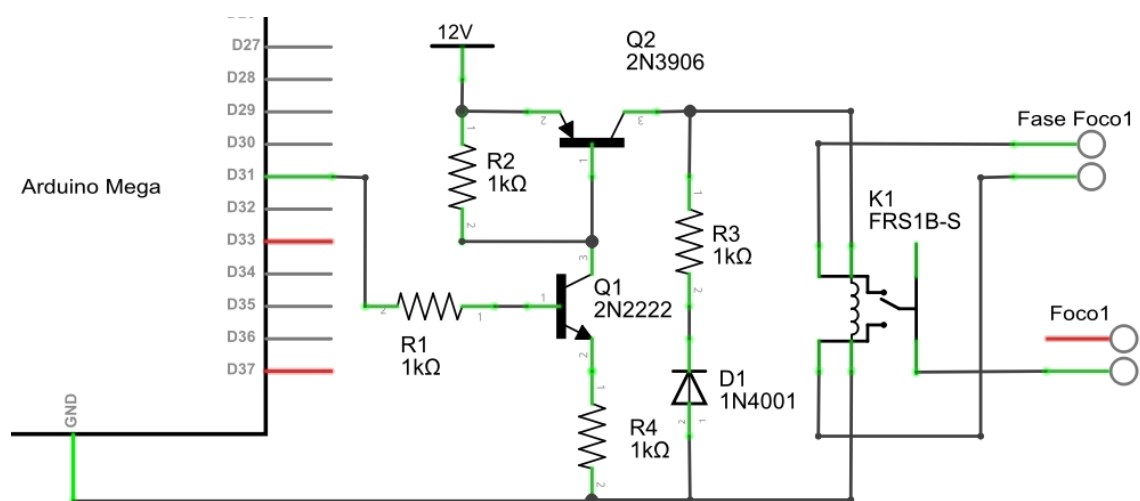
ANEXO 1.2: CIRCUITO IMPRESO DEL CONMUTADOR DE LOS FOCOS.



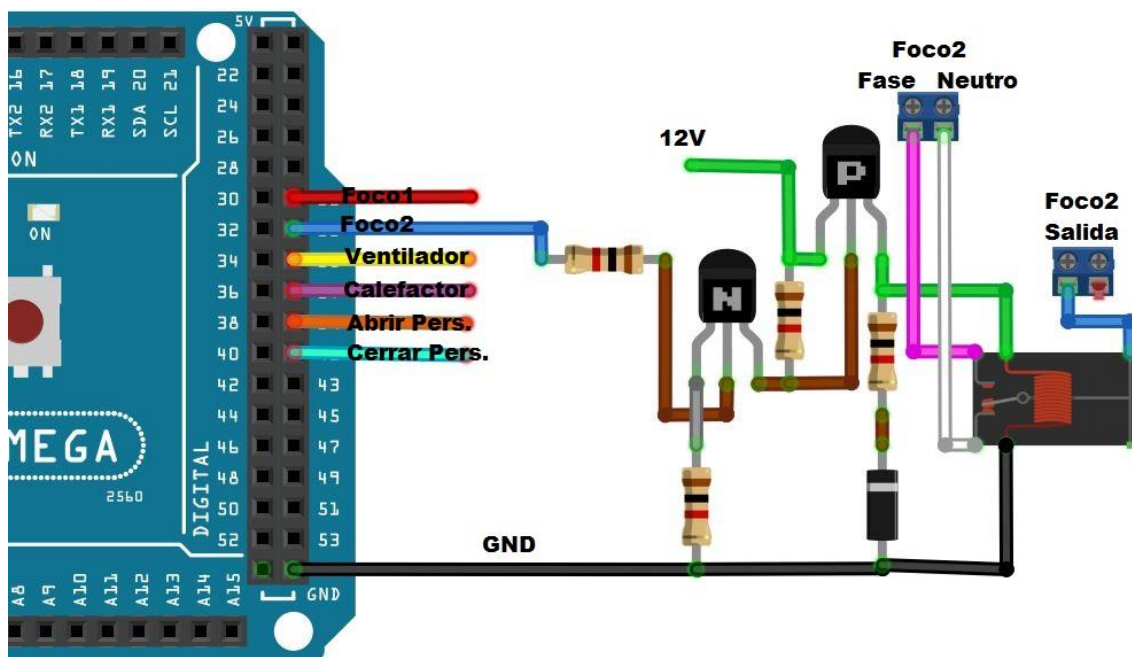
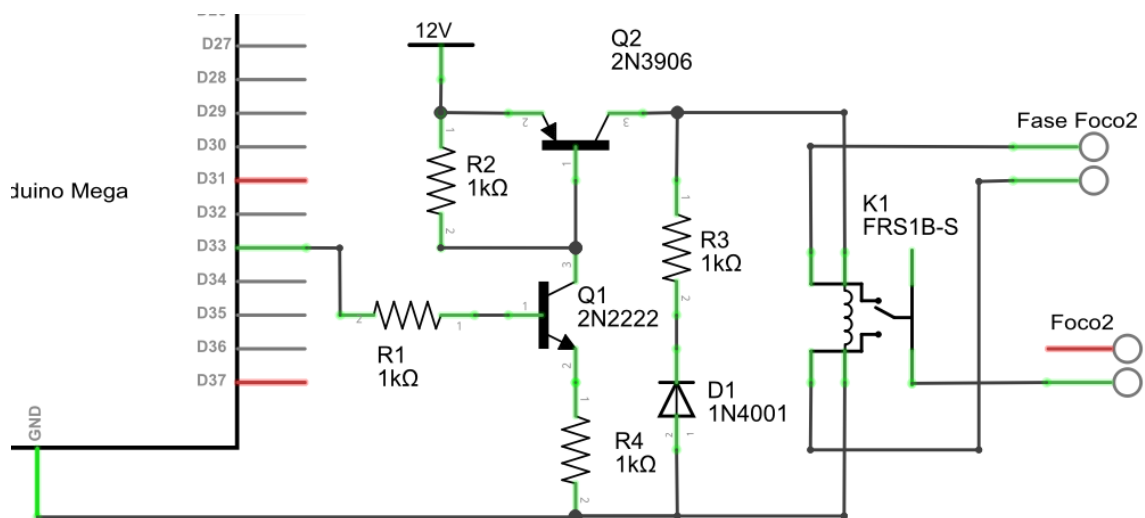
ANEXO 1.3: CIRCUITO CONTROLADOR DE LA PERSIANA



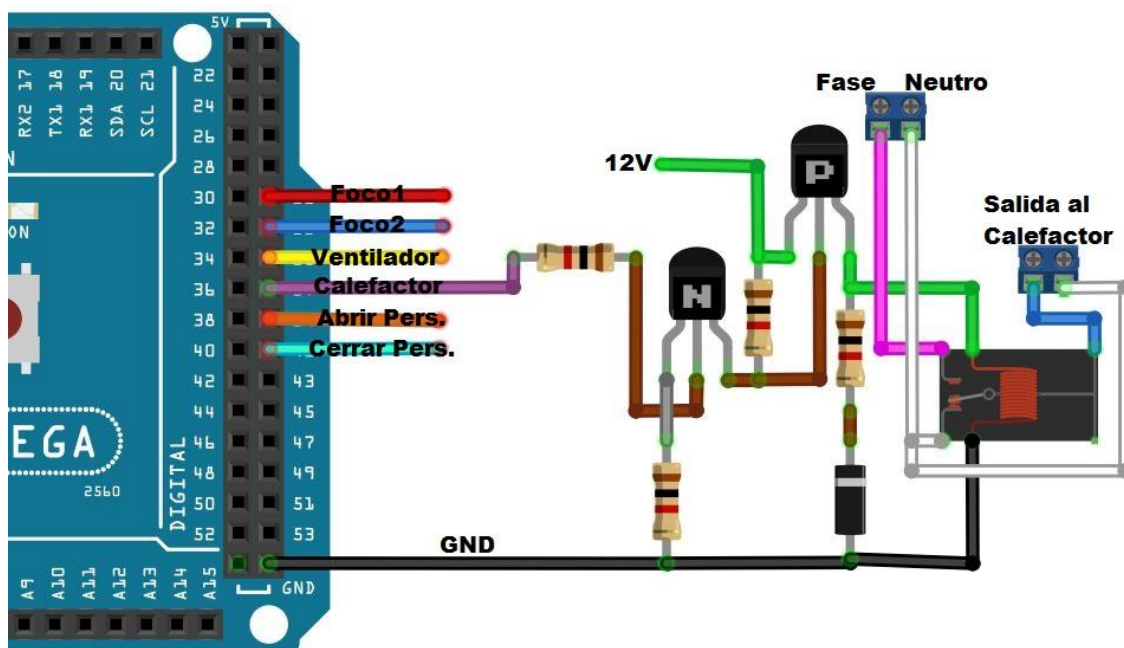
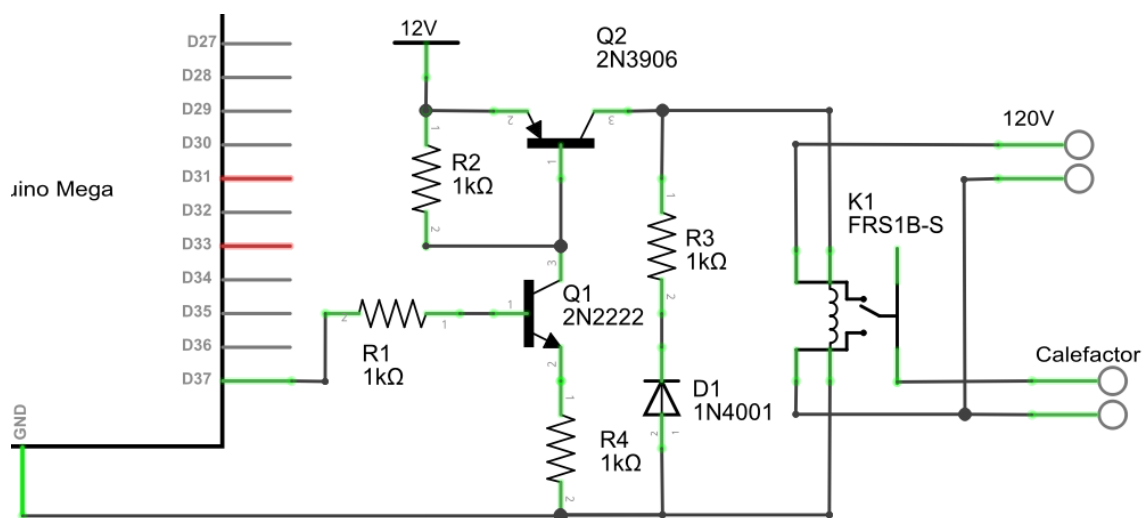
ANEXO 1.4: CIRCUITO CONTROLADOR DEL FOCO 1



ANEXO 1.5: CIRCUITO CONTROLADOR DEL FOCO 2



ANEXO 1.6: CIRCUITO CONTROLADOR DEL CALEFACTO



ANEXO 2. PROGRAMACIÓN ARDUINO

ANEXO 2.1: PROGRAMACIÓN BOTONES CONTROLADORES

```
#include <SPI.h>
#include <Ethernet.h> //Declaración de la direcciones MAC e IP

byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED}; //MAC
IPAddress ip(192, 168, 0, 111); //IP
EthernetServer servidor(82);

int PIN_LED1=31;
int PIN_LED2=33;
int PIN_LED3=35;
int PIN_LED4=37;
int PIN_LED5=39;
int PIN_LED6=41;

int PIN_VA=32;
int PIN_VC=27;
int ventana = 0; // un 0 me indica ventana cerrada y un 1 indica ventana abierta

String readString=String(35); //lee los caracteres de una secuencia en una cadena.
//Los strings se representan como arrays de caracteres (tipo char)
String statel=String(5);
String state2=String(5);
String state3=String(5);
String state4=String(5);
String state5=String(8);

void setup() {
  Ethernet.begin(mac, ip); //Inicializamos con las direcciones asignadas
  servidor.begin();
  Serial.begin(9600); //Inicializa Serial a 9600 baudios
  Serial.println("Iniciando ...");

  pinMode(PIN_LED1,OUTPUT); //definimos como salidas los pines
  pinMode(PIN_LED2,OUTPUT);
  pinMode(PIN_LED3,OUTPUT);
  pinMode(PIN_LED4,OUTPUT);
  pinMode(PIN_LED5,OUTPUT);
  pinMode(PIN_LED6,OUTPUT);
  pinMode(PIN_VA, INPUT);
  pinMode(PIN_VC, INPUT);

  digitalWrite(PIN_LED1,LOW); //arrancan en bajo los pines "salida"
  digitalWrite(PIN_LED2,LOW);
  digitalWrite(PIN_LED3,LOW);
  digitalWrite(PIN_LED4,LOW);
  digitalWrite(PIN_LED5,LOW);
  digitalWrite(PIN_LED6,LOW);
```

```

statel="OFF";
state2="OFF";
state3="OFF";
state4="OFF";
state5="CERRADO";
}

void loop() {
  //EthernetClient Crea un cliente que se puede conectar a
  //una dirección específica de Internet IP
  EthernetClient cliente= servidor.available();
  if(cliente) {
    boolean lineaenblanco=true;
    while(cliente.connected()) {          //Cliente conectado
      if(cliente.available()) {
        char c=cliente.read();           //Leemos petición HTTP caracter a caracter
        if(readString.length()<35) {
          readString.concat(c);          //Almacenar los caracteres en la variable readString
        }
        if(c=='\n' && lineaenblanco) {    //Si la petición HTTP ha finalizado
          int LEA = readString.indexOf("LEA=");
          int LEB = readString.indexOf("LEB=");
          int LEC = readString.indexOf("LEC=");
          int LED = readString.indexOf("LED=");
          int LEF = readString.indexOf("LEF=");

          if(readString.substring(LEA,LEA+5)=="LEA=T") {
            digitalWrite(PIN_LED1,HIGH);
            statel="ON";
            Serial.println('p',DEC); }
          else if (readString.substring(LEA,LEA+5)=="LEA=F") {
            digitalWrite(PIN_LED1,LOW);
            statel="OFF";
            Serial.println("p");
          }

          if(readString.substring(LEB,LEB+5)=="LEB=T") {
            digitalWrite(PIN_LED2,HIGH);
            state2="ON";
            Serial.println('p',HEX); }
          else if (readString.substring(LEB,LEB+5)=="LEB=F") {
            digitalWrite(PIN_LED2,LOW);
            state2="OFF";
            Serial.println('p',OCT);
          }

          if(readString.substring(LEC,LEC+5)=="LEC=T") {
            digitalWrite(PIN_LED3,HIGH);
            state3="ON";
            Serial.println('p',BIN); }
          else if (readString.substring(LEC,LEC+5)=="LEC=F") {
            digitalWrite(PIN_LED3,LOW);
            state3="OFF";
            Serial.println("rb");
          }
        }
      }
    }
  }
}

```

```

        if(readString.substring(LED,LED+5)=="LED=T") {
            digitalWrite(PIN_LED4,HIGH);
            state4="ON";
            Serial.println("sa"); }
        else if (readString.substring(LED,LED+5)=="LED=F") {
            digitalWrite(PIN_LED4,LOW);
            state4="OFF";
            Serial.println("sb");
        }

        if(readString.substring(LEF,LEF+5)=="LEF=T" && ventana==0){
            digitalWrite(PIN_LED5,HIGH);
            ventana=1;
            delay(3000);
            digitalWrite(PIN_LED5,LOW);
            state5="ABIERTO"; }
        else if(readString.substring(LEF,LEF+5)=="LEF=F" && ventana==1){
            digitalWrite(PIN_LED6,HIGH);
            ventana=0;
            delay(3000);
            digitalWrite(PIN_LED6,LOW);
            state5="CERRADO";
        }

//Cabecera HTTP estándar
cliente.println("HTTP/1.1 200 OK");
cliente.println("Content-Type: text/html");
cliente.println(); //Página Web en HTML
cliente.println("<html>");
cliente.println("<head>");
cliente.println("<title>DOMOTICAGABO</title>");
cliente.println("</head>");
cliente.println("<body width=100% height=100%>");
cliente.println("<center>");
cliente.println("<h1>CONTROL DEL HOGAR</h1>");
cliente.print("<br><br>");
cliente.print("Estado del focol: ");
cliente.print(state1);
cliente.print("<br><br>");
cliente.println("<input type=submit value=ON style=width:200px;height:75px onClick=location.href='./?LEA=T\'>");
cliente.println("<input type=submit value=OFF style=width:200px;height:75px onClick=location.href='./?LEA=F\'>");

cliente.print("<br><br><br><br>");
cliente.print("Estado del foco2: ");
cliente.print(state2);
cliente.print("<br><br>");
cliente.println("<input type=submit value=ON style=width:200px;height:75px onClick=location.href='./?LEB=T\'>");
cliente.println("<input type=submit value=OFF style=width:200px;height:75px onClick=location.href='./?LEB=F\'>");

cliente.print("<br><br><br><br>");
cliente.print("Estado del ventilador: ");
cliente.print(state3);
cliente.print("<br><br>");

cliente.println("<input type=submit value=ON style=width:200px;height:75px onClick=location.href='./?LEC=T\'>");
cliente.println("<input type=submit value=OFF style=width:200px;height:75px onClick=location.href='./?LEC=F\'>");

cliente.print("<br><br><br><br>");
cliente.print("Estado del calefactor: ");
cliente.print(state4);
cliente.print("<br><br>");
cliente.println("<input type=submit value=ON style=width:200px;height:75px onClick=location.href='./?LED=T\'>");
cliente.println("<input type=submit value=OFF style=width:200px;height:75px onClick=location.href='./?LED=F\'>");

cliente.print("<br><br><br><br>");
cliente.print("Estado de la persiana: ");
cliente.print(state5);
cliente.print("<br><br>");
cliente.println("<input type=submit value=ON style=width:200px;height:75px onClick=location.href='./?LEF=T\'>");
cliente.println("<input type=submit value=OFF style=width:200px;height:75px onClick=location.href='./?LEF=F\'>");

```

```

        cliente.println("</center>");
        cliente.println("</body>");
        cliente.println("</html>");
        cliente.stop(); //Cierro conexión con el cliente
        readString="";
    }
}
}
}
}

```

ANEXO 2.2: PROGRAMACIÓN ENVÍO DATOS DE SENSORES

```

#include <SPI.h>           //Incluimos la librería SPI
#include "Ethernet.h"      //Incluimos la librería ethernet
#include "DHT.h"           //Incluimos la librería del sensor
#define DHTPIN 43         //definimos PIN 43 para leer datos DHT
#define DHTTYPE DHT11     //definimos tipo de sensor a leer
DHT dht(DHTPIN, DHTTYPE); //Configuramos la entrada y el sensor DHT

#include <Twitter.h> //importamos libreria de twitter

// Configuración Local del arduino
byte mac[] = { 0xED, 0xAE, 0xBD, 0xDF, 0xFD, 0xDE };
IPAddress ip(192, 168, 0, 112); //IP
// Configuración de ThingSpeak
char thingSpeakAddress[] = "api.thingspeak.com";
String writeAPIKey = "*****"; //Clave privada de nuestra página
const int updateThingSpeakInterval = 8 * 1000;
// Variables
long lastConnectionTime = 0;
boolean lastConnected = false;
int failedCounter = 0;
// Inicializando Arduino Ethernet
EthernetClient client;

Twitter twitter("*****"); //Código de twitter
char msg[] = "Intruso detectado N"; //Mensaje a enviar cada vez que se active
int pinEntr = 47; // Pin de entrada del sensor PIR
int valor; // valor de la lectura que hacemos del sensor
int i = 1;

void setup()
{
    Serial.begin(9600); //Para iniciar comunicación serial
    Ethernet.begin(mac, ip); // Arranque de ethernet en arduino
    dht.begin(); //Se inicia el dht
    Serial.println("Comenzando lectura");

    pinMode(38, OUTPUT); //pin que se enciende cuando vaya a enviar un mensaje
    String i;
}

```



```

void loop()
{
    int h = dht.readHumidity();    //Leer la temperatura
    int t = dht.readTemperature(); //Leer la humedad
    //Lectura del sensor DHT11 y DYP-ME003
    String analogValue0 = String(t); //Se convierte en string el valor de la temperatura
    String analogValue1 = String(h); //Se convierte en string el valor de la humedad

    valor = digitalRead(pinEntr); //se lee la entrada analoga
    Serial.println(valor, DEC); //enviamos el dato por el cable hacia el pc
    delay(1000); //esperamos 1 segundo
    if(valor == 1)
    {
        char mensaje [36];
        i=i++;
        sprintf(mensaje,"%s = %i",msg,i);
        digitalWrite(38,HIGH); // Encendemos el led
        Serial.println("conectando..."); //enviamos mensaje a pc que empezo a conectarse
        if (twitter.post(mensaje)) { //envia mensaje a twitter
            int status = twitter.wait(&Serial); //envia datos de la conexion con twtter al pc
            if (status == 200) { //el estado de envio es correcto
                Serial.println("OK"); // envia a pc
            }
            else {
                Serial.print("fallo : codigo "); // si el estado es diferente algo fallo
                Serial.println(status); // imprime el codigo del estado
            }
        }

        else
            digitalWrite(38,LOW); //si la fotocelda esta recibiendo luz del laser

    //Imprimo si el monitor serial esta disponible
    if (client.available()) {
        char c = client.read();
        Serial.print(c);
    }
    //Desconectandome de ThingSpeak
    if (!client.connected() && lastConnected) {
        Serial.println("...desconectado");
        Serial.println();
        client.stop();
        Serial.print("Lectura del sensor: ");
        Serial.print(analogValue0);
        Serial.print("grados C; ");
        Serial.print(analogValue1);
        Serial.println(" % humedad");
    }

    //Actualización ThingSpeak
    if(!client.connected() && (millis() - lastConnectionTime > updateThingSpeakInterval))
    {
        updateThingSpeak("field1="+analogValue0+"&field2="+analogValue1);
    }
    lastConnected = client.connected();
}

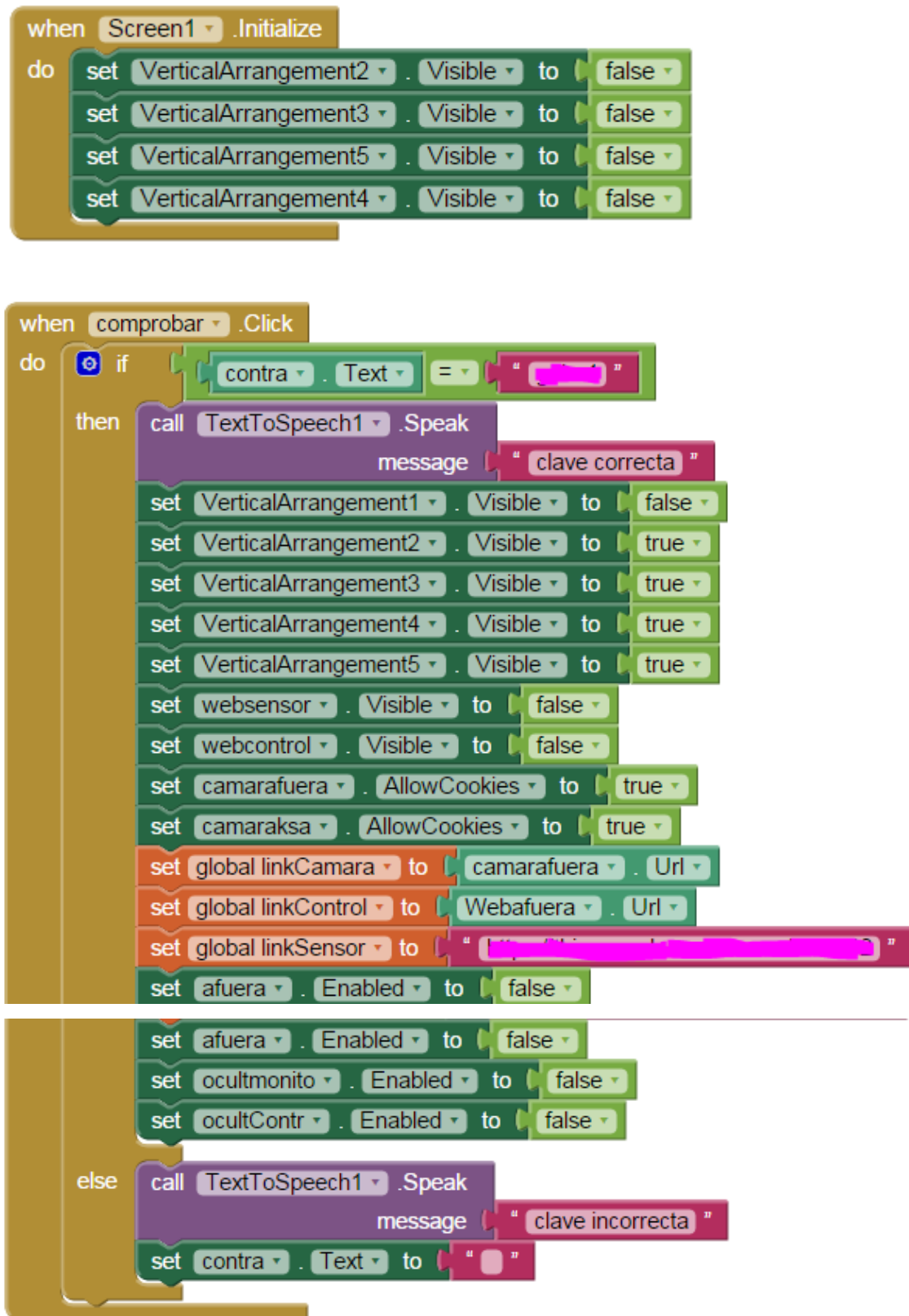
```

```

void updateThingSpeak(String tsData)
{
  if (client.connect(thingSpeakAddress, 80)) // el shield se conecta al puerto 80
  {
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: "+writeAPIKey+"\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(tsData.length());
    client.print("\n\n");
    client.print(tsData);
    lastConnectionTime = millis();
    if (client.connected())
    {
      Serial.println("Conectando con ThingSpeak...");
      Serial.println();
      failedCounter = 0;
    }
    else //Si existe algun tipo de fallo para que no se envíen los datos
    {
      failedCounter++;
      Serial.println("La conexión a fallado (" +String(failedCounter, DEC)+")");
      Serial.println();
    }
  }
  else
  {
    failedCounter++;
    Serial.println("Conexión a ThingSpeak fallo (" +String(failedCounter, DEC)+")");
    Serial.println();
    lastConnectionTime = millis();
  }
}

```

ANEXO 3. PROGRAMACIÓN DE LA APLICACIÓN ANDROID



```

when casa .Click
do
  set global linkCamara to "http://192.168.0.110:81"
  set global linkControl to "http://192.168.0.111:82"
  set casa . Enabled to false
  set afuera . Enabled to true
  set websensor . Visible to false
  set webcontrol . Visible to false
  set ocultContr . Enabled to false
  set ocultmonito . Enabled to false
  set mostvideo . Enabled to true
  set mostContr . Enabled to true
  set mostmonito . Enabled to true

```

```

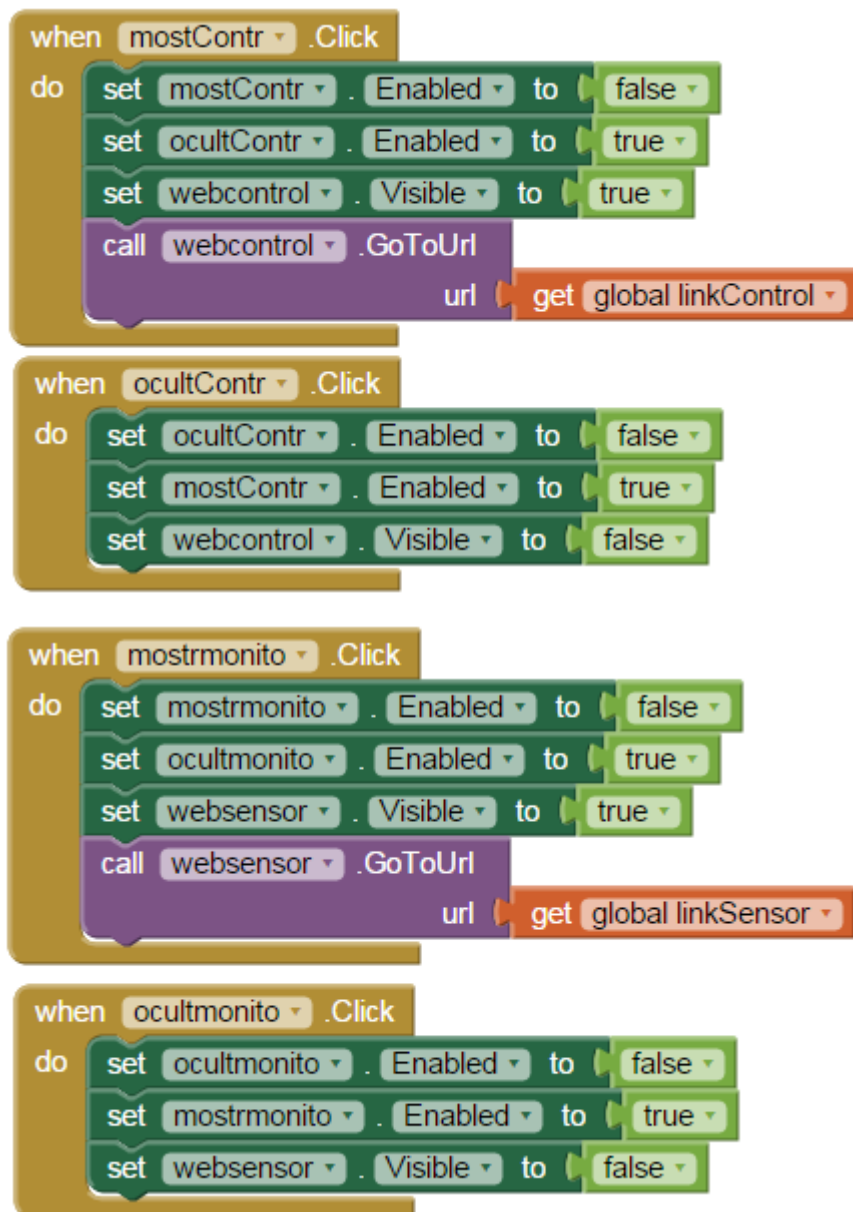
when afuera .Click
do
  set global linkCamara to camarafuera . Url
  set global linkControl to Webafuera . Url
  set afuera . Enabled to false
  set casa . Enabled to true
  set websensor . Visible to false
  set webcontrol . Visible to false
  set ocultContr . Enabled to false
  set ocultmonito . Enabled to false
  set mostvideo . Enabled to true
  set mostContr . Enabled to true
  set mostmonito . Enabled to true

```

```

when mostvideo .Click
do
  set camarafuera . AllowCookies to true
  set camaraksa . AllowCookies to true
  set ActivityStarter1 . Action to "android.intent.action.VIEW"
  set ActivityStarter1 . DataUri to get global linkCamara
  call ActivityStarter1 . StartActivity

```



ANEXO 4. PLANO DE LA CASA DOMOTIZADA EN AUTOCAD

ANEXO 5. ESPECIFICACIONES TÉCNICAS

ANEXO 5.1: ESPECIFICACIONES DEL SENSOR DHT11

Technical Specifications:

Overview:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 °C	± 5 %RH	± 2 °C	1	4 Pin Single Row

Detailed Specifications:

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

Figura 30. Datasheet sensor DHT11 [12].

ANEXO 5.2: ESPECIFICACIONES DEL SENSOR DYP-ME003


	
Tech Support: services@elec Freaks.com	
<h1>Specification of DYP-ME003</h1>	
<h2>1 Electrical parameters</h2>	
Product Type	HC--SR501 Body Sensor Module
Operating voltage range	DC 4.5-20V
Quiescent Current	<50uA
Level output	High 3.3 V /Low 0V
Trigger	L can not be repeated trigger/H can be repeated trigger(Default repeated trigger)
Delay time	5-200S(adjustable) the range is (0.xx second to tens of second)
Block time	2.5S(default)Can be made a range(0.xx to tens of seconds
Board Dimensions	32mm*24mm
Angle Sensor	<100 ° cone angle
Operation Temp.	-15-+70 degrees
Lens size sensor	Diameter:23mm(Default)

Figura 31. Parámetros eléctricos del sensor DYP-ME003 [14].

ANEXO 6. FOTOS DEL PROCESO DE MONTAJE



Figura 32. Ubicación de canaletas y cables dentro de ellas [El autor].



Figura 33. Sellado de canaletas [El autor].



Figura 34. Interruptores de los focos a modificar [El autor].



Figura 35. Caja central y motor empotrados a la pared [El autor].

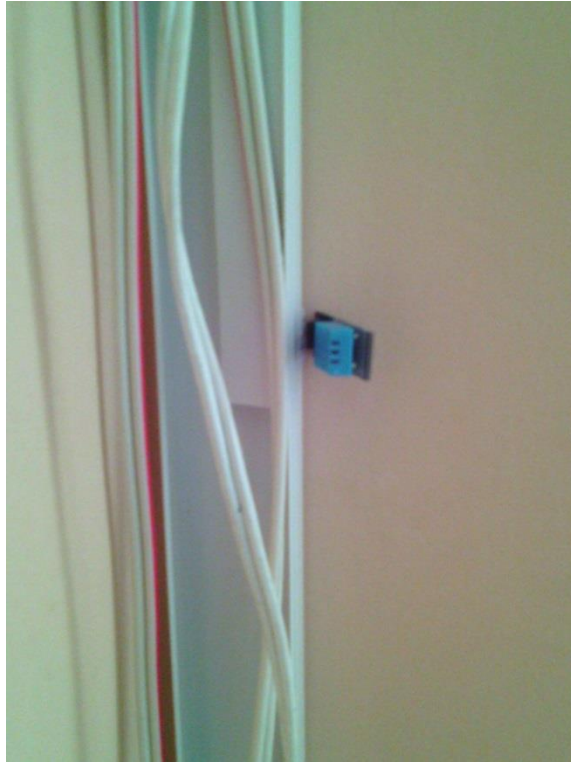


Figura 36. Sensor DHT11 ubicado en punto estratégico [El autor].



Figura 37. Ubicación del circuito controlador de las persianas en la caja central [El autor].



Figura 38. Transformador, motor y circuito controlador dentro de la caja central [El autor].



Figura 39. Caja central finalizada [El autor].



Figura 40. Cámara IP ubicada [El autor].



Figura 41. Conexiones realizadas para alimentar cámara IP [El autor].

ANEXO 7. CERTIFICADO DE TRADUCCIÓN

Lcda. Erika Pezo Badilla

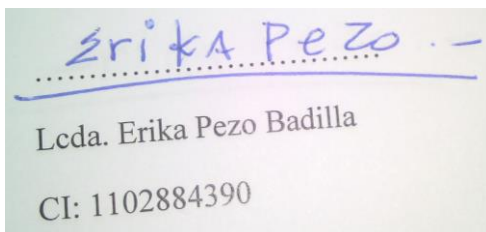
Profesora de inglés.

Certifico:

Por el presente tengo a bien certificar que, la traducción al idioma inglés del resumen de la tesis, ejecutada por el señor **GABRIEL RICARDO CHARRY VILLAMAGUA**, egresado de la carrera de Ingeniería en Electrónica y Telecomunicaciones, la misma que versa como: **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO, MONITOREADO Y CONTROLADO REMOTAMENTE EN TIEMPO REAL MEDIANTE UNA APLICACIÓN ANDROID PARA SMARTPHONE”**, fue realizada bajo mi supervisión y se encuentra correctamente traducida bajo la estructura del lenguaje inglés.

Es todo cuanto puedo decir en honor de la verdad. El interesado puede hacer uso de este certificado como mejor convenga a sus intereses.

Atentamente,



Lcda. Erika Pezo Badilla
CI: 1102884390