

**Universidad Nacional Autónoma de Nicaragua, UNAN – León
Facultad de Ciencias y Tecnología
Ingeniería en Telemática**



“Implementación de la seguridad física en una maqueta de vivienda mediante un sistema domótico inteligente Open Source con Arduino, administrado con una Aplicación móvil desarrollada en App Inventor”

Tesis para optar al título de Ingeniero en Telemática

Autor(es):

**Br. Ary Francisco López Corrales
Br. Rómulo José Miranda Fajardo**

**Tutores: M.sc. Aldo René Martínez D.
M.s.c. Arnoldo José Contreras M**

**León, Nicaragua
Abril, 2021**

**Universidad Nacional Autónoma de Nicaragua, UNAN – León
Facultad de Ciencias y Tecnología
Ingeniería en Telemática**



“Implementación de la seguridad física en una maqueta de vivienda mediante un sistema domótico inteligente Open Source con Arduino, administrado con una Aplicación móvil desarrollada en App inventor”

Tesis para optar al título de Ingeniero en Telemática

Autor(es):

**Br. Ary Francisco López Corrales
Br. Rómulo José Miranda Fajardo**

**Tutores: M.sc. Aldo René Martínez D.
M.s.c. Arnoldo José Contreras M**

**León, Nicaragua
Abril, 2021**

Dedicatoria

Dedico esta tesis de manera muy especial a:

- Nuestro Señor Jesucristo, dador de sabiduría y existencia. A nuestra madre celestial, la virgen María. Espiritualmente hicieron posible la culminación de este trabajo de tesis, ya que sin su ayuda hubiese sido imposible alcanzar nuestras metas.
- A mi familia, que con su apoyo moral y espiritual a lo largo de estos años han complementado profundamente esta especialización.
- A mis profesores quienes siempre demostraron profesionalismo, abnegación y enseñanza con una calidad a la altura de los conocimientos buscados.
- A nuestros amigos del alma y corazón.

Agradecimiento

A Dios y nuestra virgen María, quienes nos concedieron el don de la vida. Por ser fortaleza en los momentos de debilidad, por haberme brindado la capacidad de obtener los conocimientos necesarios para salir adelante.

A mi familia quienes han sido pilar fundamental en mi vida.

A mi tutor Ing. MSc. Aldo René Martínez Delgadillo, MSc. Arnoldo José Contreras por haberme brindado la oportunidad de desarrollar esta tesis bajo su supervisión ya que a pesar de sus muchas responsabilidades siempre me brindó parte de su tiempo, ayuda y apoyo.

A mis amigos por confiar y creer en mí y haber hecho de esta segunda etapa universitaria un trayecto más de vivencias que nunca olvidaré.

Índice de contenido

1	INTRODUCCIÓN.....	9
1.1	ANTECEDENTES.....	9
1.2	PLANTEAMIENTO DEL PROBLEMA.....	10
1.3	JUSTIFICACIÓN.....	12
1.3.1	<i>Originalidad.....</i>	12
1.3.2	<i>Alcance.....</i>	12
1.3.3	<i>Producto.....</i>	13
1.3.4	<i>Impacto.....</i>	13
1.4	OBJETIVOS.....	14
	OBJETIVO GENERAL.....	14
	OBJETIVOS ESPECÍFICOS.....	14
2	MARCO TEÓRICO.....	15
2.1	LA DOMÓTICA.....	15
2.1.1	<i>Historia de la Domótica.....</i>	15
2.1.2	<i>El origen de la Domótica.....</i>	15
2.2	COMPONENTES DE UN SISTEMA DOMÓTICO.....	16
2.2.1	<i>El Centro de Control.....</i>	17
2.2.2	<i>Sensores.....</i>	17
2.2.3	<i>Actuador.....</i>	18
2.2.4	<i>Bus.....</i>	18
2.3	ARDUINO.....	19
2.4	LENGUAJE PROCESSING.....	19
2.5	ARDUINO UNO R3.....	19
2.6	ESTRUCTURA DE UN PROGRAMA.....	20
2.7	¿POR QUÉ UTILIZAR ARDUINO?.....	20
2.7.1	<i>Barato:.....</i>	20
2.7.2	<i>Multiplataforma:.....</i>	20
2.7.3	<i>Entorno de programación simple y claro:.....</i>	20
2.7.4	<i>Código abierto y software extensible:.....</i>	21
2.7.5	<i>Código abierto y hardware extensible:.....</i>	21
2.8	ANDROID.....	22
2.8.1	<i>Historia de Android.....</i>	22
2.8.2	<i>Arquitectura de Android.....</i>	23
2.9	KERNEL DE LINUX.....	23
2.9.1	<i>Bibliotecas.....</i>	23
2.9.2	<i>Entorno de la Aplicación.....</i>	24
2.9.3	<i>Aplicaciones.....</i>	24
2.9.4	<i>Comunicación de Android y la Nube.....</i>	24
2.10	APP INVENTOR.....	25
2.11	SENSOR DE TEMPERATURA.....	26
2.12	SENSOR DE ALARMA.....	26
2.13	SENSOR DE APROXIMACIÓN.....	26
2.14	SERVO MOTOR.....	27
2.15	RELÉ.....	27
2.16	VENTILADORES.....	28

2.17	EL MÓDULO BLUETOOTH HC-05	29
2.17.1	Modulo bluetooth hc-05 como esclavo:	29
2.17.2	Modulo bluetooth hc-05 como Maestro:	29
2.17.3	Sensor Biométrico (Lector de Huella dactilar).....	30
2.18	ENERGÍA SOLAR	32
2.18.1	Características de la Energía Solar	32
2.18.2	Cómo Funciona la Energía Solar.....	32
2.18.3	Tipos de Energía Solar	33
2.18.4	Usos de la Energía Solar.....	33
2.18.5	Ventajas y desventajas de la Energía Solar	33
2.18.6	Beneficios de la Energía Solar.....	34
3	DISEÑO METODOLÓGICO	35
3.1	MATERIALES UTILIZADOS.....	35
3.1.1	Materiales hardware	35
3.1.2	Materiales software.....	36
3.2	ETAPAS DE LA INVESTIGACIÓN.....	37
3.2.1	Etapa I Exploración	37
3.2.2	Etapa II Diseño del equipo.....	37
3.2.3	Etapa III Creación y Configuración	37
3.2.4	Etapa IV Prueba del equipo	37
3.2.5	Etapa V Informe Final	37
4	DESARROLLO Y ANÁLISIS DE RESULTADOS.....	38
4.1	DISEÑO DEL EQUIPO	38
4.1.1	Funcionamiento del equipo.....	38
4.1.2	Definición del esquema de funcionamiento lógico	38
4.2	CREACIÓN Y CONFIGURACIÓN DEL EQUIPO	41
4.2.1	Producto Terminado	52
4.3	PRUEBAS DEL EQUIPO	53
	CONCLUSIONES	56
5	RECOMENDACIONES	57
6	REFERENCIAS BIBLIOGRÁFICAS	58
7	ANEXOS	59
7.1	CODIGO DEL PRIMER PISO.....	64
7.2	CÓDIGO DE APLICACIÓN EN APP INVENTOR.....	66
7.3	PRUEBA INDIVIDUAL	69
7.3.1	Servo motor (puerta y ventana)	69
7.3.2	Led (sistema de luz interna y externa).....	71
7.3.3	Sensor de temperatura y ventilador (sistema de aire autónomo)	73
7.3.4	Sistema de apertura de la verja de la casa (sensor de huella – servo motor)	74
7.3.5	Elemento de manejo del sistema de de la casa.....	80
7.3.6	Sistema de a horro mediante el panel solar (en nuestro proyecto).....	80

Índice de Ilustraciones

ILUSTRACIÓN 1 MODELO DOMÓTICO	16
ILUSTRACIÓN 2 COMPONENTE DE UN SISTEMA DOMÓTICO	17
ILUSTRACIÓN 3 SENSORES DE UN SISTEMA DOMÓTICO	18
ILUSTRACIÓN 4 PLACA ARDUINO	19
ILUSTRACIÓN 5 LOGO DE ANDROID	22
ILUSTRACIÓN 6 ARQUITECTURA APLICACIÓN ANDROID	23
ILUSTRACIÓN 7 COMUNICACIÓN DE ANDROID Y LA NUBE.....	24
ILUSTRACIÓN 8 LOGO DE APP INVENTOR	25
ILUSTRACIÓN 9 SENSOR DE TEMPERATURA.....	26
ILUSTRACIÓN 10 SENSOR DE ALARMA	26
ILUSTRACIÓN 11 SENSOR DE A PROXIMIDAD	27
ILUSTRACIÓN 12 SERVO MOTOR	27
ILUSTRACIÓN 13 ESQUEMA INTERNO DEL RELÉ	28
ILUSTRACIÓN 14 RELÉ.....	28
ILUSTRACIÓN 15 VENTILADORES.....	28
ILUSTRACIÓN 16 MODULO BLUETOOTH HC-05	29
ILUSTRACIÓN 17 ESQUEMA DEL SEGUNDO PISO	41
ILUSTRACIÓN 18 ESQUEMA DEL PRIMER PISO	48
ILUSTRACIÓN 19 DIAGRAMA DEL PRIMER PISO 2	50
ILUSTRACIÓN 20 CASA FINALIZADA	52
ILUSTRACIÓN 21 PRUEBA DE ABERTURA DE LA PUERTA	53
ILUSTRACIÓN 22 PRUEBA DE ABERTURA DE LA VENTANA	53
ILUSTRACIÓN 23 PRUEBA DE ENCENDIDO DE LUZ EXTERNA.....	54
ILUSTRACIÓN 24 PRUEBA DE ENCENDIDO DE LUZ INTERNA.....	54
ILUSTRACIÓN 25 PRUEBA DE ENCENDIDO DEL AIRE CON EL SENSOR DE TEMPERATURA.....	55
ILUSTRACIÓN 26 DISEÑO DE APLICACIÓN MÓVIL.....	60
ILUSTRACIÓN 27 MODELO DE LA CASA 3D.....	60
ILUSTRACIÓN 28 INTERIOR DE LA CASA COCINA	61
ILUSTRACIÓN 29 VISTA DE LA CASA SEGUNDO PISO.....	61
ILUSTRACIÓN 30 INTERIOR DE LA CASA "SALA"	62
ILUSTRACIÓN 31 VISTA DE LA CASA INTERIOR	62
ILUSTRACIÓN 32 INICIO DE CONFIGURACIÓN DE LA CASA	63
ILUSTRACIÓN 33 SISTEMA DE PUERTA Y VENTANA	69
ILUSTRACIÓN 34 SISTEMA DE LUZ INTERNA Y EXTERNA	71
ILUSTRACIÓN 35 SISTEMA DE AIRE AUTÓNOMO	73

Resumen

El término nace del neologismo francés 'Domotique', el cual procede del latín domo que significa 'casa' y del francés telematique 'telecomunicación - informática'. Dichos avances tecnológicos que han experimentado las áreas de software como las de hardware durante las últimas décadas, ha dispuesto que se pueda realizar tareas de forma automatizada, ayudando al control y supervisión de los elementos existentes en las instalaciones brindando confort, seguridad, dando de esta forma la automatización de distintas áreas que antes se realizaba de forma manual, ahorrando costos de tiempo y dinero. Estos sistemas además de posibilitar los niveles de automatización demandados han estado persiguiendo una serie de cualidades que se han llegado a considerar factores clave en el desarrollo de los mismos. Los factores determinantes son la facilidad de uso, la integración de las funciones y la interactividad tanto entre ellos mismos como son el usuario. En base a lo expuesto, hemos decidido implementar una maqueta de casa domótica controlada por una aplicación móvil elaborada con App Inventor

1 Introducción

1.1 Antecedentes

“SISTEMA DOMÓTICO CON TECNOLOGÍA EIBKONNEX PARA LA AUTOMATIZACIÓN DE SERVICIOS, CONFORT Y SEGURIDAD EN LA EMPRESA SISTELDATA S.A”, elaborado por Gustavo Israel Valle Medina en Abril -2012. En este documento se investiga sobre el grado de automatización que posee la empresa Sisteldata S.A. Se elaboró una propuesta de diseño de un sistema Domótico que garantice el control de servicios, confort y seguridad. Una de las principales causas del limitado desarrollo tecnológico en nuestro país es la falta de presupuesto. Al automatizar el edificio de Sisteldata S.A. se realizó una inversión fuerte, pero, hay que tener en cuenta que toda tecnología nueva es costosa en sus inicios y que la justificación de utilizar la misma es su desarrollo a largo plazo, debido al mejoramiento y ventajas que ofrece la domótica día a día. Al realizar la encuesta el 90% de las personas no tienen conocimiento acerca de los avances tecnológicos que se manejan hoy en día en cuanto a infraestructuras y viviendas, por lo que se puede concluir que es un campo relativamente nuevo y que irá tomando fuerza a medida que su mercado en nuestro país se expanda.

“DISEÑO DE UN MÓDULO DIDÁCTICO PARA SISTEMAS DE CONTROL DOMÓTICO CON APLICACIONES DE VIDEO VIGILANCIA SUPERVISADO POR UN TELÉFONO MÓVIL”

Elaborado por Karol Analicia Álvarez Paredes y Isaías Palaguachi Lliguichuzhca en mayo del 2015. Este proyecto se ha enfocado a un módulo didáctico que sirve como banco de trabajo para la carrera de ingeniería electrónica de la Universidad Politécnica Salesiana, que consta de: una maqueta de una vivienda hecha a una escala de 1:10 con las dimensiones de 1.35 mts de ancho por 0.60 mts de alto con una profundidad de 30 cm en los 3 ambientes que son: sala -cocina, dormitorio y patio con garaje, montada en una base rodante de 1:55 cm por 1:70cm que podrá ser utilizada por los alumnos de la carrera de ingeniería electrónica para desarrollar un conocimiento real de domótica.

Al realizar el trabajo, se puede uno dar cuenta de las grandes tendencias de la tecnología futura dando grandes pasos hacia una arquitectura con un futuro de comodidad, confort, seguridad, sin olvidar de cuidado del medio ambiente donde la domótica ayuda con el ahorro de energía dando así edificaciones inteligentes amigables con la naturaleza.

1.2 Planteamiento del problema

En la actualidad, las empresas que realizan las instalaciones del servicio de domótica utilizan controladores industriales, los cuales, por si solos, son caros y limitados, es decir requieren de módulos de extensión para los distintos tipos de dispositivos a controlar.

El presente trabajo pretende responder y aportar información a la comunidad, acerca del desarrollo de la domótica como un punto de apoyo para la comodidad y seguridad en el hogar. Se ha visto la poca información que las personas tienen acerca de automatización de los servicios. Este proyecto también plantea una solución a estas dos variables.

El uso de plataformas de desarrollo convencionales (Arduino), permite eliminar el precio como variable limitante, pues la mayoría de estos son "Open Source" o de código abierto, además su programación es muy versátil y puede realizarse de múltiples formas, lo que permite la personalización total del servicio. Asimismo, al ofrecer el servicio con interfaces gráficas de control se podrá, adicionalmente, visualizar el consumo energético de cada dispositivo para así aprovechar al máximo el rendimiento eléctrico.

Además, la mayoría de servicios de domótica en el mercado ofrecen paquetes integrales y genéricos. No existe versatilidad para controlar ciertos dispositivos ni interfaces a distancia. En el presente trabajo de investigación se ofrece una solución agradable a las familias de poco recurso.

Finalmente, se otorgará mayor control de la vivienda al desarrollar una plataforma de control remoto en forma de aplicación desde el teléfono celular, Tablet, u otro dispositivo que pueda conectarse a la vivienda.

Pregunta general:

- ¿Cómo implementar la seguridad física en una maqueta de vivienda mediante un sistema Domótico inteligente Open Source con Arduino y administrado con una aplicación móvil desarrollada con App Inventor?

Preguntas específicas:

- ¿Es posible construir una maqueta de dos plantas para la casa inteligente utilizando sensores, placa Arduino UNO, motores y actuadores?
- ¿Cómo diseñar un programa en lenguaje C++ basado en processing, para controlar la red de sensores que conforma la casa inteligente?
- ¿Cómo crear un sistema para monitorear y gestionar los diferentes ambientes y seguridad de la vivienda, mediante una aplicación móvil desarrollada en Android con App Inventor?

1.3 Justificación

La seguridad en el hogar es una de las prestaciones más interesantes de la domótica, gracias a la versatilidad de los sistemas que podemos encontrar hoy en día, los dueños de una casa pueden ausentarse de ella con tranquilidad. Una vez saliendo de nuestro hogar o estando dentro de ella en horas de descanso ya sea de día o de noche lo último que queremos al llegar a casa o al levantarnos de una siesta es encontrarnos con que nos han robado, o que hubo un incendio y la casa a quedado reducida a cenizas. Los accidentes en casa son incontrolables, pero gracias a la domótica podemos estar al tanto para poder actuar en el momento correcto. Ya que cualquier peligro que aseche en el hogar puedes ser notificado en el lugar que nos encontremos e incluso se encarga de gestionar las primeras soluciones. Los sistemas domóticos simplifican nuestra vida y minimizan las consecuencias de un posible error humano.

La domótica se ha convertido en algo más que indispensable, quién podría pensar años atrás que con la tecnología se llegarías a tener una vida más cómoda. Puesto que no solamente fue creada para darle acceso a la web por medio de cualquier dispositivo móvil u ordenador; sino que también lógicamente la cotidianidad cada vez sea más tranquila; ya que con un control se puede dominar un hogar.

Nuestro proyecto **“Mejoramiento de la seguridad física en el hogar mediante un sistema Domótico inteligente Open Source con Arduino, administrado con una Aplicación”** puede permitirnos demostrar las potencialidades de esta rama de la tecnología que está en pleno desarrollo. Presentándonos la resolución de problemas en las tareas diarias en el hogar, no como un lujo sino como una necesidad. Esto nos brindará el conocimiento para aumentar la eficiencia en relación a futuros proyectos, permitiéndonos la obtención de resultados fiables

1.3.1 Originalidad

En Nicaragua son pocos los equipos ligados al área tecnológica que se han creado para solventar el problema de ahorro económico y eléctrico en nuestra casa y seguridad del hogar

1.3.2 Alcance

Una vez instalado la placa Arduino a las conexiones eléctricas de la casa se podrán brindar los siguientes servicios:

- Encendido de luces electica de la casa mediante una aplicación móvil por botones o sistema de voz.
- Sistema autónomo de encendido de abanicos mediante el sensor de temperatura
- Sistema de alarma mediante un sensor ultra sónico

- Abertura de las puerta y ventana mediante servo motores controlado por una aplicación móvil por botones o sistema de voz
- Sistema de ahorro energético mediante un panel solar

1.3.3 Producto

Se entregará un equipo de uso simple para el usuario, que incorpore todas las funcionalidades que se requieren para brindar la mejor comodidad a todo el núcleo familiar que vivan en la casa

1.3.4 Impacto

Con la implementación de Arduino en nuestros hogares. podremos tener un gran ahorro de dinero, así como una mayor comodidad a nuestra familia pudiendo realizar las tareas del hogar de una manera más fácil desde cualquier lugar de la casa.

1.4 Objetivos

Objetivo general

- Implementar la seguridad física en una maqueta de vivienda mediante un sistema domótico inteligente Open Source con Arduino y administrado con una Aplicación móvil desarrollada en App Inventor.

Objetivos específicos

- Construir una maqueta de dos plantas para la casa inteligente utilizando sensores, placa Arduino UNO, motores y actuadores.
- Diseñar un programa en lenguaje C++ basado en processing, para controlar la red de sensores que conforma la casa inteligente.
- Crear un sistema para monitorear y gestionar los diferentes ambientes y seguridad de la vivienda, mediante una aplicación móvil desarrollada en Android con App Inventor.

2 Marco teórico

2.1 La Domótica

2.1.1 Historia de la Domótica

En la idealización del concepto de automatizar procesos se han requerido labores muy profundas de investigación, por eso este paradigma tiene muchos años de existencia como tal, desde que un interesado en el área conectó dos cables eléctricos a las manecillas de un reloj despertador, para que, movidos por dichas manecillas, los cables cerraran un circuito formado por una pila y una lámpara. En ese momento surge la idea de temporizar una función eléctrica en un ambiente doméstico (Carrasquilla Gómez & Moreno Betancourt, 1999, pág. 144).

El automatismo se inició durante el siglo XIX con el desarrollo industrial, el cual permitía controlar y establecer secuencialmente los procesos productivos. Con el paso del tiempo y hasta la actualidad, los sistemas han sido perfeccionados hasta llegar al punto en donde las industrias basan gran parte de sus fases de producción en tareas automatizadas o temporizadas. (Falco, 2008) Estados Unidos y Japón fueron los países pioneros en dar una noción de un edificio o inmueble inteligente, en el año 1977, bajo la influencia de factores tecnológicos y económicos. Desde este año se realizan estudios y análisis sobre el impacto que tiene la automatización en la sociedad y la rentabilidad que podían generar sus ideas en un período de baja productividad en el sector industrial (Carrasquilla Gómez & Moreno Betancourt, 1999, pág. 140).

2.1.2 El origen de la Domótica.

La terminología Domótica se origina del latín “domus” y define un conjunto de funciones y servicios aplicados al ámbito doméstico.

(José Manuel Huidobro, Ramón Jesús Millán Tejedor, 2010)

Sostuvo que: El origen de la domótica se remota a la década de los setenta, cuando tras muchas investigaciones aparecieron los primeros dispositivos de automatización de edificios, basados en la aun exitosa tecnología X-10, durante los años siguientes la comunidad internacional mostró un creciente interés por la búsqueda de la casa ideal, comenzando diversos ensayos con avanzados electrodomésticos y dispositivos automáticos para el hogar. (José Manuel Huidobro, Ramón Jesús Millán Tejedor, 2010)

Indican que: Los primeros sistemas comerciales fueron instalados, sobre todo, en Estados Unidos y se limitaban a la regulación de la temperatura ambiente de los edificios de oficina y poco más. Más tarde, tras el auge de los Pc (Personal Computer), a finales de la década de los ochentas y principio de la de los 90, se empezaron a incorporar en estos edificios los SCE (Sistema de Cableado Estructurado) para facilitar la conexión de todo tipo de terminales y periféricos entre sí, utilizando un cableado estándar y tomas repartidas para todo el edificio. Además de los datos, estos sistemas de cableado permitían el transporte de la voz y la conexión de algunos dispositivos de control y seguridad, por lo que, a estos edificios, que disponían de un SCE, se les empezó a llamar edificios inteligentes. Posteriormente, los automatismos destinados a edificios de oficina, junto con otros específicos, se han ido aplicando también a las viviendas en particulares u otro tipo de edificios, donde el número de necesidades a cubrir es mucho más amplio, dando origen a la vivienda domótica. Los diccionarios franceses incorporan el término *domestique* a partir de 1998. Esta palabra traducida al castellano por domótica, es originaria de la palabra latina *domus* (de la que ha derivado la raíz *domo* que quiere decir casa) y de la palabra francesa *informa tique* (de la que ha derivado la palabra informática) o, según otros autores, *robotique* (robótica).

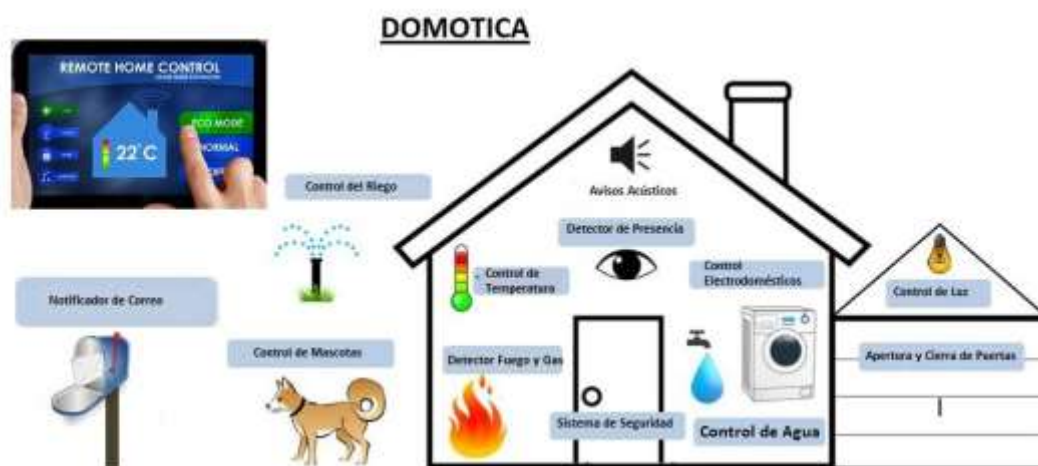


Ilustración 1 Modelo Domótico

2.2 Componentes de un Sistema Domótico.

Entre los componentes que conforman un sistema domótico podemos describir los siguientes:

El centro de control, actuadores, sensores e interfaz de usuario.

Cada uno de ellos juega un papel significativo para el correcto funcionamiento del sistema domótico.



Ilustración 2 Componente de un Sistema Domótico

2.2.1 El Centro de Control.

(Rodríguez, 2014) expresa que es el dispositivo que gestiona toda la información que recibe del sistema, la procesa y toma decisiones de acción, de acuerdo a lo que tenga programado.

Existen dos tipos de centro control, el integrado en un panel y el lógico programable.

El control integrado en un panel, su principal característica es la rapidez con que procesa la información. Los módulos lógicos programables son los más usados debido a que tienen más capacidad de programación y pueden ser adaptables a 21 condiciones futuras. Su principal ventaja es que sus medios de comunicación son estandarizados.

2.2.2 Sensores.

(Rodríguez, 2014) Expresa que el sensor es el dispositivo que monitoriza el entorno captando información que transmite al sistema (sensores de agua, gas, humo, temperatura, viento, humedad, lluvia, iluminación, etc.).



Ilustración 3 Sensores de un sistema Domótico

2.2.3 Actuador. (Rodríguez, 2014)

Expresa que el actuador es un dispositivo capaz de recibir una orden y ejecutarla, así cambiando las características del entorno domótico (encendido/apagado, subida/bajada, apertura/cierre, etc.).

2.2.4 Bus.

Es el medio de comunicación que transporta la información entre los distintos dispositivos, ya sea por una red propia, o por las redes de otros sistemas (red eléctrica, red telefónica, red de datos), también puede ser de forma inalámbrica.

Por cableado propio: la transmisión por un cableado propio es el medio más común para los sistemas de domótica, y son principalmente del tipo:

Par apantallado, par trenzado, coaxial o fibra óptica. Por cableado compartido:

Cuando se utilizan cables compartidos con redes existentes para la transmisión de la información, por ejemplo, la red eléctrica, la red telefónica o la red de datos (Internet).

Inalámbrico:

Hay muchos sistemas de domótica que utilizan soluciones de transmisión inalámbrica entre los distintos dispositivos, basadas principalmente en tecnologías de radiofrecuencia o infrarrojo. Interfaz. –La interfaz son los dispositivos (pantallas, móvil, Internet, interruptores) en que se muestra la información del sistema para los usuarios y donde ellos mismos pueden interactuar con el sistema.

2.3 Arduino

Arduino es una placa programable de código abierto (open source) basado en una tarjeta con un micro controlador básico integrado, posee un ambiente de desarrollo que permite la producción de programas para luego ser cargarlos en la memoria del mismo. Esta tarjeta se caracteriza por tener entradas y salidas tanto analógicas como digitales, la cual permite el control y lectura de los diferentes sensores, puede controlar las luces, motores y numerosas cosas diferentes, se fundamenta en el lenguaje de programación Processing. Este dispositivo asocia el mundo físico con el mundo virtual.



Ilustración 4 Placa Arduino modelo UNO.

2.4 Lenguaje Processing

Es un IDE diseñado por Ben Fry y Casey Reas en el año 2001, este ambiente de trabajo se diseñó con el objetivo inicial para servir como “sketchbook” (cuaderno de dibujo) en proyectos de imágenes, animaciones interactivas así mismo para facilitar el aprendizaje de programación en un contexto visual, es considerado como una herramienta de producción profesional. (Adrian MB, 2014) Inicialmente las versiones del IDE de Processing solo operaban con un modo de uso standard, a partir la versión 1.5.1 se adicionó el modo Android, con el propósito de permitir desarrollar aplicaciones para dispositivos móviles que dispongan de este sistema operativo, actualmente se liberó la versión 2.0alpha1 de Processing complementando el modo JavaScript, Esto permite que la aplicación funcione en un entorno web.

2.5 Arduino Uno R3

Es una tarjeta electrónica que implementa un micro controlador (ATmega328). Tiene 14 pines digitales E/S, 6 entradas analógicas, un resonador cerámico 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el micro controlador; basta con conectarlo al ordenador con un cable USB para empezar. (Arduino Uno, 2015).

2.6 ESTRUCTURA DE UN PROGRAMA

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones.

```
void setup() //Primera Parte
{
  estamentos;
}

void loop() //Segunda Parte
{
  estamentos;
}
```

2.7 ¿Por qué Utilizar Arduino?

(Arduino, Arduino.cl, n.d.) Indica que: Existe una gran variedad de micro controladores y plataformas micro controladoras adecuados para computación física. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, y muchas otras que realizan las mismas funciones. A diferencia de los otros, Arduino simplifica el proceso de trabajo con micro controladores, pero ofrece algunas ventajas para profesores, estudiantes e interesados sobre otros sistemas, además posee características que lo hacen único y que se puedan emplear en proyectos de bajos costos. (Arduino, Arduino CI, 2015)

2.7.1 Barato:

Las placas Arduino son relativamente baratas comparadas con otras plataformas micro controladoras e incluso los módulos de Arduino pre ensamblados cuestan menos de 50\$.

2.7.2 Multiplataforma:

El software de Arduino se ejecuta en sistemas operativos Windows, MacintoshOSX y GNU/Linux. La mayoría de los sistemas micro controladores están limitados Windows.

2.7.3 Entorno de programación simple y claro:

El entorno de programación de Arduino es fácil de usar para principiantes, está basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.

2.7.4 Código abierto y software extensible:

El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado.

2.7.5 Código abierto y hardware extensible:

El Arduino está basado en micro controladores ATMEGA8y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender cómo funciona y ahorrar dinero.

2.8 Android

(Blázquez, 2011) Explica que “Android es una pila de software de código abierto para dispositivos móviles que incluye sistema operativo, middleware y aplicaciones básicas, cimentado en Linux”.

Esta especialmente esquematizado para teléfonos inteligentes y tabletas, que poseen pantalla táctil. La preeminencia de Android, es que permite utilizar de todas las tenacidades se ingeniería de Google.

Una característica esencial de este sistema operativo es que es completamente libre, puede ser descargado libremente, y para los desarrolladores es altamente importante, puesto que ayudaría a recortar costos, al momento de desarrollar una aplicación.



Ilustración 5 Logo de Android

2.8.1 Historia de Android

En su origen la idea de Android fue creada bajo el nombre de una compañía, este sistema operativo fue desarrollado por Android Inc. Sus socios eran Andy Rubin, Rich Miner, Nick Sears y Chris White, nunca se imaginaron que años más tarde Google se fijaría en ellos; Fue así como en el 2005 paso a formar parte de Google.

Junto con la fundación Open Handset Alliance, Android fue presentado oficialmente en el año 2007, la fundación es un consorcio de compañías que se encargan de desarrollar estándares para dispositivos móviles. El primer dispositivo móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. (Carlosnuel, 2013)

2.8.2 Arquitectura de Android

A continuación, haremos una breve descripción de la arquitectura del Sistema Operativo Android, donde relataremos acerca de los cuatro niveles por los cuales está conformado y como está distribuido.

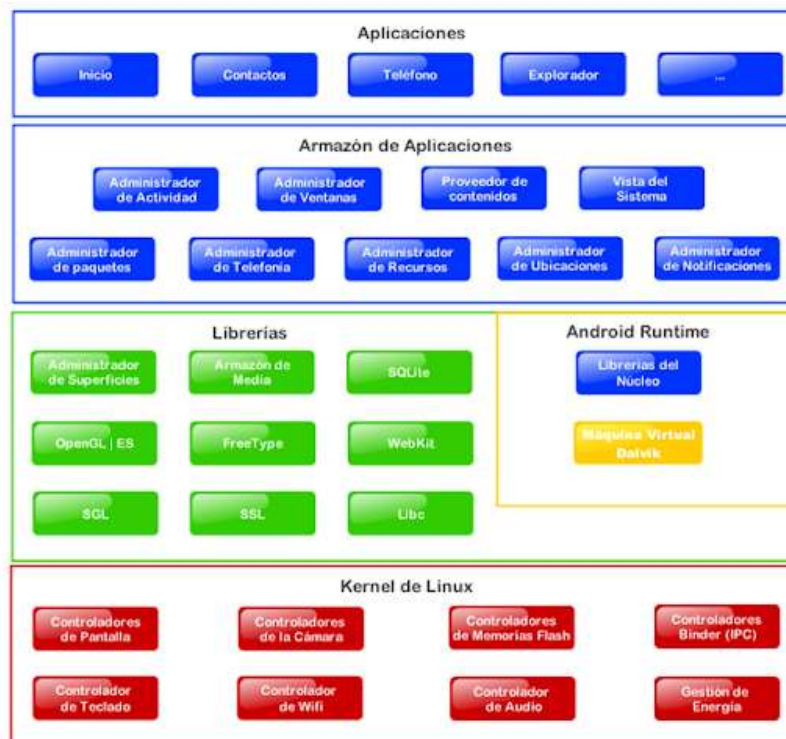


Ilustración 6 Arquitectura Aplicación Android

2.9 Kernel de Linux

El núcleo del sistema operativo Android es un kernel Linux versión 2.6, esta capa proporciona servicios como seguridad, mapeo de la memoria, el multiproceso y el soporte de drivers para los dispositivos. Esta capa provee la abstracción entre el hardware y el resto de la pila, por lo tanto, es la única que es dependiente del hardware.

2.9.1 Bibliotecas

Esta capa está compuesta por las bibliotecas nativas de Android. Estas bibliotecas están escritas en C o C++ y compiladas para la arquitectura hardware específica del teléfono. Su labor es suministrar funcionalidad a las aplicaciones, para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se lleven a cabo de la forma más eficiente.

2.9.2 Entorno de la Aplicación

Esta capa representa básicamente el conjunto de herramientas de desarrollo de cualquier aplicación. Toda aplicación que se desarrolle para Android, ya sean las propias del dispositivo, las desarrolladas por Google o terceras compañías, o incluso las que el propio usuario cree, utilizan el mismo conjunto de API y el mismo "framework", representado por este nivel.

2.9.3 Aplicaciones

Esta capa comprende, tanto las contenidas por defecto de Android como aquellas que el usuario vaya añadiendo posteriormente, ya sean de terceras empresas o de su propio desarrollo. Todas estas aplicaciones utilizan los servicios, las API y librerías de los niveles anteriores.

2.9.4 Comunicación de Android y la Nube

Bibliotecas Esta capa está compuesta por las bibliotecas nativas de Android. Estas bibliotecas están escritas en C o C++ y compiladas para la arquitectura hardware específica del teléfono. Su labor es suministrar funcionalidad a las aplicaciones, para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan acabo de la forma más eficiente.

Entorno de la Aplicación Esta capa representa básicamente el conjunto de herramientas de desarrollo de cualquier aplicación. Toda aplicación que se desarrolle para Android, ya sean las propias del dispositivo, las desarrolladas por Google o terceras compañías, o incluso las que el propio usuario cree, utilizan el mismo conjunto de API y el mismo "framework", representado por este nivel.

Aplicaciones Esta capa comprende, tanto las contenidas por defecto de Android como aquellas que el usuario vaya añadiendo posteriormente, ya sean de terceras empresas o de su propio desarrollo. Todas estas aplicaciones utilizan los servicios, las API y librerías de los niveles anteriores.

Comunicación de Android y la Nube Android proporciona librerías, que nos permite comunicar la aplicación con la nube, estas son encargadas de gestionar el envío y la recepción de data desde un servicio web específico, entre las librerías tenemos las siguientes: HttpClient, HttpPost, HttpGet.

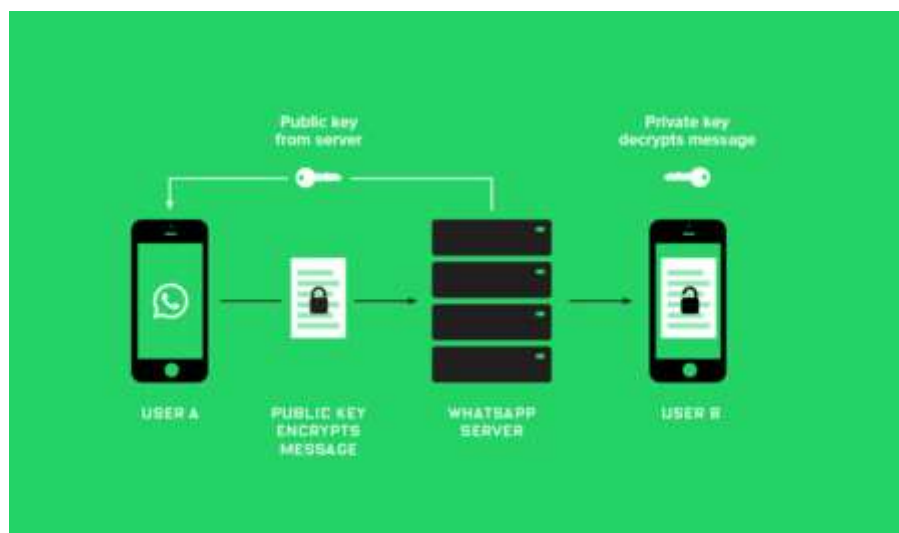


Ilustración 7 Comunicación de Android y la Nube

HttpClient.

Su función es implementar clientes HTTP, es encargada de encapsular los objetos y ejecutar solicitudes HTTP, para la autenticación y gestión de conexión y otras características de seguridad en la red. (Google, 2010)

HttpPost

Este método se utiliza para solicitar que el servidor de origen acepta la entidad incluida en la solicitud identificado por el Request-URI en el Request-Line.

HttpGet

Se usa para obtener información del servidor, puede ser algún archivo HTML, una imagen, un archivo de texto, un XML, etc. Este método solo debe usarse para obtener información del servidor de acuerdo a los estándares de HTTP.

2.10 App Inventor



Ilustración 8 logo de app inventor

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

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

2.11 Sensor de Temperatura

Un **sensor de temperatura** es un componente que recoge la **temperatura** y/o humedad del exterior y lo transforma en señal digital o electrónica que envía a una placa electrónica como puede ser una placa **Arduino**. Existen muchos tipos de **sensores** y para muchos ámbitos.



Ilustración 9 Sensor de temperatura

2.12 Sensor de Alarma

Sistema de alarma basado en Arduino. Con dos sensores uno de movimiento y otro de luz se detecta la alarma. Después de detectar alarma, se entra en un estado de pre-alarma que si no se introduce la clave correcta en 10 segundos, paso a un estado de alarma. ... Detectar un movimiento (activar el sensor tilt)

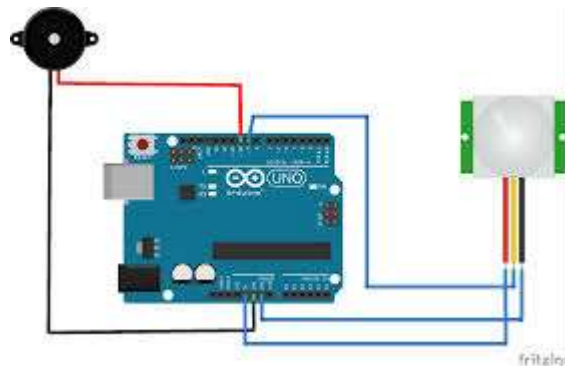


Ilustración 10 Sensor de Alarma

2.13 Sensor de Aproximación

Los sensores de proximidad (el HC-SR04 en este caso) utilizan ultrasonidos para detectar el tiempo que tardan en rebotar las ondas a la velocidad del sonido con los obstáculos que pueda tener delante.

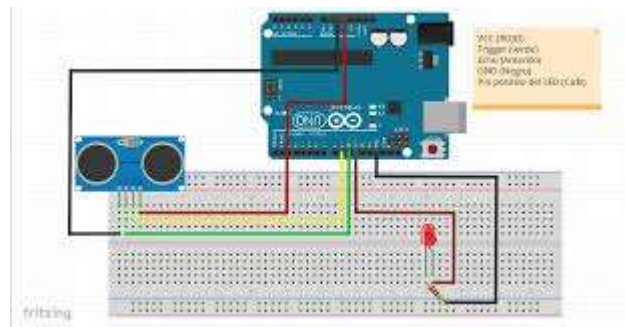


Ilustración 11 Sensor de a proximidad

2.14 Servo Motor

Servo es un tipo de motor DC con reductora que sólo puede girar 180 grados. Se controla mediante el envío de impulsos eléctricos de Arduino. Este pulso le dice al servo a qué posición se debe mover.

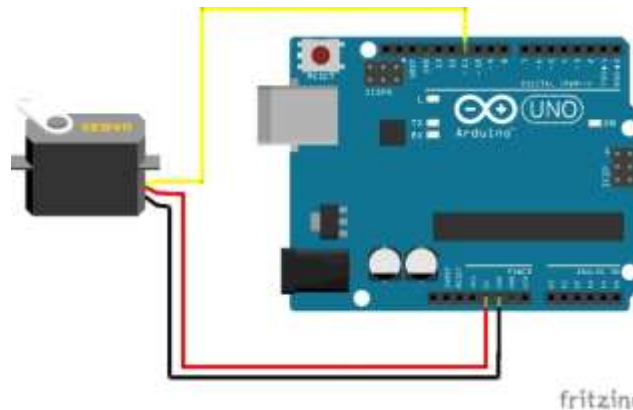


Ilustración 12 Servo Motor

2.15 Relé.

El relé es un dispositivo electromecánico, que funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes para activar otros dispositivos.

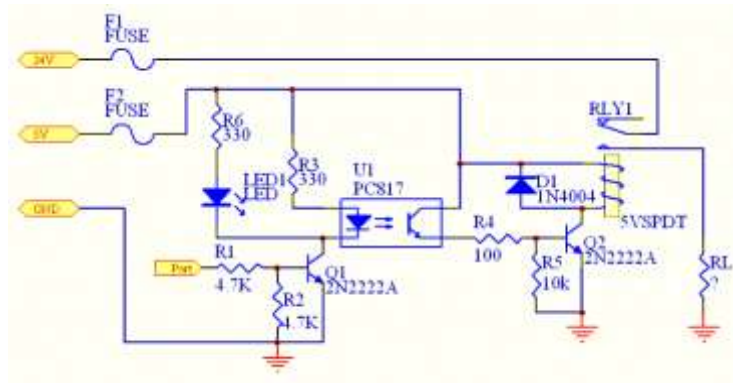


Ilustración 13 Esquema interno del relé

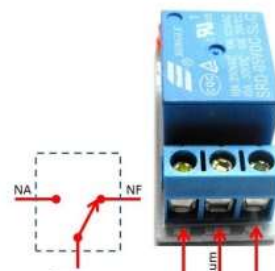


Ilustración 14 relé

2.16 Ventiladores.

Los ventiladores son motores de corriente continua con aspas para ingresar y extraer aire del invernadero, el cual sirve para refrescar el interior del área bajo demanda. Esto significa que, por medio de un sensor de temperatura, se podrá controlar el encendido.



Ilustración 15 Ventiladores

2.17 El módulo Bluetooth HC-05

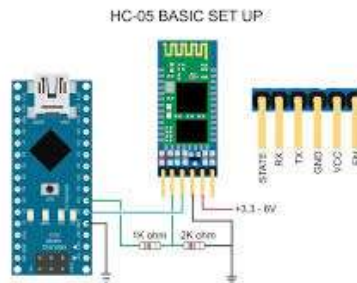


Ilustración 16 Módulo bluetooth HC-05

viene configurado de fábrica como "Esclavo" (slave), pero se puede cambiar para que trabaje como "maestro" (master), además al igual que el HC-06, se puede cambiar el nombre, código de vinculación, velocidad y otros parámetros más.

Definamos primero que es un dispositivo bluetooth maestro y dispositivo esclavo:

2.17.1 Modulo bluetooth hc-05 como esclavo:

Cuando está configurado de esta forma, se comporta similar a un HC-06, espera que un dispositivo bluetooth maestro se conecte a este, generalmente se utiliza cuando se necesita comunicarse con una PC o Celular, pues estos se comportan como dispositivos maestros.

2.17.2 Modulo bluetooth hc-05 como Maestro:

En este modo, EL HC-05 es el que inicia la conexión. Un dispositivo maestro solo se puede conectarse con un dispositivo esclavo. Generalmente se utiliza este modo para comunicarse entre módulos bluetooth. Pero es necesario antes especificar con que dispositivo se tiene que comunicar, esto se explicará más adelante

El módulo HC-05 viene por defecto configurado de la siguiente forma:

- Modo o role: Esclavo
- Nombre por defeco: HC-05
- Código de emparejamiento por defecto: 1234
- La velocidad por defecto (baud rate): 9600

2.17.3 Sensor Biométrico (Lector de Huella dactilar).

El concepto biometría proviene de las palabras bio (vida) y metría (medida), por lo tanto con ello se infiere que todo equipo biométrico mide e identifica alguna característica propia de la persona. Biometría es el conjunto de características fisiológicas y de comportamiento que pueden ser utilizadas para verificar la identidad del individuo, lo cual incluye huellas digitales, reconocimiento del iris, geometría de la mano, reconocimiento visual y otras técnicas. La medición biométrica se ha venido estudiando desde tiempo atrás y es considerada en la actualidad como el método ideal de identificación humana.



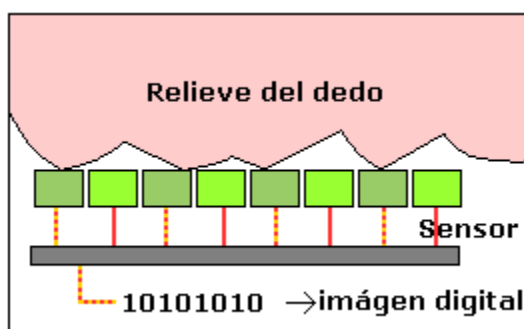
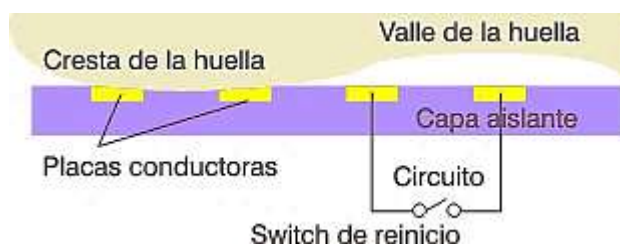
Para este proyecto se utiliza el sensor biométrico para detección de huellas dactilares, el cual funciona con un micro DSP (Procesador de señal digital) que hace la representación y cálculo de la imagen. Se conecta a un serial TTL y manda la información para detectar y comprobar las huellas dactilares. Puede guardar hasta 162 huellas diferentes en la memoria flash y cuenta con un led que indica su estado de funcionamiento.

Un lector óptico funciona con un dispositivo CCD (Charged Coupled Device), como el usado en las cámaras digitales, que tienen un arreglo de diodos sensible a la luz que genera una señal eléctrica en respuesta a fotones de luz. Cada diodo graba un pixel, un pequeño punto que representa la luz que le es reflejada. Colectivamente, la luz y perfiles oscuros forman una

imagen de la huella leída. El proceso de lectura comienza cuando la persona coloca su dedo sobre la ventana del lector, el cual tiene su propia fuente de iluminación, típicamente un arreglo de LEDs, para iluminar las crestas de la huella digital. El CCD genera, de hecho, una imagen invertida del dedo, con áreas más oscuras que representan más luz reflejada (las crestas del dedo) y áreas más claras que representan menos luz reflejada (los valles entre las crestas).

Antes de comparar la información obtenida con la almacenada, el procesador del lector se asegura de que el CCD ha capturado una imagen clara. Checa la oscuridad promedio de los píxeles, o los valores generales en una pequeña muestra, y rechaza la lectura si la imagen general es demasiado oscura o demasiado clara. Si la imagen es rechazada, el lector ajusta el tiempo de exposición para dejar entrar más o menos luz, e intenta leer la huella de nuevo.

Si el nivel de luz es adecuado, el lector revisa la definición de la imagen (que tan precisa es la imagen obtenida). El procesador busca varias líneas rectas que se mueven horizontal y verticalmente sobre la imagen, y si esta tiene buena definición, una línea que corre perpendicular a las crestas será hecha de secciones alternantes de píxeles muy claros y muy oscuros. (biotech.aikons.com), (tec-mex.com.mx), (tecelectronica.com.mx)



2.18 Energía Solar

Es aquella fuente de energía renovable, natural, limpia y **alternativa que es aprovechada de la radiación del sol**. Por medio de diferentes tecnologías o captadores heliostatos, células fotoeléctricas, o colectores solares y llegando a generar la producción de **energía eléctrica o térmica**.

En sí, es la energía que el ser humano **obtiene de la luz y calor del sol**.



2.18.1 Características de la Energía Solar

- Energía renovable y natural que proviene del sol.
- El sol genera más de 4.000 veces más energía de la que puede consumir la humanidad.
- Energía alternativa ante el impacto de los combustibles fósiles y otros problemas ambientales.
- Fuente de energía desarrollada e ilimitada.
- Energía que sirve para generar electricidad y se captura por medio de paneles solares.
- Energía que es limpia por no generar contaminación.
- Promueve el desarrollo sostenible
- Es totalmente ecológica y sustentable.

2.18.2 Cómo Funciona la Energía Solar

Sin duda alguna, que gracias a la tecnología esta energía **se transforma en energía eléctrica como en térmica** y la misma se aprovecha a partir de la luz y el calor que emite el sol.

Para capturar esta energía, se emplean **paneles solares o fotovoltaicos** para transformar la energía luminosa (fotones) en energía eléctrica (electrones).

2.18.3 Tipos de Energía Solar

Entre los principales tipos de energía solar se destacan:

Activa; es la que utiliza la tecnología y equipos mecánicos como bombas y ventiladores para transformar la energía solar para la ventilación, refrigeración y almacenar el calor a futuro.

Pasiva; se obtiene directamente sin transformar la misma sin equipos tecnológicos, mecánicos, reflexiones o refracciones intermedias.

Térmica; energía termosolar para producir calor y calentar alimentos, agua, calefacción, energía mecánica y posteriormente la energía eléctrica.

Fotovoltaica; es aquella energía renovable que se obtiene especialmente de la radiación solar y por célula fotovoltaica para la generación de electricidad.

Híbrida; aquella que se combina con otras energías renovables como biomasa, energía eólica y las no renovables como los combustibles fósiles.

Energía eólico solar; aquella que por medio de una torre que calienta la energía solar y al mismo tiempo el aire, creando una convención que se utiliza para ventilar y enfriar el edificio o casa.

2.18.4 Usos de la Energía Solar

- Para calentar el agua.
- Suministro de energía eléctrica.
- Iluminación de lugares con faros especiales.
- Telecomunicaciones
- Para el transporte y navegación marítima.
- Otros

2.18.5 Ventajas y desventajas de la Energía Solar

2.18.5.1 Ventajas

- Es una fuente de energía inagotable.
- Conserva el medio ambiente por no contaminar el mismo.
- Es una energía independiente.
- Energía económica y de fácil mantenimiento.
- Es modular y de gran aplicación.
- Genera empleo.
- Es sostenible.
- Mejora la calidad de vida.

2.18.5.2 Desventajas

- Inversión inicial.
- Tecnología poca eficiente.
- Intermitente
- Climatología
- Espacio
- Transporte
- Disponibilidad y almacenamiento.
- Impacto ambiental de los paneles solares y células fotovoltaicas.

2.18.6 Beneficios de la Energía Solar

Definitivamente, los principales beneficios de fomentar y promover la energía solar van desde la **seguridad energética de los países**, aumento de la sostenibilidad, reducción de la contaminación ambiental.

Así como el mantenimiento de los precios en los combustibles fósiles, **mitigación al calentamiento global, cambio climático**, mejoramiento en calidad ambiental y de vida de los seres humanos.

3 Diseño metodológico

3.1 Materiales Utilizados

Para llevar a cabo la realización de nuestro proyecto hicimos uso de los siguientes materiales:

3.1.1 Materiales hardware

Materiales	Descripción	Costo
Placa Arduino Uno Protoboard Servo motores Resistencia Led	Estos materiales los adquirimos de un Kit Arduino que contiene varios de los elementos que necesitábamos para nuestro proyecto.	\$ 35
Sensor ultra sónico Sensor de alarma Sensor de temperatura Sensor de teclado Sensor de huellas (fpm10a)	Estos materiales los adquirimos de un Kit sensores Arduino que contiene varios de los elementos que necesitábamos para nuestro proyecto.	\$ 20
Fan	Este material es el que hace la simulación de abanicos	\$ 6
Regleta de relé	Control de la corriente eléctrica	\$ 6
Modulo bluetooth	Módulo utilizado para la comunicación entre casa y la aplicación	\$ 8
Panel solar	Sistema usado para el ahorro de energía	\$ 40
Caja múltiple de conexión USB	Sistema utilizado para la distribución de energía en el sistema de seguridad	\$ 5
Maqueta de la casa	Maqueta utilizada para el sistema domótica	\$ 40
Computadora	Procesador AMD A8, 2.50GHz. RAM 8GB	\$300
Total		\$ 455

3.1.2 Materiales software

Software	Descripción	Costo
Arduino	Compilador de software libre que utilizamos para programar nuestro proyecto.	Gratuito
Android	Sistema que utilizaremos para aplicación móvil	Gratuito
App Inventor	Programa utilizado para el desarrollo de la aplicación que ayuda a la manipulación de la maqueta	Gratuito
Sketchup	Programa utilizado para creación de modelo de la casa en 3D	Gratuito
LucidSpark	Programa utilizado para la creación de los diagramas de flujo	Gratuito
TinkerCard	Programa utilizado para la creación del modelo gráfico de los circuitos utilizado en la maqueta	Gratuito

3.2 Etapas de la investigación

Para cumplir nuestros objetivos planteado en nuestro trabajo lo dividimos en las siguiente etapas.

3.2.1 Etapa I Exploración

En Esta etapa se recopilo información teórica y práctica acerca del tema, haciendo búsqueda de libro y sitios web.

3.2.2 Etapa II Diseño del equipo

En esta etapa definimos cuáles son los equipos a utilizar durante la creación de nuestro escenario propuesto

3.2.3 Etapa III Creación y Configuración

En esta etapa del proyecto comenzamos a armar y configurar la maqueta de nuestro sistema Domótico a presentar

3.2.4 Etapa IV Prueba del equipo

En esta etapa del proyecto hacemos las últimas pruebas ante de presentarlo el proyecto final

3.2.5 Etapa V Informe Final

En esta etapa del trabajo terminamos de documentar todo lo trabajado durante este periodo

4 Desarrollo y análisis de resultados.

En esta sección se aborda el desarrollo de las diversas etapas del proyecto.

4.1 Diseño del Equipo

Para dar inicio a nuestro proyecto lo principal fue diseñar un esquema lógico en el que plasmamos la idea del equipo que deseábamos crear.

4.1.1 Funcionamiento del equipo

En este apartado explicaremos la estructura interna y el funcionamiento del equipo final.

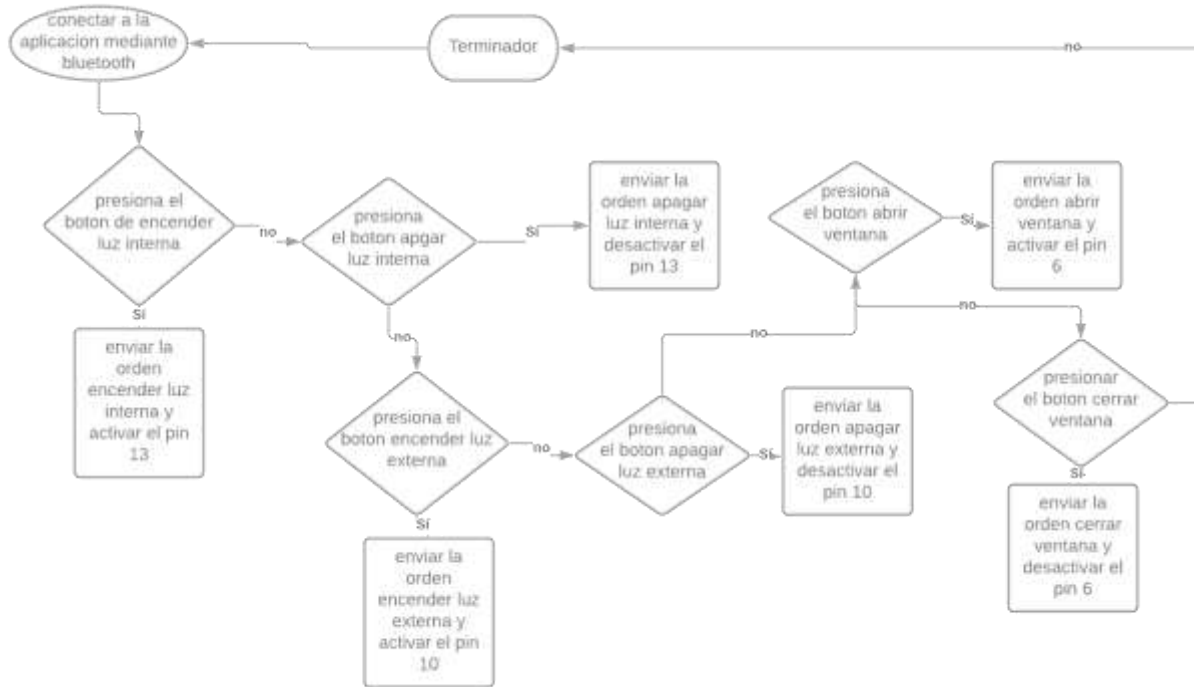
4.1.2 Definición del esquema de funcionamiento lógico

El funcionamiento básico del equipo se basa en que la maqueta cuenta con dos pisos:

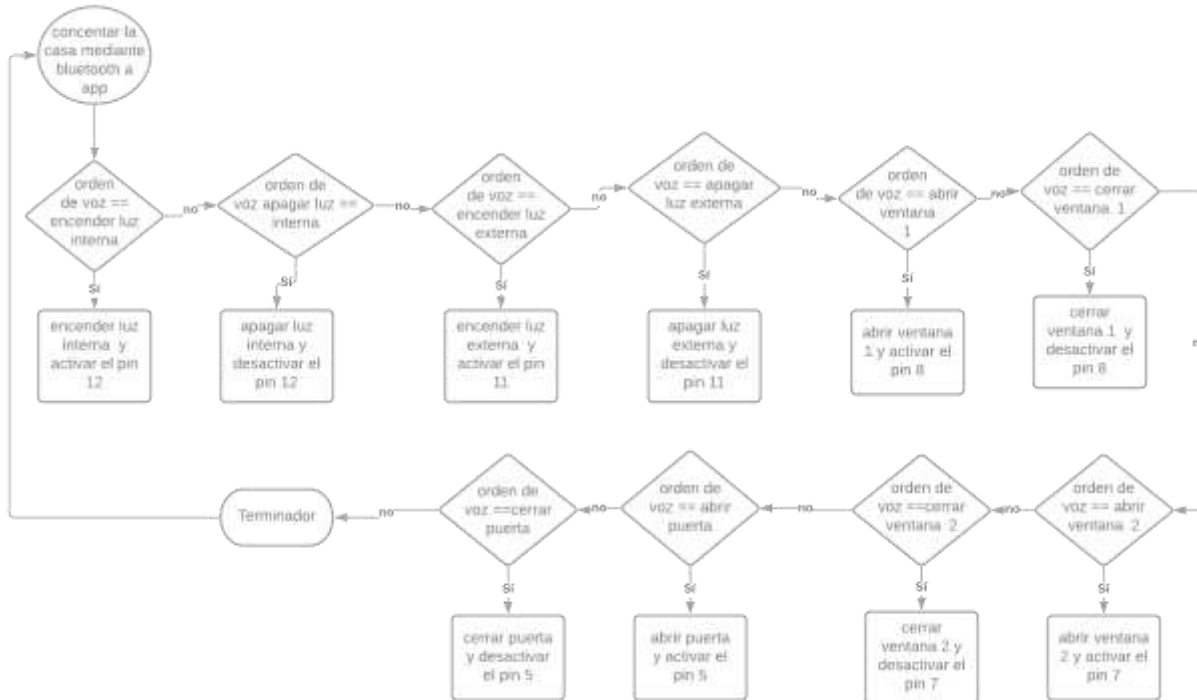
el primer piso cuenta con luz interna y externa serán encendida o pagada mediante una aplicación móvil mediante bluetooth, una puerta la cual será abierta o cerrada mediante servo motor y un módulo keypad 4x4, una ventana la cual será abierta o cerrada mediante servo motor controlada por la aplicación y un sistema de alarma el cual está conformada por un buzzer y un sensor ultrasónico

El segundo piso de la casa está conformado por dos ventanas una puerta que serán abierta o cerrada por servo motor y la aplicación móvil, sistema de luces interna y externa encendida mediante la aplicación móvil y un sistema de aire autónomo el cual es encendido por un sensor de temperatura.

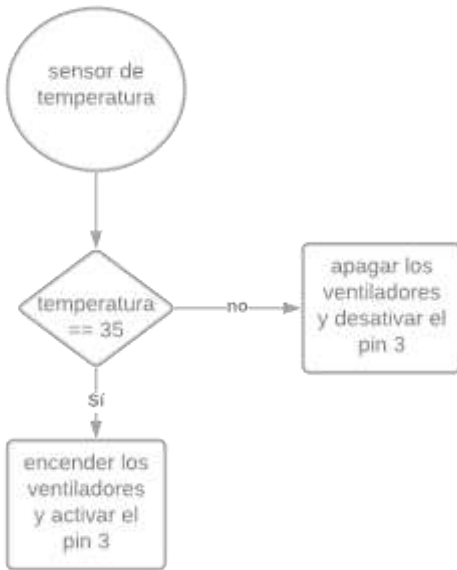
4.1.2.1 Diagrama lógico del primer piso



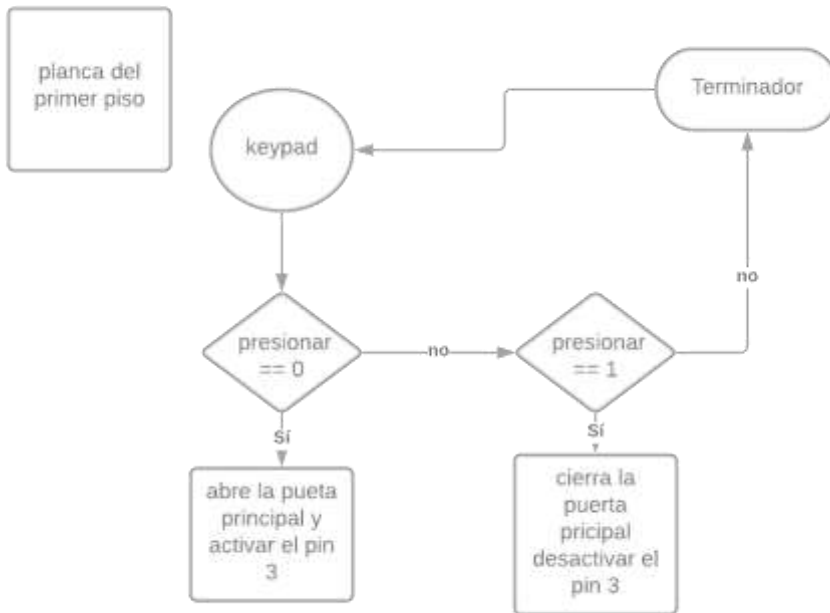
4.1.2.2 Diagrama lógico del segundo piso



4.1.2.3 Diagrama lógico del sistema autónomo del aire



4.1.2.4 Diagrama lógico del sistema de puerta principal



4.2 Creación y configuración del equipo

En la siguiente imagen se muestra el diseño físico de nuestro equipo y las conexiones necesarias para su funcionamiento. En él se ven reflejadas las conexiones correspondientes:

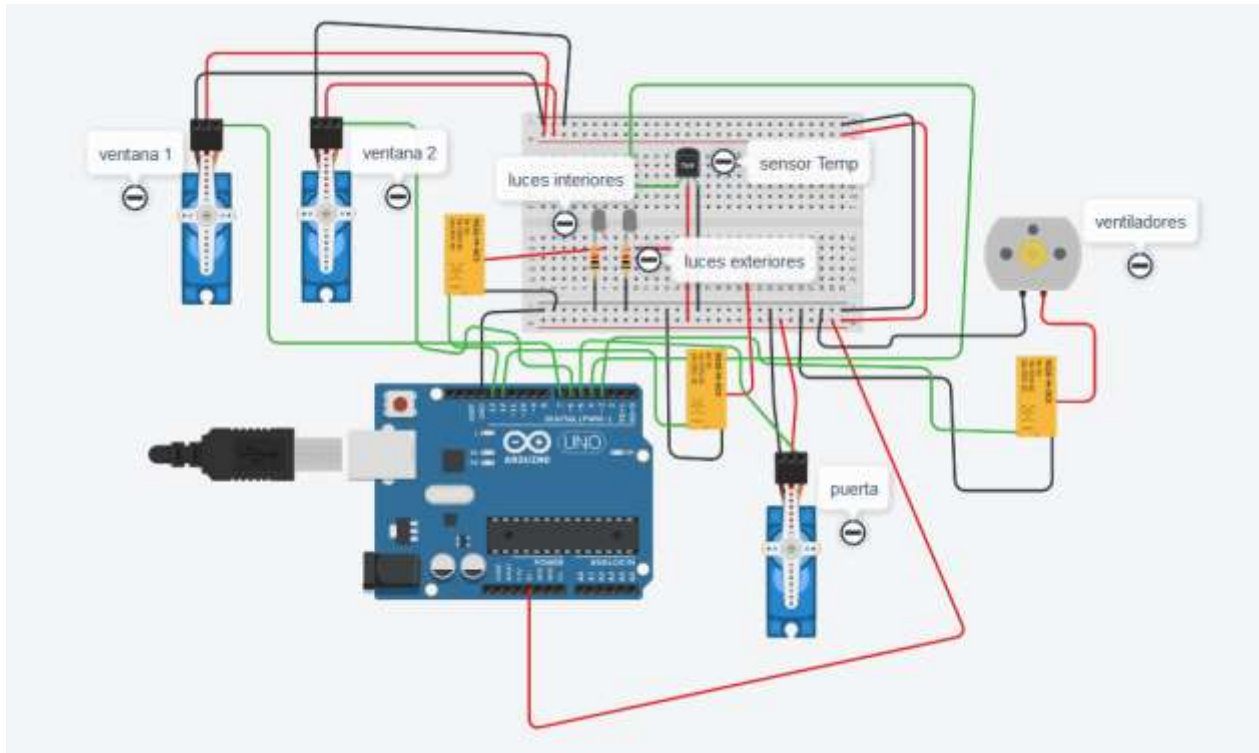


Ilustración 17 Esquema del segundo piso

En este esquema tenemos toda conectado el segundo piso el cual tiene la puerta, ventana, luces internas, luces externas, sensor de temperatura que activa los ventiladores manipulado mediante un bluetooth y conectado a una aplicación móvil.

Código completo del segundo piso

```
#include <dht11.h>
#include <Servo.h>
#define DHT11PIN 4

#define RELAY_ON 0
#define RELAY_OFF 1
#define RELAY_ON1 0
#define RELAY_OFF1 1
#define RELAY_ON2 0
#define RELAY_OFF2 1
#define RELAY_ON3 0
#define RELAY_OFF3 1
#define RELAY_ON4 0
#define RELAY_OFF4 1

Servo ventana2;
Servo ventana1;
Servo ventana;
Servo Puerta;
char leer;
int vel = 0;
dht11 DHT11;

void setup()
{
  // inicializaciòn

  Serial.begin(9600);

  digitalWrite (13, RELAY_OFF); // pin de entrada
```

```
pinMode (13, OUTPUT); // pin de salida

digitalWrite (12, RELAY_OFF2); // pin de entrada

pinMode (12, OUTPUT); // pin de salida

digitalWrite (3, RELAY_OFF1);

pinMode(3, OUTPUT);

digitalWrite (11, RELAY_OFF3); // pin de entrada

pinMode (11, OUTPUT); // pin de salida

digitalWrite (10, RELAY_OFF4); // pin de entrada

pinMode (10, OUTPUT); // pin de salida

ventana2.attach(8);

ventana1.attach(7);

ventana.attach(6);

Puerta.attach(5);
}

void loop()
{

Serial.println();

leer = Serial.read();

if ((leer == '1'))

{

vel = 110;

ventana1.write(vel);

delay(1500);

}

else if ((leer == '2')){

vel = 0;
```

```
ventana1.write(vel);  
  
delay(1500);  
  
}  
  
else if ((leer == '3')){  
  
    vel = 110;  
  
    ventana.write(vel);  
  
    delay(1500);  
  
}  
  
else if ((leer == '4')){  
  
    vel = 0;  
  
    ventana.write(vel);  
  
    delay(1500);  
  
}  
  
    else if ((leer == '3')){  
  
        vel = 110;  
  
        ventana2.write(vel);  
  
        delay(1500);  
  
    }  
  
else if ((leer == '4')){  
  
    vel = 0;  
  
    ventana2.write(vel);  
  
    delay(1500);  
  
}  
  
else if ((leer == '9'))  
  
{  
  
    vel = 110;  
  
    Puerta.write(vel);  
  
    delay(1500);  
  
}  
  
else if ((leer == 'a'))
```

```
{  
  vel = 0;  
  Puerta.write(vel);  
  delay(1500);  
}  
else if ((leer == '5'))  
{  
  digitalWrite (13, RELAY_ON); //Activa relé 1  
}  
else if ((leer == '6'))  
{  
  digitalWrite (13, RELAY_OFF); //Desactiva relé 1;  
}  
else if ((leer == '7'))  
{  
  digitalWrite (12, RELAY_ON2); //Activa relé 3  
}  
else if ((leer == '8'))  
{  
  digitalWrite (12, RELAY_OFF2); //Desactiva relé 3;  
}  
  
else if ((leer == '7'))  
{  
  digitalWrite (11, RELAY_ON3); //Activa relé 4
```

```
}  
  
else if ((leer == '8'))  
{  
  digitalWrite (11, RELAY_OFF3); //Desactiva relé 4;  
  
}  
  
else if ((leer == '7'))  
{  
  digitalWrite (10, RELAY_ON4); //Activa relé 5  
  
}  
  
else if ((leer == '8'))  
{  
  digitalWrite (10, RELAY_OFF4); //Desactiva relé 5;  
  
}  
  
int chk = DHT11.read(DHT11PIN);  
  
Serial.print("Humedad: ");  
Serial.println((float)DHT11.humidity, 2);  
  
Serial.print("Temperatur: ");  
Serial.println((float)DHT11.temperature, 2);  
if (DHT11.temperature >= 25)  
{  
  digitalWrite(3, RELAY_ON1);  
  
}  
else  
{
```

```
digitalWrite (3, RELAY_OFF1);  
}  
delay(2000);  
}
```

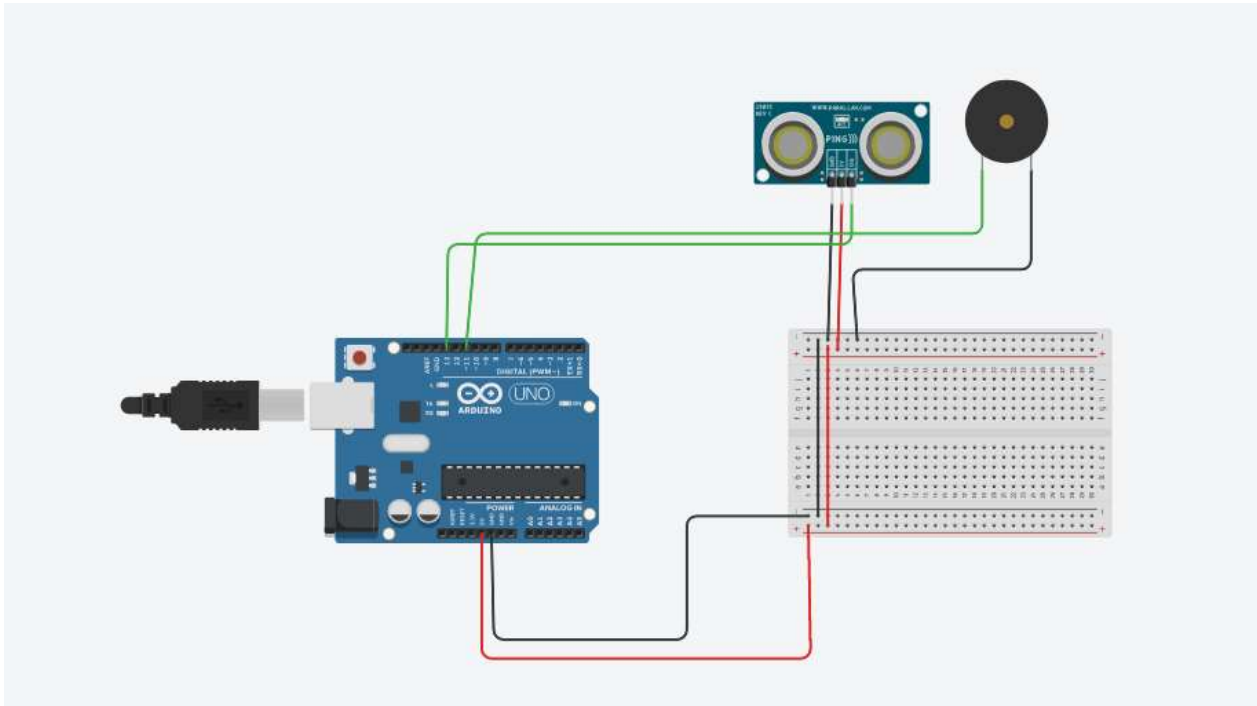


Ilustración 18 Esquema del primer piso

Aquí tenemos el esquema del sistema de alarma que está compuesto por un sensor ultra sónico y un buzzer el cual es activado cuando el sensor ultrasónico detecta movimiento cercano al activa el buzzer y emite un sonido

Código de sistema de alarma

```
int Trigger=2; //PIN usado para trigger
int Echo=3; //PIN usado para echo
int Buzzer = 13; //PIN usado para buzzer / led
int t,d; //variables para valores del sensor

void setup() {
  Serial.begin(9600); //Incializamos la comunicación
  pinMode(Trigger,OUTPUT); //Declaramos a trigger de tipo Salida (emisor)
  pinMode(Echo,INPUT); //Declaramos a echo de tipo Entrada (receptor)
  pinMode (Buzzer, OUTPUT); //Declaramos al buzzer de tipo salida
  digitalWrite(Trigger,LOW); //Inicializamos el sensor en cero (apagado)
}
```



```
void loop() {
  delay(1000);
  digitalWrite(Triiger, HIGH);
  delayMicroseconds(10);
  digitalWrite(Triiger,LOW);
  t= pulseIn(Echo,HIGH); //es el tiempo que tarda el eco del ultrasonico
  d= 0.01715*t; // Agregamos la formula
  //Agregamos un IF para validar, si se detecta algo o alguien a una distancia menor a 10cm encendemos la alarma.
  if (d < 5){
    digitalWrite (Buzzer, HIGH); // Buzzer will be on
    delay (4000); // Waitmode for 4 seconds
    digitalWrite (Buzzer, LOW); // Buzzer will be off
    delay (2000); // Waitmode for another 2 seconds in which the buzzer will be off
  }

  //Mostramos en el monitor la distancia que esta midiendo el sensor.
  Serial.print ("Distancia: ");
  Serial.print (d);
  Serial.print ("cm");
  Serial.println("");
  delay (100);
}
```

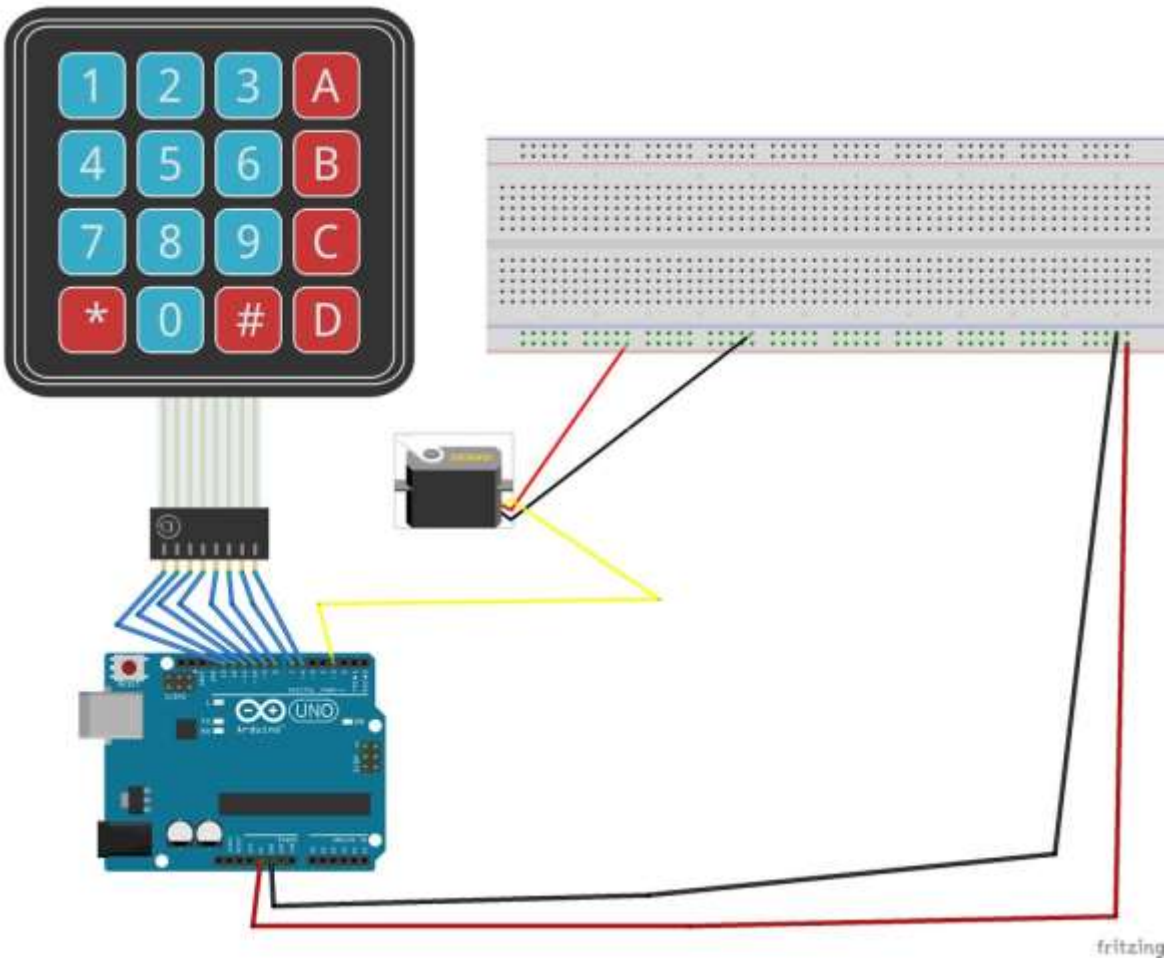


Ilustración 19 diagrama del primer piso 2

En este esquema grafico tenemos un keypad y servo motor que cuando ponemos una clave abre la puerta y otra clave cierra la puerta.

Código del sistema de puerta mediante el teclado 4 x 4

```
#include <Keypad.h>
#include <Servo.h>

const byte rows = 4; //four rows since the keypad is full
const byte columns = 4; //four columns, same as above

char buttons[rows][columns] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowP[rows] = {13, 12, 11, 10}; //row pins of the keypad
byte columnsP[columns] = {9, 8, 7, 6}; //column pins of the keypad

Keypad pad = Keypad( makeKeymap(buttons), rowP, columnsP, rows, columns ); //create keypad

Servo myServo; //

void setup(){
  Serial.begin(9600);
  myServo.attach(5); // attaches the servo on pin 9 to the servo object
}

void loop(){
  char key = pad.getKey(); //get the char from the keypad
  if(key == '0')
  {
    myServo.write(11);
  }
}
```

```
    delay(15);  
  }  
  if(key == '1')  
  {  
    myServo.write(0);  
  
    delay(15);  
  }  
}
```

4.2.1 Producto Terminado



Ilustración 20 Casa finalizada

4.3 Pruebas del equipo



Ilustración 21 Prueba de abertura de la puerta



Ilustración 22 Prueba de abertura de la ventana



Ilustración 23 Prueba de encendido de luz externa

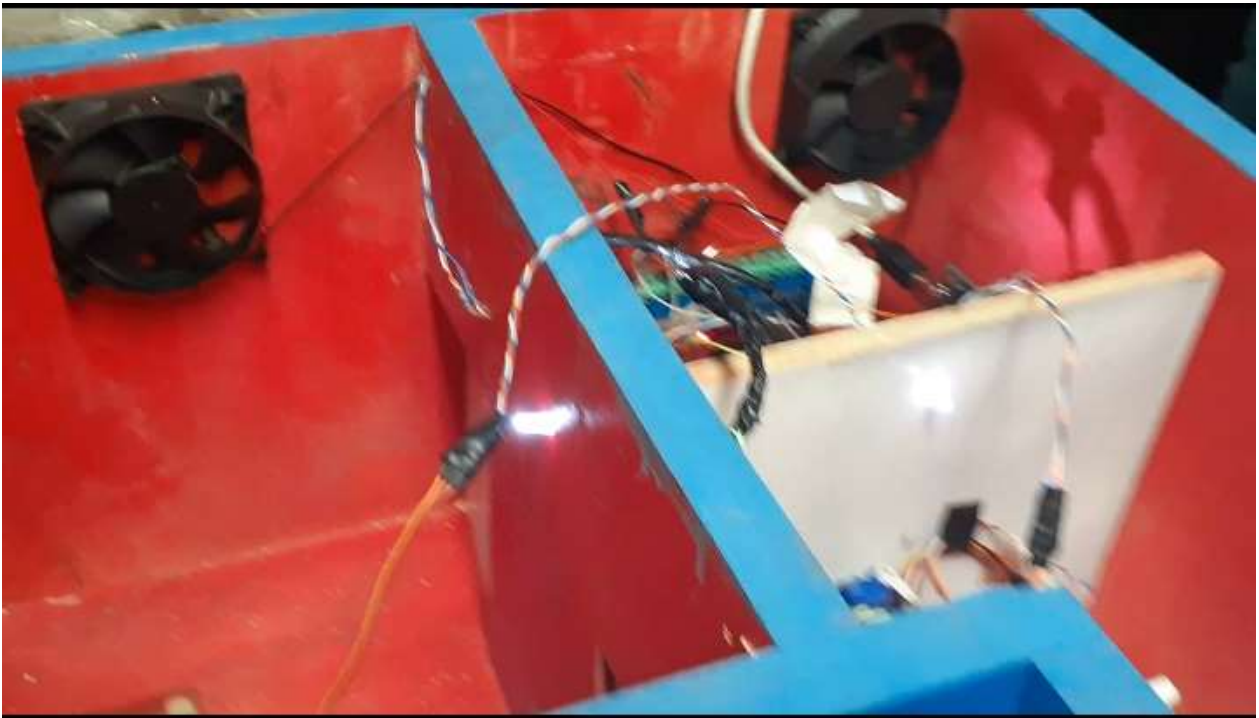


Ilustración 24 Prueba de encendido de luz interna



Ilustración 25 Prueba de encendido del aire con el sensor de temperatura

Conclusiones

- La construcción de una maqueta prototipo de una casa inteligente, integrada con elementos domóticos como placas de Arduino, sensores, actuadores, leds, etc, nos da una idea clara de cómo se pueden escalar estas propuestas hacia construcciones reales para hacer más eficientes y económicas, la gestión de los hogares modernos.
- El lenguaje de programación C++ basado en processing, nos ha resultado de mucha utilidad para probar cada componente domótico de manera individual, así como también, para probar el sistema en su conjunto.
- La creación de una aplicación móvil para el control y monitoreo de la casa inteligente, resulta muy conveniente y oportuna, debido a su facilidad de uso y a su rápido despliegue usando tecnologías de fácil acceso como App Inventor.

5 Recomendaciones

- Instalar más dispositivos de un sistema domótico, como sensores de reconocimiento de audio y facial, sensores de humo, de ruido, de humedad, etc,
- Incentivar a los estudiantes de Telemática, Sistemas y carreras afines, a seguir desarrollando trabajos de domótica con nuevas técnicas de inteligencia artificial, etc.
- Mejorar el diseño gráfico de la aplicación móvil utilizando flutter u otro framework.

6 Referencias bibliográficas

<http://www.monografias.com/trabajos35/domotica/domotica.shtml>

DOMÓTICA

Autor: PEREZ SAUCEDO JOSE RAMON REYES PADILLA KARENH GYSSEL

Fecha: 12 de julio del 2006

<http://es.scribd.com/doc/6935759/Trabajo-Domotica-Completo>

http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/sanchez_a_d

DOMÓTICA: DISEÑO DE UNA CASA INTELIGENTE BASADO EN LA TECNOLOGÍA JINI

Autor: SANCHEZ ARIAS DAVID

Fecha: 28 de mayo del 2004

http://www.cib.espol.edu.ec/Digipath/D_Tesis_PDF/D-35310.pdf

DISEÑO INMOTICO PARA AHORRO ENERGETICO, SEGURIDAD Y CONTROL DE LAS INSTALACIONES PARA EL NUEVO EDIFICIO DE LA FIEC.

Autor: COBOS FRANCO

MARIA JOSÉ LOAYZA INTRIAGO

ANDREA ALEJANDRA GARAY CONTRERAS FRANCISCO ANTONIO

Fecha: 2006

7 Anexos

Actividades	2019			
	Septiembre	Octubre	Noviembre	Diciembre
-Elección del Tema				
-Antecedentes				
-Planteamiento del problema				
-Justificación				
-Objetivos				
-Marco Metodológico				
-Marco Teórico				
-Anexos/Cronograma de actividades				

Actividades	2020 - 2021			
	Noviembre	Diciembre	Enero-febrero	Marzo-Abril
Reparar y montar la maqueta				
Organizar el documento monográfico				
Montar todos los sensores a la casa domótica				
Hacer prueba a la casa domótica				
Diseñar la aplicación de monitorización				
Hacer prueba a la casa con la aplicación				
Últimas pruebas de la casa				
Exponer la tesis				



Ilustración 26 diseño de aplicación móvil



Ilustración 27 Modelo de la casa 3D



Ilustración 28 Interior de la casa cocina



Ilustración 29 Vista de la casa segundo piso



Ilustración 30 Interior de la casa "sala"

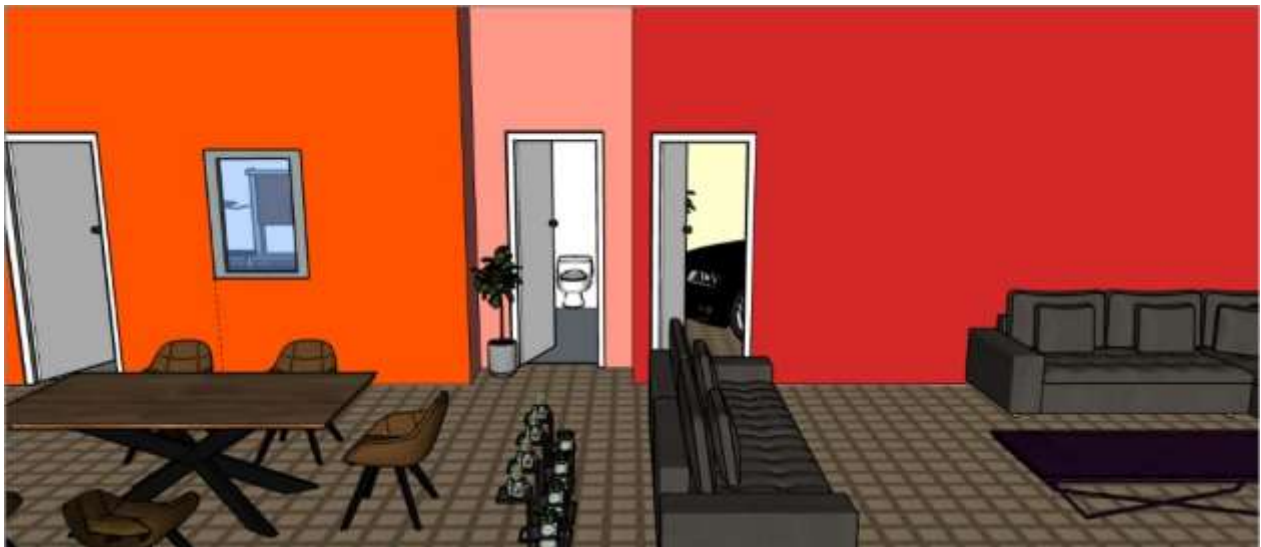


Ilustración 31 Vista de la casa interior



Ilustración 32 Inicio de configuración de la casa

7.1 Código del primer piso

```
#include <Keypad.h>
#include <Servo.h>

int Trigger=2; //PIN usado para trigger
int Echo=3; //PIN usado para echo
int Buzzer = 4; //PIN usado para buzzer / led

int t,d; //variables para valores del sensor

const byte rows = 4; //four rows since the keypad is full
const byte columns = 4; //four columns, same as above

char buttons[rows][columns] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowP[rows] = {13, 12, 11, 10}; //row pins of the keypad
byte columnsP[columns] = {9, 8, 7, 6}; //column pins of the keypad
Keypad pad = Keypad( makeKeymap(buttons), rowP, columnsP, rows, columns ); //create keypad

Servo myServo; //

void setup() {
  Serial.begin(9600); //Inicializamos la comunicación
  pinMode(Trigger,OUTPUT); //Declaramos a trigger de tipo Salida (emisor)
  pinMode(Echo,INPUT); //Declaramos a echo de tipo Entrada (receptor)
  pinMode (Buzzer, OUTPUT); //Declaramos al buzzer de tipo salida
  digitalWrite(Trigger,LOW); //Inicializamos el sensor en cero (apagado)
  myServo.attach(5); // attaches the servo on pin 9 to the servo object
}
```



```
void loop() {
delay(1000);
digitalWrite(Trip, HIGH);
delayMicroseconds(10);
digitalWrite(Trip,LOW);
t= pulseIn(Echo,HIGH); //es el tiempo que tarda el eco del ultrasonico
d= 0.01715*t; // Agregamos la formula
//Agregamos un IF para validar, si se detecta algo o alguien a una distancia menor a 10cm encendemos la alarma.
if (d < 5){
    digitalWrite (Buzzer, HIGH); // Buzzer will be on
    delay (4000); // Waitmode for 4 seconds
    digitalWrite (Buzzer, LOW); // Buzzer will be off
    delay (2000); // Waitmode for another 2 seconds in which the buzzer will be off

    char key = pad.getKey(); //get the char from the keypad
    if(key == '0')
    {
        myServo.write(110);

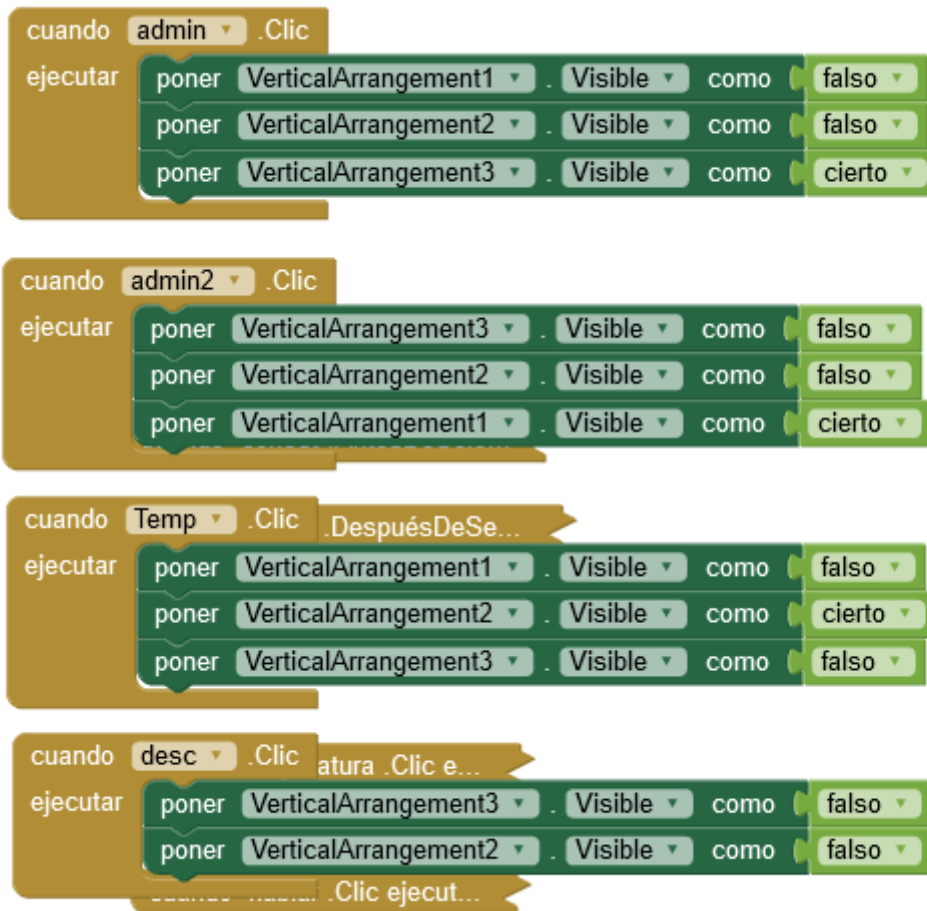
        delay(15);
    }
    if(key == '1')
    {
        myServo.write(0);
        delay(15);
    }
}
}
```

7.2 Código de aplicación en app inventor

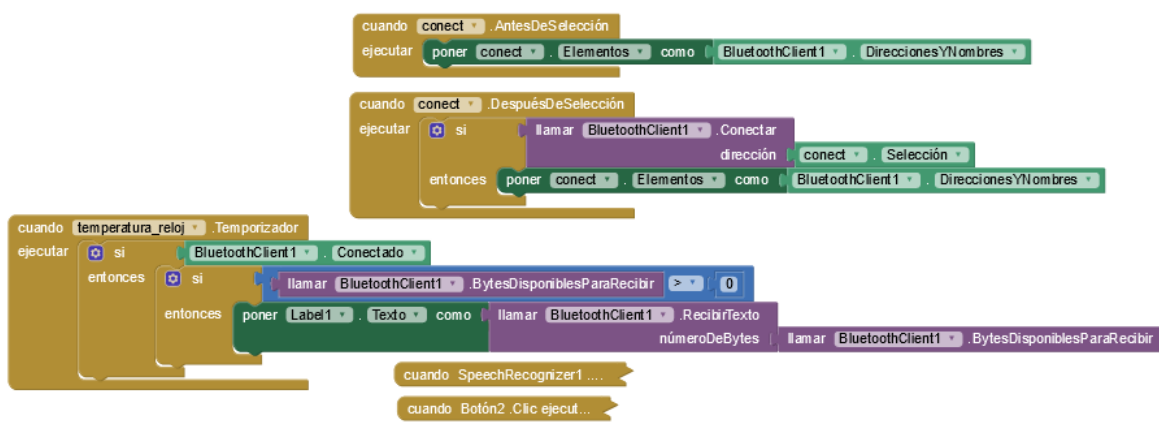
```
inicializar global bt_menu como 0

cuando Botón1 .Clic
ejecutar poner Reloj1 . TemporizadorHabilitado como cierto

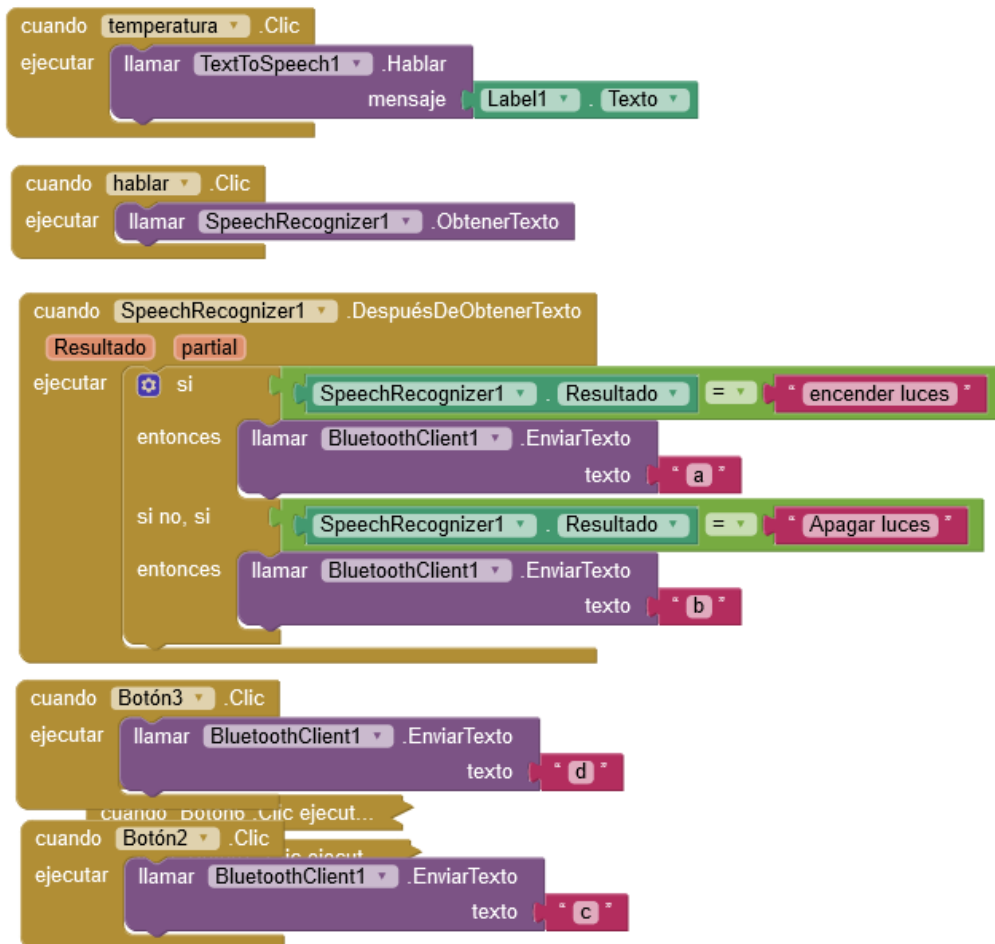
cuando Reloj1 . Temporizador
ejecutar si
  si tomar global bt_menu = 0
  entonces poner Botón1 . Texto como ←
  si menu_lateral . Ancho < Screen1 . Ancho / 2
  entonces poner menu_lateral . Ancho como menu_lateral . Ancho + 40
  sino poner Reloj1 . TemporizadorHabilitado como falso
  poner global bt_menu a 1
  si no, si tomar global bt_menu = 1
  entonces poner Botón1 . Texto como ≡
  si menu_lateral . Ancho > Screen1 . Ancho / 10
  entonces poner menu_lateral . Ancho como menu_lateral . Ancho - 80
  sino poner Reloj1 . TemporizadorHabilitado como falso
  poner global bt_menu a 0
  Clic ejecut...
```



En esta parte de código es la parte del menú desplegable



En esta parte de código está la configuración del botón conectar mediante bluetooth a la casa



Aquí está la configuración del primer, segundo piso y la forma del modelo temperatura

7.3 Prueba individual

7.3.1 Servo motor (puerta y ventana)

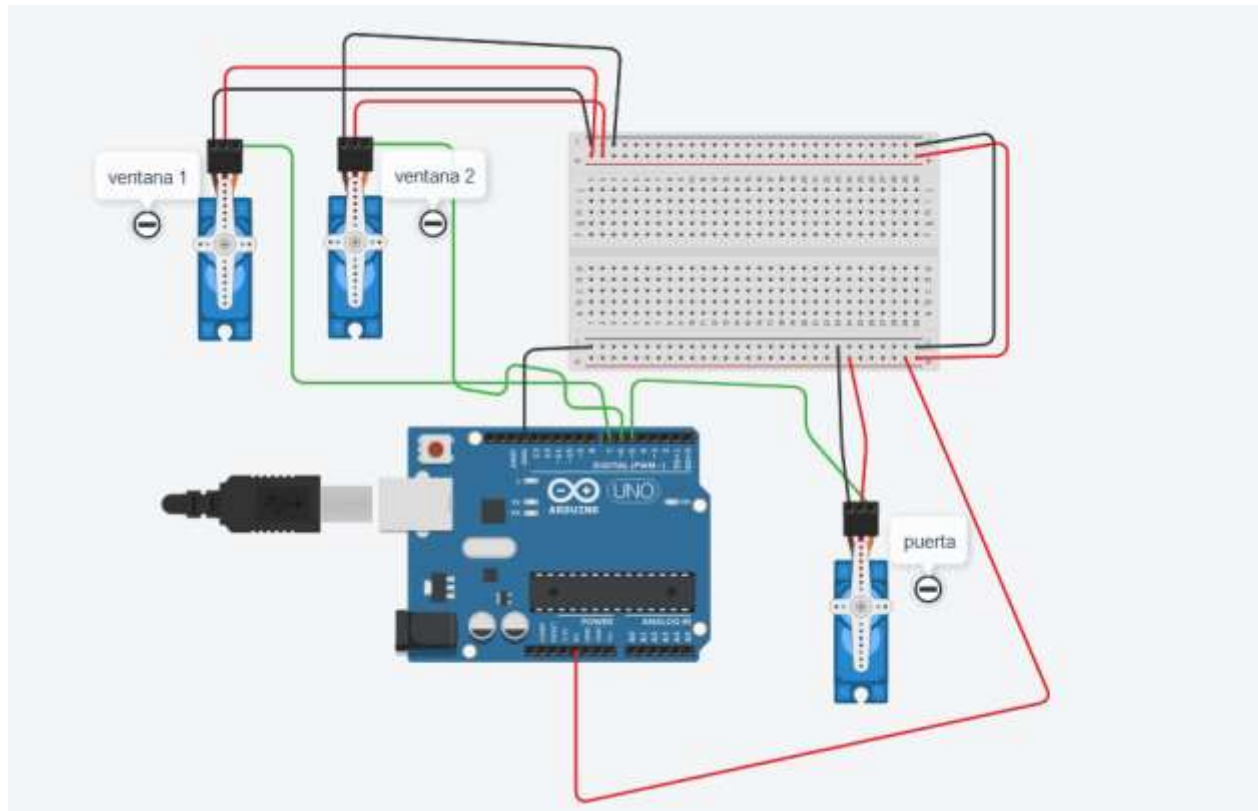


Ilustración 33 Sistema de puerta y ventana

```
#include <Servo.h>

Servo puerta;
Servo ventana;

char leer;
int vel = 0;

void setup() {
  Serial.begin(9600);
  puerta.attach(9);
  ventana.attach(8);
```

```
}  
  
void loop() {  
  
  leer = Serial.read();  
  if ((leer == '1')){  
    vel = 110;  
    puerta.write(vel);  
    delay(1500);  
  }  
  else if ((leer == '2')){  
    vel = 0;  
    puerta.write(vel);  
    delay(1500);  
  }  
  else if ((leer == '3')){  
    vel = 110;  
    ventana.write(vel);  
    delay(1500);  
  }  
  else if ((leer == '4')){  
    vel = 0;  
    ventana.write(vel);  
    delay(1500);  
  }  
}
```

7.3.2 Led (sistema de luz interna y externa)

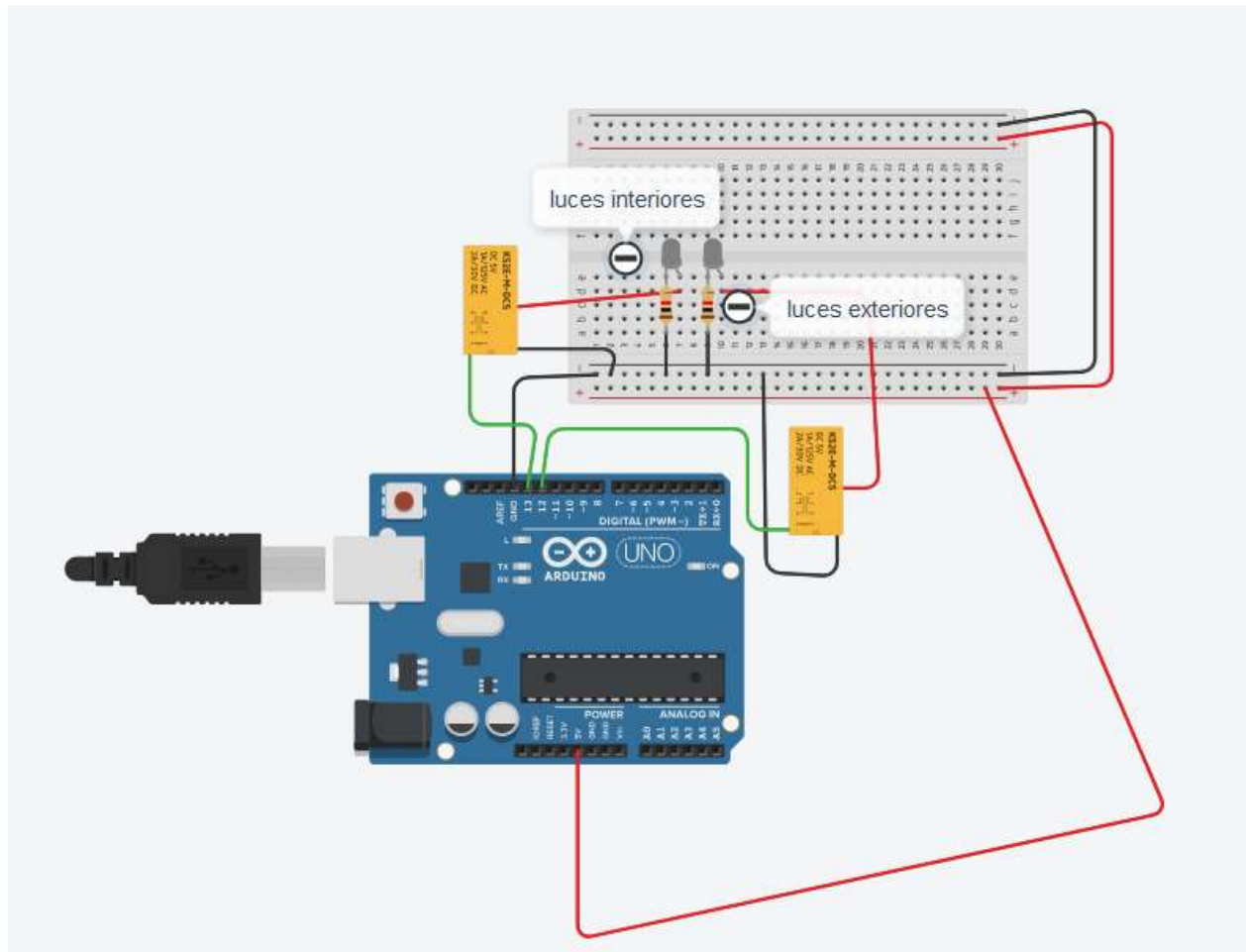


Ilustración 34 Sistema de luz interna y externa

```
//Definiciones
//Recordar que los relés se activan con nivel BAJO (0)

#define RELAY_ON 0
#define RELAY_OFF 1

char leer;

void setup () {

//Inicialización
Serial.begin(9600);

//Asegurar nivel ALTO en cada entrada de rele
```

```
digitalWrite (13, RELAY_OFF);

//Definir los pines como salida
pinMode (13, OUTPUT);

}

void loop () {

leer = Serial.read();
if ((leer == '1'))
{
digitalWrite (13, RELAY_ON); //Activa relé 1

}else if ((leer == '2')){
digitalWrite (13, RELAY_OFF); //Desactiva relé 1;

}

}
```


7.3.3 Sensor de temperatura y ventilador (sistema de aire autónomo)

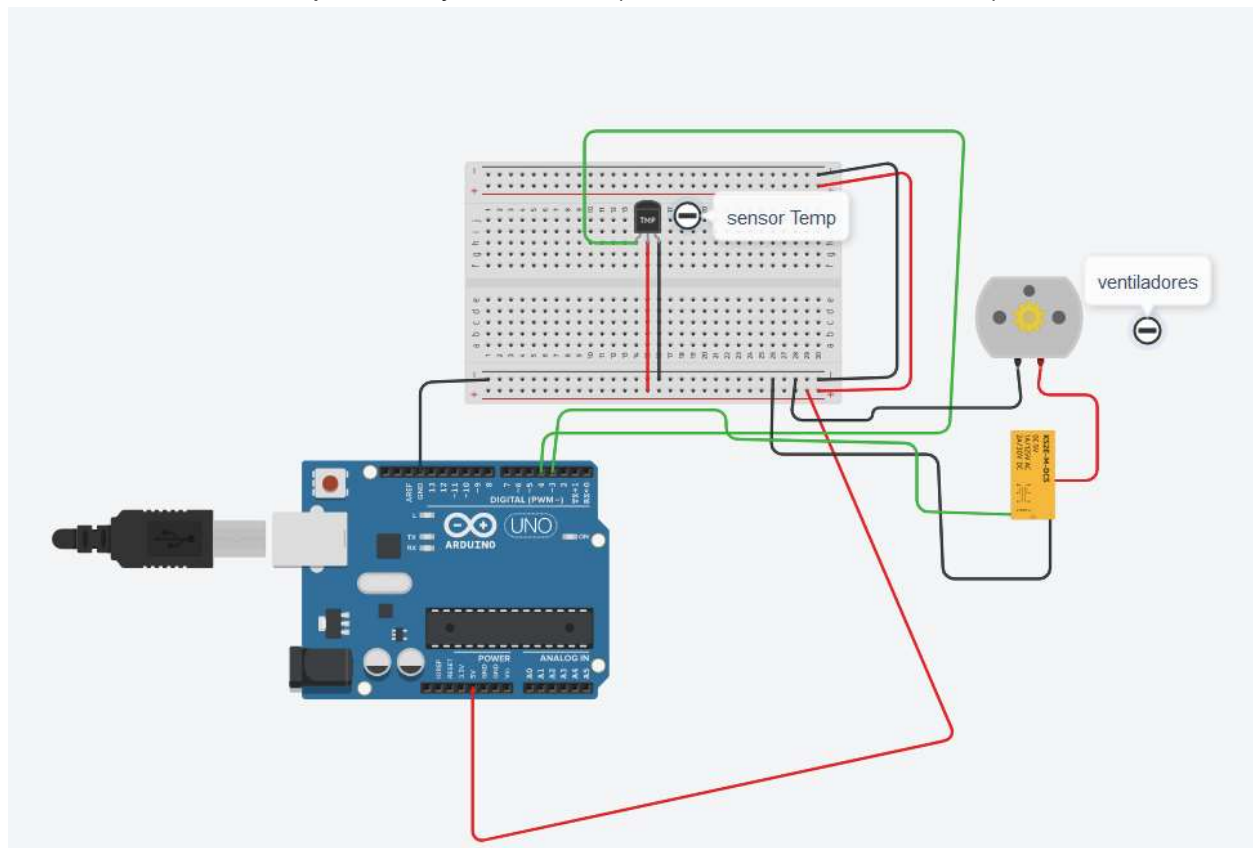


Ilustración 35 sistema de aire autónomo

```

#include <dht11.h>

#define DHT11PIN 4

#define RELAY_ON1 0
#define RELAY_OFF1 1

dht11 DHT11;

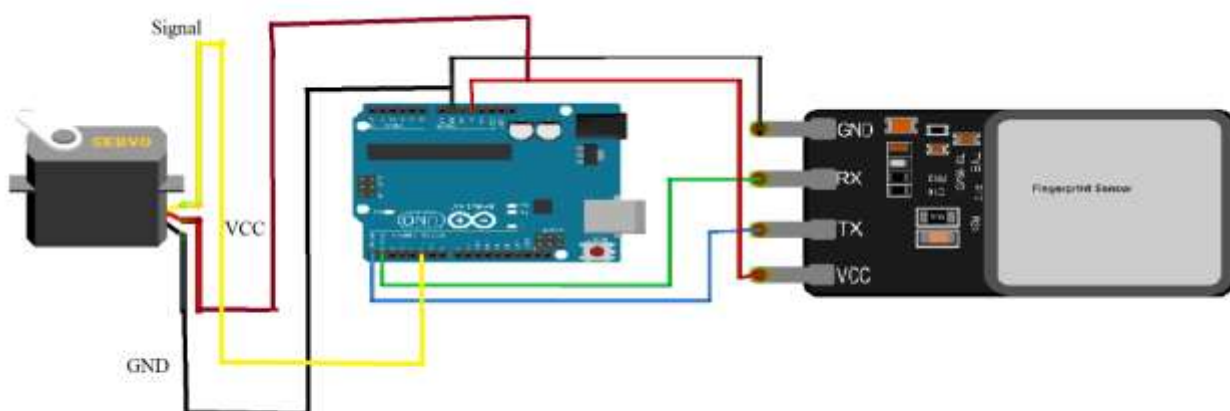
void setup()
{
  Serial.begin(9600);
  digitalWrite (3, RELAY_OFF1);
  pinMode(3, OUTPUT);
}

void loop()

```

```
{  
  int chk = DHT11.read(DHT11PIN);  
  
  if (DHT11.temperature >= 25)  
  {  
    digitalWrite(3, RELAY_ON1);  
    Serial.print("ventilador activo con Temperature (C): ");  
    Serial.println((float)DHT11.temperature, 2);  
    Serial.print("Humidity (%): ");  
    Serial.println((float)DHT11.humidity, 2);  
  }  
  else  
  {  
    digitalWrite (3, RELAY_OFF1);  
  }  
  delay(2000);  
}
```

7.3.4 Sistema de apertura de la verja de la casa (sensor de huella – servo motor)



Código del sistema de apertura de la verja de la casa mediante sensor de huella – servo motor

```
#include <Adafruit_Fingerprint.h>

#include <Servo.h>

int vel = 0;

Servo Puerta;

int Shock = 8; //define shock port

int val;//define digital variable val

// On Leonardo/Micro or others with hardware serial, use those! #0 is green wire, #1 is white
// uncomment this line:
// #define mySerial Serial1

// For UNO and others without hardware serial, we must use software serial...
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
// comment these two lines if using hardware serial

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(9600);
  pinMode(Shock, INPUT); //define shock sensor as a output port
  Puerta.attach(5);
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(100);
```

```
Serial.println("\n\nAdafruit finger detect test");

// set the data rate for the sensor serial port
finger.begin(57600);

if (finger.verifyPassword()) {
  Serial.println("Found fingerprint sensor!");
} else {
  Serial.println("Did not find fingerprint sensor :(");
  while (1) { delay(1); }
}

finger.getTemplateCount();
Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");
Serial.println("Waiting for valid finger...");
}

void loop()          // run over and over again
{
  getFingerprintIDez();
  delay(50);         //don't ned to run this at full speed.
}

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
```

```
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
    Serial.println("No finger detected");
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    return p;
case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

// OK success!

p = finger.image2Tz();
switch (p) {
case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
```

```
    Serial.println("Communication error");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}
```

```
// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);

return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) return -1;

    // found a match!
    vel = 0;
    Puerta.write(vel);
    Serial.print(" puerta abierta ");
    Serial.print("");
    delay(2000);
    vel = 110;
    Puerta.write(vel);
```

```

Serial.print(" cerrada ");

Serial.print("");

Serial.print("Found ID #"); Serial.print(finger.fingerID);

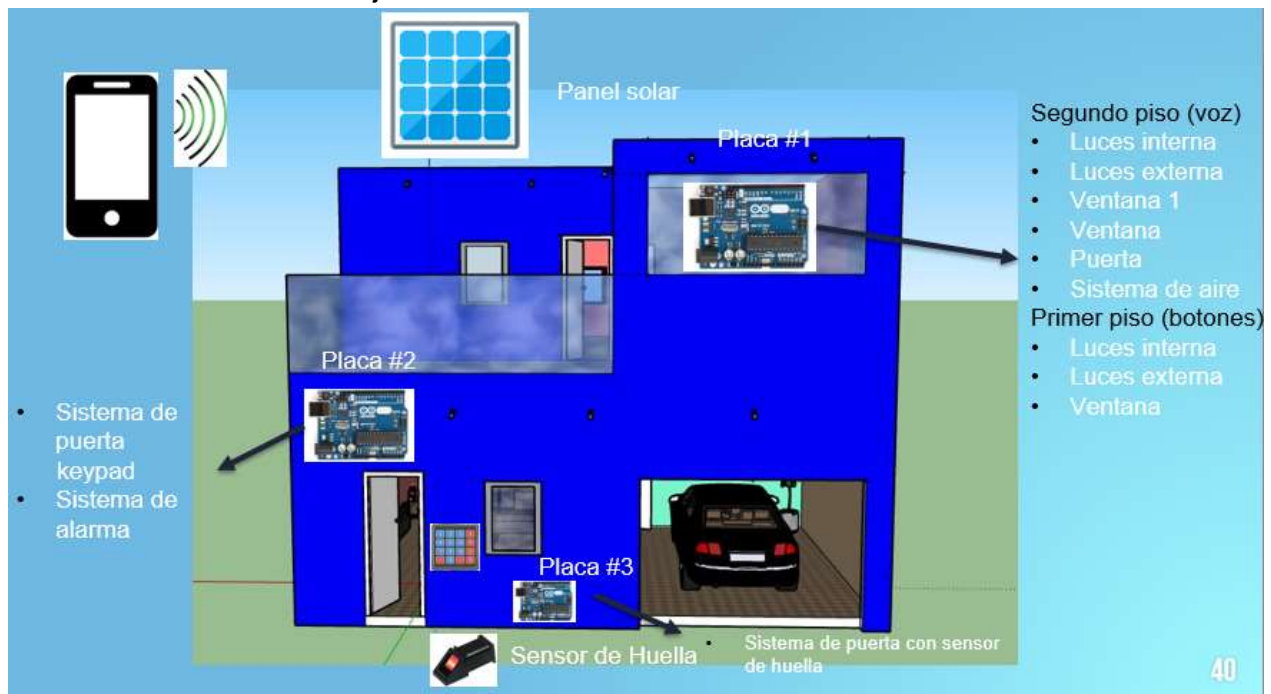
Serial.print(" with confidence of "); Serial.println(finger.confidence);

return finger.fingerID;

}

```

7.3.5 Elemento de manejo del sistema de de la casa



7.3.6 Sistema de a horro mediante el panel solar (en nuestro proyecto)

