



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA,

TELECOMUNICACIONES Y REDES

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE
CONTROL DOMÓTICO DE BAJO COSTO ACTIVADO POR
VOZ PARA PERSONAS CON DISCAPACIDAD MOTRIZ”**

TRABAJO DE TITULACIÓN: PROYECTO TÉCNICO

Para optar el Grado Académico de:

INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

AUTORES: VEINTIMILLA OCAÑA ALVARO FABIÁN

YUNGA SÁNCHEZ CARLOS ANDRÉS

TUTOR: Ing. Edwin Altamirano

Riobamba-Ecuador

2018

©2018, Alvaro Fabián Veintimilla Ocaña, Carlos Andrés Yunga Sánchez.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO
FACULTAD DE INFORMATICA Y ELECTRONICA ESCUELA DE INGENIERIA
ELECTRONICA EN TELECOMUNICACIONES Y REDES

El Tribunal de trabajo de titulación certifica que: El trabajo de titulación: **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE CONTROL DOMÓTICO DE BAJO COSTO ACTIVADO POR VOZ PARA PERSONAS CON DISCAPACIDAD MOTRIZ.**, de responsabilidad del señor Alvaro Fabián Veintimilla Ocaña y el señor Carlos Andrés Yunga Sánchez, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación quedando autorizado su presentación.

ING. WASHINGTON LUNA
DECANO DE LA FACULTAD DE
INFORMÁTICA Y ELECTRÓNICA _____

ING. FRANKLIN MORENO
DIRECTOR DE LA ESCUELA DE
INGENIERIA ELECTRÓNICA,
TELECOMUNICACIONES Y REDES _____

ING. EDWIN ALTAMIRANO
DIRECTOR DE TESIS _____

ING. ALBERTO ARELLANO
MIEMBRO DEL TRIBUNAL _____

Nosotros, Alvaro Fabián Veintimilla Ocaña y Carlos Andrés Yunga Sánchez, declaramos ser los autores del presente trabajo de titulación: ***“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE CONTROL DOMÓTICO DE BAJO COSTO ACTIVADO POR VOZ PARA PERSONAS CON DISCAPACIDAD MOTRIZ.”*** , que fue elaborado en su totalidad por nosotros, bajo la dirección del Ingeniero Edwin Altamirano, haciéndonos totalmente responsables por las ideas, criterios, doctrinas y resultados expuestos en este Trabajo de Titulación, y el patrimonio de la misma pertenece a la Escuela Superior Politécnica de Chimborazo.

ALVARO FABIÁN VEINTIMILLA OCAÑA

CARLOS ANDRÉS YUNGA SÁNCHEZ

DEDICATORIA

A Dios por haberme bendecido siempre, y a mis padres por ser el pilar principal, no solo en el ámbito académico sino en cada aspecto de mi vida, brindándome su apoyo incondicional y confianza en cada momento.

Alvaro

A mis padres por ser el principal apoyo en mi vida, por su ejemplo de superación ante las adversidades, retos que se presentan en el diario vivir y a mis maestros que gracias a sus enseñanzas supieron indicarme el camino hacia el éxito y el triunfo.

Carlos

AGRADECIMIENTO

A mi familia los cuales han sido el apoyo incondicional durante toda mi vida en especial a mis padres Fabián y María los cuales han sido mi ejemplo de superación y esfuerzo constante.

Alvaro

Agradezco principalmente a mi familia por haber brindado todo el apoyo posible durante todo este camino hacia el conocimiento y por siempre estar presentes conmigo en todo momento.

A mis amigos que fueron parte de toda esta travesía hacia la meta, entre risas, esfuerzo y dedicación hicieron que este largo camino sea lo más ameno posible.

Carlos

RESUMEN

En el presente trabajo de titulación se diseñó e implementó un prototipo de control domótico de bajo costo activado por voz para personas con discapacidad motriz, el cual permite controlar los actuadores colocados en los diferentes ambientes de una maqueta de un hogar como: baño, comedor, dormitorio y sala, por medio de una aplicación web que receipta comandos de voz, los convierte a texto, los compara con órdenes previamente establecidas y controla los eventos como: encendido-apagado de luces , apertura cierre de puerta y ventana, encendido-apagado del ventilador o calefactor. La alimentación para los actuadores son de 3.5 V para los diodos led que simulan las luces y de 9 V para el ventilador o calefactor. La aplicación web se desarrolló con lenguaje natural, para el encendido y apagado de los actuadores en los distintos ambientes del hogar. El sistema trabaja bajo una red local con acceso a internet, tanto los actuadores como la aplicación web funcionan a través de un servidor web Apache montado sobre la tarjeta de desarrollo Raspberry Pi modelo 3B. La aplicación web fue desarrollada bajo los lenguajes de programación: html, javascript, php y Python. Los actuadores que controlan los eventos de las luces son asistidos por un circuito elevador de potencia e implementados y conectados a los diversos puertos GPIO de la Raspberry Pi. El tiempo de respuesta de todos los actuadores en las distintas pruebas realizadas y en intervalo de tiempo de una hora durante 10 horas fue óptimas ya que no sufrieron cambios significativos. El prototipo es de gran ayuda para personas con alguna discapacidad motriz ya que su control depende en su mayoría de la voz evitando el movimiento limitado que poseen dichas personas. Para la aplicación web es recomendable utilizar el navegador de Google Chrome ya que presenta las mejores prestaciones para su correcto funcionamiento.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERIA>, <RASPBerry PI MODELO 3B>, <RED LOCAL>, <SERVIDOR WEB>, <APLICACIÓN WEB>, <COMANDOS DE VOZ>, <ACTUADORES>, <DISCAPACIDAD MOTRIZ>, <EVENTOS FISICOS>.

ABSTRACT

A prototype of low cost control home automation activated by voice to people with motor disability was designed and implemented in the present degree work. It allows controlling the actuators placed in the different environments of a home model such as bathroom, dining room, bedroom and living room through a web application that receives voice commands, converts them to text, compares them with previously established orders and controls events such as on-off lights, opening-closing door and window, on-off the fan or heater. The power to the actuators are 3.5 V to the LED diodes that simulate the lights and 9 V to the fan or heater. The web application was developed with natural language for turning on and off the actuators in the different home environments. The system works under a local network with internet access, both the actuators and the web application work through an Apache web server set on the Raspberry Pi development card model 3B. The web application was developed under the programming languages html, javascript, php and Python. The actuators that control the events of the lights are assisted by a circuit power elevator, implemented and connected to the various GPIO ports of the Raspberry Pi. The response time of all the actuators in the different tests carried out and in the time interval of one hour for 10 hours was optimal because they did not suffer significant changes. The prototype is of great help for people with some motor disability because their control depends mostly on the voice avoiding the limited movement that these people have. It is advisable to use the Google Chrome browser to the web application because it presents the best features to its correct functioning.

KEY WORDS: <TECHNOLOGY AND ENGINEERING SCIENCES>, <RASPBERRY PI MODEL 3B>, <LOCAL NETWORK>, <WEB SERVER>, <VOICE COMMANDS>, <ACTUATORS>, <MOTOR DISABILITY>, <PHYSICAL EVENTS>.

TABLA DE CONTENIDOS

	Páginas
DERECHOS DE AUTOR	i
CERTIFICACIÓN	ii
DECLARACIÓN DE RESPONSABILIDAD	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
RESUMEN	vi
ABSTRACT	vii
INTRODUCCIÓN	1
ANTECEDENTES	2
FORMULACIÓN DEL PROBLEMA	3
JUSTIFICACIÓN TEÓRICA	3
JUSTIFICACIÓN APLICATIVA	4
OBJETIVOS	5
CAPITULO I	6
1. MARCO TEORICO	6
1.1 Domótica	6
1.1.1 Aplicaciones	7
1.1.2 El sistema domótico	8
1.1.3 Ámbitos de la domótica	10
1.1.4 Plataformas del sistema domótico	11
1.1.5 Fabricantes de sistemas domóticos	14
1.2 Controladores de propósito general	16
1.2.1 Raspberry Pi	16
1.2.1.1 Características	17
1.2.1.2 Modelos de la placa Raspberry Pi	17
1.2.1.3 Componentes de la Raspberry Pi	18
1.2.2 Arduino	21
1.2.2.1 Hardware	21
1.2.2.2 Componentes de la placa Arduino Uno	21

1.2.2.3	<i>Comparativa de la placa Arduino</i>	23
1.2.2.4	<i>Software</i>	25
1.2.2.5	<i>Entorno Arduino</i>	25
1.2.3	<i>Otras tarjetas de desarrollo</i>	26
1.2.4	<i>Arduino vs Raspberry Pi</i>	30
1.3	Sistema operativo Raspbian	32
1.3.1	<i>El Sistema Operativo Debian</i>	32
1.3.2	<i>Software de código abierto</i>	33
1.3.3	<i>Componentes de Raspbian</i>	33
1.4	Reconocimiento de voz	36
1.4.1	<i>Tamaño del vocabulario de reconocimiento de voz</i>	37
1.4.2	<i>Formas de habla usuario-máquina</i>	37
1.4.3	<i>Caracterización de los sistemas de reconocimiento de voz</i>	38
1.4.4	<i>Aplicaciones para el reconocimiento de voz</i>	38
1.4.5	<i>Conocimiento de los patrones de voz del usuario</i>	42
1.4.6	<i>Grado de diálogo usuario-máquina</i>	42
1.4.7	<i>Grado de conocimiento acústico-lingüístico del sistema</i>	42
CAPITULO II		43
2	MARCO METODOLÓGICO	43
2.1	Diseño del prototipo del control domótico.	43
2.1.1	<i>Estructura del Prototipo</i>	44
2.1.2	<i>Etapa de adquisición de datos</i>	44
2.1.3	<i>Etapa de interpretación de datos</i>	44
2.1.4	<i>Etapa de control de los puertos GPIO</i>	45
2.1.5	<i>Etapa de control de los actuadores</i>	45
2.2	Etapa Software.	46
2.2.1	<i>Configuración de la RASPBERRY PI</i>	46
2.2.1.1	<i>Instalación del sistema operativo Raspbian en la Raspberry pi</i>	46
2.2.1.2	<i>Conexión a través del protocolo SSH</i>	47
2.2.1.3	<i>Instalación y configuración del servidor WEB</i>	48
2.2.1.4	<i>Certificado SSL</i>	49
2.2.1.5	<i>Desarrollo de la aplicación WEB</i>	49
2.3	Etapa de Hardware	50
2.3.1	<i>Diseño de la maqueta</i>	50
2.3.2	<i>Diseño del circuito del prototipo</i>	55

2.3.4	<i>Elementos Hardware</i>	57
2.3.4.1	<i>Instalación de los Diodos Led</i>	57
2.3.4.2	<i>Instalación ventilador</i>	59
2.3.4.3	<i>Instalación Calefactor</i>	60
2.3.4.4	<i>Instalación Puerta</i>	61
2.3.4.5	<i>Instalación ventana</i>	63
2.4	Funcionamiento	64
CAPÍTULO III		69
3	PRUEBAS, COSTOS Y RESUSLTADOS	69
3.1	Pruebas de funcionamiento	69
3.1.1	<i>Prueba luces</i>	69
3.1.2	<i>Prueba ventilador</i>	71
3.1.3	<i>Prueba calefactor</i>	71
3.1.4	<i>Prueba Puerta</i>	72
3.1.5	<i>Prueba ventana</i>	72
3.2	Resultados	73
3.2.1	<i>Confiabilidad del prototipo</i>	74
3.2.2	<i>Tiempo de respuesta del prototipo</i>	77
CONCLUSIONES		82
RECOMENDACIONES		83
GLOSARIO		
BIBLIOGRAFÍA		

ÍNDICE DE TABLAS

	Páginas
Tabla 1-1. Ámbitos de la domótica	10
Tabla 2-1. Plataformas del Sistema Domótico.....	12
Tabla 3-1. Fabricantes de Sistemas Domóticos.....	15
Tabla 4-1: Modelos de la placa Raspberry Pi.	18
Tabla 5-1: Comparativa de la placa Arduino.	23
Tabla 6-1: Otras tarjetas de desarrollo	27
Tabla 7-1. Arduino vs Raspberry Pi.....	30
Tabla 8-1. Aplicaciones para el reconocimiento de voz	38
Tabla 1-2. Materiales utilizados.	51
Tabla 1-3 Costos materiales del prototipo.....	73
Tabla 2-3 Costos materiales del prototipo.....	75
Tabla 3-3 Tiempo de respuesta de encendido y apagado a 2 m.	78
Tabla 3-4 Tiempo de respuesta de encendido y apagado a 15 m.	79
Tabla 3-5 Tiempo de respuesta de encendido y apagado a 2 m.	80
Tabla 3-6 Tiempo de respuesta de encendido y apagado a 15 m.	81

ÍNDICE DE FIGURAS

	Páginas
Figura1-1. Domótica en el hogar	6
Figura 2-1. Aplicaciones de la domótica	7
Figura 3-1. Sistema domótico	8
Figura 4-1: Raspberry Pi 3	18
Figura 5-1: Distribución de los pines del GPIO.....	21
Figura 6-1: Ubicación del GPIO en la placa Raspberry Pi	21
Figura 7-1: Arduino Uno	22
Figura 8-1: Plataformas que soporta el Software Arduino	26
Figura 9-1: Interfaz shell.....	36
Figura 10-1: Escritorio de Raspbian	37

ÍNDICE DE GRÁFICOS

	Páginas
Gráfico 1-1: Esquema del prototipo a desarrollar.	4
Gráfico 2-1: Componentes de la Raspberry Pi.....	19
Gráfico 3-1: Componentes de la placa Arduino Uno.....	22
Gráfico 4-1: Entorno Arduino.....	26
Gráfico 1-2: Diagrama de bloques general del prototipo.....	43
Gráfico 2-2: Diagrama de bloques adquisición de datos.	44
Gráfico 3-2: Diagrama de bloques interpretación de datos.....	45
Gráfico 4-2: Diagrama de bloques control de puertos GPIO.....	45
Gráfico 5-2: Diagrama de bloques control de los actuadores.	46
Gráfico 6-2: Diagrama de bloques general del prototipo.....	47
Gráfico 7-2: Conexión de forma remota SSH.....	48
Gráfico 8-2: Dirección ip estática asignada.	48
Gráfico 9-2: Pagina web por defecto de Apache.	49
Gráfico 10-2: Verificación de HTTPS y certificado SSL.	49
Gráfico 11-2: Interfaz gráfica de la aplicación web.....	50
Gráfico 12-2: Diseño Maqueta.....	51
Gráfico 13-2: Circuito general.	56
Gráfico 14-2: Circuito de potencia diodo led.....	57
Gráfico 15-2: Circuito Potenciador de intensidad de corriente.....	58
Gráfico 16-2: Conexiones Diodo led.	58
Gráfico 17-2: Cableado diodos led.	59
Gráfico 18-2: Cableado recubierto en la maqueta.....	59
Gráfico 19-2: Circuito Ventilador.....	59
Gráfico 20-2: Ventilador.....	60
Gráfico 21-2: Circuito calefactor.	61
Gráfico 22-2: Calefactor.	61
Gráfico 23-2: Motor con piñón.	61
Gráfico 24-2: Pieza puerta.	62
Gráfico 25-2: Circuito motor puerta.	62

Gráfico 26-2: Motores de la ventana.....	63
Gráfico 27-2: Montaje de la ventana.....	64
Gráfico 28-2: Circuito motores ventana.....	64
Gráfico 29-2: Activación lenguaje y micrófono.	65
Gráfico 30-2: Evento onresult.....	66
Gráfico 31-2: Elementos input.....	66
Gráfico 32-2: Refresca la aplicación web.	67
Gráfico 33-2: Comando de envió automático.	67
Gráfico 34-2: Ordenes establecidas del fichero procesa.php.	68
Gráfico 35-2: Fichero prendedormitorio.py.....	68
Gráfico 36-2: Diagrama de flujo del prototipo.	69
Gráfico 1-3: Encendido y apagado de la luz del dormitorio.	70
Gráfico 2-3: Encendido y apagado de la luz del baño.	71
Gráfico 3-3: Encendido y apagado de la luz del comedor.	71
Gráfico 4-3: Encendido y apagado de la luz de la sala.	72
Gráfico 5-3: Encendido y apagado del ventilador.....	72
Gráfico 6-3: Encendido y apagado del calefactor.	73
Gráfico 7-3: Apertura y cerradura de la puerta.	73
Gráfico 8-3: Apertura y cerradura de la ventana.....	74
Gráfico 9-3: Facilidad de manejo del prototipo.	77
Gráfico 10-3: Eficacia del prototipo.	77
Gráfico 11-3: Utilidad del prototipo.	78
Gráfico 12-3: Señal digital apagado GPIO.	83
Gráfico 13-3: Señal digital encendido GPIO.	84

INDICE DE ANEXOS

- Anexo A:** Tarjeta de desarrollo Raspberry Pi
- Anexo B:** Modulo L298N
- Anexo C:** Interfaz gráfica raspbian
- Anexo D:** Terminal raspbian
- Anexo E:** Pruebas diodos led
- Anexo F:** Planos maqueta
- Anexo G:** Instalación luces
- Anexo H:** Instalación ventilador
- Anexo I:** Instalación calefactor
- Anexo J:** Maqueta terminada
- Anexo K:** Conexiones de los actuadores
- Anexo L:** Case circuito
- Anexo M:** Diagrama del circuito
- Anexo N:** Aplicación web
- Anexo Ñ:** Instalación sistema operativo Raspbian
- Anexo O:** Instalación conexión SSH
- Anexo P:** Instalación y configuración servidor del servidor WEB
- Anexo Q:** Instalación y configuración certificado SSL
- Anexo R:** Desarrollo de la aplicación WEB
- Anexo S:** Fichero index.html
- Anexo T:** Fichero webspeech.js
- Anexo U:** Fichero procesa.php
- Anexo V:** Ficheros .py
- Anexo W:** Solicitud de presupuesto, para casa domótica
- Anexo X:** Presupuesto del prototipo

INTRODUCCIÓN

El crecimiento abismal del servicio de internet en el transcurso de los años ha obligado a las empresas ofertar cada vez una mayor cobertura, esto ha sido el motor principal para que se desarrollen diferentes productos capaces de conectarse a la red, con la finalidad de brindar control en casi todos los aspectos del hogar; para ello la domótica brinda las herramientas necesarias que nos permiten una interacción del usuario con su casa, y así brindar mayor seguridad y confort; sin embargo muchos de estos beneficios no son orientado a personas con algún tipo de discapacidad motriz, por lo cual existe la necesidad de desarrollar un entorno adaptable para dicho grupo de personas.(Vargas, 2016)

Planteado esto, se ha visto la necesidad de diseñar y a la vez implementar un prototipo domótico de bajo costo, el cual servirá como modelo para proyectos futuros con el fin de ayudar a mejorar el estilo de vida de las personas con algún tipo de discapacidad motriz.

Por lo cual el presente escrito consta de tres capítulos documentando el diseño e implementación de un prototipo de control domótico de bajo costo activado por voz para personas con discapacidad motriz con el objetivo de controlar la mayoría de elementos que habitualmente se encuentran en una habitación (luces, calefacción, ventilación, puertas y ventanas), convergiendo en una red inalámbrica bajo el standard IEEE 802.11, y cuyo control basado en comandos de voz.

Para poder controlar el sistema se utilizó comandos de voz, los cuales mediante una aplicación web utilizan los ficheros de Web Speech API (Aplicación de código abierto) específica de Google Chrome, activa el micrófono del terminal y traduce la voz a texto facilitando el procesamiento de los comandos. El Capítulo I puntualiza los conceptos desarrollados en el presente trabajo de titulación tales como: Domótica, Controladores de propósito general, Sistema Operativo Raspbian y Reconocimiento de voz. El Capítulo II describe los procedimientos llevados a cabo para la programación del software y montaje del hardware correspondiente al prototipo domótico. En el Capítulo III se presentan las pruebas, costos y resultados correspondientes a la evaluación del rendimiento del sistema.

ANTECEDENTES

Las innovaciones tecnológicas siempre han sido aplicadas y utilizadas en las viviendas. Recientemente la domótica, o el uso y adopción de las nuevas tecnologías de la información y la comunicación en el hogar, está empezando a inducir cambios en el uso y la función de la vivienda, acentuando las alteraciones en la percepción del espacio-tiempo que ya se detectan en otras instancias de la vida cotidiana.(Chaparro, 2003)

La Historia de la domótica inició con X10 en 1975, creado para el telecontrol y basado en corrientes portadoras o Power Line (PL). Este sistema de protocolo estándar se extendió mucho por Estados Unidos y en Europa (sobretudo Reino Unido y España). Estándares como KNX, Insteon, X10, etc permitieron múltiples posibilidades, pero son más costosos de integrar. Pero con el pasar de los años la domótica ha empezado a interesar al usuario y a grandes compañías como Google o Apple.(Domoprac, 2009)

Los aspectos claves para el futuro de la domótica en el hogar son el uso de asistentes de voz y la inteligencia de los dispositivos. La llegada del uso de los asistentes de voz a la domótica, más allá de simples comandos ya predefinidos, llegará los años próximos con HomeKit y Siri, Google Now y también Cortana de Microsoft. (Santamaria, 2014)

Debido al crecimiento tecnológico que se evidencia en la ciudad de Riobamba, la provincia y el país, surge la necesidad de crear recursos que potencialicen las características de la domótica, no cabe duda que estará presente en nuestra vida cotidiana incluso para cosas tan sencillas como prender la luz, abrir puertas y ventanas, regular la calefacción, entre otros. La nueva generación tecnológica cuenta con una infinidad de sensores, los cuales nos permitirán que todos los dispositivos se conecten a la red.

Actualmente según el CONADIS en Riobamba el 1.8 % de la población registrada posee alguna discapacidad motriz por tal motivo hemos planteado un control domótico activado por voz para las personas mencionadas.

El presente trabajo se pretende dar una opción de control domótico controlado por voz para personas con alguna discapacidad motriz, debido a que les resulta complejo movilizarse de un lado a otro para realizar tareas aparentemente sencillas como encender las luces, abrir o cerrar puertas, abrir o cerrar ventanas, controlar el encendido del ventilador y calefactor.

FORMULACIÓN DEL PROBLEMA

¿Cómo se puede realizar la activación de eventos físicos mediante el uso de un prototipo de control domótico activado por voz para personas con deficiencia motriz?

JUSTIFICACIÓN TEÓRICA

Según el trabajo de titulación del repositorio de la Universidad técnica de Valencia “Diseño de un sistema de control domótico basado en la plataforma Arduino”. El cual detalla los conocimientos básicos para entender que es y cómo funciona un sistema domótico y cómo utilizando el hardware libre de Arduino se puede crear un sistema estable con un presupuesto muy inferior al de las viviendas de alta categoría.

Debido al giro que viene dando en la actualidad el control domótico en la infraestructura de viviendas y edificios viendo la necesidad de las personas con alguna discapacidad motriz, se familiaricen de un modo más práctico y sencillo

Por lo cual se propone realizar un diseño funcional de un control domótico de bajo costo activado por voz para personas con alguna discapacidad motriz de esta forma se busca mejorar el estilo de vida de este grupo de personas usando los avances tecnológicos que se ha venido desarrollando en los últimos años, ya que el principal objetivo es la inclusión.

El funcionamiento constará de 4 bloques previos bien definidos antes de llegar a los actuadores (Figura 1-1):

- Bloque 1.- Este bloque (Bloque de interconexión) tiene como función brindar conexión hacia la red para la convergencia de los dispositivos.
- Bloque 2.- Este bloque (Bloque matriz) tiene como función alojar toda la programación que sea necesaria para que el sistema funcione de forma eficiente.

- Bloque 3.- Este bloque (Bloque de acceso) tiene como función brindar acceso mediante una interfaz compatible con Android para el control de voz.
- Bloque 4.- Este bloque (Bloque de transición) tiene como función convertir todas las señales emitidas por el bloque matriz a una señal eléctrica que accionara los distintos actuadores.
- Finalmente, el usuario será capaz de controlar dichos actuadores sin la necesidad de moverse.

JUSTIFICACIÓN APLICATIVA

En los últimos años si bien es cierto la tecnología ha ido avanzando constantemente no podemos dejar a un lado el hecho de que muchos de los prototipos están enfocados en general a usuarios comunes sin tomar en cuenta a los usuarios con capacidades especiales según el CONADIS en el Ecuador en la ciudad de Riobamba existen un aproximadamente de 2632 personas con discapacidad motriz registrados.

Actualmente la tecnología ha evolucionado al punto de poder controlar varios elementos remotamente desde diferentes terminales (Computador, celulares, tablets, etc.) utilizando el estándar IEEE 802.11.

En el esquema propuesto podemos observar el funcionamiento del control domótico el cual consta de los siguientes procesos.

1. Conectar la red local.
2. Ingresar a la interfaz desde un dispositivo móvil
3. Controlar los diferentes terminales mediante comandos de voz.

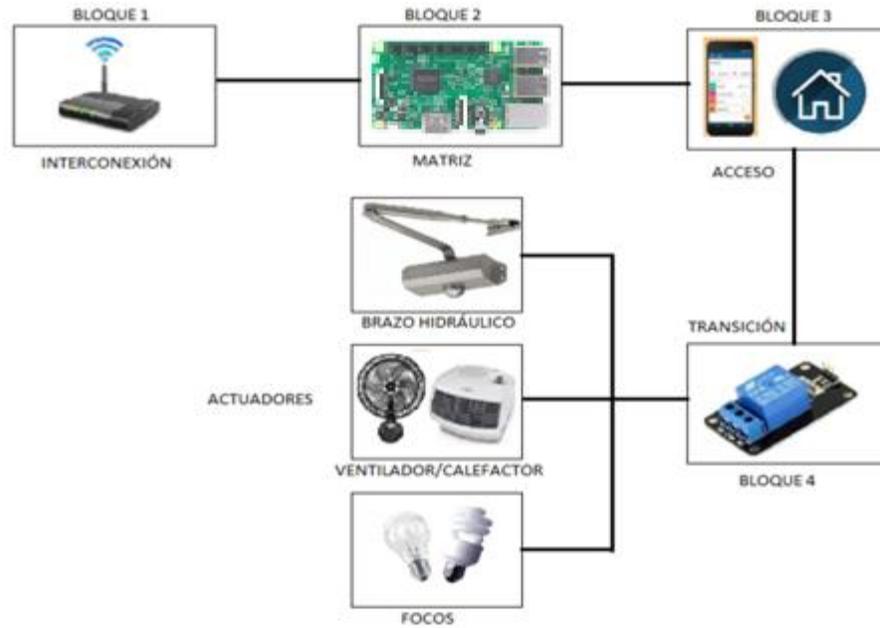


Gráfico 1-1: Esquema del prototipo a desarrollar.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

OBJETIVOS

OBJETIVO GENERAL

Diseñar e implementar un prototipo de control domótico de bajo costo activado por voz para personas con discapacidad motriz.

OBJETIVOS ESPECIFICOS

- Estudiar las plataformas utilizadas para el control domótico en la actualidad y los módulos de comunicación para el control domótico.
- Diseñar la activación de eventos físicos mediante voz para el prototipo de control domótico.
- Implementar una interfaz hardware-software para el control de eventos físicos mediante la activación de voz.
- Evaluar los datos obtenidos al inicio de la implementación y después de un tiempo determinado para el análisis comparativo.

CAPITULO I

1. MARCO TEORICO

1.1 Domótica

El término domótica proviene de la fusión de las palabras *domus* (casa en latín) e informática (DRAE). La domótica surgió en la década de los setenta, tras un sin número de investigaciones, que dieron paso a los primeros dispositivos de automatización, los cuales fueron implementados en sus inicios en edificaciones bajo el protocolo de comunicación X-10.

Definimos a la domótica como una agrupación de sistemas, que proporcionan las herramientas necesarias para automatizar una vivienda, como se muestra en la Figura 1-1.

Los aspectos principales de la domótica son los siguientes:

- Gestión de energía
- Seguridad
- Bienestar
- Comunicación



Figura1-1. Domótica en el hogar

Fuente: <https://www.trisacor.com/es/domotica-la-tecnologia-hogar/>

Un sistema domótico puede integrarse ya sea por medio de redes interiores como exteriores de comunicación, cableadas o inalámbricas, desde el interior como fuera del hogar, un sistema domótico que tenga acceso a internet puede ser controlada desde cualquier parte del mundo dependiendo de las necesidades del usuario. (Falconi & Jimenez, 2009)

1.1.1 Aplicaciones

En la actualidad la domótica tiene un sin número de aplicaciones, en la Figura 2-1 se detalla las más relevantes:

- Confort
- Seguridad
- Ahorro de energía
- Comunicación
- Accesibilidad
- Eficacia



Figura 2-1. Aplicaciones de la domótica

Fuente: <https://sites.google.com/a/correo.unimet.edu.ve/fgtce0410-02-tgi/domotica-u-hogar-inteligente/aplicaciones-de-la-domotica>

- **Seguridad.-** Es una de las principales aplicaciones de la domótica, ya que a más de brindar seguridad a los bienes materiales también brinda seguridad a los habitantes en el hogar.
- **Confort.-** Esta aplicación se ha ido desarrollando debido al incremento de la población de personas de la tercera edad y personas con algún tipo de discapacidad permitiendo controlar la vivienda desde un solo punto y con el menor esfuerzo posible, realizando tareas como:
 - ✓ Regulación de la temperatura.
 - ✓ Control de puertas y ventanas
 - ✓ Apagado y encendido de luces
 - ✓ Encendido y apagado de dispositivos conectados a la red eléctrica
- **Ahorro de energía.-** Los sistemas de control permiten manipular remotamente varios aspectos en relación a la red eléctrica como pueden ser:

- ✓ Optimizar el sistema de calefacción por habitaciones
 - ✓ Intensidad de la luz de acuerdo a la luz ambiental
 - ✓ Desactivación de grifos después de un periodo de inactividad
 - ✓ Uso de electrodomésticos en horas de menor tarifa
- **Comunicaciones.-** Consiste en el intercambio de información entre el usuario y los distintos aparatos electrónicos, sea dentro o fuera de la vivienda.
 - **Accesibilidad.-** Podemos acceder a los dispositivos desde cualquier parte dentro o fuera de la habitación mientras esté conectado el usuario a la red principal encargada de controlar el sistema. (Camargo, García, & Gaona, 2012)

1.1.2 El sistema domótico

El sistema domótico está compuesto de una familia de elementos que detectan los cambios de estado en una variable y a la vez, enviar señales a todos los dispositivos del sistema encargados de llevar a cabo las órdenes siguiendo las reglas establecidas por el usuario, en la Figura 3-1 podemos apreciar los componentes de un sistema domótico.

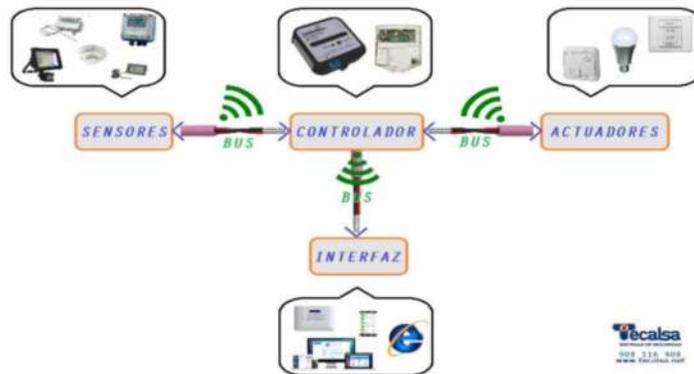


Figura 3-1. Sistema domótico

Fuente: http://tecalsa.net/wp-content/uploads/2015/09/componentes_domotica_tecalsa_alarmas_baratas_para_casa_21.png

Un sistema domótico está compuesto por:

- Sensores
- Actuadores
- Unidades de control
- Interfaces de usuario

- Pasarelas de comunicación

- **Sensores.-** Tienen como función detectar los cambios que presentan las variables y al mismo tiempo transmitirlos al bloque de control para ser procesadas. El uso de los distintos sensores va a depender del tipo de control que requiera el usuario o de los parámetros a medir. Los sensores se clasifican en:
 - ✓ De luminosidad.
 - ✓ De temperatura.
 - ✓ Volumétricos de presencia.
 - ✓ Detectores de incendios.
 - ✓ Detectores de inundación
 - ✓ Detectores de gas.
 - ✓ Detectores de puertas y ventanas abiertas.
 - ✓ Anemómetros.
 - ✓ Interruptores de lluvia.
 - ✓ Contadores de energía.
 - ✓ Otros.

- **Actuadores.-** Se comportan como interfaces de potencia, son los encargados de llevar a cabo las acciones mecánicas del sistema, se clasifican en dos grupos: Según el tipo de señal de entrada y según la magnitud controlada.
- **Unidades de control.-** Es el elemento encargado de procesar todas las señales emitidas por los sensores, y así poder emitir las órdenes a cada uno de los actuadores; se dice que es el elemento que posee cierto grado de inteligencia en el sistema.
- **Interfaces de usuario.-** Es el medio por el cual el usuario puede interactuar con el sistema domótico, debe ser lo más simple posible para que el usuario se sienta a gusto con su entorno.
- **Pasarelas de comunicación.-** Es el dispositivo encargado de brindar la conexión entre la infraestructura de comunicaciones de la vivienda y una red de datos; es el responsable de controlar los elementos desde cualquier parte de la vivienda. (Alvarez & Cajas, 2016)

1.1.3 Ámbitos de la domótica

A continuación se describe los principales ámbitos de la domótica en la actualidad

Tabla 1-1. Ámbitos de la domótica

ÁMBITOS	DESCRIPCIÓN
Del ahorro energético	<ul style="list-style-type: none">• Programación y zonificación de la climatización.• Racionalización de cargas eléctricas: desconexión de equipos de uso no prioritario en función del consumo energético en un momento dado. Reduce la potencia contratada.• Gestión de tarifas, derivando el funcionamiento de algunos aparatos a horas de tarifa reducida.
De nivel de confort	<ul style="list-style-type: none">• Apagado general de todas las luces de la vivienda.• Automatización del apagado/encendido en cada punto de luz.• Regulación de la iluminación según el nivel de luminosidad ambiente.• Automatización de todos los distintos sistemas/ instalaciones /equipos dotándolos de control eficiente y de fácil manejo.

	<ul style="list-style-type: none"> • Integración del portero al teléfono, o del video portero al televisor. • Reproducción de audio y video.
De la protección familiar y patrimonial	<ul style="list-style-type: none"> • Detección de posible intruso. • Simulación de presencia. • Detección de conatos de incendio, fugas de gas, escapes de agua. • Alerta médica. • Tele asistencia. • Cerramiento de persianas puntual y seguro.
En comunicaciones	<ul style="list-style-type: none"> • Control remoto. • Comunicación mediante internet • Transmisión de alarmas. • Intercomunicaciones. • Controlador mediante mando a distancia

Fuente: <http://hdl.handle.net/2099.1/2859>

1.1.4 Plataformas del sistema domótico

En la actualidad existe un sin número de plataformas dedicadas a la domótica, en la Tabla 1-1 se detallan algunos de los más importantes.

Tabla 2-1. Plataformas del Sistema Domótico

NOMBRE	LOGO/IMAGEN	DESCRIPCIÓN
Vivint Sky		<p>Vivint Sky es una plataforma desarrollada por Vivint que pretende ofrecer todo un sistema completo de control para dispositivos del hogar tan variados como las luces, termostato, cierre de puertas, sistemas de seguridad, etc. Todo ello controlable desde nuestro smartphone, tablet u ordenador. Y con la capacidad de aprender de los comportamientos del usuarios.(Santamaría, 2014)</p>
Domoticz		<p>Domoticz es un software libre de control domótico disponible para las plataformas Windows y Linux y caracterizado por consumir muy pocos recursos del sistema lo que lo convierten en una solución muy interesante si queremos combinarlo con un algún controlador Low Cost como puede ser una Raspberry Pi.(Vega, 2017)</p>
Jeedom		<p>Jeedom es un software francés mucho más moderno en cuanto a sus planteamientos y que se ha convertido en una solución relativamente conocida dentro del mercado doméstico europeo. Este software es parte de una solución completa e integrada de la compañía que lo desarrolla y gracias a la cual podríamos comprar un dispositivo con el software instalado y listo para funcionar. Sin embargo,</p>

		<p>el software es libre y podemos crear nuestro propio sistema basándonos en él.(Vega, 2017)</p>
OpenHAB		<p>OpenHAB presume de ser un sistema domótico Open Source totalmente agnóstico en cuanto a tecnología se refiere. Esto quiere decir que cubre una gran cantidad de sistemas: Windows, OSX, Linux, sistemas embebidos como NAS, etc.</p> <p>Su diseño se basa en una arquitectura totalmente modular donde tenemos el core del sistema y sobre el que se construyen alrededor los diferentes módulos que le añaden funcionalidad. Está escrito en Java y su modularidad ha permitido que cuente con muchos módulos diferentes que cubren gran cantidad de elementos hardware y protocolos.(Vega, 2017)</p>
Home Assistant		<p>Es un proyecto gratuito y de código abierto diseñado para permitirnos gestionar todos nuestros dispositivos IoT de domótica desde una única plataforma. Este proyecto está creado en Python 3 y diseñado para que, una vez montado en un dispositivo como un Raspberry Pi, podamos centralizar el control de todos estos aparatos que, de otra forma, tendrían que controlarse de forma independiente.(Velasco, 2017)</p>
		<p>Desarrollada por Google "Brillo", está basada en Android pero creada directamente para el "Internet de las Cosas". No se trata de su sistema</p>

Brillo		<p>móvil propiamente dicho, sino un software derivado de él que permite aprovecharse de cosas como una conectividad total, flexibilidad para adaptarse a cualquier aparato.(Bolaños, 2015)</p>
HomeMaestro		<p>Desarrollada por Microsoft y basada en la creación de casas y edificios automáticos con la finalidad de mejorar su gestión energética y la calidad de vida de sus habitantes. Ésta, así mismo, está evolucionando hacia la integración de todos los sistemas de control en una única red desde la que se dan las órdenes, creando lo que se denomina hogar digital, con lo que se consigue reducir el trabajo doméstico, aumentar la seguridad de la casa y optimizar los recursos de la misma.(Santamarín, 2012)</p>

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

1.1.5 Fabricantes de sistemas domóticos.

Actualmente existe una gran variedad de sistemas domóticos disponibles en el mercado, cuya mayoría están centralizados en cubrir grandes superficies, y más pequeños para aplicaciones en el hogar.(Sánchez, 2015)

A continuación citaremos algunas de las empresas que brindan este tipo de servicios:

Tabla 3-1. Fabricantes de Sistemas Domóticos

NOMBRE	LOGO	PAÍS DE ORIGEN
Loxone		Australia
Simon		España
Legrang		España
Fagor		España
Somfy		Francia
Makita		Japón
Terasaki		España
Fermax		España

Climastar		España
Sony	SONY	Japón

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

Los presupuestos pueden variar dependiendo de las necesidades del usuario (número de pisos, baños, habitaciones, mano de obra), los cuales pueden ser consultados sin costo alguno en la siguiente página: <https://marcas.habitissimo.es/domotica> (Anexo W).

1.2 Controladores de propósito general

Son aquellos dispositivos con la capacidad de recibir información, realizar cálculos sobre ella y dar una respuesta como salida con el fin de resolver un sistema. A continuación describiremos algunos de ellos:

1.2.1 Raspberry Pi.

La Fundación Raspberry Pi surge alrededor del año 2008, pero no fue hasta 2012 que lanzaron el primer Raspberry comercial, en respuesta a la necesidad de los usuarios de una placa de desarrollo de bajo costo y obteniendo así un reconocimiento a nivel mundial. Las prometedoras propuestas de dicha fundación, les permitió estar en la mira de prestigiosas universidades entre ellas la Universidad de Cambridge que conjuntamente con la Corporación Broadcom, una de las pioneras en el desarrollo de circuitos integrados para la comunicación en banda ancha de los Estados Unidos brindaron ayuda para desarrollar el diseño final, el cual estaba enfocado para el uso de estudiantes de primaria secundaria ya que es la etapa donde los estudiantes desarrollan sus destrezas de programación.(Calle, 2014)

Es considerado un computador que tiene el tamaño de una tarjeta de crédito, y de un costo reducido, consta de una conexión de red y HDMI, puede comunicarse con diferentes componentes electrónicos para facilitar la enseñanza de la informática en general.

Una Raspberry Pi es orientada hacia personas con intereses en programación, sistemas embebidos o simplemente para personas que quieren saciar su curiosidad. Como complemento posee un puerto GPIO que permite conectar periféricos o componentes electrónicos, la Figura 4-1 señala la estructura de la placa Raspberry Pi 3, ya que es la más utilizada actualmente.

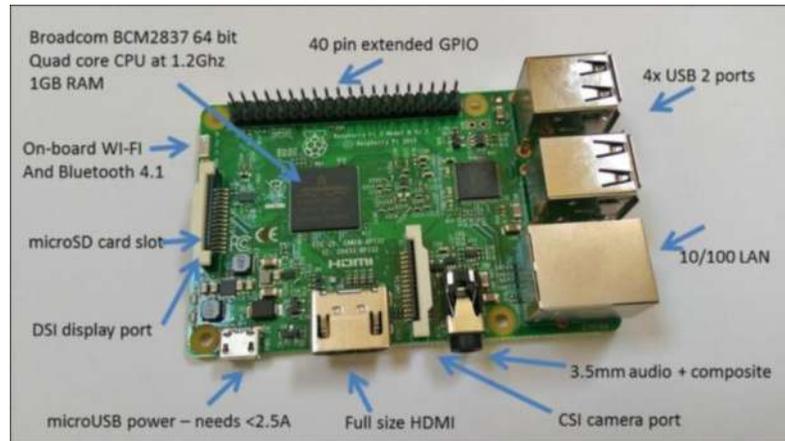


Figura 4-1: Raspberry Pi 3

Fuente: <https://tech.scargill.net/raspberry-pi-3-grand-opening/>

1.2.1.1 Características

- Consta de una ranura SD donde debe ser instalado el sistema operativo.
- Memoria RAM fija.
- Posee varios puertos USB expandibles.
- Es compatible con módulos WIFI para su conexión en versiones que no la tienen incorporado.
- No cuenta con un reloj de tiempo real.
- Consta con un puerto Ethernet para su conexión.
- Posee una salida HDMI para conectarla a un monitor.
- Es considerado como un dispositivo de bajo costo.

1.2.1.2 Modelos de la placa Raspberry Pi.

La placa Raspberry Pi está disponible en dos versiones (Modelo A y Modelo B), entre estos dos modelos hay una diferencia. El modelo A tiene 256MB de RAM y un puerto USB. No es compatible con la conexión de red ya que no tiene ningún puerto Ethernet. Por otro lado, el Modelo B tiene un puerto Ethernet, 512 MB de RAM y dos puertos USB, como se aprecia en la siguiente tabla:

Tabla 4-1: Modelos de la placa Raspberry Pi.

	Modelo A	Modelo B
RRP	\$25	\$35
System on a Chip	Broadcom BCM2835	
CPU	700 MHz ARM1176JZF-S core	
GPU	Broadcom VideoCore IV, OpenGL ES 2.0 Device capable of MPEG-2 and VC-1, 1080p30 h.264/MPEG-4 AVC decoding and encoding.	
Memory (SDRAM)	256 MB, shared with GPU	512 MB (models build since October 15th 2012)
USB 2.0	1(Expandable)	2(Integrated USB hub)
Video Out	Composite RCA (PAL and NTSC), HDMI (also Display Serial Interface for LCD panels)	
Audio Out	3.5 mm jack, HDMI	
Storage	SD/MMC/SDIO card slot	
Network	No connector	RJ45 Ethernet through integrated USB hub
Peripheral connectors	8 x GPIO, UART, IC bus, SPI bus	
Power rating	300 mA(1.5 W)	700 mA (3.5 W)
Power source	5 volt via MicroUSB or GPIO header	

Fuente: <https://www.thefruitycomputer.com/forums/topic/923-a-beginner%E2%80%99s-guide-to-the-raspberry-pi/>

1.2.1.3 Componentes de la Raspberry Pi

Raspberry Pi tiene todo los componentes necesarios para equiparse a un ordenador, a continuación en el Gráfico 2-1 se detalla cada uno de ellos:

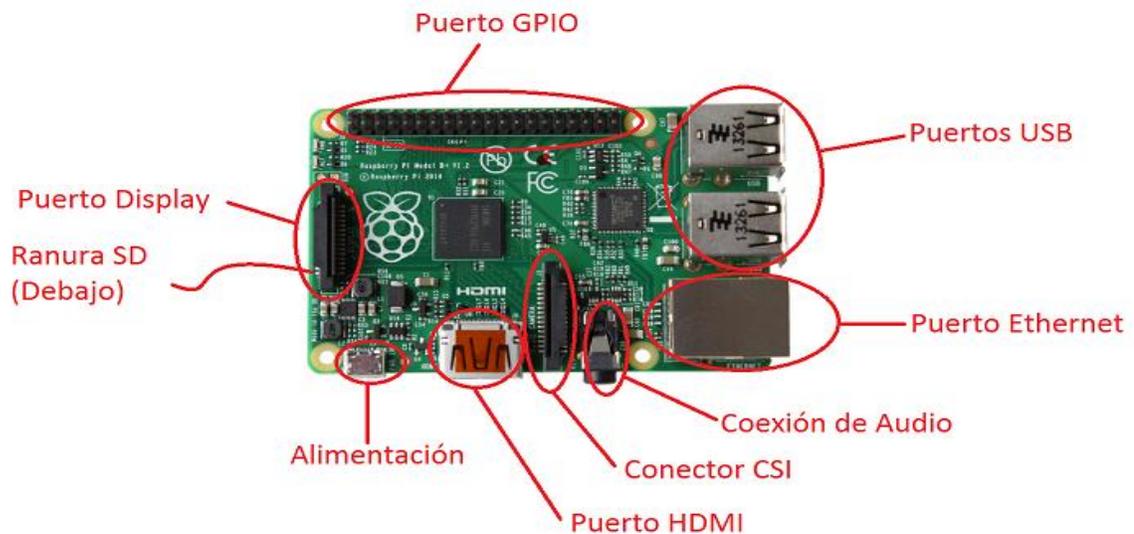


Gráfico 2-1: Componentes de la Raspberry Pi

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

- **Puerto USB (Universal Serial Bus)**

Dependiendo del modelo de la placa, el número de puertos puede variar, permitiendo conectar periféricos como teclado y ratón, podemos extender el número de puertos dependiendo las necesidades del usuario.

- **Puerto Ethernet**

Posee un puerto Ethernet que permite conectar la Raspberry Pi a la red por medio de cable, lo cual hace posible acceder a internet o incluso poder acceder a la Raspberry Pi desde cualquier otro dispositivo conectado a la red.

- **Conexión de Audio**

Está equipado por una salida/entrada de audio, en esta ranura podemos conectar auriculares para entrada y salida de audio, esta conexión es innecesaria al momento que utilizamos el puerto HDMI.

- **Conector CSI (Camera Serial Interface)**

Es un conector de 15 pines tipo bus, en él podemos conectar la cámara de la Raspberry Pi (también compatible con la interfaz CSI-2).

- **Puerto HDMI**

Este puerto es de gran ayuda al momento de utilizar la interfaz gráfica de la Raspberry Pi, nos permite transmitir audio y video hacia monitores que soporten la tecnología HDMI.

- **Alimentación**

Consta de un conector micro USB que suministra 5 Voltios y mínimo 0.7 Amperios, cabe recalcar que no posee con un botón de encendido/apagado por lo cual se enciende automáticamente al momento que lo alimentamos.

- **Ranura SD**

Debido a que no posee un disco duro la Raspberry Pi está equipada con una ranura SD compatible con casi cualquier tarjeta de memoria, dicha tarjeta debe ser mayor a 2 GB ya que es la encargada de albergar el sistema operativo de la Raspberry Pi.

- **Puerto Display**

Son conectores que permiten integrar displays específicos para la Raspberry Pi que se encuentran en el mercado con la finalidad de hacerlo más compacto.

- **Puerto GPIO**

Es un puerto de expansión de entradas/salidas de propósito general (GPIO) que posee la Raspberry pi, podemos acceder al GPIO mediante el terminal llamado P1 en la placa. Podemos configurar casi todos los puertos excepto los de alimentación, dependiendo la necesidad de usuario. También posee puertos designados por defecto como podemos apreciar en la Figura 5-1 que hace referencia a la distribución de los pines del GPIO.

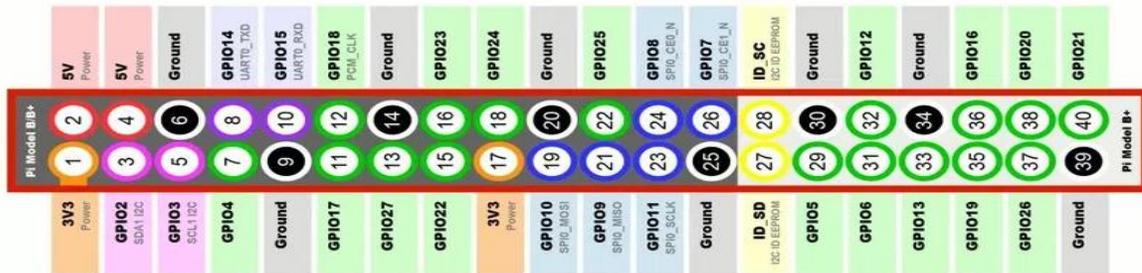


Figura 5-1: Distribución de los pines del GPIO.

Fuente: <https://comohacer.eu/gpio-raspberry-pi/>

El puerto GPIO de la Raspberry Pi no soporta cortocircuitos o tensiones muy elevadas, éstos pines trabajan con 3.3V y una corriente máxima de 50mA, es decir que para tensiones superiores es necesario realizar previamente un acople como protección a la placa (Calle, 2014), la Figura 6-1 muestra la ubicación de dicho puerto en la placa Raspberry Pi.

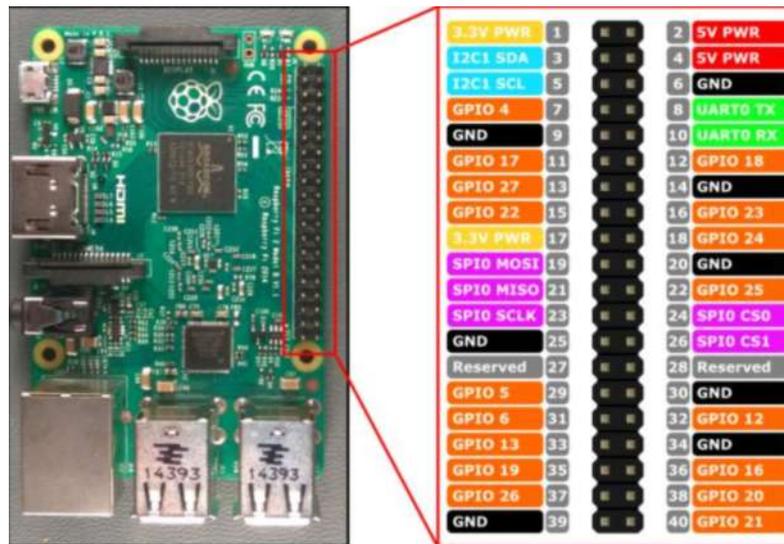


Figura 6-1: Ubicación del GPIO en la placa Raspberry Pi

Fuente: <https://www.prometec.net/rpi-gpio/>

1.2.2 Arduino

Es una tarjeta electrónica diseñada para proyectos multidisciplinarios basada en software y hardware flexibles y fáciles de usar, puede recolectar información de su entorno dependiendo las necesidades del usuario, a través de sus pines de entrada usando una gama de sensores con la finalidad de afectar aquello que lo rodea (actuadores).

Un microcontrolador integrado en la placa se encarga de alojar la programación realizada con un lenguaje basado en Wiring y su entorno de desarrollo Arduino basado en Processing. Los proyectos basados en Arduino pueden ejecutarse sin la necesidad de conectarse a un ordenador, esto no quiere decir que no se pueda hacerlo. (Flores & Robles, 2012)

1.2.2.1 Hardware

En la actualidad hay diferentes versiones de la placa Arduino (Uno, Tre, 101, Zero, Zero pro, Yun, Leonardo, Mega, etc.), la placa más utilizada es la placa Arduino Uno (Figura 7-1) basada en Atmel ATmega328 a diferencia del Arduino Mega que usa ATmega1280.



Figura 7-1: Arduino Uno

Fuente: <https://www.generationrobots.com/blog/wp-content/uploads/2016/09/arduino-uno-rev3-generation-robots.jpg>

1.2.2.2 Componentes de la placa Arduino Uno

Esta placa consta de 14 pines (in/out) digitales de los cuales podemos usar 6 como salidas PWM , 6 entradas analógicas, un cristal oscilador a 16 Mhz, conexión USB, entradas de alimentación DC, una cabecera ICSP, y un botón de reset, como se muestra en el Gráfico 3-1.

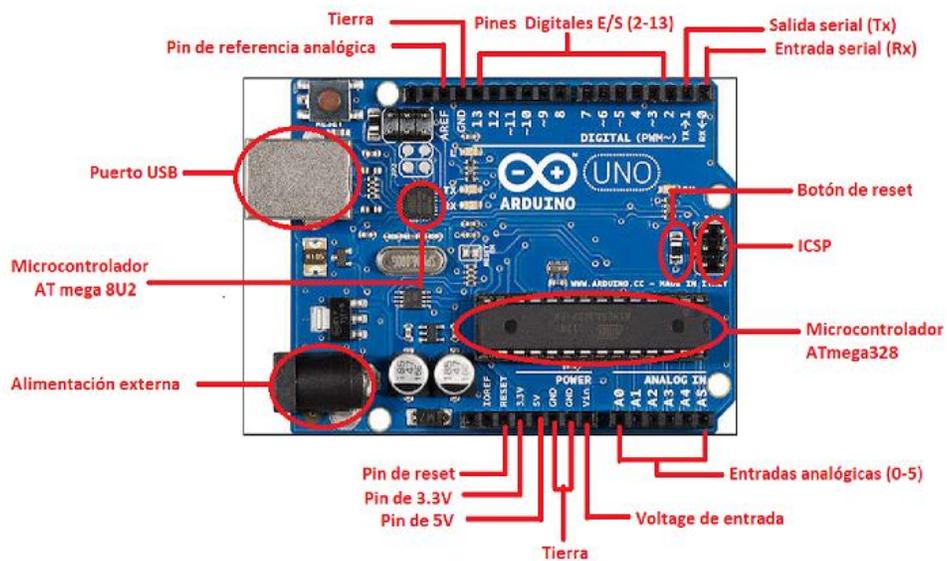


Gráfico 3-1: Componentes de la placa Arduino Uno

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

1.2.2.3 Comparativa de la placa Arduino

Actualmente existen varios modelos de dicha placa, a continuación en la Tabla 4-1 se detalla las más importantes:

Tabla 5-1: Comparativa de la placa Arduino.

Versión	Imagen	Microcontrolador	Voltaje de funcionamiento	Memoria flash	Sram	EEPROM	Voltage de salida	Pines	Intensidad
Uno		Atmel ATmega320 de 8 bits a 16Mhz	5 V	32 KB	2 KB	1 KB	6-20 V	14 Digitales 6 Analógicos	40 mA
Tre		Atmel ATmega320 de 8 bits a 16Mhz	5 V	32 KB	2.5 KB	1 KB	6-20 V	14 Digitales 6 Analógicos	40 mA
101		SoC Intel Quark de 32 bits	5 V	384 KB	80 KB		6-20 V	14 Digitales 6 Analógicos	40 mA
Zero		Atmel SAMD21 MCU de 32 bits a 48 MHz	5 V	256 KB	32 KB	Más de 16 KB	6-20 V	14 Digitales 6 Analógicos	40 mA

Zero Pro		Cortex M0+ de 32 bits a 48 MHz	5 V	256 KB	32 KB	Más de 16 KB	6-20 V	14 Digitales 6 Analógicos	40 mA
Yun		ATmega32u4 en un chip Atheros AR9331	5 V	32 KB	2,5 KB	1 KB	6-20 V	20 Digitales 12 Analógicos	40 mA
Mega		ATmega2560 de 8 bits a 16 Mhz	5 V	256 KB	8 KB	4 KB	6-20 V	54 Digitales 16 Analógicos	40 mA

FUENTE: <https://comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles/>

1.2.2.4 Software

Soporta plataformas como: Windows, Linux, Android y Mac OS X, (Figura 8.-1) debido a que es un código abierto y a que su entorno es fácil de usar, nos facilita la programación y cargado a la placa de E/S. El entorno está basado en Processing, avr-gcc y otros programas de código abierto y a la vez escrito en Java. (Flores & Robles, 2012)

Download the Arduino Software



Figura 8-1: Plataformas que soporta el Software Arduino.

Fuente: <https://i2.wp.com/tecnologiारobotica.com/wp-content/uploads/2016/06/Actualizar-Instalar-Arduino.jpg>

1.2.2.5 Entorno Arduino

También conocido como IDE para Arduino, este entorno está compuesto por un editor de texto donde se escribe el código, una barra de mensajes, una consola de textos, una barra de herramientas y una serie de menús como se detalla en el Gráfico 4-1.

Permite la conexión con el hardware de Arduino mediante el puerto USB, el cual sirve para cargar los programas y la comunicación entre ellos.

Un programa en Arduino se denomina sketch, estos programas son escritos en el editor de texto en el cual podremos: cortar, pegar y buscar/reemplazar texto. En el área que se muestra los mensajes se puede observar información mientras se carga el programa o a la vez mostrar errores en la programación. La barra de herramientas permite constatar el proceso de carga, crear programas, abrir o cargar programas existentes.

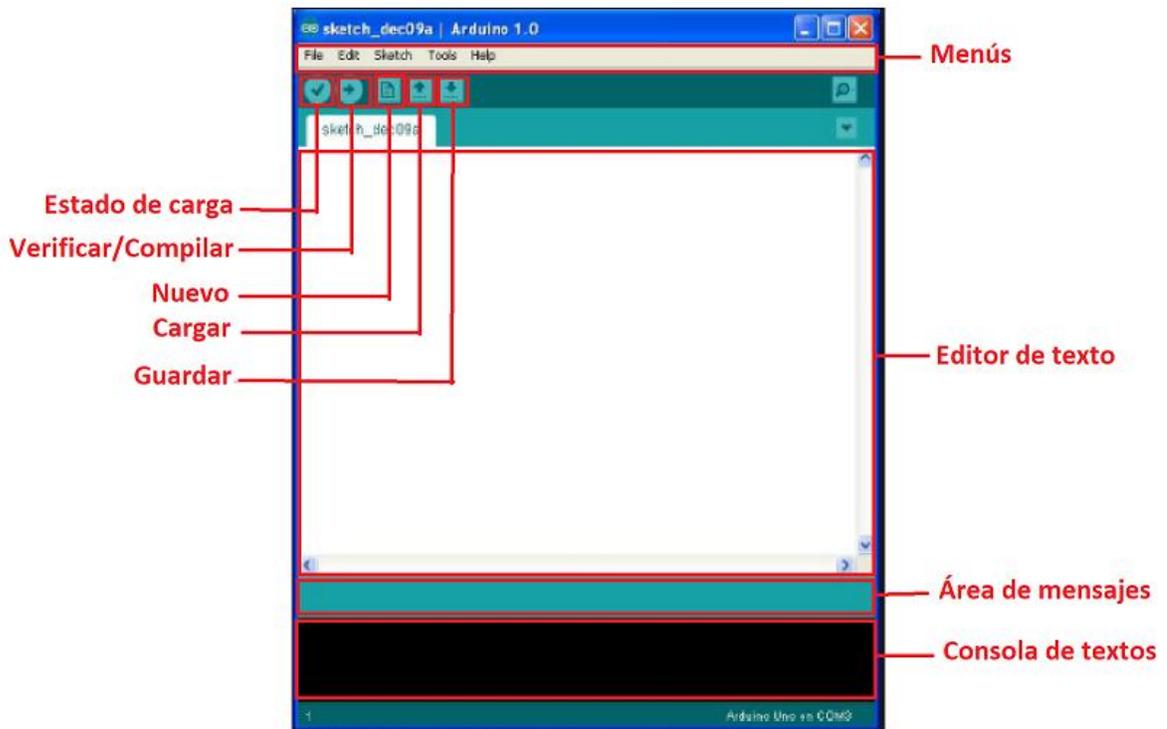


Gráfico 4-1: Entorno Arduino

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

Ventajas de usar Arduino

- Listo para usar, no necesita de ninguna parte adicional.
- Es de bajo costo.
- Comunidad grande y activa (foros).

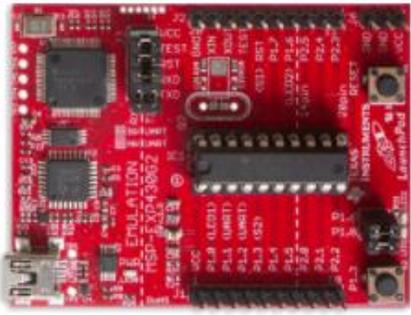
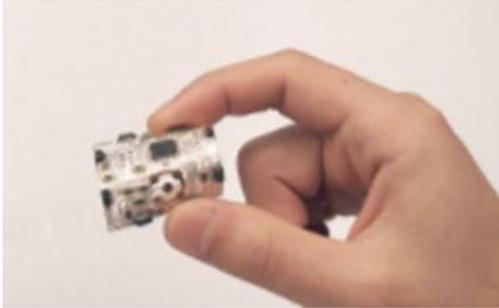
Desventajas de usar Arduino

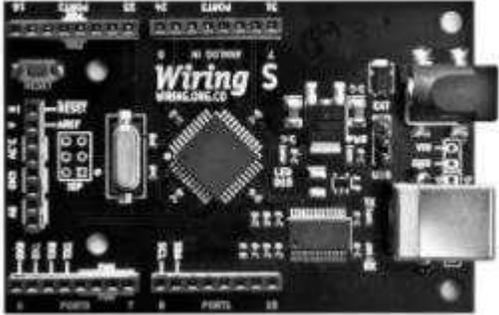
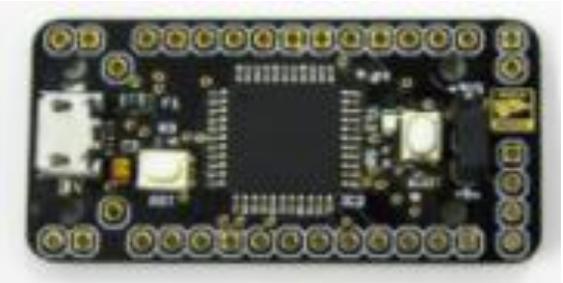
- Baja eficiencia, debido a su facilidad no siempre es la mejor opción.
- Estructura, ocupa un espacio considerable.

1.2.3 Otras tarjetas de desarrollo

También existe varias tarjetas de desarrollo no muy conocidas, que pueden variar dependiendo de las necesidades del usuario, a continuación se detalla algunas de ellas:

Tabla 6-1: Otras tarjetas de desarrollo

NOMBRE	IMAGEN	DESCRIPCIÓN
Tessel		<p>Esta tarjeta de desarrollo maneja el lenguaje de programación JAVASCRIPT compatible con el API de NODE.JS. Es una tarjeta que se programa de manera familiar para desarrolladores de software.(Quispe, 2017)</p>
Launchpad MSP 430		<p>Esta tarjeta está compuesta por un microcontrolador MSP430 de la familia de Texas Instrument, tiene varias tarjetas dependiendo al tipo de microcontrolador.(Quispe, 2017)</p>
Printoo		<p>Printoo es una plataforma de papel delgado basada en Arduino, su particularidad es que es de baja potencia y sus módulos permiten a los desarrolladores, lograr nuevos niveles de flexibilidad creativa.(Quispe, 2017)</p>

Wiring		<p>El entorno de programación de Wiring es compatible con cualquier hardware basado en la serie de procesadores AVR de Atmel, no sólo las propias tarjetas Wiring. Y esto incluye la línea de Arduino.(Quispe, 2017)</p>
Parallax Propeller		<p>Esta placa tiene un procesador multi-núcleo paralelo arquitectura de ordenador microcontrolador de chip con ocho de 32 bits denominado Parallax Propeller, esta placa puede ejecutar aplicaciones en paralelo en lugar de utilizar interrupciones tiene la forma similar a un Arduino.(Quispe, 2017)</p>
NoMADA Go		<p>NoMADA Go es una laca que permite crear aplicaciones mecatrónicas utilizando el microcontrolador AVR® ATmega32U4 de la marca Atmel a través de una</p>

		<p>tarjeta compacta, la cual puede ser programada desde la interfaz USB mediante el uso de la aplicación FLIP desarrollada por Atmel.(Quispe, 2017)</p>
<p>DigiSpark</p>		<p>Es una otra tarjeta pequeña compatible con Arduino. Está construida alrededor del microcontrolador ATtiny85 tiene una variedad de kits de shields interesantes que te permitirán extender sus capacidades.(Quispe, 2017)</p>
<p>Jaguar One</p>		<p>El JaguarBoard nació recientemente de un proyecto de Kickstarter y ya tiene su primer modelo, la Jaguar One. Su mayor atractivo es la compatibilidad con X86 y la posibilidad de instalar Windows 8 o 10 en su versión completa.(Quispe, 2017)</p>

Fuente: <http://www.lightpath.io/tarjetas-de-desarrollo/>

1.2.4 Arduino vs Raspberry Pi

Desde un punto de vista general y básico, la principal diferencia es el poder de procesamiento. Un Arduino con un ATmega328P, un microcontrolador de 8 bits, vs la Raspberry Pi 3, con un BCM2837, un Quad-core de 64 bits. Para muchos usuarios el poder de procesamiento de la placa Arduino no tiene comparación con la de la Raspberry Pi (Torres, 2017), sin embargo eso no lo es todo, en la siguiente tabla compararemos las características de dichas tarjetas.

Tabla 7-1. Arduino vs Raspberry Pi.

	Arduino	Raspberry Pi	Observación
Procesamiento	ATmega328P con 20 MIPS (millones de instrucciones por segundo)	BCM2837 Cortex A7 con arquitectura ARM tipo Quad Core a 1.2 Ghz.	Para la Raspberry Pi el procesamiento de los datos es vital.
Periféricos y protocolos	UART, I2C, SPI, GPIO, PWM, ADC y Comparador.	UART, I2C, SPI, GPIO, PWM, USB, Ethernet, WiFi, HDMI.	Si deseamos desplegar una interfaz gráfica, comunicarte mediante WiFi o tener la capacidad de manipular un a cámara web mediante USB, la Raspberri Pi es la mejor opción. Si lo que necesitas solo es leer datos de un sensor I2C, escribir en una memoria SPI, controlar un GPS o un módulo de comunicación GSM mediante UART, la mejor opción es el Arduino.
Costo y puesta en marcha	El costo aproximado es de \$15, considerando que se quiere programar el “hola mundo” nos tomaría un par de	El costo es de \$70 aproximadamente, considerando que se quiere programar el “hola mundo” , este proceso es más lento ya que debemos	Sin duda el Arduino tene un menor costo, y sería la mejr opción para proyectos sencillos.

	minutos.	cargar el sistema operativo en una memoria SD y conectar los periféricos, nos tomaría alrededor de 30 minutos.	
Entorno de desarrollo	El entorno de desarrollo gráfico (IDE, por sus siglas en inglés) del Arduino es muy limitado. Aparte de la interfaz de programación solo tiene un botón de validación y uno de carga.	La Raspberry Pi es compatible con muchos programas y lenguajes de programación, como C++ y Python. Además puedes realizar compilación cruzada, instalar IDEs como Eclipse o QT.	Raspberry Pi, simplemente tienes más variedad y entornos más completos.
Comunidad y hardware compatible	La cantidad de tutoriales Arduino y marcas o compañías que diseñan sistemas compatibles con Arduino es muy grande.	Existen muchas tarjetas y foros relacionados Raspberry Pi. La comunidad y compatibilidad de esta tarjeta, ha ido creciendo, sin embargo no al paso de Arduino.	Sin duda la comunidad Arduino es más amplia en la red.

Fuente: <https://hetpro-store.com/TUTORIALES/arduino-vs-raspberry-pi/>

Se puede afirmar que ambas tarjetas son las más relevantes en su ámbito tecnológico, sin embargo para el desarrollo del prototipo se optó por la placa Raspberry Pi, debido a su capacidad de procesamiento el cual es vital ya que se desarrolla en tiempo real y además es compatible con varios lenguajes de programación, permitiendo así la eficacia del sistema.

1.3 Sistema operativo Raspbian

No importa cuán bueno sea el hardware de la Raspberry Pi, sin un sistema operativo, es solo una pieza de silicio, fibra de vidrio y algunos otros materiales. Hay varios sistemas operativos diferentes para Raspberry Pi, incluyendo RISC OS, Pidora Arch Linux y Raspbian.

Actualmente, Raspbian es el sistema operativo basado en Linux más popular para Raspberry Pi. Raspbian es un sistema operativo libre basado en Debian optimizado para el hardware Raspberry Pi. Un sistema operativo está compuesto por el conjunto de programas básicos y utilidades que conjuntas hacen que funcione la Raspberry Pi. Sin embargo, Raspbian ofrece más que un sistema operativo puro; consta con más de 35.000 paquetes, entre ellos software recompilado en un formato que hace más fácil la instalación en su Raspberry Pi.

En el año 2012 se envió al mercado el sistema operativo Raspbian con 35.000 paquetes optimizados para un buen rendimiento en la Raspberry Pi. Sin embargo este sistema operativo aún está en desarrollo activo con la finalidad de mejorar la estabilidad y el rendimiento como sea posible.

Raspbian está diseñado para ser fácil de usar y es el sistema operativo recomendado para los principiantes con su Raspberry Pi. Cabe recalcar que Raspbian no pertenece a la Fundación Raspberry Pi, este sistema operativo fue creado por un pequeño equipo dedicado al desarrollador los cuales son fans del Hardware Raspberry Pi, y el proyecto Debian.(Barahona & Huilcapi, 2015)

1.3.1 El Sistema Operativo Debian

Fue creado en agosto de 1993 por Ian Murdock y es una de las distribuciones originales de Linux, como Raspbian se basa en el sistema operativo Debian, comparte casi todas las funciones de Debian, incluido su gran repositorio de paquetes de software. Hay más de 35000 paquetes de software gratuitos disponibles para su Raspberry Pi, y están disponibles para su uso en este momento.

1.3.2 Software de código abierto

La mayoría del software que compone Raspbian en la Raspberry Pi es de código abierto. El software de código abierto es un software cuyo código fuente ya está disponible para modificaciones o mejoras.

El kernel de Linux y la mayoría de otro software que compone Raspbian se maneja bajo la Licencia GPLv2. Esto significa que el software está disponible para usted sin costo alguno y que el código fuente que lo compone está disponible para cuando usted lo desee; La licencia GPLv2 también elimina cualquier reclamo o garantía.

1.3.3 Componentes de Raspbian

Hay muchos componentes que componen una distribución de Linux. Estos componentes trabajan juntos para proporcionarle todas las características que espera de la computadora.

Hay varios componentes clave de los que se construye Raspbian. Estos componentes son:

- **Gestor de arranque Raspberry Pi**

Cuando tu Raspberry Pi está encendida, suceden muchas cosas detrás de la escena. El rol del gestor de arranque es inicializar el hardware en el Raspberry Pi a un estado conocido, y luego comenzar a cargar el kernel de Linux. En el caso de la Raspberry Pi, esto lo hacen los cargadores de arranque de primera y segunda etapa. El gestor de arranque de la primera etapa se programa en la ROM de Raspberry Pi durante la fabricación y no se puede modificar. Los cargadores de arranque de la segunda y tercera etapa se almacenan en la tarjeta SD y son ejecutados automáticamente por el gestor de arranque de la etapa anterior.

- **Kernel de Linux**

El kernel de Linux fue creado por Linus Torvalds, quien comenzó a trabajar en Kernel en abril de 1991. Desde entonces, grupos de voluntarios y organizaciones han trabajado juntos para continuar el desarrollo del kernel y hacerlo lo que es hoy. Es una de las partes más fundamentales de Raspbian. Gestiona cada parte de la operación de tu Raspberry Pi, desde mostrar texto en la pantalla hasta recibir pulsaciones de teclas cuando escribes en tu teclado

Si desea usar un dispositivo de hardware y conectarlo a su Raspberry Pi, el kernel necesita saber de qué se trata y cómo usarlo. La gran mayoría de los dispositivos en el mercado son compatibles con

el kernel de Linux, y cada vez se agregan más. Un buen ejemplo de esto es cuando conecta una unidad USB y notifica a un daemon que automáticamente pone a su disposición los archivos.

Cuando el kernel ha terminado de cargarse, automáticamente ejecuta el programa calle `init`. Este programa está diseñado para finalizar la inicialización de la Raspberry Pi y luego cargar el resto del sistema operativo. Este programa comienza cargando todos los daemons en segundo plano, seguidos por la interfaz gráfica de usuario.

- **Daemons**

Es una pieza de software que detrás de escena le brinda al sistema operativo diferentes funciones. Algunos ejemplos de un daemon incluyen el servidor web Apache, Cron, un programador de trabajos que se usa para ejecutar programas automáticamente en diferentes momentos, y `Autofs`, un daemon que monta automáticamente dispositivos de almacenamiento removibles como unidades USB.

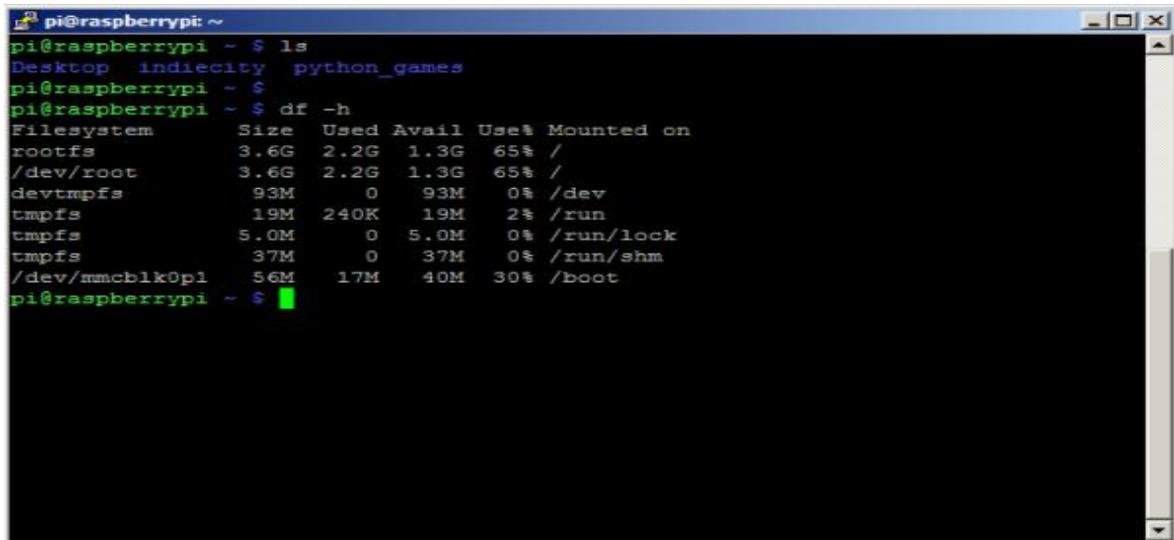
Una distribución como Raspbian no solo necesita que el kernel funcione. También necesita otro software que le permita al usuario interactuar con el kernel y administrar el resto del sistema operativo. El sistema operativo principal consiste en una colección de programas y scripts que hacen que esto suceda.

- **Shell**

Después de que todos los daemons se hayan cargado, `init` lanza un shell. Un shell es una interfaz para su Raspberry Pi que le permite monitorear y controlar utilizando comandos escritos con un teclado como se muestra en la Figura 9-1. No se deje engañar por esta interfaz, a pesar de que se ve exactamente como se usaba en las computadoras hace 30 años. El shell es una de las partes más poderosas de Raspbian.

Hay varios shells disponibles en Linux. Raspbian usa Bourne again shell (`bash`). Este shell es, con mucho, el shell más común usado en Linux.

`Bash` es una pieza de software extremadamente poderosa. Una de las características más potentes de `bash` es su capacidad para ejecutar scripts. Un script es simplemente una colección de comandos almacenados en un archivo que puede ejecutar un programa, leer claves del teclado y muchas otras cosas.



```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ ls  
Desktop indiecity python_games  
pi@raspberrypi ~ $  
pi@raspberrypi ~ $ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
rootfs          3.6G  2.2G  1.3G   65% /  
/dev/root       3.6G  2.2G  1.3G   65% /  
devtmpfs        93M   0    93M   0%  /dev  
tmpfs           19M   19M  19M   2%  /run  
tmpfs           5.0M   0   5.0M   0%  /run/lock  
tmpfs           37M   0   37M   0%  /run/shm  
/dev/mmcblk0p1  56M   17M  40M   30% /boot  
pi@raspberrypi ~ $
```

Figura 9-1: Interfaz shell

Fuente: <http://blogs.arcssoftwareconsultancy.com/pi/2013/03/07/windows-networking/>

- **Utilidades de Shell**

Un intérprete de comandos no es muy útil sin ningún comando para ejecutar. Mientras que bash proporciona algunos comandos muy básicos, todos los demás comandos son las utilidades de shell. Ofrecen muchas funciones que van desde la copia de archivos, la creación de directorios, hasta Advanced Packaging Tool (APT), una aplicación de administrador de paquetes que le permite instalar y eliminar software de su Raspberry Pi.

- **Servidor gráfico X.Org**

Después de cargar el shell y los daemons, de forma predeterminada, el servidor gráfico X.Org se inicia automáticamente. La función de X.Org es proporcionarle una plataforma común desde la cual crear una interfaz gráfica de usuario. X.Org maneja todo, desde mover el puntero del mouse, escuchar y responder a las pulsaciones de teclas para dibujar realmente las aplicaciones que está ejecutando en la pantalla.

- **Entorno del escritorio**

Es difícil usar cualquier computadora sin un entorno de escritorio. Un entorno de escritorio le permite interactuar con su computadora utilizando más que solo su teclado, navegar por Internet, ver imágenes y películas, y muchas otras cosas. Una GUI normalmente usa Windows, menús y un mouse para hacer esto.

Raspbian incluye una interfaz gráfica de usuario llamada Lighthouse X11 Desktop Environment o LXDE. LXDE se utiliza en Raspbian ya que fue diseñado específicamente para funcionar en dispositivos como el Raspberry Pi, que solo tiene recursos limitados (Harrington, 2015), en la Figura 10-1 se muestra el escritorio de Raspbian.

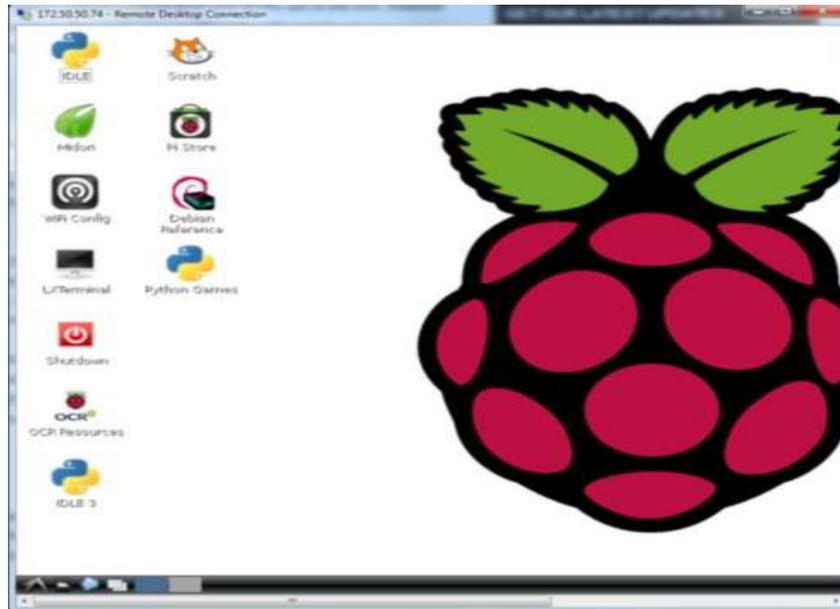


Figura 10-1: Escritorio de Raspbian

Fuente: <https://jadasupport.files.wordpress.com/2015/02/remote2.png>

1.4 Reconocimiento de voz

Es una técnica con la cual podemos, interpretar el significado de nuestras palabras, con el objetivo de ejecutar órdenes o tomar decisiones. El reconocimiento de voz está ligado a un conjunto de aplicaciones, entre las cuales podemos mencionar las siguientes:(Falconi & Jimenez, 2009)

- Identificación del locutor
- Selección de locutor
- Reconocimiento de una palabra clave
- Reconocimiento del habla
- Verificación del locutor

Ya que el propósito del habla es la comunicación, es muy común que en la actualidad una persona necesite comunicarse con una máquina como en el caso de las computadoras, la cual utiliza periféricos como el teclado o el mouse.

Teniendo presente que el habla es el medio más eficiente y natural con el cual las personas se comunican, se puede decir que el Reconocimiento de Voz permite que la comunicación entre máquina y hombre de manera natural, sencilla y simple. (Falconi & Jimenez, 2009)

1.4.1 Tamaño del vocabulario de reconocimiento de voz

Se considera tres rangos tomando en cuenta el número de palabras:

- **Pequeño:** Capaz de reconocer hasta 100 palabras.
- **Mediano:** Entre 100 y 1000 palabras.
- **Grande:** Más de 1000 palabras.

1.4.2 Formas de habla usuario-máquina.

Para realizar un sistema de reconocimiento de voz hay que tener en cuenta diferentes aspectos. Como:

1. Palabra aislada.

Cuando el usuario habla palabras individuales o frases tomadas de un vocabulario previamente determinado.

2. Palabras conectadas.

Cuando el usuario habla de forma fluida una serie de palabras alojadas en un vocabulario restringido, el cual podría ser un número telefónico.

3. Habla continua.

Cuando el usuario habla de forma fluida teniendo un vocabulario muy amplio usualmente ilimitado.(Falconi & Jimenez, 2009)

1.4.3 Caracterización de los sistemas de reconocimiento de voz

Según Kurzweil (1999), el reconocimiento de voz es el proceso de convertir, por medio de una computadora, una señal acústica a una secuencia de palabras representadas en forma de texto. Las palabras reconocidas pueden servir de entrada a otros sistemas que los requieran para realizar alguna reacción, como activar ciertos dispositivos.

1.4.4 Aplicaciones para el reconocimiento de voz.

En la siguiente tabla se detalla las aplicaciones más significativas que emplean el reconocimiento de voz:

Tabla 8-1. Aplicaciones para el reconocimiento de voz

NOMBRE	LOGO	DESCRIPCIÓN
Vlingo	 The logo for Vlingo, featuring the word "Vlingo" in a blue sans-serif font with three orange curved lines to the right representing signal waves.	Esta aplicación permite controlar varios aspectos de los teléfonos inteligentes (apertura de aplicaciones, enviar mensajes de texto, realizar llamadas) por medio del uso de la voz. Disponible gratuitamente en varias plataformas (iOS, Android y Nokia)
Voice Dial	 A circular icon with a blue center containing a white speech bubble and a person silhouette, surrounded by a silver metallic ring with a mesh-like texture.	Aplicación que permite al usuario crear atajos de voz (ordenes de voz), seleccionar contactos de su agenda y efectuar llamadas por órdenes de voz pre programadas. Disponible en el idioma inglés, compatible solo con iOS y no es gratuito.

<p>Cyberon Voice Commander</p>		<p>Esta aplicación de paga se puede controlar todas las funciones principales de tu teléfono utilizando la voz marcación, búsqueda de contactos, abrir aplicaciones, notificaciones de la hora. Disponible y compatible para la plataforma Android 2.2 y versiones posteriores.</p>
<p>Dragon Dictation</p>		<p>Dragon Dictation tiene la particularidad de transcribir instantáneamente un texto de un dictado de voz ya sea para mensajes de texto, notas correos electrónicos además de contar con soporte de idioma múltiple (francés, inglés, español e italiano). Disponible y compatible para iOS desde la versión 4.0 en adelante, es gratuito.</p>
<p>Cortana</p>		<p>Cortana es una aplicación gratuita con la funcionalidad de dictar mails, búsqueda en internet, realizar llamadas, abrir aplicaciones entre otras cosas. Con Cortana es fácil interactuar con el reconocimiento de voz. Esta aplicación viene incorporada en la plataforma Windows 10 para escritorio y móviles.</p>
		<p>La aplicación Siri permite preguntar a tu iPhone ciertos datos u ordenar que</p>

Siri		<p>realice ciertas funciones, a través del reconocimiento de voz. Siri no tiene la necesidad de pre programar comandos especiales, ya que puede interpretar un lenguaje coloquial. Esta aplicación viene incorporada en iOS 4.0 en adelante</p>
Web Speech		<p>La API de JavaScript permite a los desarrolladores de páginas web incorporar el reconocimiento de voz. Esta api posee un control flexible sobre las capacidades de reconocimiento de voz, de generar salidas de texto a voz (viceversa), reconocimiento de voz para entrada de formularios, dictado y otras más.</p>

Fuente: <http://es.ccm.net/faq/7819-las-aplicaciones-de-reconocimiento-de-voz-para-ios-y-android>

Las librerías de Web Speech son herramientas poderosas para tener disponibles en las computadoras, y se han generalizado bastante en esta era moderna, herramientas como Cortana, Dictation y Siri en sistemas operativos modernos populares, y herramientas de accesibilidad como lectores de pantalla.(Graham, 2017)

Pero, ¿qué hay de la Web? Poder emitir comandos de voz directamente a una página web y hacer que el navegador lea el contenido de texto directamente sería muy útil, debido a esto y a que en la actualidad las nuevas tecnologías IOT están presentes en todo el mundo orillándonos a que todo esté conectado a la web, el presente trabajo utilizará las librerías de Web Speech.

Actualmente los único navegadores que soportan esta tecnología son Chrome y Firefox, ahora, ¿Cuáles son las diferencias entre los dos navegadores?

Chrome y Chrome mobile han sido compatibles con la síntesis y el reconocimiento desde la versión 33 utilizando los prefijos de webkit que no son más que etiquetas propias de cada navegador el cual interpreta de forma diferente al no haber un estándar por el que se tengan que regir.(Graham, 2017)

Antes de hablar de la Web Speech de Firefox debemos tener claro el concepto de Gecko el cual no es más que el nombre del motor de presentación desarrollado por la Fundación Mozilla. En un principio se llamó NGLayout. La función de Gecko es leer el contenido de la web, tanto HTML, CSS, XUL, como JavaScript, y presentarlo en pantalla o imprimirlo.(Alegsa, 2009)

Ahora, Firefox por otro lado tiene soporte para ambas partes de la API sin prefijos, aunque hay algunas cosas a tener en cuenta:

- Incluso a través del reconocimiento se implementa en Gecko, actualmente no se puede usar en el escritorio, Android porque el UX y el UI para permitir que los usuarios otorguen permiso a una aplicación para usarlo todavía no se implementa.
- La síntesis de voz no funciona en Android todavía.
- En Firefox OS, para que una aplicación use reconocimiento de voz necesita ser privilegiada, e incluye el permiso de captura de audio y reconocimiento de voz.
- Firefox actualmente no es compatible con la propiedad continua.
- El controlador de eventos onnomatch es actualmente de uso limitado; no se activa porque el motor de reconocimiento de voz Gecko integrado, Pocketsphinx, no admite una medida de confianza para cada reconocimiento. Por lo tanto, no informa "lo siento, no es ninguna de las anteriores", sino que dice "de las opciones que me diste, esto se ve mejor".(Graham, 2017)

Dicho esto se puede concluir que Chrome no parece tratar con gramáticas específicas; en su lugar, solo devuelve todos los resultados, y puede tratarlos como lo desee. Esto se debe a que el reconocimiento de voz del lado del servidor de Chrome tiene más poder de procesamiento disponible que la solución del lado del cliente que usa Firefox.

1.4.5 Conocimiento de los patrones de voz del usuario.

- **Sistemas dependientes del locutor:** Dirigido hacia locutores particulares, principalmente utilizado en diseños orientados a personas minusválidas. El margen de reconocimiento de estos sistemas debe ser más alto ya que la mayoría de los parámetros de representación del habla son sensibles a las características de un locutor en particular.
- **Sistemas independientes del locutor:** Emplean a una población de locutores más grandes, generalmente desconocidos por el sistema; este tipo de sistemas tiene varias ventajas a pesar de ser más complejos, ya que el usuario no debe entrenar los comandos de voz cada vez que lo utiliza (puede ser utilizado por cualquier persona), los comandos solo deben ser grabados una sola vez para posteriormente utilizarlos.
- **Sistemas adaptables:** Se adaptan a cualquier locutor mientras el sistema está en uso. (Falconi & Jimenez, 2009)

1.4.6 Grado de diálogo usuario-máquina.

- **Pasivo o unidireccional:** El usuario habla mientras la máquina realiza una acción como respuesta.
- **Sistema de diálogo activado por la máquina:** El sistema inicia el diálogo, y requiere información del usuario (entrada verbal).
- **Sistema de diálogo natural:** La máquina “conversa” con el locutor, le solicita entradas, actúa en función de las entradas y trata de despejar ambigüedades.

1.4.7 Grado de conocimiento acústico-lingüístico del sistema.

- **Solo conocimiento acústico:** No usan conocimiento acústico.
- **Integración de conocimiento acústico y lingüístico:** El conocimiento lingüístico está usualmente representado por restricciones sintácticas y semánticas sobre salida del sistema de reconocimiento.

CAPITULO II

2 MARCO METODOLÓGICO

2.1 Diseño del prototipo del control domótico.

El presente capítulo consiste en una guía del diseño e implementación de un prototipo de control domótico de bajo costo activado por voz para personas con discapacidad motriz.

Para una mayor comprensión se presenta un diagrama de bloques del funcionamiento como se muestra en el Gráfico 1-2 el cual sintetiza las distintas etapas que va a seguir nuestro prototipo domótico:

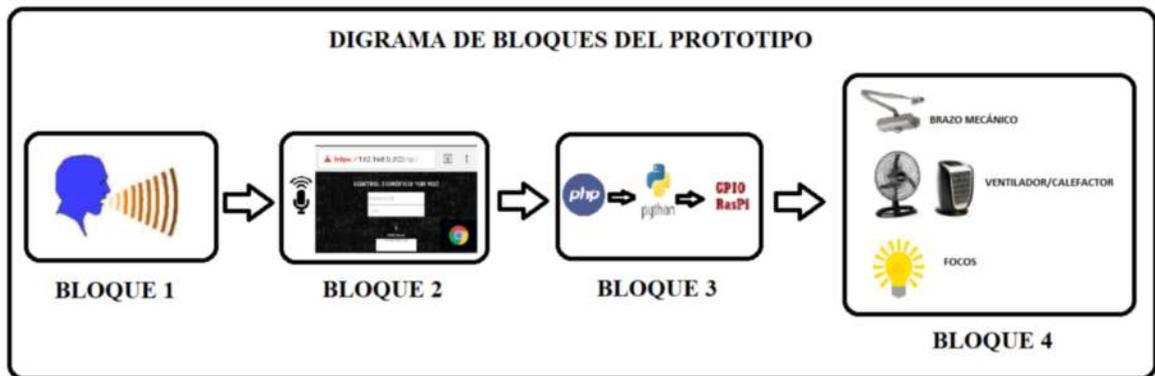


Gráfico 1-2: Diagrama de bloques general del prototipo.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017.

Bloque 1.- Adquisición de datos a través de voz

Bloque 2.- Interfaz para la interpretación de comandos de voz,

Bloque 3.- Control de los puertos GPIO de la Raspberry pi.

Bloque 4.- Control de los actuadores.

2.1.1 Estructura del Prototipo.

El prototipo será instalado en una maqueta que simulará una vivienda, previamente se realizaron pruebas con todos los actuadores conectados para analizar su funcionamiento.

La activación o desactivación de eventos físicos (encender- apagar actuadores) se realizó bajo una conexión inalámbrica de una red local con acceso a internet.

Para la alimentación del prototipo se usó dos fuentes de alimentación, una de 5 V para alimentar la Raspberry y una de 9 V para alimentar el módulo L298N.

EL prototipo y sus actuadores se comunican a través de cable UTP que están conectados a los distintos puertos GPIO, los cuales envían un pulso de activación o desactivación dependiendo de la orden que se envié.

2.1.2 Etapa de adquisición de datos.

Los datos de voz serán obtenidos a través de un teléfono inteligente (Smartphone) con acceso a internet en una red local específica, con el uso de su micrófono, el cual a través de una aplicación web lo mantiene activo y permite enviar los datos al servidor web alojado en la Raspberry Pi para su interpretación. En el gráfico 2-2 se puede apreciar el diagrama de adquisición de datos.

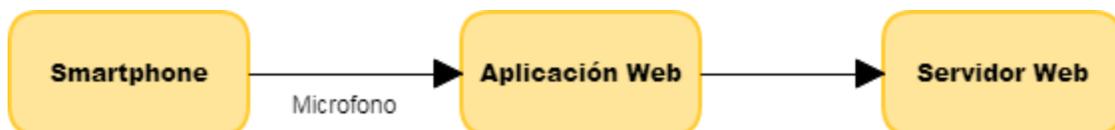


Gráfico 2-2: Diagrama de bloques adquisición de datos.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017.

2.1.3 Etapa de interpretación de datos.

Para la interpretación de comandos de voz se utilizó la aplicación Web Speech API de Google Chrome que permite digitalizar e interpretar la voz, después traducirla en cadena de caracteres. En el gráfico 3-2 se puede observar el diagrama de bloques de la interpretación de datos.

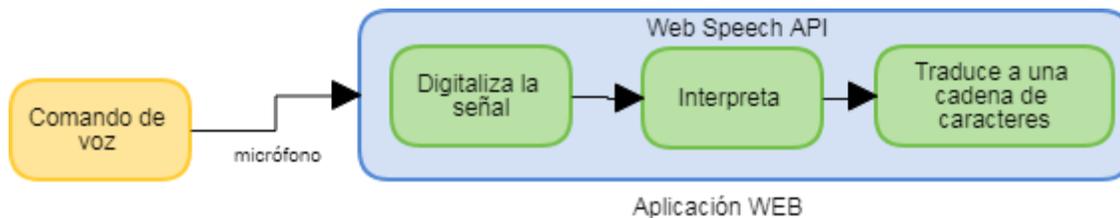


Gráfico 3-2: Diagrama de bloques interpretación de datos.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017.

2.1.4 Etapa de control de los puertos GPIO.

Los puertos GPIO van a ser controlados desde la aplicación WEB alojada en la Raspberry Pi la cual debe estar conectada al internet, la señal que active cada puerto dependerá del comando de voz emitida a la aplicación. Por medio de varios archivos programados en el servidor comparara la cadena de caracteres para la activación de los puertos respectivos. El grafico 4-2 muestra el diagrama de bloques de control de puertos GPIO.

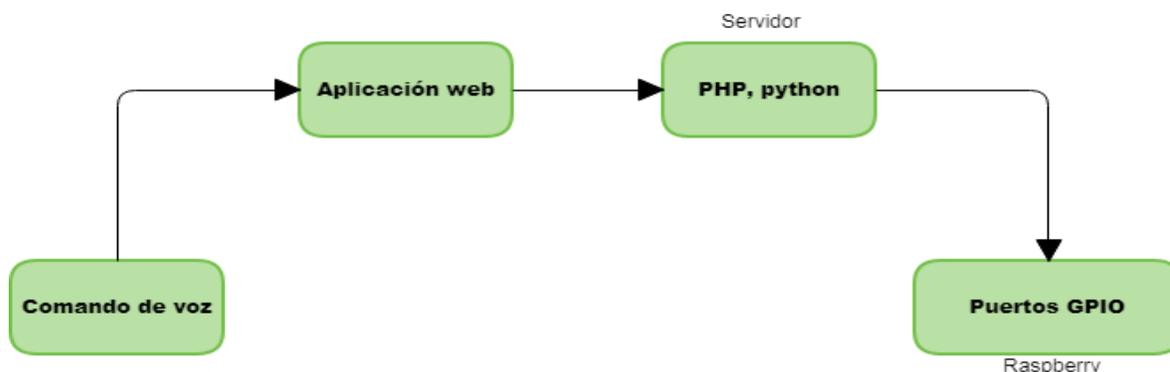


Gráfico 4-2: Diagrama de bloques control de puertos GPIO.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017.

2.1.5 Etapa de control de los actuadores

El control de los actuadores dependerá de la señal que emitan los puertos GPIO, debido a que cada actuador está conectado a un puerto específico y a su vez a un circuito elevador de potencia del

gráfico 15-2, ya que el amperaje de los puertos no es el necesario para su correcto funcionamiento. En el gráfico 5-2 se detallara el diagrama de bloques de control de los actuadores.



Gráfico 5-2: Diagrama de bloques control de los actuadores.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017.

Para el diseño del prototipo se dividió en dos partes tanto para el hardware como software.

2.2 Etapa Software.

2.2.1 Configuración de la RASPBERRY PI

El siguiente contenido incluye:

- Instalación del sistema operativo Raspbian.
- Conexión a través del protocolo de comunicación SSH.
- Instalación y configuración de un servidor web.
- Instalación del certificado SSL.
- Desarrollo de la aplicación web.

2.2.1.1 Instalación del sistema operativo Raspbian en la Raspberry pi

Se procedió a la instalación sistema operativo Raspbian debido a las prestaciones, características, servicios y recomendaciones de la plataforma Raspberry ya que explotan las capacidades al máximo de la Raspberry para su funcionamiento. Los pasos para su instalación están detalladas en el anexo Ñ.

El gráfico 6-2 muestra la interfaz gráfica del sistema operativo Raspbian.



Gráfico 6-2: Entorno gráfico Raspbian.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017.

2.2.1.2 Conexión a través del protocolo SSH.

Esta conexión tiene como objetivo facilitar el manejo y configuración de la Raspberry Pi, ya que brinda la posibilidad de conectarse de forma remota desde un ordenador evitando el uso de periféricos adicionales como teclado, televisor y un monitor. Los pasos de la instalación están detallados en el anexo O.

En el gráfico 7-2 se puede apreciar una conexión de forma remota por medio del protocolo SSH.

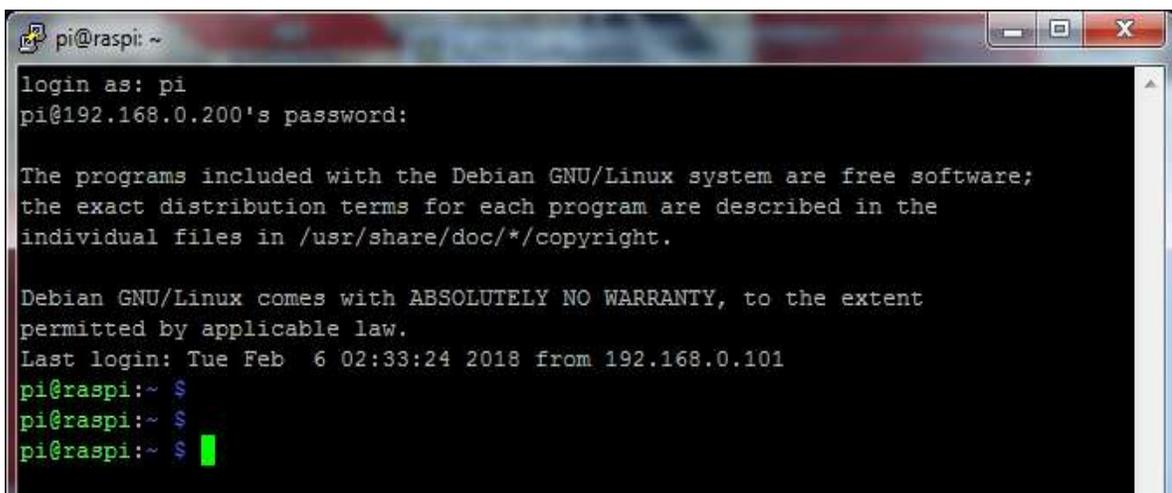
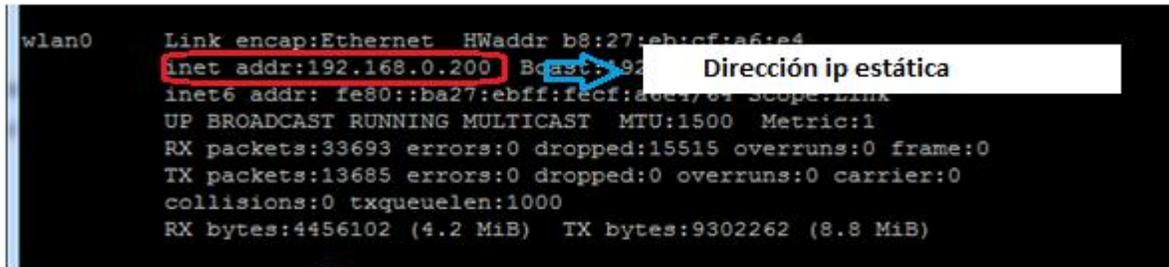


Gráfico 7-2: Conexión de forma remota SSH.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017.

Se recomienda configurar una dirección ip estática en la Raspberry pi para facilitar y la conexión de forma remota como se observa en el gráfico 8-2.



```
wlan0    Link encap:Ethernet  HWaddr b8:27:eb:cf:a6:e4
inet addr:192.168.0.200  Bcast:192.168.0.255  Mask:255.255.255.0
inet6 addr: fe80::ba27:ebff:fece:ae61/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:33693 errors:0 dropped:15515 overruns:0 frame:0
TX packets:13685 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:4456102 (4.2 MiB)  TX bytes:9302262 (8.8 MiB)
```

Gráfico 8-2: Dirección ip estática asignada.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017.

2.2.1.3 Instalación y configuración del servidor WEB

El servidor Web nos permitirá alojar los diferentes ficheros que se encargarán del procesamiento de los comandos de voz para la activación de los distintos puertos GPIO, como también el alojamiento de la aplicación web. Para mayor comprensión dirigirse al anexo P el cual se detalla los pasos de la instalación y configuración del servidor.

Se instaló el servidor Apache debido a que es el servidor más estable en la actualidad. En el gráfico 9-2 se puede observar el servidor Apache correctamente instalado.



Gráfico 9-2: Pagina web por defecto de Apache.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017.

2.2.1.4 Certificado SSL

El certificado SSL sirve para brindar seguridad al usuario de la aplicación web. SSL (Secure Socket Layer), es un protocolo de seguridad en donde los datos que son transmitidos entre el usuario y el servidor web son totalmente cifradas o encriptadas permitiendo que los datos viajen de manera íntegra y segura. En el anexo Q se detalla los pasos para su instalación.

En nuestra aplicación web el certificado SSL permite que el acceso al micrófono del navegador Google Chrome sea concedido ya que algunos de sus gadgets (virtuales) no se pueden activar si la navegación no cumple con un certificado de seguridad como https. En el gráfico 10-2 se puede apreciar la página web por defecto de Apache con un certificado de seguridad.

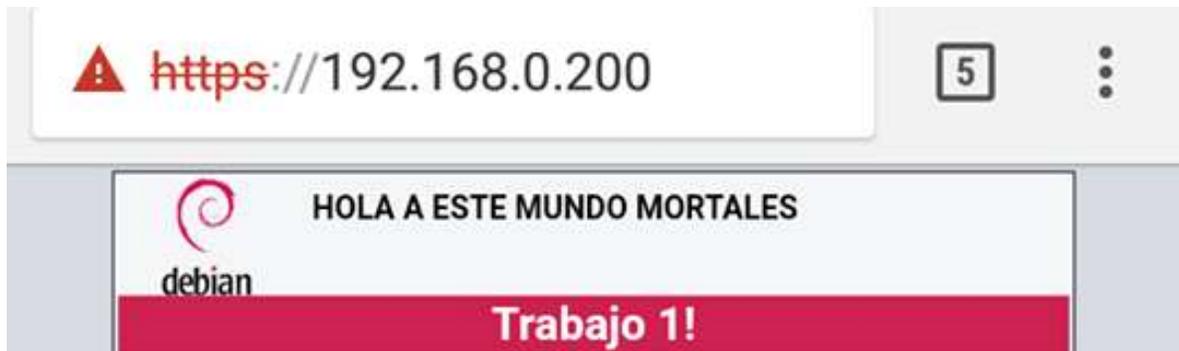


Gráfico 10-2: Verificación de HTTPS y certificado SSL.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

2.2.1.5 Desarrollo de la aplicación WEB

La aplicación web fue desarrollada bajo el código de programación HTML y JavaScript, tiene la función de ser la interfaz de comunicación entre el usuario y los distintos actuadores. El diseño de esta aplicación web permite al usuario interactuar casi en su totalidad por medio de comandos de voz.

En el gráfico 11-2 se puede observar la interfaz gráfica de la aplicación web.



Gráfico 11-2: Interfaz gráfica de la aplicación web.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

Algo importante de todo sistema es la seguridad, pero ¿Cuál sería el mejor método que se puede ofrecer a una persona con discapacidad motriz?

La respuesta sería por medios de comandos de voz, es por eso que en el presente prototipo se ha programado una palabra clave, que puede ser cualquiera que el usuario elija, con este comando el usuario de ser necesario podría bloquear el sistema deshabilitando el servidor apache, el cual puede ser reiniciado manualmente sólo por personas que conozcan las contraseñas de acceso a la Raspberry pi.

Nota: La seguridad de la red va a depender del administrador de la misma.

En el anexo M se detalla los pasos del desarrollo de la aplicación WEB.

2.3 Etapa de Hardware

2.3.1 Diseño de la maqueta.

Se diseñó y elaboró una maqueta emulando un departamento, como se puede observar en el gráfico 12-2 con distintos ambientes para montar los actuadores a controlar:

- Sala
- Comedor
- Dormitorio
- Baño

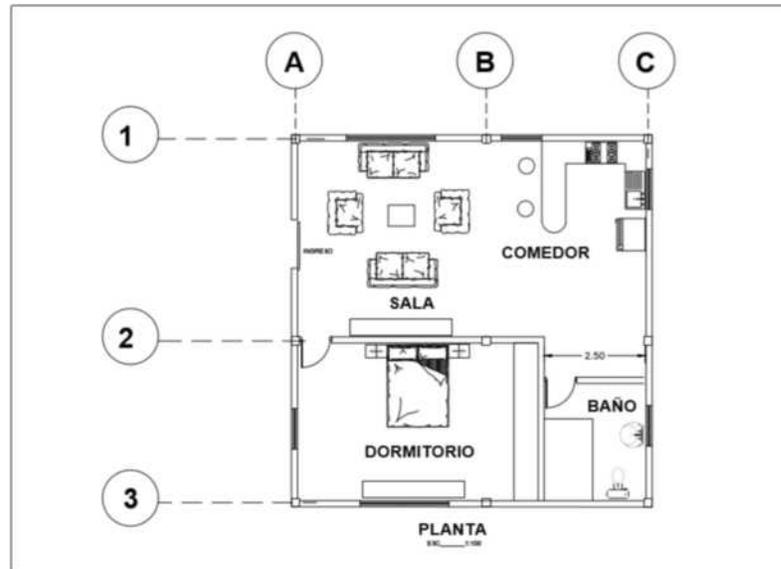
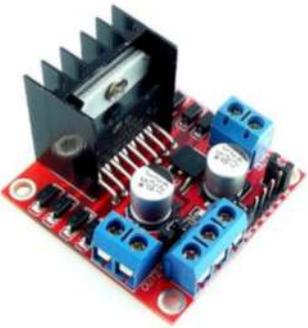


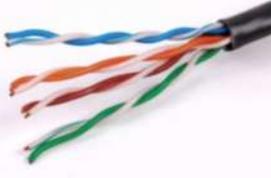
Gráfico 12-2: Diseño Maqueta.

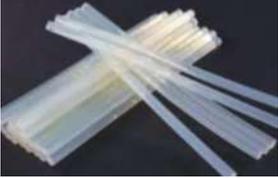
Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

En el ensamblaje del prototipo se han utilizado varios materiales y componentes electrónicos para lograr un correcto funcionamiento. En la siguiente tabla se muestra los materiales utilizados en el ensamblaje.

Tabla 1-2. Materiales utilizados.

NOMBRE	Componente	Características
Led 10mm white		<ul style="list-style-type: none"> • Voltaje: 3.0 - 3.4V • Corriente: 20 mA • Dimensión: 10mm (10 milímetros) • Color Blanco (alta luminosidad)
Resistencia 1kΩ		<ul style="list-style-type: none"> • Resistencia: 1kΩ • Potencia: ½ W • Tolerancia: 5% • Tipo: Carbón
Diodo 1N4148		<ul style="list-style-type: none"> • Corriente de avance de 4^a en una anchura de impulso 1us y 1^a en anchura de impulso 1s • La disipación de energía es 500mW • Voltaje inverso máximo de 100V • Voltaje de 1V a 10mA
Transistor 2N222A	 <p style="text-align: center;">2N222A</p>	<ul style="list-style-type: none"> • Polaridad del transistor: NPN • Rango de corriente: 0.8 A • Voltaje nominal: 40 V
Puente H L298N		<ul style="list-style-type: none"> • Canales: 2 (soporta 2 motores DC) • Voltaje lógico: 5V • Voltaje de operación: 5V-35V • Consumo de corriente (digital): 0 a 36mA • Capacidad de corriente: 2 A • Potencia máxima: 25W • Dimensiones: 43* 43* 27 mm

<p>Motor corriente DC 0-12V</p>		<ul style="list-style-type: none"> • Voltaje nominal: 0-12V • Velocidad : 4800 rpm • Corriente: 0.025 A
<p>Motor Corriente DC 5V</p>		<ul style="list-style-type: none"> • Voltaje nominal: 5V • Velocidad: 2650 rpm • Corriente : 0.01 A
<p>Cable UTP Categoría 5</p>		<ul style="list-style-type: none"> • Es el cable de pares trenzados más utilizado en la implementación de Redes, no posee ningún tipo de protección adicional a la recubierta de PVC y tiene una impedancia de 100 Ohm y utiliza el conector RJ-45
<p>Soldadura de cables aleación estaño/plomo</p>		<ul style="list-style-type: none"> • Diámetro: 1.0 mm • Aleación: Estaño/plomo
<p>Pasta para soldar</p>		<ul style="list-style-type: none"> • Peso : 25gr • Cuenta con fundente rápido y agresivo, que limpia, proporcionando la adherencia de la soldadura.
<p>Cautín Pistola de madera</p>		<ul style="list-style-type: none"> • Voltaje 110 V • Tipo lápiz/pistola • Potencia 30W • Mango de madera

<p>Cables Jumpers Macho Hembra</p>		<ul style="list-style-type: none"> • Conectores: Plug-jack (macho-hembra) • Calibre cable: 26(AWG) • Largo cable: 7" (17.78 cm) • Largo conector: 1" (2.54 mm)
<p>Barra de silicona</p>		<ul style="list-style-type: none"> • Barra de silicona transparente, para pistola aplicadora. • Diámetro: 7 mm.
<p>Pistola</p>		<ul style="list-style-type: none"> • Pistola termofusible de silicona.
<p>Termocontraible</p>		<ul style="list-style-type: none"> • Termocontraible • Diámetro: 3mm • Dimensión: 1mt
<p>Piñón de plástico</p>		<ul style="list-style-type: none"> • Piñón recto • Diámetro alojamiento del eje: 4.20 mm • Diámetro externo del piñón: 12.15 mm
<p>Cremallera para engranajes</p>		<ul style="list-style-type: none"> • Numero de dientes : 83
<p>Electrodo 2.5mm</p>		<ul style="list-style-type: none"> • Diametro: 2.5 mm • Dimensiones: 36x6x3 cm

<p>Tubo flexible de Agua</p>		<ul style="list-style-type: none"> • Diámetro externo: 4 mm • Diámetro interno : 2.9 mm
----------------------------------	---	---

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

2.3.2 Diseño del circuito del prototipo.

El gráfico 13-2 muestra el circuito completo del prototipo y los distintos elementos electrónicos que se conectan a los puertos GPIO.

El circuito está dividido en tres partes:

1. Circuito de control de giro de los motores
2. Circuito elevador de potencia para las luces
3. Circuito elevador de potencia ventilador y calefactor

Para cada actuador le corresponde una señal de entrada por parte de los puertos GPIO.

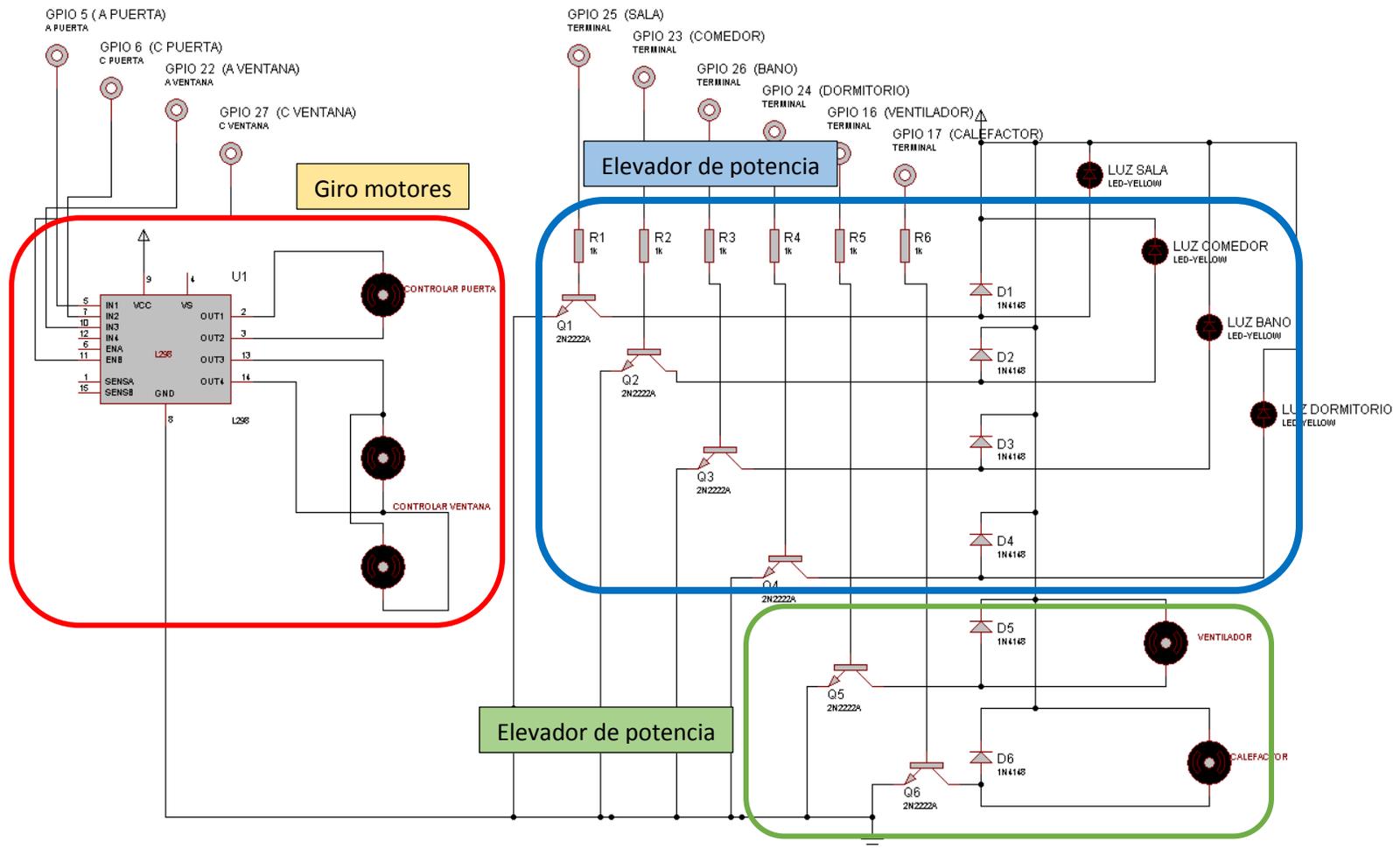


Gráfico 13-2: Circuito general.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

2.3.4 Elementos Hardware.

2.3.4.1 Instalación de los Diodos Led

Los diodos led representan la iluminación de cada ambiente (dormitorio, sala, baño y comedor), simulando a un departamento, en lugar de usar instalaciones eléctricas de 110 V se reemplaza por diodos led blancos (alta luminosidad) de 3V-3.4V para esta maqueta.

Debido a que el voltaje de los puertos GPIO no es suficiente para abastecer una completa luminosidad a los diodos led se implementó un circuito elevador de potencia como se muestra en el gráfico 14-2.

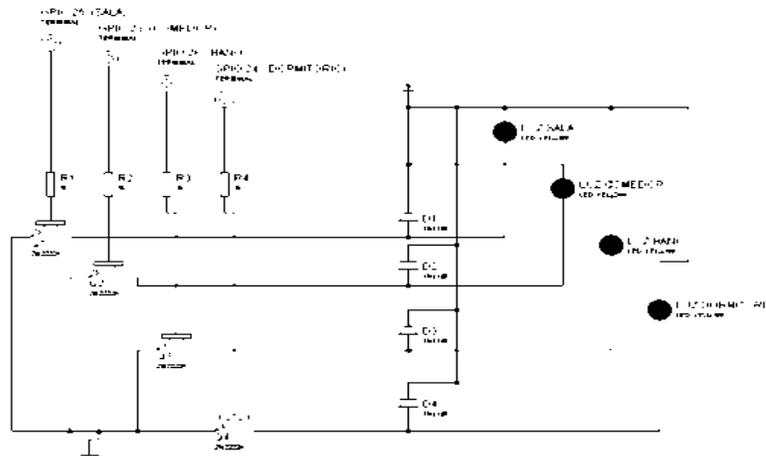


Gráfico 14-2: Circuito de potencia diodo led.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

Debido a que la intensidad proporcionada por los GPIOs es de 16 mA es necesario un circuito que eleve la intensidad, la cual se describe en la figura 15-2

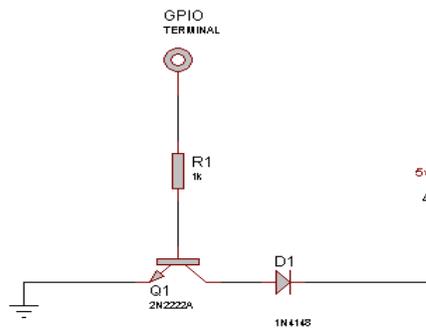


Gráfico 15-2: Circuito Potenciador de intensidad de corriente.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

La Raspberry y los actuadores (Motor, led) se alimentan con fuentes independientes de 5 V y 3.4 V respectivamente. La Raspberry envía un pulso a la base del transistor el cual dejará pasar corriente entre su colector y emisor, cerrando el circuito y permitiendo que los actuadores se enciendan o se apaguen dependiendo de la señal emitida por los GPIOs. Cuando se reduce el voltaje que produce un pico de voltaje elevado que se convierte en un riesgo para la Raspberry, este diodo evita dicho efecto conocido como Flyback Diode.

Una vez que el circuito se encuentra listo, se realizan las conexiones a cada puerto GPIO asignado para iluminar cada ambiente de la maqueta. Los hilos del cable UTP se encuentran soldados a cada uno de los leds, para cada pin positivo de los leds se hace punto de suelda con un cable de diferente color y para el pin negativo se usó un cable en común de color blanco. Como se muestran en los gráficos 16-2 y 17-2



Gráfico 16-2: Conexiones Diodo led.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017



Gráfico 17-2: Cableado diodos led.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

Los cables deben ser recubiertos con el termocontraible y como se puede apreciar en el gráfico 18-2



Gráfico 18-2: Cableado recubierto en la maqueta.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

2.3.4.2 Instalación ventilador

Para simular un ventilador en la maqueta se dispuso de un motor DC de 0-5 V. La conexión del circuito se puede observar en los gráficos 19-2 y 20-2.

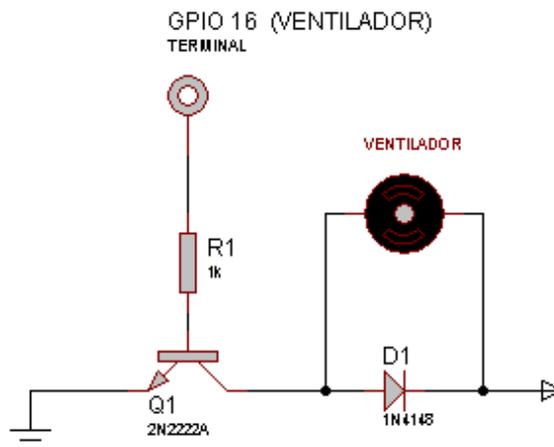


Gráfico 19-2: Circuito Ventilador.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017



Gráfico 20-2: Ventilador.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

2.3.4.3 Instalación Calefactor

En el calefactor se utilizó el mismo circuito para encender el led, cabe recalcar que el calefactor se simulo con un diodo led de color rojo como se puede apreciar en los gráficos 21-2 y 22-2.

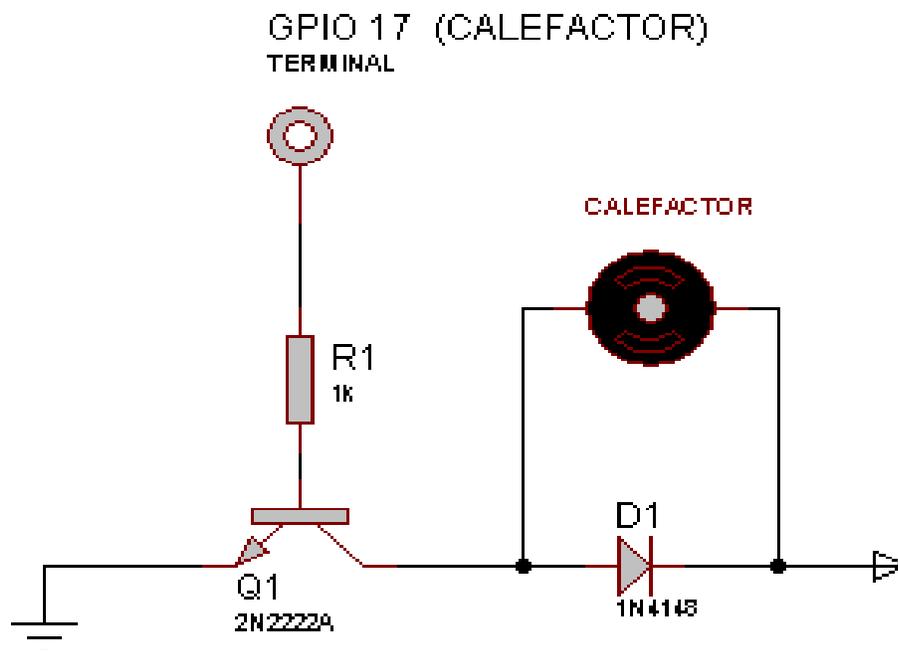


Gráfico 21-2: Circuito calefactor.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017



Gráfico 22-2: Calefactor.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

2.3.4.4 Instalación Puerta

Para la instalación de la puerta en la maqueta se realizó los siguientes pasos:

1. Colocar un piñón en la punta del motor de 12 V como se muestra en el gráfico 23-2.



Gráfico 23-2: Motor con piñón.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

2. Cortar el tubo flexible una distancia de 15 cm (ancho de la puerta), también un corte con la misma medida a la cremallera de engranajes. Como se muestra en el gráfico 24-2



Gráfico 24-2: Pieza puerta.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

Para el funcionamiento de los motores se utilizó el módulo L298N el cual utiliza una fuente externa de 9 V que es conectada a la entrada de 6 -12 V del módulo. Posee dos canales puente H para controlar el giro y la activación de los motores que se usan en puerta y ventana.

En el gráfico 25-2 se muestra el circuito del motor que controla la puerta.

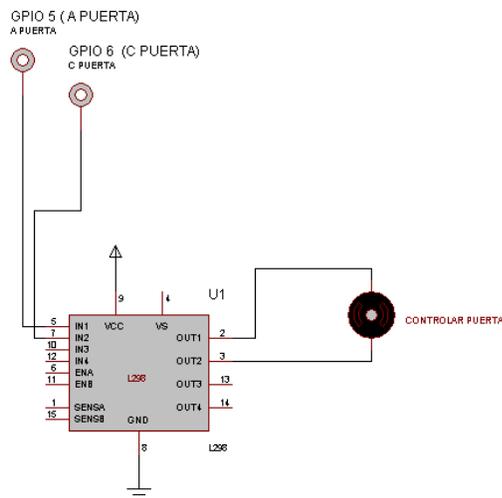


Gráfico 25-2: Circuito motor puerta.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

Las salidas del motor están conectados al módulo L298N donde sus pines IN1 e IN2 tienen la función de controlar el motor a través de la señal que emita los puertos GPIO a la que estén conectadas.

2.3.4.5 Instalación ventana.

La instalación de la ventana en la maqueta es similar al de la puerta, para lo cual se tomó en cuenta los siguientes aspectos:

1. Se utilizó dos motores DC de 9 V en vez de uno como se muestra en gráfico 26-2.



Gráfico 26-2: Motores de la ventana.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

2. Para la ventana se realizó el corte de dos piezas con el fin de simular la apertura y cerrado de la misma. En el grafico 27-2 se muestra el proceso.



Gráfico 27-2: Montaje de la ventana.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

En el gráfico 28-2 se muestra el circuito de los motores que controlan la ventana.

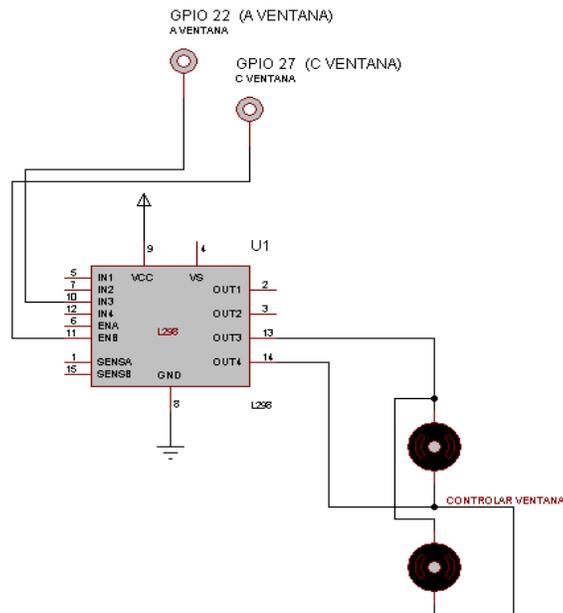


Gráfico 28-2: Circuito motores ventana.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

2.4 Funcionamiento.

Para el desarrollo de las aplicaciones del prototipo se utilizaron los lenguajes de programación html, php, Python y javascript.

El funcionamiento del prototipo se detalla en los siguientes pasos:

1. El usuario debe acceder a la aplicación web desde el navegador Google Chrome con la URL “https://192.168.0.200/rpi”.
2. El fichero WebSpeech.js activa el micrófono y selecciona el idioma a trabajar el gráfico 29-2 muestra las líneas de comando que se mencionó.

```
window.onload = function () {  
    var recognition = new webkitSpeechRecognition();  
    recognition.continuous = true;  
    recognition.interimResults = true;  
    recognition.lang = "es";  
    recognition.start();  
}
```

lenguaje Español

Activa el micrófono

Gráfico 29-2: Activación lenguaje y micrófono.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

3. WebSpeech.js posee un evento llamado `recognition.onresult` el cual tiene la particularidad de que la información emitida al micrófono sea reconocida y almacenada. En el gráfico 30-2 se muestra el evento `onresult`.

```
recognition.onresult = function (event) {  
    console.log("onresult");  
    var interimResult = '',  
        finalResult = '';  
    for (var i = event.resultIndex; i < event.results.length; ++i) {  
        if (event.results[i].isFinal) {  
            finalResult = event.results[i][0].transcript;  
        } else {  
            interimResult += event.results[i][0].transcript;  
        }  
    }  
    finalTranscript.value = finalResult;  
    areaResult.value = interimResult;  
}
```

Evento onresult

Gráfico 30-2: Evento onresult.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

4. WebSpeech.js es llamado por el fichero `index.html`, el cual posee 2 elementos `input` donde el primer `input` va reconociendo y el segundo muestra el resultado final de la orden emitida por el usuario. Las líneas mencionadas se puede observar en el gráfico 31-2.

```

</script>
<script src="WebSpeech.js"></script>
<center>
<section> <br />
<input placeholder="ESCUCHANDO" name="nombre1" id="areaResult"><br />
<input placeholder="TU ORDEN" name="nombre" style="text-transform:uppercase;" id="finalTranscript" ><br />
<br />
<br />
</section>
<h4 align="center">RESPUESTA</h4>
<div id="resultado"></div>

```

Llamado del fichero WebSpeech.js

Orden emitida

Gráfico 31-2: Elementos input.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017,

- En este paso el usuario puede empezar a dictar órdenes o simplemente hablar. Si el micrófono no detecta ninguna señal durante los primeros 6 segundos este se desactiva por esta razón en el fichero index.html se implantó la siguiente línea la que se resalta en el gráfico 32-2.

```

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<meta http-equiv="refresh" content="6">
<title>Reconocimiento de voz </title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
<link rel="stylesheet" type="text/css" href="css/estilos.css" />
<script src="js/jquery-1.10.1.min.js"></script>

```

Refresca la aplicación web

Gráfico 32-2: Refresca la aplicación web.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

- Una vez que el usuario haya emitido su orden a la aplicación esta empezara a receptor y enviar la orden. La orden debe compararse con el fichero procesa.php. El comando que envía la orden es " var intervalo=setInterval(función a enviar, tiempo de envío)" como se puede apreciar en el gráfico 33-2.

```

ajax.setRequestHeader("content-type","application/x-www-form-urlencoded");
ajax.send("varp2="+ valor )
}
var intervalo=setInterval('Enviar()',4000);

```

Envía la variable cada 4s

Gráfico 33-2: Comando de envío automático.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

7. Si el usuario no emite las ordenes establecidas para activar los actuadores, la aplicación web le responderá con un mensaje “comando no valido”, caso contrario responderá con los mensajes establecidos en fichero procesa.php. En el gráfico 34-2 se observa una parte del código procesa.php

```
case 'encender ventilador':
    echo "VENTILADOR ENCENDIDO";
    exec('sudo python /var/www/html/rpi/prendeventilador.py');
    break;
case 'Apagar ventilador':
    echo "VENTILADOR APAGADO";
    exec('sudo python /var/www/html/rpi/apagaventilador.py');
    break;
default:
    echo 'COMANDO NO VALIDO';
    break;
}
```

Gráfico 34-2: Ordenes establecidas del fichero procesa.php.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

8. En el caso de que la orden emitida coincida con las opciones establecidas por el fichero procesa.php, se ejecutaran los archivos Python los cuales activaran los actuadores correspondientes. En el gráfico 35-2 se puede observar un archivo Python.

```
pi@raspi: ~
GNU nano 2.2.6 Fichero: prenedormitorio.py

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(24, GPIO.OUT)
GPIO.output(24, True)
```

Gráfico 35-2: Fichero prenedormitorio.py.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

El proceso de funcionamiento se puede observar en el siguiente diagrama de flujo:

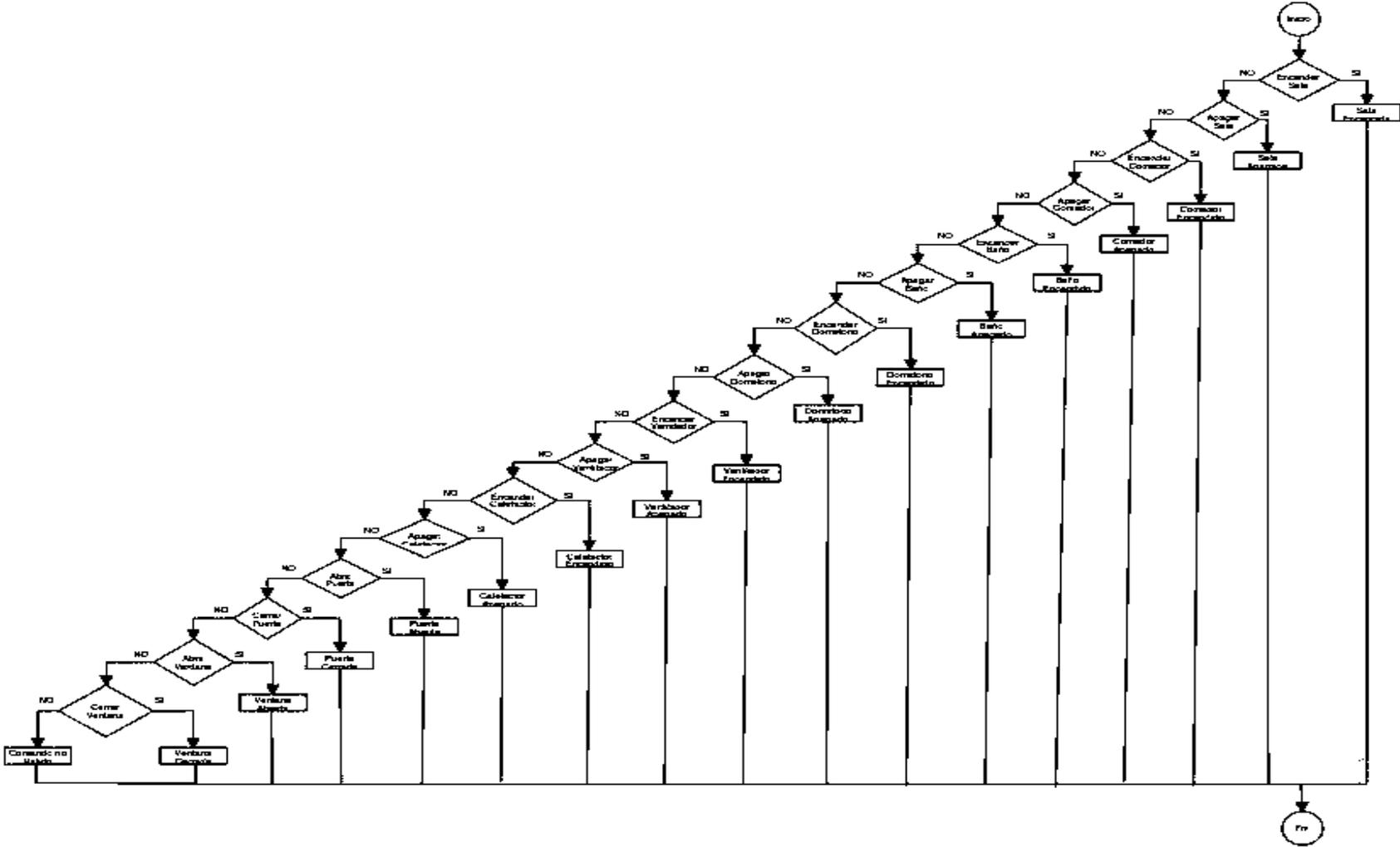


Gráfico 36-2: Diagrama de flujo del prototipo.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

CAPÍTULO III

3 PRUEBAS, COSTOS Y RESULTADOS

3.1 Pruebas de funcionamiento

Una vez terminado el prototipo y el montaje del mismo, se realizó las respectivas pruebas para comprobar su funcionamiento. Se conectó el prototipo a la red Wifi “TPLINK”, los diversos actuadores a su circuito y a sus puertos GPIO correspondientes se hizo prueba de funcionamiento de la siguiente manera:

A. Prueba luces.

- Encendido y apagado de la luz del dormitorio.

Como se puede apreciar en el gráfico 1-3 el usuario emitió la orden “Encender dormitorio” y “Apagar dormitorio”.

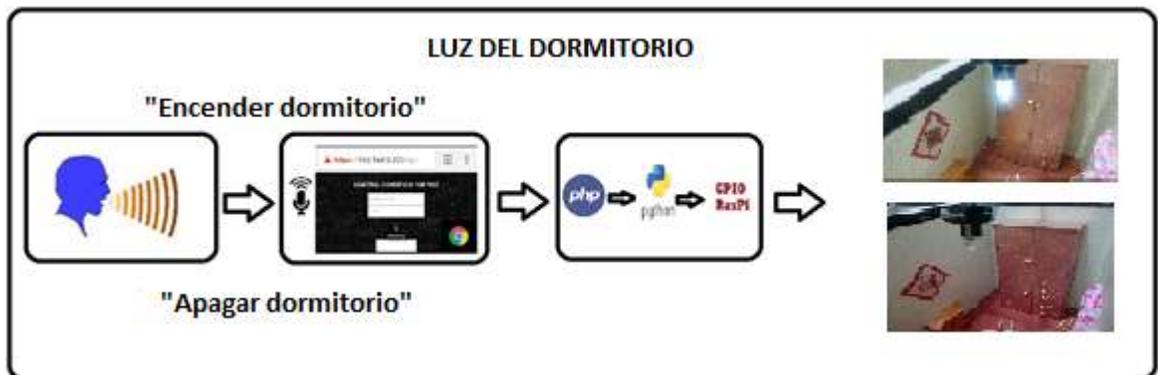


Gráfico 1-3: Encendido y apagado de la luz del dormitorio.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

- Encendido y apagado de la luz del baño.

Como se puede apreciar en el gráfico 2-3 el usuario emitió la orden “Encender baño” y “Apagar baño”.

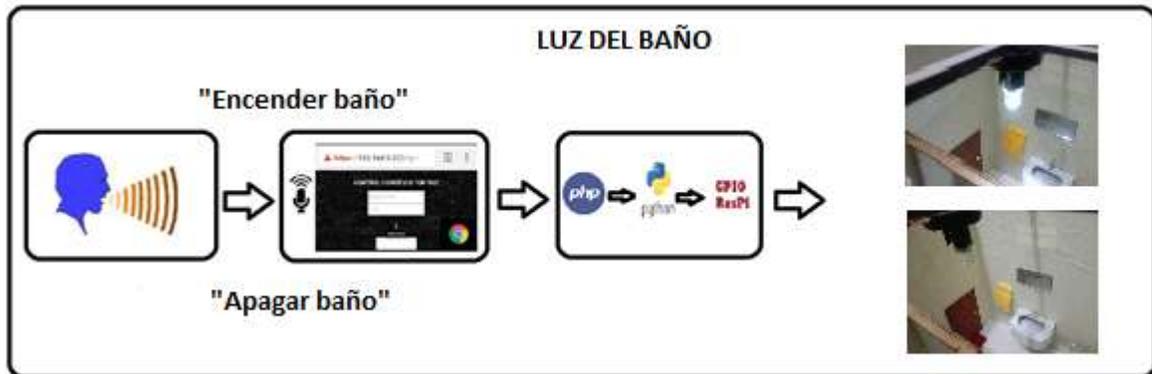


Gráfico 2-3: Encendido y apagado de la luz del baño.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

- Encendido y apagado de la luz del comedor.

Como se puede apreciar en el gráfico 3-3 el usuario emitió la orden “Encender comedor” y “Apagar comedor”.

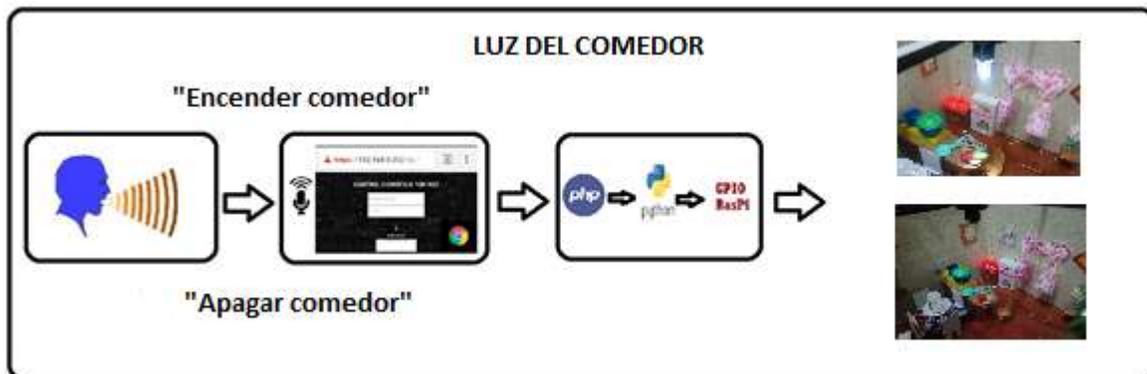


Gráfico 3-3: Encendido y apagado de la luz del comedor.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

- Encendido y apagado de la luz de la sala.

Como se puede apreciar en el gráfico 4-3 el usuario emitió la orden “Encender sala” y “Apagar sala”



Gráfico 4-3: Encendido y apagado de la luz de la sala.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

B. Prueba ventilador

Para encendido y apagado del ventilador el usuario emitió la orden “Encender ventilador” y “Apagar ventilador”. En el gráfico 5-3 se puede apreciar su funcionamiento.

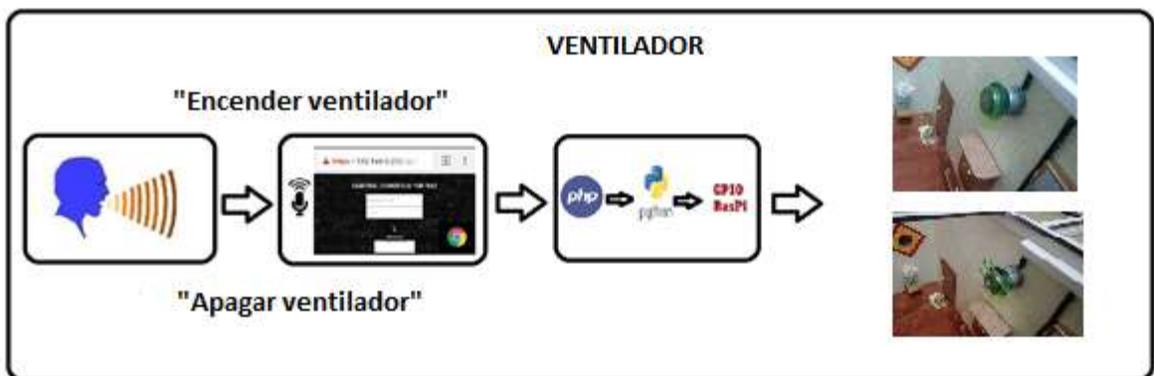


Gráfico 5-3: Encendido y apagado del ventilador.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

C. Prueba calefactor

Para encendido y apagado del calefactor el usuario emitió la orden “Encender calefactor” y “Apagar calefactor” como se puede observar en el gráfico 6-3.

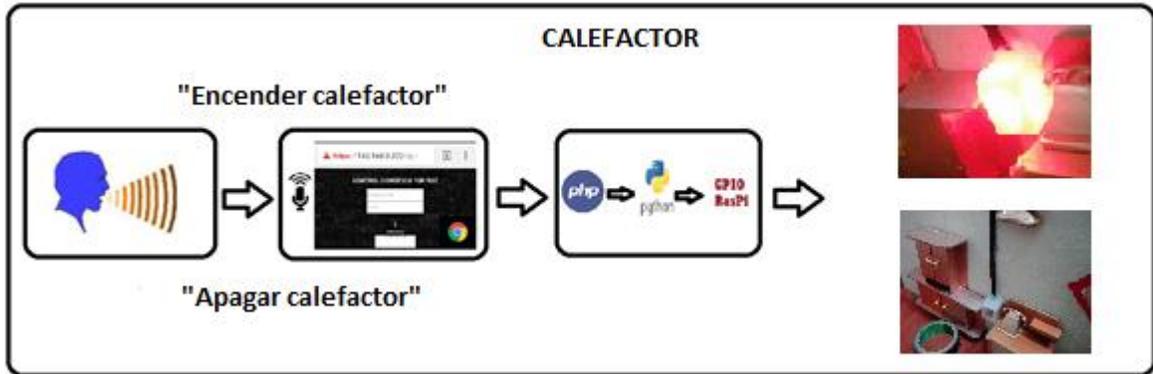


Gráfico 6-3: Encendido y apagado del calefactor.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

D. Prueba Puerta

Para abrir y cerrar la puerta el usuario emitió la orden “Abrir puerta” y “Cerrar puerta” como se puede observar en el gráfico 7-3.

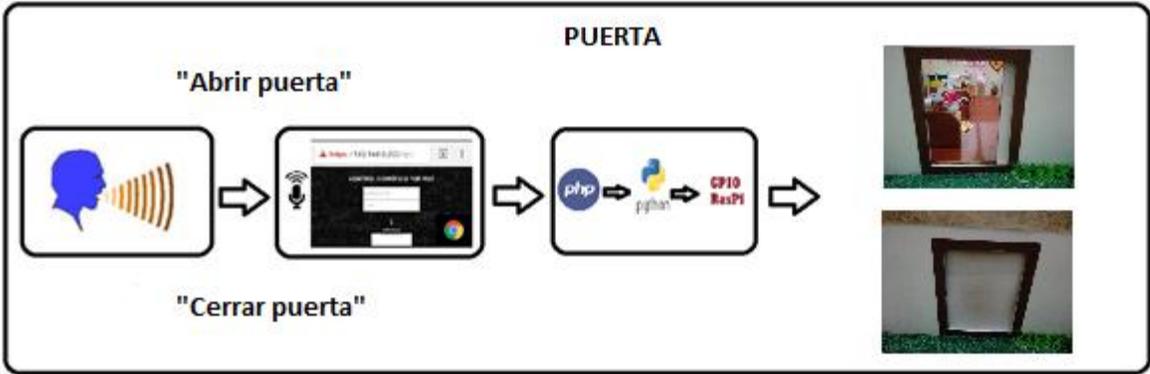


Gráfico 7-3: Apertura y cerradura de la puerta.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

E. Prueba ventana

Para abrir y cerrar la ventana el usuario emitió la orden “Abrir ventana” y “Cerrar ventana” como se puede observar en el gráfico 8-3.

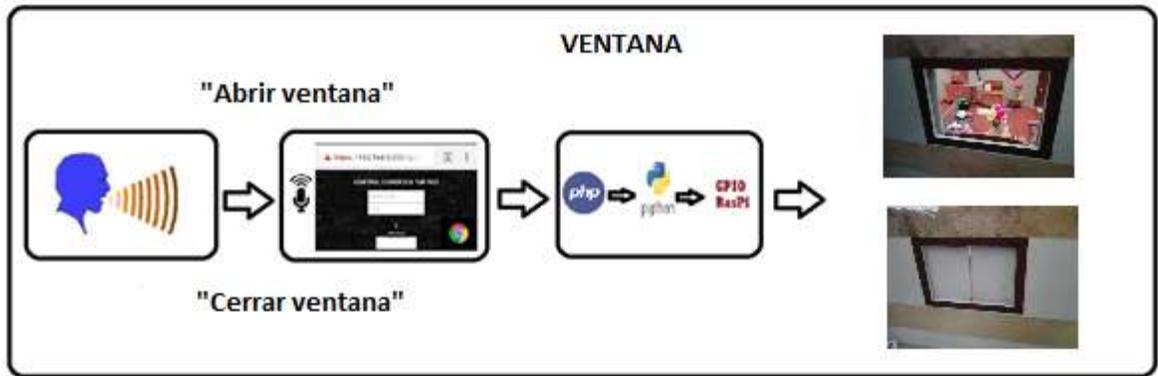


Gráfico 8-3: Apertura y cerradura de la ventana.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

3.2 Resultados

Según el Consejo Nacional para la Igualdad de Discapacidades en el Ecuador existen 187708 personas con alguna discapacidad física comprendidos entre 7 años en adelante, con una discapacidad de 30% a 84%, de los cuales 5121 se encuentran en la provincia de Chimborazo y 2632 en la ciudad de Riobamba, de esta población se calculó la muestra para realizar una encuesta sobre la confiabilidad del prototipo.

Datos:

$N = 2632$ (La población)

Nivel de confianza del 80%

$k = 1,28$

$p = 0,5$

$q = 1 - p = 0,5$

$e = 15\% = 0,15$ (margen de error de 15%)

$$n = \frac{k^2 p q N}{(e^2(N - 1)) + k^2 p q}$$

$$n = \frac{(1,28)^2 (0,5) (0,5) 2632}{((0,15)^2(2632 - 1)) + (2632)^2 (0,5) (0,5)}$$

$$n = \frac{(1,64) (0,25) 2632}{((0,0225)(2631)) + (1,64) (0,25)}$$

$$n = \frac{1079,12}{59,1975 + 0,41}$$

$$n = \frac{1079,12}{59,6075}$$

$$n = 18,1$$

El tamaño de la muestra es de 18,1 se optó por escoger una muestra de 18 personas para la encuesta.

3.2.1 Confiabilidad del prototipo

La encuesta se realizó a 18 personas con y sin discapacidad motriz, de las cuales la mayoría no posee alguna discapacidad motriz debido a que fue difícil contactar a personas que tengan discapacidad motriz.

Las valoraciones del prototipo estuvieron enfocadas a estos parámetros

- Facilidad de manejo.
- Eficiencia del prototipo.
- Utilidad del prototipo.

Las pruebas que se realizaron fueron las siguientes:

1. En la ejecución de los comandos de voz para el encendido y apagado de las luces de los diferentes ambientes (dormitorio, baño, sala, comedor).
2. En la ejecución de los comandos de voz para el encendido y apagado del ventilador.
3. En la ejecución de los comandos de voz para el encendido y apagado del calefactor).
4. En la ejecución de los comandos de voz para abrir y cerrar la puerta.
5. En la ejecución de los comandos de voz para abrir y cerrar ventanas.

Tabla 1-3 Costos materiales del prototipo

Número de Personas	Facilidad de manejo			Eficiencia del prototipo			Utilidad del prototipo	
	Fácil	Regular	Difícil	Excelente	Regular	Deficiente	SI	NO
1	x			x			x	
2	x			x			x	
3	x			x			x	
4		x		x			x	
5	x			x			x	
6	x			x			x	
7	x				x			x
8	x			x			x	
9		x		x			x	
10		x		x				x
11	x			x			x	
12		x		x			x	
13	x			x			x	
14	x			x			x	
15	x			x				x
16		x		x			x	
17	x			x			x	
18	x				x		x	
Total	13	5	0	16	2	0	15	3
Porcentaje	72%	28%		89%	11%		83%	17%

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

- Según la calificación de los usuarios la facilidad de manejo del prototipo tuvo un 72% Fácil, un 28 % índico que es regular y un 0% de que difícil, por lo que el prototipo no tiene una dificultad mayor en su manejo. Revisar gráfico 9-3.



Gráfico 9-3: Facilidad de manejo del prototipo.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

- Ante la eficiencia los usuarios indicaron que el prototipo cuenta con una eficiencia de 89 %, 11 % regular y 0% deficiente esto indica que el prototipo es eficiente. Revisar el gráfico 10-3.



Gráfico 10-3: Eficacia del prototipo.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

- Por último se cuestionó si le parece útil que el prototipo sea implantado en un hogar teniendo un nivel de aceptación del 83 % y un 17% de oposición. Revisar el gráfico 11-3.



Gráfico 11-3: Utilidad del prototipo.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

3.2.2 Tiempo de respuesta del prototipo.

A. Tiempo de respuesta día 1

Tabla 2-3 Tiempo de respuesta de encendido y apagado a 2 m.

Tiempo de respuesta del prototipo a 2 m																
Tiempo (s)	Luz del baño		Luz del dormitorio		Luz del comedor		Luz de la sala		Ventilador		Calefactor		Puerta		Ventana	
Pruebas	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado
1	2,14	3,04	2,21	2,4	2,29	2,18	2,65	2,51	3,27	2,8	2,67	2,46	2,53	2,51	2,93	2,7
2	2,89	2,64	2,71	2,66	2,41	2,07	2,43	2,5	2,63	2,74	2,3	2,07	2,68	2,98	2,9	2,76
3	2,82	1,82	1,98	2,11	2,57	2,33	2,8	2,51	2,79	2,97	2,28	2,33	2,74	2,72	2,58	2,35
4	2,41	2,39	2,12	2,41	2,52	2,29	2,55	2,52	3,19	2,91	2,6	2,5	2,77	2,72	2,47	2,54
5	2,65	2,74	2,55	2,48	2,39	2,33	2,47	2,71	2,92	2,55	2,33	2,48	2,72	2,8	2,56	2,82
6	2,97	3,02	2,43	2,23	2,42	2,28	2,3	2,48	2,81	2,46	2,12	2,15	2,44	2,98	2,75	2,49
7	2,76	2,73	2,49	2,18	2,17	1,95	2,7	2,62	3,14	2,32	2,37	2,55	2,92	2,59	2,44	3
8	2,59	2,71	2,12	2,13	2,08	1,99	2,78	2,52	3,26	2,71	2,45	2,1	2,58	2,65	2,61	2,72
9	2,33	2,55	2,11	2,39	2,46	2,24	2,79	2,86	2,74	2,4	2,44	2,25	2,99	3	2,35	2,78
10	3,08	2,7	2,66	2,02	2,53	2,39	2,9	2,4	2,96	2,32	2,66	2,28	2,4	2,89	2,35	2,59
Promedio	2,66	2,63	2,34	2,30	2,38	2,21	2,64	2,56	2,97	2,62	2,42	2,32	2,68	2,78	2,59	2,68

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

Como se puede observar de los datos medidos en la tabla 2-3 se obtuvieron los promedios de cada uno de los actuadores tanto encendido como en el apagado.

Tabla 3-3 Tiempo de respuesta de encendido y apagado a 15 m.

Tiempo de respuesta del prototipo a 15 m																
Tiempo (s)	Luz del baño		Luz del dormitorio		Luz del comedor		Luz de la sala		Ventilador		Calefactor		Puerta		Ventana	
Pruebas	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado
1	2,84	2,63	2,29	2,41	2,26	2,28	2,9	2,83	2,49	2,44	2,74	2,74	2,72	2,7	2,42	2,52
2	2,69	2,95	2,56	2,7	2,3	2,15	2,62	2,55	2,61	4,74	2,64	2,69	2,7	3,19	2,57	2,72
3	2,87	2,8	2,4	2,36	2,33	1,93	2,35	2,68	2,77	2,74	3,15	2,58	1,76	2,59	2,16	2,54
4	2,78	2,63	2,4	2,14	2,67	2,3	2,6	2,27	2,3	2,99	2,71	2,83	2,69	3,16	2,61	2,42
5	2,66	2,63	2,55	2,72	2,58	2,57	2,55	2,7	2,54	2,72	2,76	3,18	2,57	2,43	2,6	2,73
6	2,44	2,79	2,59	2,27	2,41	2,43	2,57	2,41	2,78	2,69	2,47	2,44	2,73	1,78	2,89	2,87
7	2,95	2,44	2,26	1,95	2,47	1,79	2	2,98	2,62	3,55	2,49	2,77	2,54	2,66	2,8	2,66
8	2,4	2,76	2,29	2,39	2,34	2,52	3,48	2,56	2,89	2,87	2,54	2,75	2,72	2,54	2,67	2,88
9	2,69	2,54	2,12	2,48	2,6	2,57	2,58	2,65	2,8	2,95	2,37	2,8	3,19	2,29	3,49	2,3
10	2,57	2,67	2,13	2,61	2,55	2,31	2,61	3,50	2,8	3,3	2,25	2,45	2,66	2,66	2,5	2,67
Promedio	2,69	2,68	2,36	2,40	2,45	2,29	2,63	2,63	2,66	3,10	2,61	2,72	2,63	2,60	2,67	2,63

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

Como se puede observar de los datos medidos en la tabla 3-3 se obtuvieron los promedios de cada uno de los actuadores tanto encendido como en el apagado

B. Tiempo de respuesta día 2

Tabla 4-3 Tiempo de respuesta de encendido y apagado a 2 m.

Tiempo de respuesta del prototipo a 2 m																
Tiempo (s)	Luz del baño		Luz del dormitorio		Luz del comedor		Luz de la sala		Ventilador		Calefactor		Puerta		Ventana	
Pruebas	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado
1	2,81	2,93	2,41	2,12	2,51	2,1	2,27	2,5	2,12	2,19	2,38	2,69	2,16	2,59	2,89	2,39
2	2,18	2,29	2,44	2,79	2,6	2,58	2,54	2,76	2,37	2,64	2,45	2,87	2,98	2,69	2,52	2,25
3	2,46	2,24	2,86	2,13	2,33	2,9	2,15	2,52	2,35	2,1	2,42	2,62	2,61	2,34	2,6	2,43
4	2,73	2,36	2,68	2,46	2,44	2,42	2,26	2,33	2,26	2,95	2,77	2,65	2,51	2,19	2,92	2,23
5	2,44	2,53	2,5	2,3	2,61	2,36	2,78	2,96	2,37	2,64	2,78	2,1	2,92	2,93	2,7	2,74
6	2,73	2,58	2,78	2,77	2,17	2,12	2,45	2,61	2,18	2,96	2,44	2,48	2,61	2,26	2,57	2,39
7	2,58	2,85	2,13	2,21	2,94	2,63	2,43	2,66	2,69	2,93	2,24	2,24	2,47	2,68	2,2	2,16
8	2,28	2,32	2,53	2,95	2,29	2,63	2,67	2,18	2,4	2,72	2,3	2,63	2,81	2,13	2,61	2,19
9	2,93	2,62	2,1	2,41	2,43	2,13	2,33	2,56	2,62	2,29	2,85	2,24	2,82	2,12	2,14	2,98
10	2,66	2,4	2,64	2,93	2,76	2,3	2,87	2,3	2,27	2,1	2,85	2,97	2,34	2,84	2,26	2,53
Promedio	2,58	2,51	2,51	2,51	2,51	2,42	2,48	2,54	2,36	2,55	2,55	2,55	2,62	2,48	2,54	2,43

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

Como se puede observar de los datos medidos en la tabla 4-3 se obtuvieron los promedios de cada uno de los actuadores tanto encendido como en el apagado

Tabla 5-3 Tiempo de respuesta de encendido y apagado a 15 m.

Tiempo de respuesta del prototipo a 15 m																
Tiempo (s) Pruebas	Luz del baño		Luz del dormitorio		Luz del comedor		Luz de la sala		Ventilador		Calefactor		Puerta		Ventana	
	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado	Encendido	Apagado
1	2,35	2,63	2,16	2,94	2,19	2,52	2,8	2,29	2,5	2,33	2,97	2,41	2,51	2,79	2,61	2,85
2	2,19	2,73	2,22	2,19	2,45	2,69	2,67	2,7	2,37	2,98	2,26	2,66	2,58	2,84	2,83	2,43
3	2,48	2,42	2,69	2,69	2,63	2,21	2,33	2,68	2,32	2,75	2,36	2,95	2,82	2,58	2,12	2,56
4	2,14	2,08	2,14	2,31	2,71	2,59	2,58	2,8	2,4	2,49	2,42	2,32	2,45	2,23	2,27	2,63
5	2,67	2,72	2,11	2,1	2,89	2,38	2,89	2,17	2,85	2,58	2,87	2,32	2,63	2,77	2,52	2,9
6	2,11	2,27	2,8	2,45	2,07	2,69	2,62	2,45	2,72	2,5	2,09	2,77	2,82	2,78	2,61	2,21
7	2,97	2,37	2,83	2,12	2,71	2,81	2,26	2,21	2,85	2,57	2,74	2,92	2,87	2,5	2,21	2,36
8	2,1	2,6	2,38	2,1	2,93	2,57	2,6	2,74	2,33	2,4	2,44	2,4	2,57	2,61	2,18	2,27
9	2,91	2,9	2,66	2,16	2,71	2,46	2,77	2,5	2,63	2,26	2,42	2,82	2,72	2,76	2,76	2,78
10	2,77	2,97	2,4	2,2	2,86	2,86	2,9	2,28	2,97	2,12	2,48	2,1	2,86	2,45	2,72	2,81
Promedio	2,47	2,57	2,44	2,33	2,62	2,58	2,64	2,48	2,59	2,50	2,51	2,57	2,68	2,63	2,48	2,58

Realizado por: VEINTIMILLA, *Alvaro* & YUNGA, *Carlos*, 2017

Como se puede observar de los datos medidos en la tabla 5-3 se obtuvieron los promedios de cada uno de los actuadores tanto encendido como en el apagado

Para el análisis estadístico del tiempo de respuesta del prototipo se obtuvo datos del encendido y apagado de los actuadores utilizados y medidos en segundos. Los datos fueron tomados cada hora durante 10 horas por un lapso de dos días.

Con los datos obtenidos se realizó un promedio del tiempo de respuesta de los distintos actuadores para luego comparar el funcionamiento del prototipo a distintas distancias y comprobar su eficiencia.

La medición de la señal de los puertos GPIO se realizó mediante el programa brainbox el cual permite monitorear en tiempo real la actividad de los puertos GPIO.

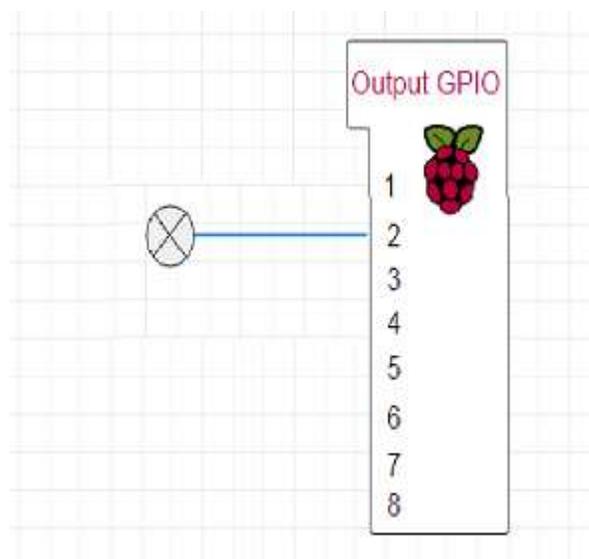


Gráfico 12-3: Señal digital apagado GPIO.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

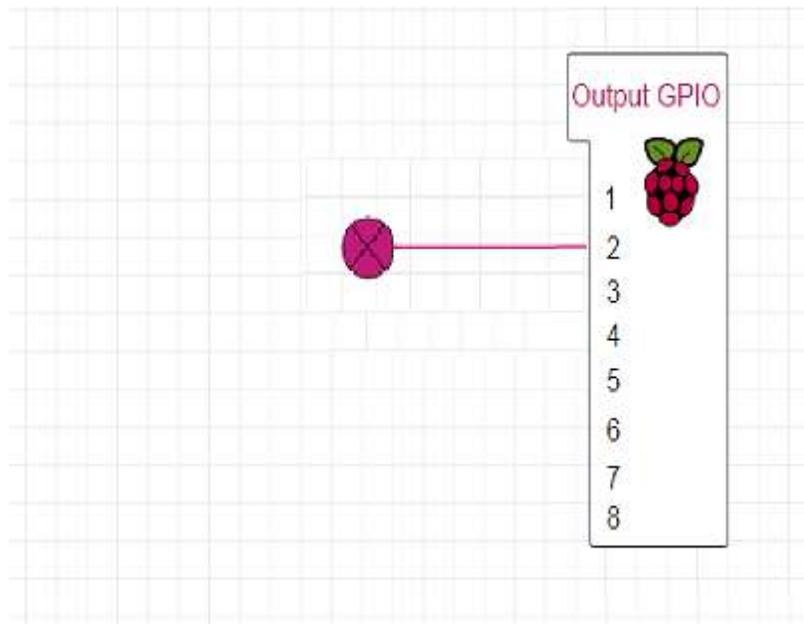


Gráfico 13-3: Señal digital encendido GPIO.

Realizado por: VEINTIMILLA, Alvaro & YUNGA, Carlos, 2017

3.3 CONCLUSIONES

El diseño e implementación del prototipo está desarrollado con la placa Raspberry Pi 3 modelo B, que permite montar servidores web de distribución libre como es Raspbian, y a su vez tener una comunicación inalámbrica con los puertos GPIO conectados a los actuadores.

Después de estudiar las diferentes plataformas del control domótico, se concluyó que existen un sin número de dichas plataformas que cumplen distintas funciones dependiendo de las necesidades del usuario, sin embargo muchas de ellas no están 100% orientadas a personas con alguna discapacidad motriz o tienen un costo elevado en el mercado, por lo cual para este prototipo se desarrolló su propia plataforma domótica.

En la implementación de la interfaz hardware-software, la aplicación web desarrollada actúa como puente de comunicación entre el usuario y los actuadores, encargada de interpretar las órdenes mediante el fichero WebSpeech.js el cual transforma la voz a texto y activa el micrófono del terminal.

Para el diseñar la activación de los eventos físicos se programó las órdenes en Python, las cuales activan los puertos GPIO, dichas órdenes se encuentran alojadas el fichero procesa.php.

Para evaluar los datos obtenidos se realizó pruebas de funcionamiento y el análisis estadístico del prototipo domótico, para así concluir que el tiempo de respuesta es óptimo tanto al inicio de su implementación como después de un tiempo de funcionamiento. Obteniéndose tiempos de respuesta muy similares hasta una distancia de 15 m, con una variación promedio de 0.07 segundos.

3.4 RECOMENDACIONES

Para realizar las pruebas de funcionamiento se recomienda realizar un circuito elevador de potencia para cada actuador ya que la intensidad suministrada por los puertos GPIO no abastece a dichos elementos.

Es necesario conectar una fuente de alimentación estable de 5V con un amperaje superior a los 700 mA para el funcionamiento óptimo de la Raspberry y a su vez una fuente de alimentación externa de 9V-1A para el módulo L298N.

Al momento de realizar las diferentes conexiones es de vital importancia conectar el pin GND del módulo L298N al negativo de la fuente de 9V y a su vez al GND de la Raspberry para cerrar el circuito.

Se recomienda programar en el fichero index.html un tiempo de recarga de la aplicación Web cada intervalo de tiempo dependiendo de las necesidades del usuario, en este caso 6s debido a que si el micrófono no recepta ninguna señal de audio se desactiva automáticamente, evitando que el usuario tenga la necesidad de recargar la aplicación web manualmente.

Es necesario programar en el fichero index.html la sección de envío de la variable “finalTrascript”, con un tiempo de envío menor al tiempo de recarga, de esta manera el usuario no tiene la necesidad de presionar ningún botón para enviar su orden.

Para controlar la Raspberry sin necesidad de usar periféricos, se recomienda establecer una dirección ip estática, para acceder remotamente siempre con la misma dirección ip.

En el control de los motores utilizados en el prototipo es necesario contar con un PWN digital, este se debe programar en los ficheros de Python que controlan los GPIOs de activación y desactivación de los mismos, colocando una potencia de salida adecuada para evitar un giro excesivo de los motores.

Es recomendable tener un tiempo de encendido y apagado adecuado en la programación de los GPIOs que controlan los motores para evitar que sigan activos una vez que se ejecutó la orden.

Se recomienda para la programación del fichero index.html la utilización del navegador Google Chrome por que para esta aplicación presenta las mejores prestaciones que los demás.

Se recomienda para futuros trabajos usar librerías de reconocimiento de voz locales ya que al usar las librerías de los servidores de Google Chrome el trabajo es limitado por la disponibilidad del servicio de internet.

Se recomienda tener un sistema de respaldo para el control de puertas y ventanas, en el caso de fallar el sistema eléctrico.

GLOSARIO

Actuadores.- Se comportan como interfaces de potencia, son los encargados de llevar a cabo las acciones mecánicas del sistema, se clasifican en dos grupos.

Confort.- Bienestar o comodidad material.

Certificado SSL.- Se puede definir como una forma de seguridad que mantiene la información encriptada y segura de páginas y aplicaciones web. Acrónimo de Secure Socket layer.

Discapacidad motriz.- Es una condición de vida que afecta el control y movimiento del cuerpo, generando alteraciones en el desplazamiento, equilibrio, y manipulación de las personas que la padecen.

Domótica.- Se puede definir a la domótica como una agrupación de sistemas, que proporcionan las herramientas necesarias para automatizar una vivienda.

Fichero.- Conjunto de información que se almacena de forma virtual para ser leído por un ordenador .

GPIO.- Puerto que posee la Raspberry pi para la comunicación con dispositivos externos.
Acronimo de General Purpose input/output.

Habla continua.- Facultad de una persona para hablar de forma fluida teniendo un vocabulario muy amplio usualmente ilimitado.

Html.- Lenguaje que se utiliza comúnmente para establecer, configurar la estructura y contenido de páginas web. Acrónimo de Hiper Text Markup language (lenguaje de marcación de hipertexto).

Plataforma domótica.- Son los servicios que ofrecen las distintas empresas dedicadas al expendio de sistemas domóticos para viviendas.

Protocolo SSH.- Protocolo que facilita las comunicaciones seguras entre dos sistemas por medio de la arquitectura cliente servidor permitiendo al usuario la comunicación remota con un host determinado. Acrónimo de Secure Shell.

Raspberry Pi.- Placa de desarrollo de proporciones pequeñas que soporta varios componentes de un ordenador común.

Raspbian.- Sistema operativo libre basado en Debian más utilizado para la tarjeta de desarrollo Raspberry Pi.

Reconocimiento de voz.- Es una técnica con la cual podemos, interpretar el significado de nuestras palabras, con el objetivo de ejecutar órdenes o tomar decisiones.

Servidor web.- Programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales, unidireccionales y síncronas con el cliente.

Unidireccional.- Acción donde el usuario habla mientras la máquina realiza una acción como respuesta.

URL.- Es la dirección específica que se asigna a cada uno de los recursos disponibles en la red. Acrónimo de Uniform Resource Locator (Localizador uniforme de recursos).

BIBLIOGRAFÍA

Alegsa, L. "Definición de Motor de renderizado web (de interpretación)", *Informática y Electrónica* [En línea], 2018, (Argentina) 12(2), pp 132-136. [Consulta: 23 Marzo 2018]. Disponible en: <http://www.alegsa.com.ar/Dic/motor_de_renderizado_web.php>

Alvarez, C., & Cajas, M. Desarrollo de una interfaz para el reconocimiento de comandos de voz aplicado al control de sistemas domóticos [En línea] (Trabajo de titulación). (Ingeniería) Universidad Politécnica Salesiana, Cuenca, Ecuador. 2016. pp 93-95. [Consulta: 2017-12-18]. Disponibl en: <https://dspace.ups.edu.ec/bitstream/123456789/12363/1/UPS-CT006460.pdf>

Barahona, E., & Huilcapi, D. (2015). Diseño e implementación de un sistema de control domótico supervisado por un teléfono móvil mediante la utilización de ASTERISK [En línea] (Trabajo de titulación). (Ingeniería) Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2015. pp 35-38. [Consulta: 2017-12-20]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/5051>

Calle, P. Desarrollo de un sistema didáctico aplicado a inversores trifásicos, empleando el sistema embebido raspberry pi mediante la técnica de modulación vectorial espacial [En línea] (Trabajo de titulación). (Ingeniería) Universidad Politécnica Salesiana, Cuenca, Ecuador. 2014. pp 67-70. [Consulta: 2017-12-18]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/7063/1/UPS-CT003798.pdf>

Camargo, J., García, L., & Gaona, E. Reconocimiento de voz humana aplicado a la domótica [En línea] (Trabajo de titulación). (Ingeniería) Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. 2012. pp 97-106. [Consulta: 2017-11-26]. Disponible en: <https://www.revistavirtualpro.com/biblioteca/reconocimiento-de-voz-humana-aplicado-a-la-domotica>

Falconi, L., & Jimenez, C. Estudio e implementación de domótica activada por comandos de voz y comunicación en zigbee [En línea] (Trabajo de titulación). (Ingeniería) Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2009. pp 92-96. [Consulta: 2018-01-10]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/165>

Flores, F., & Robles, J. Estudio, diseño e implementación de módulos de entrenamiento sobre plataforma Arduino para el Laboratorio de Microcontroladores de la Facultad de Ingeniería Electrónica de la Universidad Tecnológica Israel [En línea] (Trabajo de titulación). (Ingeniería)

Universidad Tecnológica Israel, Quito, Ecuador. 2012. pp 102-106. [Consulta: 2018-01-12]

Harrington, W. *Learning Raspbian* [En línea] Birmingham-Inglaterra: Packt Publishing, 2015.[Consulta: 20 enero 2018]. Disponible en: <https://books.google.com/books?hl=en&lr=&id=O6HNBgAAQBAJ&oi=fnd&pg=PP1&dq=raspbian&ots=ZG2PnhATWF&sig=0QwaUR3tOkvYgRWReEo5sgesoHo>

Quispe, O. "Tarjetas Para Desarrollo De Hardware". *Lightpath Technologies* [En línea], 2017, (Estados Unidos) 97(2), pp. 27-29. [Consulta: 16 enero 2018]. Disponible en: <http://www.lightpath.io/tarjetas-de-desarrollo/>

Sánchez, A. Diseño de un sistema domótico accionado por voz, Jarvis [En línea] (Trabajo de titulación). (Ingeniería) Universidad Tecnológica de Pereira, Pereira, Colombia. 2015. pp 154-158. [Consulta: 2018-01-18]. Disponible en: <http://repositorio.utp.edu.co/dspace/handle/11059/5785>

Santamaria, P. "Domótica y el futuro que siempre está llegando". *Xataka-Tecnología* [En línea], 2014, (México) 153(2), pp. 15-18. [Consulta: 30 enero 2018]. Disponible en: <https://www.xataka.com/domotica-1/domotica-y-el-futuro-que-siempre-esta-llegando>

Santamaría, P. "Vivint Sky, una nueva plataforma para la domótica del hogar". *Xataka-Tecnología* [En línea], 2014, (México) 148(2), pp. 65-69. [Consulta: 26 enero 2018]. Disponible en: <https://www.xatakahome.com/domotica/vivint-sky-una-nueva-plataforma-para-la-domotica-del-hogar>

Santamarín, D. "Domótica de la mano de Microsoft". *HomeMaestro* [En línea], 2014, (México) 87(2), pp. 56-60. [Consulta: 16 diciembre 2017]. Disponible en: <https://www.xatakamovil.com/futuro/homemaestro-domotica-de-la-mano-de-microsoft>

Torres, H. "Arduino vs Raspberry Pi una comparativa". *Hetpro* [En línea], 2017, (Colombia) 74(2), pp 17-22. [Consulta: 24 enero 2018]. Disponible en: <https://hetpro-store.com/gtkd-df/arduino-vs-raspberry-pi/>

Vargas, J. Modelo de desarrollo abierto para un sistema demótico de tipo centralizado enfocado en el apoyo de perssonas con discapacidad motriz [En línea] (Trabajo de titulación). (Ingeniería) Universidad Autonoma de Aguascalientes, Aguascalientes, México. 2016. pp. 45-47. [Consulta 2017-12-16]. Disponible en: <http://bdigital.dgse.uaa.mx:8080/xmlui/bitstream/handle/123456789/1248/416487.pdf?sequence=1&isAllowed=y>

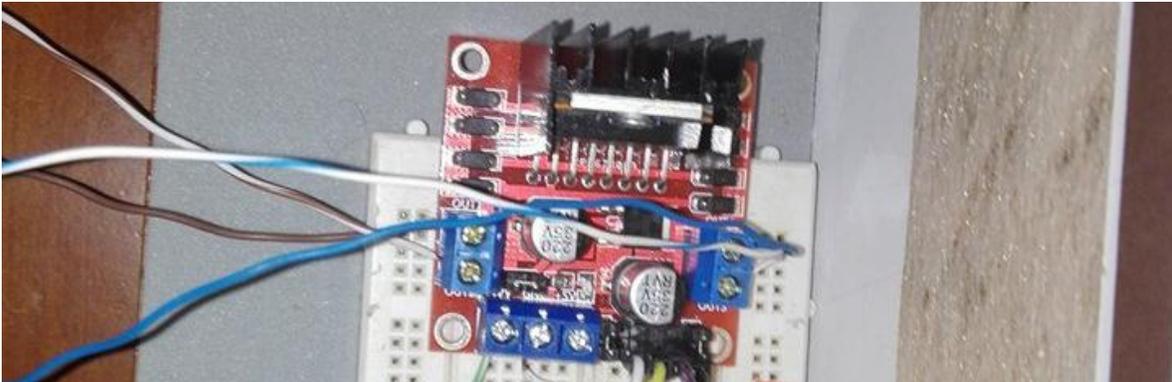
Velasco, R. "Home Assistant, un software OpenSource para controlar la domótica de tu hogar". *Redes@zone* [En línea], 2017, (Colombia) 36(2), pp. 77-81. [Consulta: 20 enero 2018]. Disponible en: <https://www.redeszone.net/2017/11/05/home-assistant-domotica-hogar/>

ANEXOS

Anexo A: Tarjeta de desarrollo Raspberry Pi.



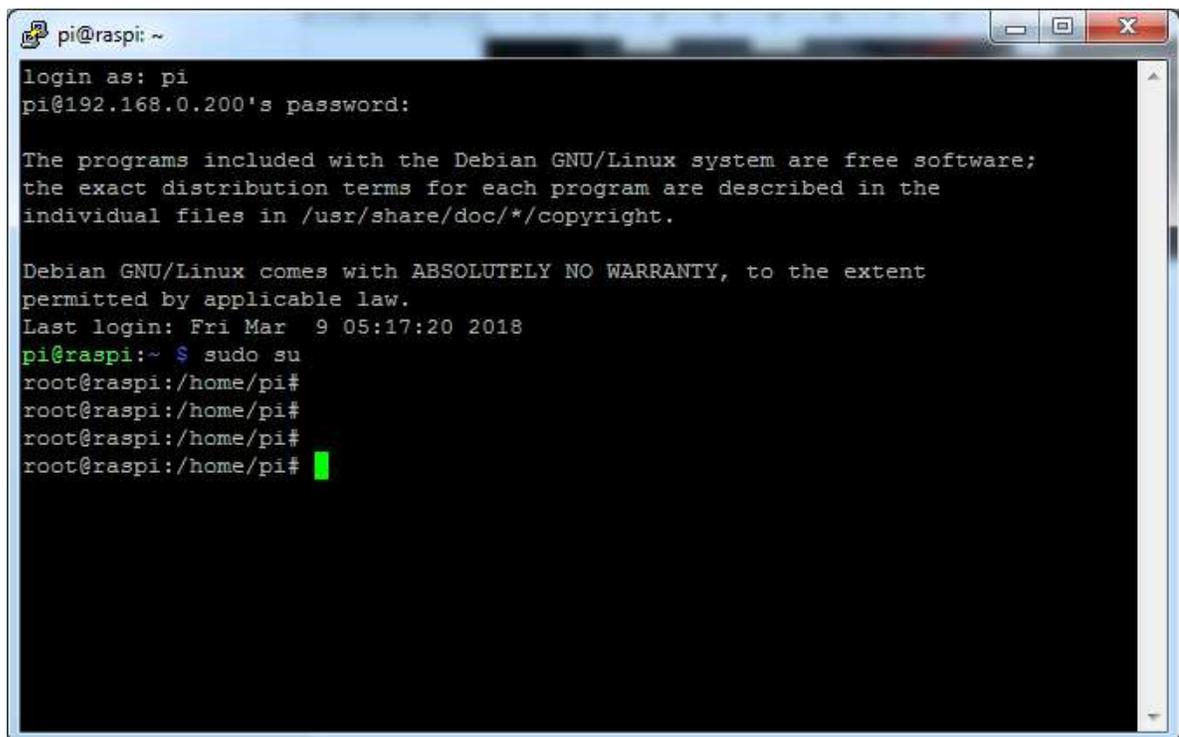
Anexo B: Modulo L298N.



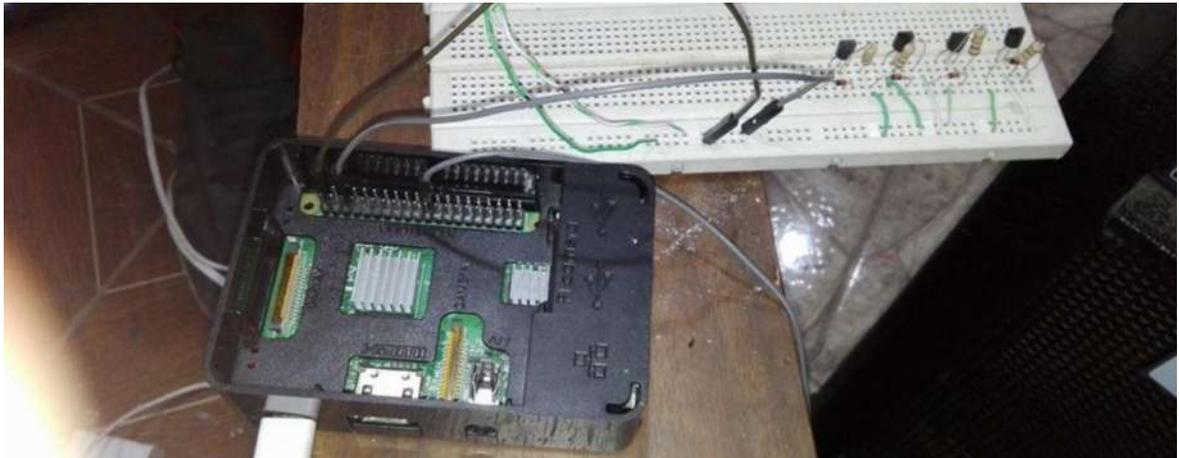
Anexo C: Interfaz gráfica Raspbian.



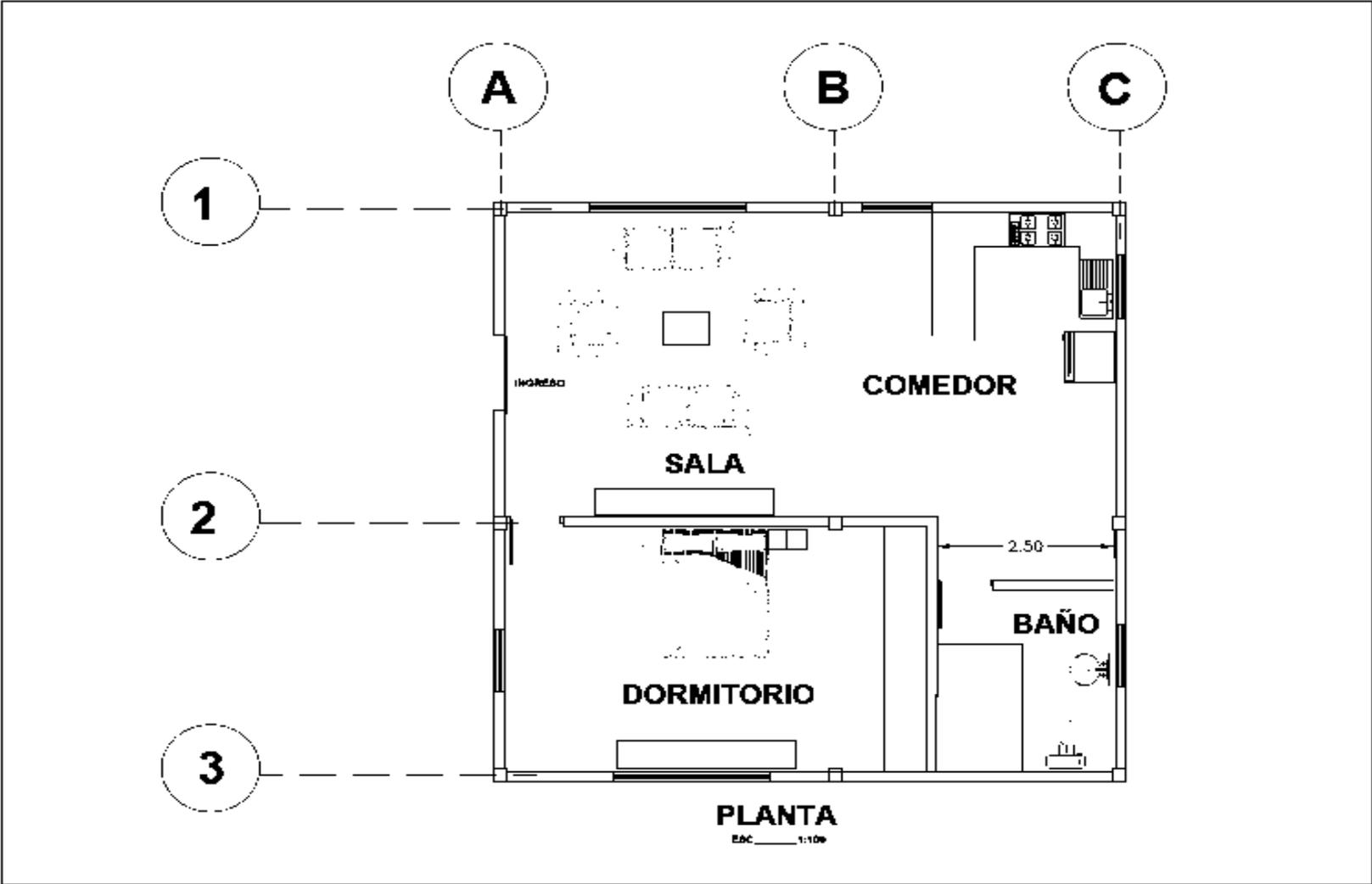
Anexo D: Terminal Raspbian.



Anexo E: Pruebas diodos led.



Anexo F: Planos maqueta



Anexo G: Instalación luces



Anexo H: Instalación ventilador



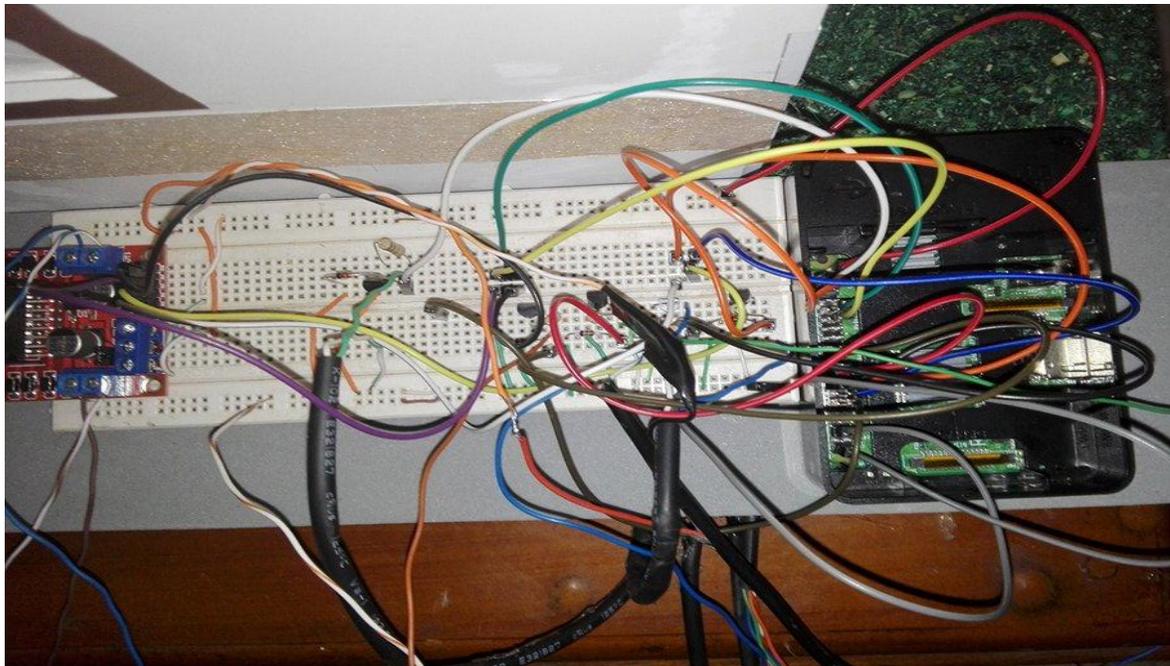
Anexo I: Instalación calefactor.



ANEXO J: Maqueta terminada



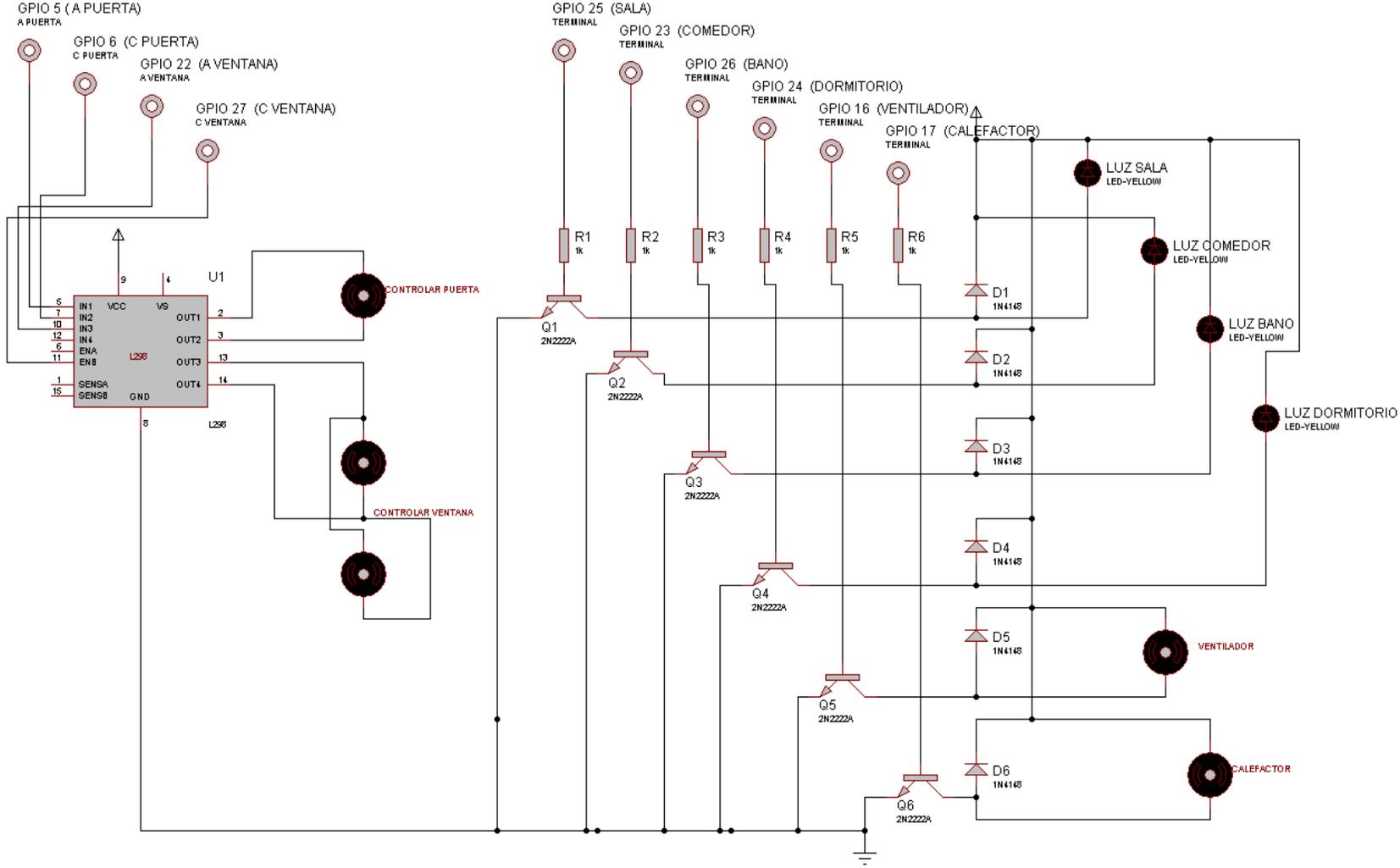
Anexo K: Conexiones de los actuadores.



Anexo L: Case circuito.



Anexo M: Diagrama del circuito.



Anexo N: Aplicación web.



Anexo Ñ: Instalación sistema operativo Raspbian.

Instalación del sistema operativo Raspbian en la Raspberry pi 3B.

Antes de la instalación del Sistema Operativo Raspbian es necesario contar con una tarjeta micro SD de mínimo 2 Gb de almacenamiento y realizar los siguientes pasos:

1. Descargar la última actualización del software alojada en la página oficial de Raspberry Pi, en el siguiente enlace.
<https://www.raspberrypi.org/downloads/raspbian/>.
2. Descomprimir el archivo, para obtener la imagen (.iso) y preparamos la tarjeta micro SD para la instalación.
3. Descargar e instalar SD CARD FORMATER, disponible en el siguiente sitio.
<https://sd-card-formatter.uptodown.com/windows>.

SD Formater sirve para formatear memorias micro SD.

4. Formatear la memoria micro SD usando SD Card Formatter, seleccionar el nombre de la unidad (G:), seleccionar Quick format y dar click en Format. Observar la figura 1



Figura 1: Formateo de la micro SD.

5. Descargar e instalar Win32DiskImager, disponible en el siguiente sitio.

<https://win32-disk-imager.uptodown.com/windows>.

Win32DiskImager es una aplicación de código abierto desarrollada para grabar imágenes CD o DVD en memorias micro SD o en memorias USB y de esta manera crea lectores de disco virtuales.

Para montar la imagen iso basta con colocar la ruta donde se guardó el archivo y la unidad donde se almacenará (G:\), por ultimo dar clic en Write como se muestra en la figura 3

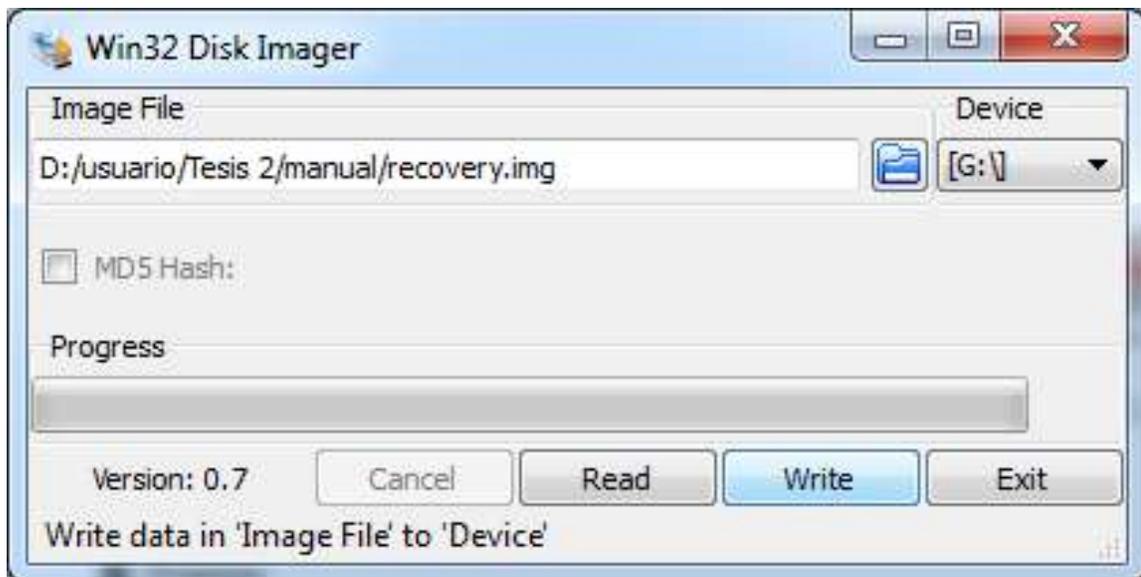


Figura 3: Montaje .iso Win 32.

6. Extraer y colocar la micro SD en el puerto SD de la Raspberry pi.
7. Conectar los diferentes periféricos (monitor, mouse, teclado), y el cable de alimentación (5V - 2A), el montaje de los periféricos debe quedar como se aprecia en la figura 4.



Figura 4: Cables HDMI y cargador de alimentación conectados a la Raspberry.

8. Seleccionar el Sistema Operativo a instalar, en este caso Raspbian y el idioma como se muestra en la figura 5.

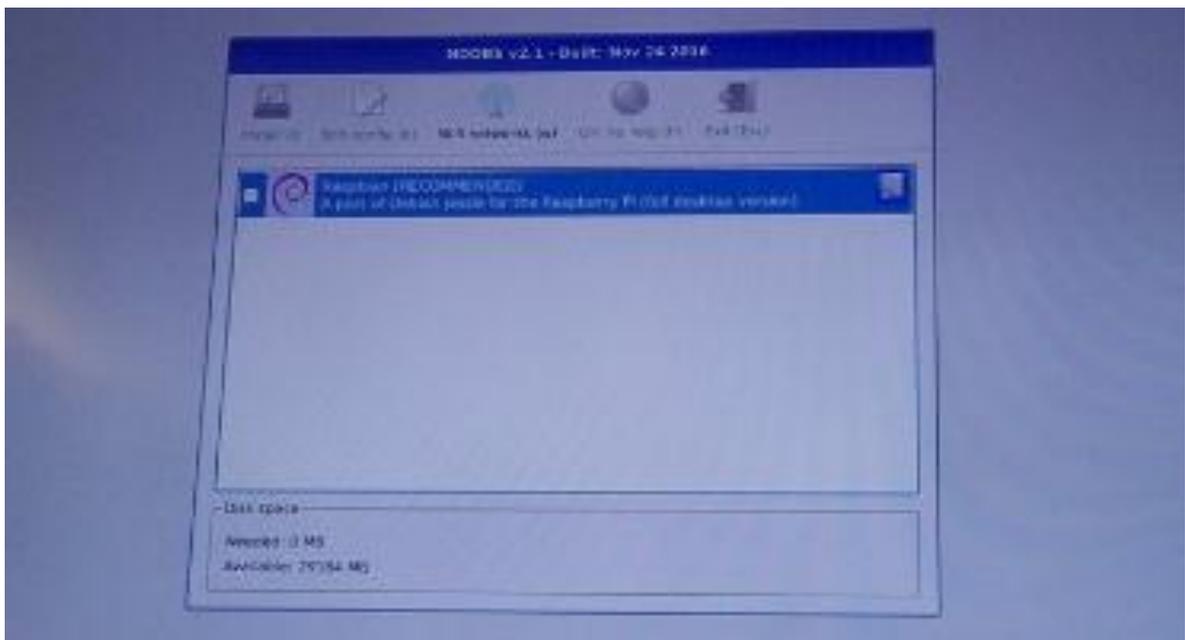


Figura 5: Sistema operativo Raspbian e idioma.

9. Finalmente se instalará el Sistema Operativo, si no existe ningún inconveniente nos mostrará un mensaje de instalación exitosa como en la figura 6.

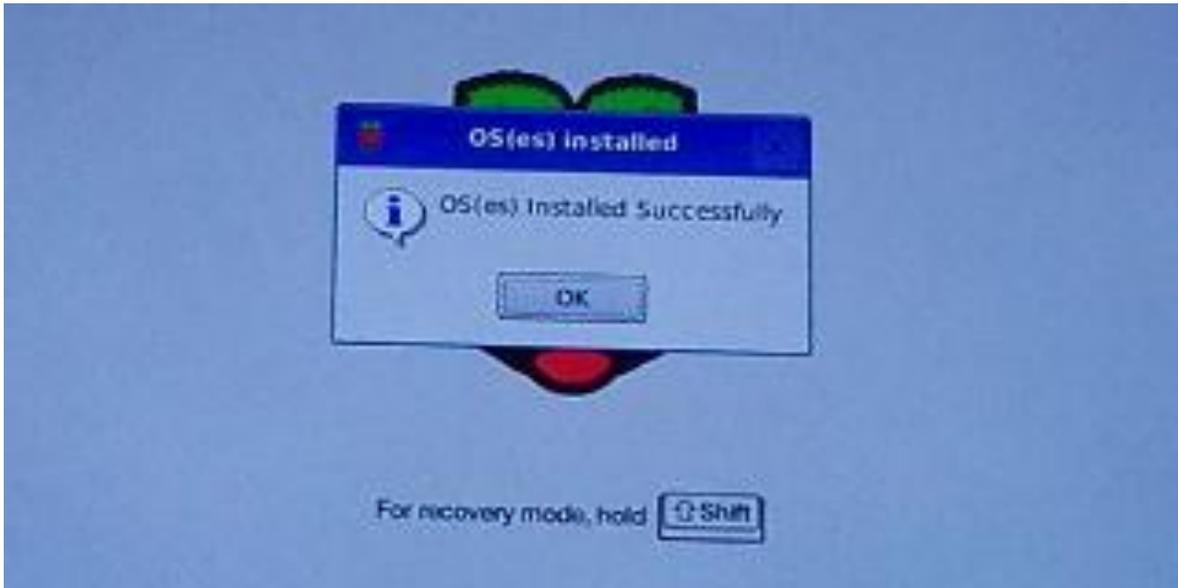


Figura 6: Instalación Raspbian completa.

10. Entorno de la Raspbian.



Figura 7: Entorno grafico Raspbian.

Anexo O: Instalación conexión SSH.

Conexión a través del protocolo SSH

Pasos para instalación del protocolo SSH.

1. Descargar e instalar Advanced IP Scanner, con esta aplicación se puede identificar la dirección IP que tiene asignada la Raspberry Pi, disponible en el siguiente enlace.
<http://www.advanced-ip-scanner.com/es/>

En la figura 8 se puede observar la interfaz del programa Advanced IP Scanner en la cual la IP local asignada a la Raspberry pi es 192.168.1.8.

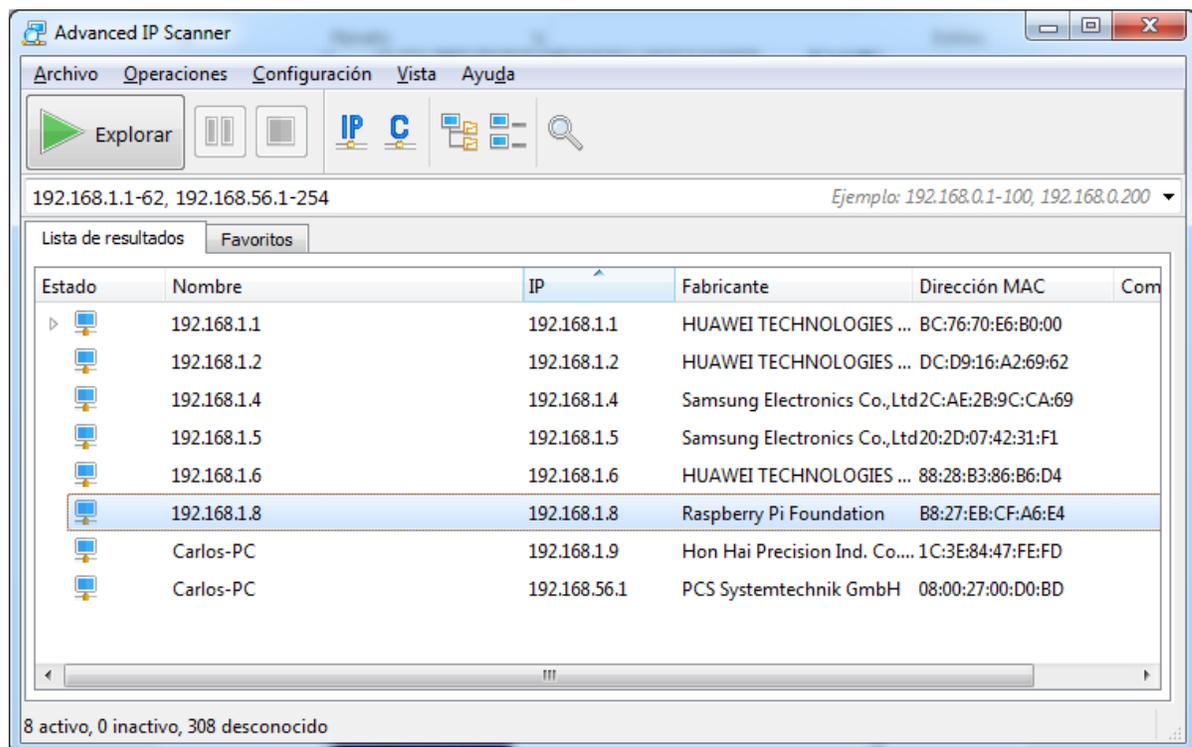


Figura 8: Dirección IP la Raspberry Pi.

2. Descargar e instalar el programa Putty, para establecer una conexión a través de SSH con la Raspberry, disponible en el siguiente enlace.

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>



Figura 9: Conexión con la Raspberry a través de SSH.

- Para lograr la conexión SSH con el programa Putty (Figura 9) colocar la dirección IP local de la Raspberry en la casilla de Host name, seleccionar SSH en Connection Type y damos click en Open, lo cual nos desplegará la interfaz donde se debe colocar el usuario y contraseña de la Raspberry, que por defecto son: nombre de usuario pi, y contraseña: Raspberry cabe recalcar que tanto la contraseña como el usuario no se mostraran en la pantalla como se aprecia en la figura 10.

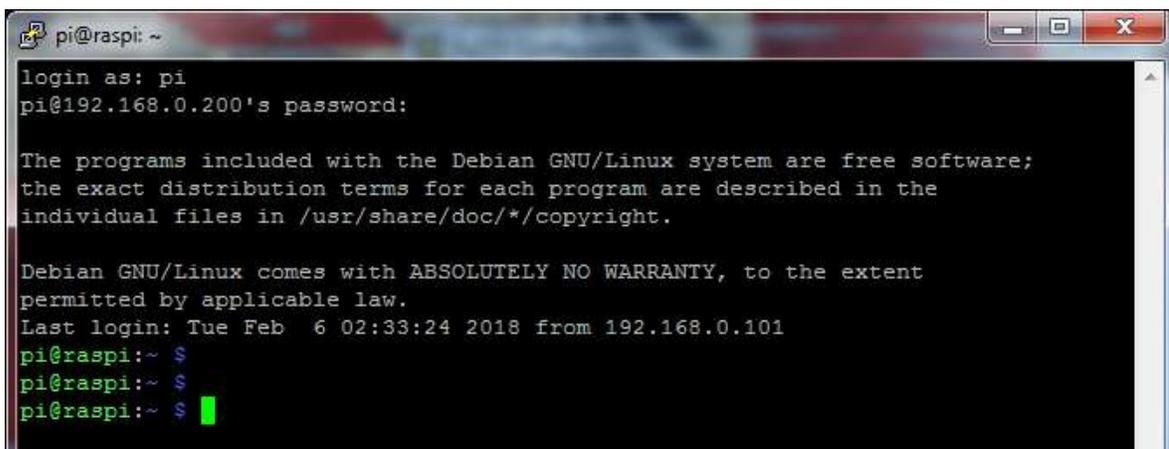
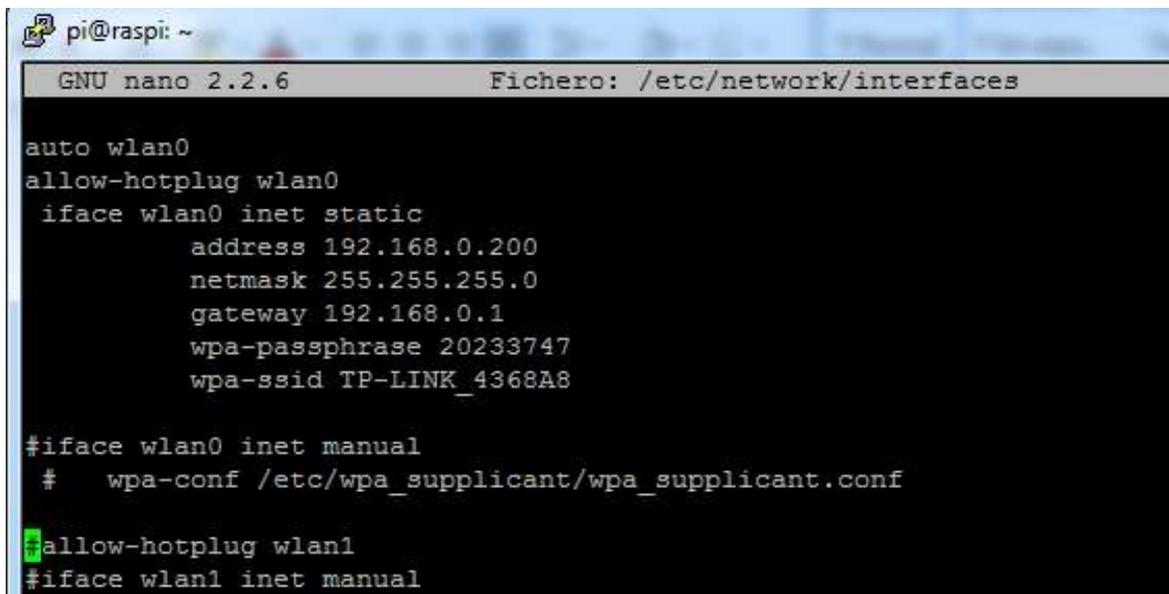


Figura 10: Conexión con la Raspberry a través de SSH con el software PuTTY.

- Para mejorar la comunicación entre la Raspberry y el programa PuTTY a través del protocolo SSH es recomendable asignar una dirección estática en la Raspberry, mediante el siguiente código en el terminal.
 - sudo su
 - nano /etc/network/interfaces
- Ahora comentar estas líneas utilizando #:
 - #iface wlan0 inet manual
 - #wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

6. Escribir los siguientes comandos, donde address es la dirección estática que se asigna a la Raspberry, netmask es la máscara, gateway es la puerta de enlace, wpa-passphrase es la clave de la red y wpa-ssid es el nombre la red como se muestra en la figura 11.

- auto wlan0
- allow-hotplug wlan0
- iface wlan0 inet static
- address 192.168.0.200
- netmask 255.255.255.
- gateway 192.168.0.1
- wpa-passphrase 20233747
- wpa-ssid TP-LINK_4368A8



```
pi@raspi: ~
GNU nano 2.2.6          Fichero: /etc/network/interfaces
auto wlan0
allow-hotplug wlan0
  iface wlan0 inet static
    address 192.168.0.200
    netmask 255.255.255.0
    gateway 192.168.0.1
    wpa-passphrase 20233747
    wpa-ssid TP-LINK_4368A8

#iface wlan0 inet manual
#   wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
#iface wlan1 inet manual
```

Figura 11: Asignación de la dirección estática de la Raspberry.

7. Reiniciar la red con el comando:

- /etc/init.d/networking restart

8. Verificar el cambio realizado utilizando el comando:

- Ifconfig

Se mostrará los cambios realizados como en la figura 12.

```
wlan0    Link encap:Ethernet  HWaddr b8:27:eb:cf:a6:e4
         inet addr:192.168.0.200  Bcast:192.168.0.255  Mask:255.255.255.0
         inet6 addr: fe80::ba27:ebff:febf:a6e4/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:33693 errors:0 dropped:15515 overruns:0 frame:0
         TX packets:13685 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:4456102 (4.2 MiB)  TX bytes:9302262 (8.8 MiB)
```

Figura 12: Dirección ip estática asignada.

Anexo P: Instalación y configuración servidor del servidor WEB.

Instalar y configurar el servidor WEB

Después de realizar la conexión SSH, lo primero es actualizar los repositorios de la Raspberry Pi, para ello se usa el comando “sudo” este comando brinda privilegios de seguridad al momento de ejecutar programas, y seguir los siguientes pasos:

1. Colocar el siguiente comando para actualizar el programa y buscar paquetes requeridos por el usuario a través de repositorios.
 - Sudo apt-get- update
2. Instalar Apache, PHP y libApache2-mod-php5, con la ejecución del siguiente comando:
 - Sudo apt-get install *Apache2* php5 lib*Apache2*-mod-php5
3. Reiniciar el servidor Apache con el comando:
 - sudo /etc/init.d/apache2 restart
4. Para comprobar si efectivamente el servidor esta levantado, direccionar la ip estática en la barra url de un navegador, se muestra la página por defecto del servidor Apache. En la figura 13 se puede apreciar la página por defecto de Apache.

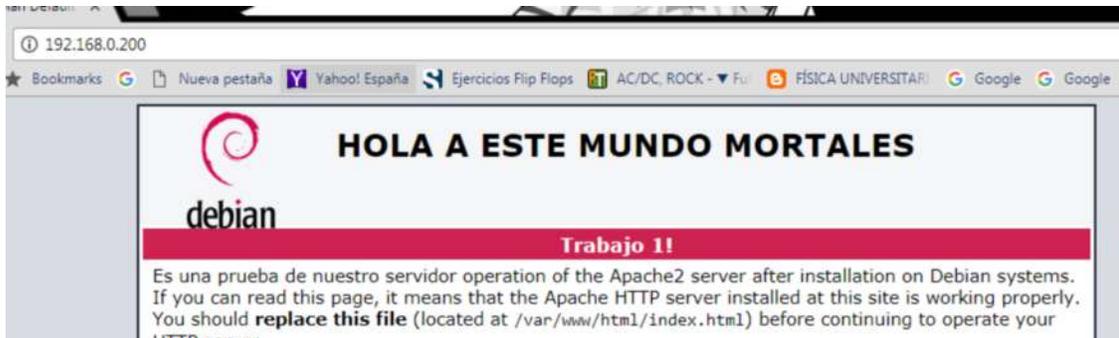


Figura 13: Pagina web por defecto de Apache.

5. Reiniciado el servidor Apache se deben otorgar permisos de ejecución, lectura y escritura al directorio que tiene por defecto. Apache al ser instalado se crea el usuario, un grupo “www-data” y el directorio /var/www aquí es donde se almacena el contenido del servidor, en primera el grupo, usuario y directorio tienen bloqueados los permisos de ejecución, lectura y escritura por eso se procede a desbloquearlos.

Comando para dar permisos al usuario y grupo www-data:

- `sudo chown www-data:www-data /var/www/`

Permisos de ejecución, lectura y escritura al directorio:

- `sudo chmod 775 /var/www/`

Incluir el usuario pi al usuario www-data:

- `sudo usermod -a -G www-data pi`

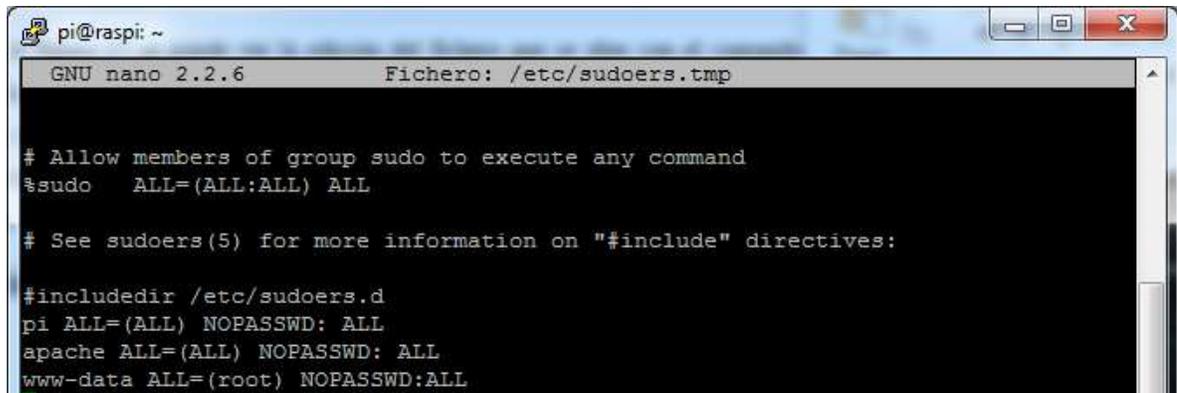
6. Proporcionar permisos root o administrador al usuario www-data con el comando:

- `sudo visudo`

7. En la figura 14 se muestra la edición del fichero que se ejecutó con el comando `sudo visudo`, se puede notar que el servidor Apache y el usuario Pi tienen permisos de administrador, ahora lo que se debe hacer es otorgar los permisos al usuario www-data añadiendo el siguiente comando:

- `www-data ALL=(root) NOPASSWD:ALL`

8. Guardar el archivo con `ctrl+o` y después `ctrl+x` para salir.



```
pi@raspi: ~
GNU nano 2.2.6      Fichero: /etc/sudoers.tmp

# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
pi ALL=(ALL) NOPASSWD: ALL
apache ALL=(ALL) NOPASSWD: ALL
www-data ALL=(root) NOPASSWD:ALL
```

Figura 14: Permiso de administrador al usuario www-data.

9. Descargar e instalar la librería para controlar los pines del GPIO de la Raspberr Pi para posteriormente programar los pines con Python. Escribimos el siguiente código en la consola
 - `wget http://downloads.sourceforge.net/project/raspberrypi-gpio-python/RPi.GPIO-0.6.3.tar.gz`
10. Descomprimir e instalar, utilizar los siguientes comandos:
 - `tar xzf RPi.GPIO-0.6.3.tar.gz`
 - `cd RPi.GPIO-0.6.3`
 - `sudo python setup.py install`

Anexo Q: Instalación y configuración certificado SSL.

Instalar servidor Https con certificado SSL

Para instalar el certificado SSL se debe seguir los siguientes pasos:

1. Instalar el modulo SSL para Apache con el siguiente comando:
 - `Sudo a2enmod ssl`
2. Cargar la configuración por defecto de apache para SSL
 - `sudo a2ensite default-ssl`

3. Reiniciar el servidor apache:
 - `sudo /etc/init.d/apache2 restart`

Con estos comandos se obtiene el HTTPS, pero el navegador nos reconoce como conexión no verificada es decir que nos falta el certificado SSL para este enlace:

4. Para obtener un certificado SSL se necesita crear una llave, primero dirigirse a la carpeta del servidor apache:

- `cd /etc/apache2`

5. Crear la llave con el nombre `server.key` con 2048 bytes de cifrado, utilizar el comando:

- `sudo openssl genrsa -des3 -out server.key 2048`

6. El servidor requiere una contraseña para la llave, colocar una contraseña fácil de recordar:
Al crear el certificado, tener en cuenta colocar el nombre de la llave `server.key` con el siguiente comando:

- `sudo openssl req -new -key server.key -out server.csr`

7. Una vez ejecutado el comando colocar la contraseña de la llave. Ahora requiere llenar un formulario con los datos del certificado.

En este caso el formulario requiere un código de país (colocar `ec`), el país (Ecuador), ciudad (Riobamba), organization name (colocar cualquier nombre), organization Unit name (colocar cualquier nombre) common Name (colocar el nombre de organization name) , Email Address(su correo electrónico)

8. Generar archivos con el certificado, para esto utilizar el comando:

- `sudo openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt`

Después de ejecutar y colocar la contraseña de la llave, ya tendremos creado el certificado.

9. Copiar tanto la llave como el certificado a la carpeta correspondiente de Apache, mediante el siguiente comando:

- sudo cp server.key /etc/ssl/private/
- sudo cp server.crt /etc/ssl/certs/

10. Abrir y modificar el archivo por defecto de apache para redirigir la configuración a estos archivos. Abrir el archivo:

- cd / etc/apache2/sites-available

11. Ejecutar el comando para editar el fichero:

- nano default-ssl.conf

12. Buscar el comando SSLCertificateFile y colocar al final server.crt, también el comando SSLCertificateKeyFile y colocar al final server.key como se observa en la figura 15

```
# SSLCertificateFile directive is needed.  
SSLCertificateFile /etc/ssl/certs/server.crt  
SSLCertificateKeyFile /etc/ssl/private/server.key
```

Figura 15: Edición de las líneas del certificado y llave.

13. Para activar las opciones de SSL, lo único que se debe hacer es borrar el # como se ve en la figura 16, guardar y salir.

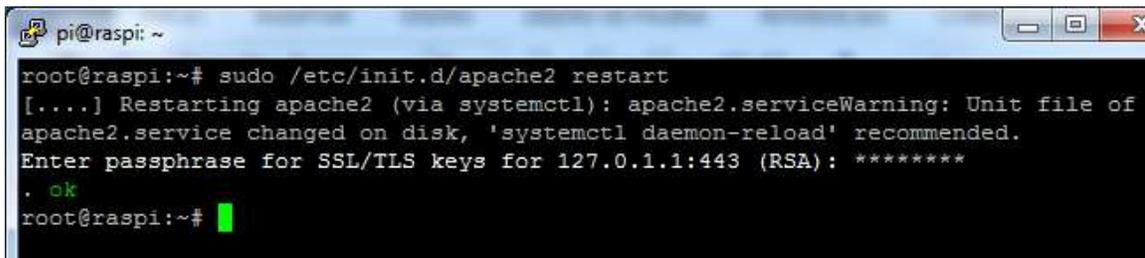
```
# directives are used in per-directory context.  
SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire  
<FilesMatch "\.(cgi|shtml|phtml|php)$">  
    SSLOptions +StdEnvVars  
</FilesMatch>  
<Directory /usr/lib/cgi-bin>
```

Figura 16: Habilitación de las opciones SSL.

14. Ahora habilitar el archivo nuevamente:

- `sudo a2ensite default-ssl.conf`

15. Por ultimo para aplicar los cambios se debe reiniciar el servidor Apache, se debe colocar la contraseña de la llave para que finalmente el certificado SSL y HTTPS estén habilitados como se muestra en la figura 17.



```
pi@raspi: ~  
root@raspi:~# sudo /etc/init.d/apache2 restart  
[...] Restarting apache2 (via systemctl): apache2.serviceWarning: Unit file of  
apache2.service changed on disk, 'systemctl daemon-reload' recommended.  
Enter passphrase for SSL/TLS keys for 127.0.1.1:443 (RSA): *****  
. ok  
root@raspi:~# █
```

Figura 17: Reinicio del servidor Apache.

16. Verificar cargando una página con la dirección de nuestro servidor en el navegador como en la figura 18.

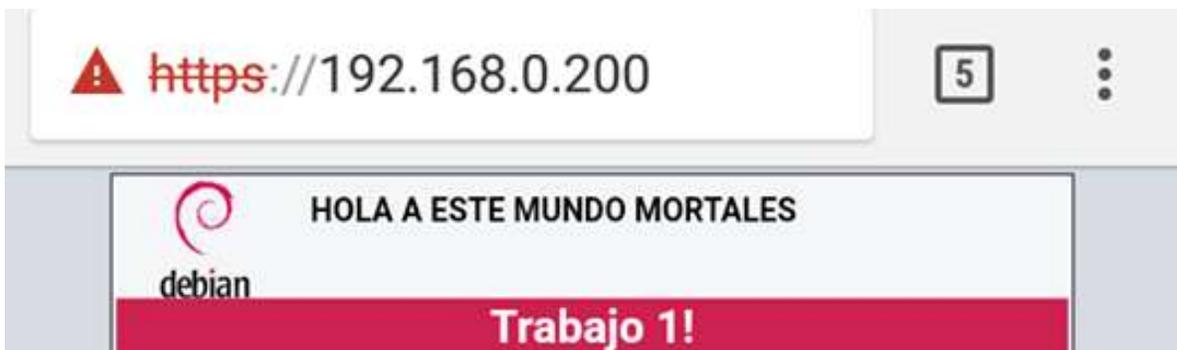


Figura 18: Verificación de HTTPS y certificado SSL.

Anexo R: Desarrollo de la aplicación WEB.

Desarrollo de la aplicación WEB

Para desarrollar la aplicación WEB hay que tomar en cuenta los distintos ficheros que se utilizan en la página que son creados en lenguajes de programación: HTML, PHP, JavaScript, y Python, para lo cual se siguen los siguientes pasos:

1. Para la elaboración de los ficheros o archivos de la aplicación Web se utiliza el editor de texto que viene instalado en el ordenador o utilizar Sublime Text2, disponible en el siguiente enlace:

<https://www.sublimetext.com/download>

Sublime text2 es un editor de código de texto con una interfaz amigable que soporta el uso de sistemas de construcción de códigos html como se muestra en la figura 19

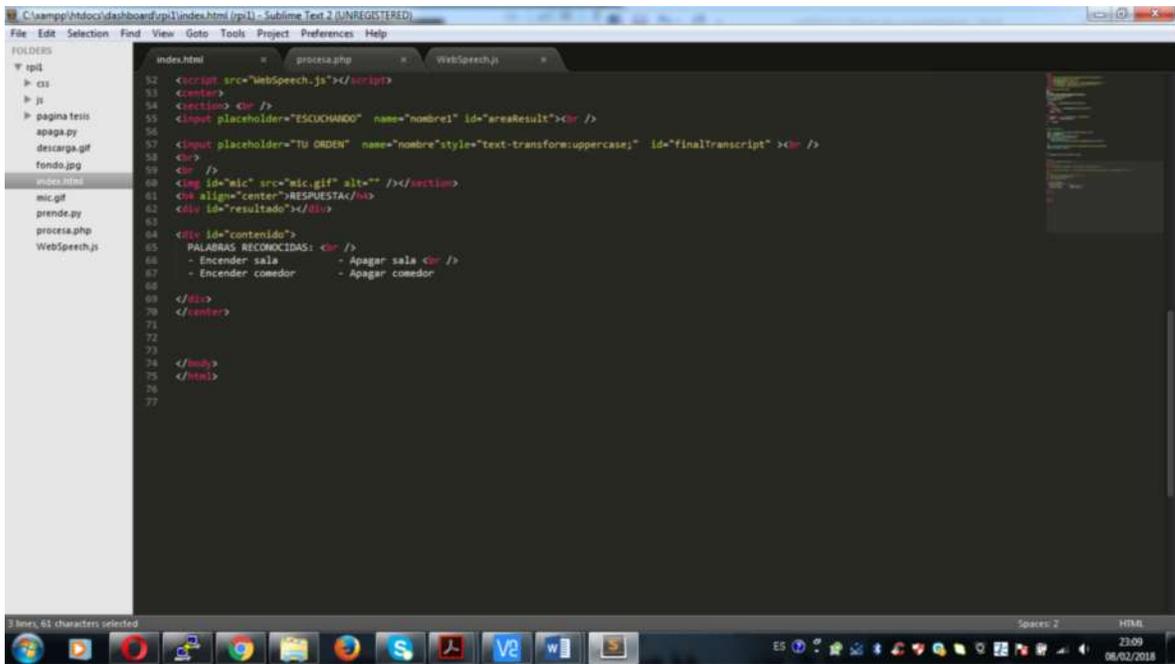


Figura 19: Interfaz de Sublime text2.

Cada vez que se desarrolla un código se guarda el archivo con un nombre y con las extensiones (.html, .php, .js, .py) al momento de montar en la Raspberry Pi. Todos los ficheros de la aplicación web se tienen que almacenar en el directorio rpi como se muestra la figura 20.

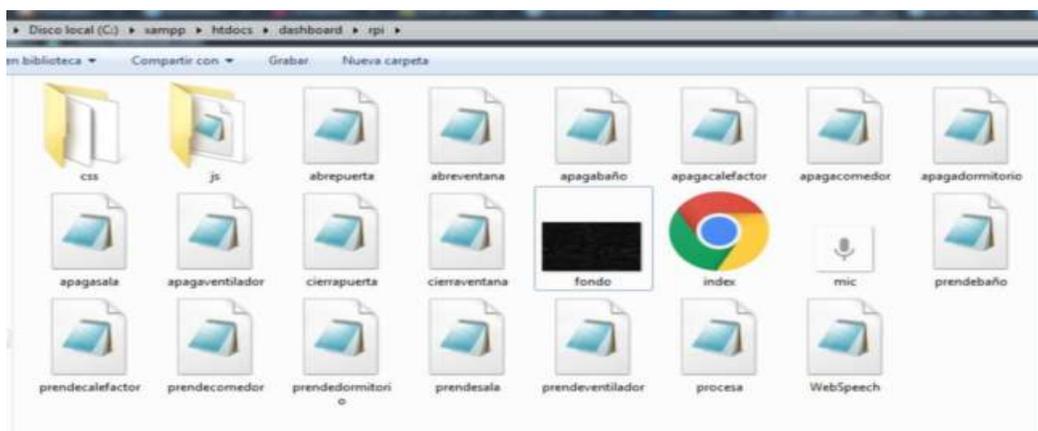


Figura 20: Ficheros de la aplicación Web.

2. A continuación se describirá cada uno de los ficheros de la aplicación Web del prototipo.

- **Index.html:**

El fichero index.html es uno de los más importantes ya que es el puente entre el usuario y el control de los distintos actuadores, el desarrollo de éste fichero se basó del código fuente de la aplicación Web Speech API de Google. El código fuente de esta aplicación es un archivo de texto creado mediante el software sublime text 2. Para la edición de la aplicación web se utilizó el motor de reconocimiento por voz de la aplicación Web speech api.

Esta aplicación es sencilla y compacta con el fin de almacenar una variable llamada “*finalTranscript*” generada por lo reconocido del motor de búsqueda por voz . Esta variable es transformada a texto y posteriormente enviada por el método POST a comparar en el fichero procesa.php la orden emitida por voz, dependiendo de la orden se ejecutan acciones para activar los diferentes actuadores en el prototipo.

Los ficheros con extensión “.gif”, “.jpg” son recursos usadas por index.html para el diseño de la aplicación web.

- **Procesa.php:**

Procesa.php recibe la variable “finalTranscript” proveniente del fichero index.html mediante el método POST . Compara luego la variable mediante una instrucción case con las distintas opciones preestablecidas y si la variable es igual a una de estas opciones ejecuta un script de Python determinado.

- **Predebaño.py:**

El presente fichero a igual que los demás con extensión “.py” alojados en la carpeta rpi es un script de Python, mediante las librerías de Python del GPIO previamente instaladas en la Raspberry Pi son las que se encargan de la activación o desactivación de los distintos pines GPIO. Si el usuario pronuncia la palabra “encender baño” el fichero WebSpeech.js lo convierte a texto, la variable “finalTranscript” se le asigna ese dato, la cual es enviada al procesa.php, que compara con los distintas opciones del case, si la opción coincide con ‘encender baño’ ejecuta el script predebaño.py y activa el GPIO correspondiente; así mismo se procede con las demás ordenes que dispone el prototipo.

- **Apagabaño.py:**

Si el usuario pronuncia la palabra “Apagar baño” el fichero WebSpeech.js lo convierte a texto, la variable “finalTranscript” se le asigna ese dato. Esta variable es enviada al procesa.php, se compara con los distintas opciones del case, si la opción coincide con ‘apagar baño’ ejecuta el script apagabaño.py y desactiva el GPIO correspondiente.

NOTA : EL resto de ficheros con extención py tiene la misma lógica solo cambia el nombre.

- **Abrepuerta.py:**

Si el usuario pronuncia la palabra “Abrir Puerta” el fichero WebSpeech.js lo convierte a texto, la variable “finalTranscript” se le asigna ese dato. Esta variable es enviada al procesa.php, se compara con los distintas opciones del case, si la opción coincide con ‘encender calefactor’ ejecuta el script abrepuerta.py y activa el GPIO correspondiente por un tiempo determinado.

- **Cierrapuerta.py:**

Si el usuario pronuncia la palabra “Cerrar Puerta” el fichero WebSpeech.js lo convierte a texto, la variable “finalTranscript” se le asigna ese dato. Esta variable es enviada al procesa.php, se compara con los distintas opciones del case, si la opción coincide con ‘cerrar puerta’ ejecuta el script cierrapuerta.py y activa el GPIO correspondiente por un tiempo determinado.

- **Abreventana.py:**

Si el usuario pronuncia la palabra “Abrir ventana” el fichero WebSpeech.js lo convierte a texto, la variable “finalTranscript” se le asigna ese dato. Esta variable es enviada al procesa.php, se compara con los distintas opciones del case, si la opción coincide con ‘encender calefactor’ ejecuta el script abrirventana.py y activa el GPIO correspondiente por un tiempo determinado.

- **Cierraventana.py:**

Si el usuario pronuncia la palabra “Cerrar ventana” el fichero WebSpeech.js lo convierte a texto, la variable “finalTranscript” se le asigna ese dato. Esta variable es enviada al procesa.php, se compara con los distintas opciones del case, si la opción coincide con ‘cerrar ventana’ ejecuta el script cierraventana.py y activa el GPIO correspondiente por un tiempo determinado.

Montaje de la aplicación WEB a la Raspberry PI

Para montar nuestra carpeta rpi en la Raspberry Pi se debe seguir los siguientes pasos:

1. Copiar la carpeta en un pendrive y luego se inserta en uno de los puertos USB de la Raspberry Pi como.
2. Copiar la carpeta rpi en el escritorio de Raspbian tal como se observa en las figura 21.

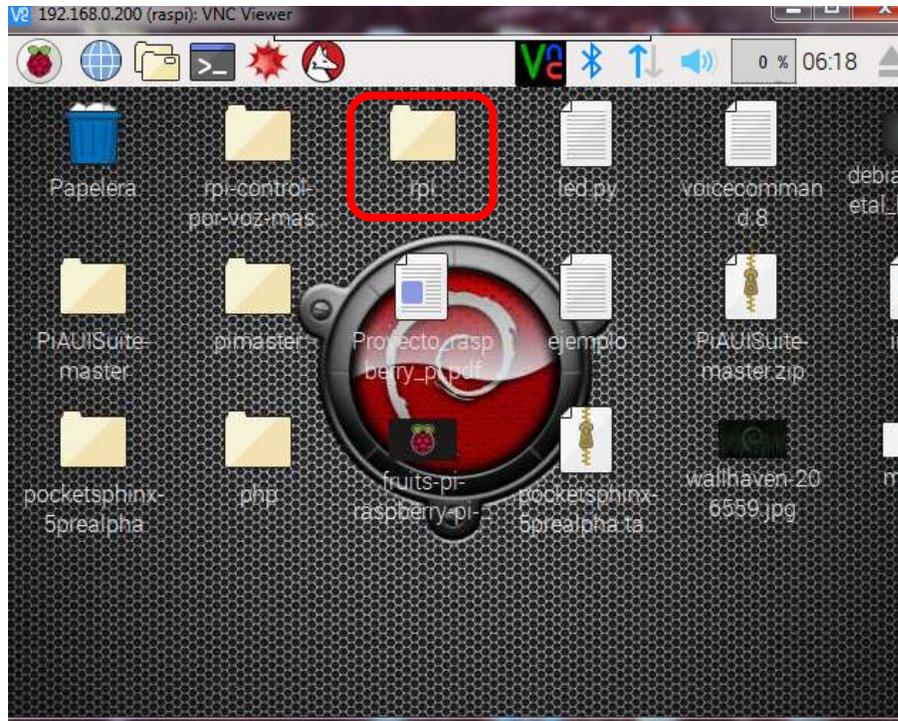


Figura 21: Carpeta rpi en el escritorio de Raspbian.

3. Una vez que la carpeta está en la Raspberry Pi se debe cambiar la dirección de ubicación de los ficheros del servidor Apache a la dirección /var/www/html.
 - `cd Desktop`
4. Para que muestre los archivos del escritorio se debe escribir el siguiente comando:
 - `Ls`
5. La carpeta rpi está ubicada en el escritorio. Finalmente escribir el siguiente código para mover el fichero a la ubicación del servidor Apache.
 - `mv rpi /var/www/html`
6. Para comprobar si el fichero fue movido con éxito nos dirigimos a la ubicación del servidor Apache con el siguiente código.
 - `cd /var/www/html`
 - `ls`

7. Para la comprobación de la aplicación Web, se debe ubicar la dirección: <https://192.168.0.200/rpi/> desde un Smartphone conectado a una red de área local, en el navegador de Google Chrome como se muestra en la figura 22



Figura 22: Acceso a la aplicación WEB.

Anexo S: fichero index.html.

```
pi@raspi: ~
GNU nano 2.2.6 Fichero: index.html

<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <meta http-equiv="refresh" content="6">
  <title>Reconocimiento de Voz</title>
  <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
  <link rel="stylesheet" type="text/css" href="css/estilos.css" />
  <script src="js/jquery-1.10.1.min.js"></script>
</head>
<body background="fondo.jpg">
<br>
<br />
<h1 align="center">CONTROL DOMÓTICO POR VOZ</h1>
<script type="text/javascript">
function objetoAjax(){
  var xmlhttp=false;
  try{
    xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
  } catch (e){
    try {
      xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (E){
      xmlhttp =false;
    }
  }
  if (!xmlhttp && typeof XMLHttpRequest!="undefined"){
    xmlhttp = new XMLHttpRequest();
  }
  return xmlhttp;
}
function Enviar() {
  valor = document.getElementById('finalTranscript').value;
  ajax = objetoAjax();
  ajax.open("POST", "procesa.php", true);
  ajax.onreadystatechange= function(){
    if (ajax.readyState==4) {
      document.getElementById("resultado").innerHTML=ajax.responseText;
    }
  }
}
```

```

    }
  }
  ajax.setRequestHeader("content-type","application/x-www-form-urlencoded");
  ajax.send("varp2="+ valor )
}
var intervalo=setInterval('Enviar()',4000);
</script>
<script src="WebSpeech.js"></script>
<center>
<section><br />
<input placeholder="ESCUCHANDO TU ORDEN" name="nombre1" id="areaResult"><br />
<input placeholder="TU ORDEN" name="nombre" style="text-transform:uppercase;" id="finalTranscript"><br />
<br>
<br />
</section>
<h4 align="center">RESPUESTA</h4>
<div id="resultado"></div>
<div id="contenido">
PALABRAS RECONOCIDAS: <br />
- Encender sala          - Apagar sala <br />
- Encender comedor       - Apagar comedor
</div>

</center>

</body>
</html>

```

Anexo T: Fichero WebSpeech.js

pi@raspi: ~

GNU nano 2.2.6

Fichero: WebSpeech.js

```
window.onload = function () {  
  
    var recognition = new webkitSpeechRecognition();  
    recognition.continuous = true;  
    recognition.interimResults = true;  
    recognition.lang = "es";  
  
    recognition.start();  
    mic.addEventListener('click', function () {  
        areaResult.focus();  
  
    });  
  
    //events  
  
    recognition.onaudiostart = function (event) {  
        console.log("onaudiostart");  
    };  
  
    recognition.onsoundstart = function (event) {  
        console.log("onsoundstart");  
    };  
  
    recognition.onspeechstart = function (event) {  
        console.log("onspeechstart");  
    };  
  
    recognition.onspeechend = function (event) {  
        console.log("onspeechend");  
    };  
  
    recognition.onsoundend = function (event) {  
        console.log("onsoundend");  
    };  
  
    recognition.onaudioend = function (event) {  
        console.log("onaudioend");  
    };  
};
```

```
};

recognition.onresult = function (event) {
  console.log("onresult");
  var interimResult = '',
      finalResult = '';
  for (var i = event.resultIndex; i < event.results.length; ++i) {
    if (event.results[i].isFinal) {
      finalResult = event.results[i][0].transcript;
    } else {
      interimResult += event.results[i][0].transcript;
    }
  }

  finalTranscript.value = finalResult;
  areaResult.value = interimResult;

  if (interimResult.indexOf('terminar') != -1)
    recognition.stop();
};

recognition.onnomatch = function (event) {
  console.log("onnomatch");
};

recognition.onerror = function (event) {
  console.log("onerror: " + event);
};

recognition.onstart = function (event) {
  console.log("onstart");
};

recognition.onend = function (event) {
  console.log("onend");
};
};
```

Anexo U: Fichero procesa.php

```
pi@raspi: ~
GNU nano 2.2.6 Fichero: procesa.php

<?php
switch ($_POST['varp2'])
{
    case 'encender dormitorio':
        echo "DORMITORIO ENCENDIDO";
        exec('sudo python /var/www/html/rpi/prendedormitorio.py');
        break;
    case 'Apagar dormitorio':
        echo "DORMITORIO APAGADO";
        exec('sudo python /var/www/html/rpi/apagadormitorio.py');
        break;
    case 'abrir ventana':
        echo "VENTANA ABIERTA";
        exec('sudo python /var/www/html/rpi/abreventana.py');
        break;
    case 'cerrar ventana':
        echo "VENTANA CERRADA";
        exec('sudo python /var/www/html/rpi/cierraventana.py');
        break;
    case 'abrir puerta':
        echo "PUERTA ABIERTA";
        exec('sudo python /var/www/html/rpi/abrepuerta.py');
        break;
    case 'cerrar puerta':
        echo "PUERTA CERRADA";
        exec('sudo python /var/www/html/rpi/cierrapuerta.py');
        break;
    case 'encender sala':
        echo "SALA ENCENDIDA";
        exec('sudo python /var/www/html/rpi/prendesala.py');
        break;
    case 'Apagar sala':
        echo "SALA APAGADA";
        exec('sudo python /var/www/html/rpi/apagasala.py');
        break;
    case 'encender baño':
        echo "BAÑO ENCENDIDO";
        exec('sudo python /var/www/html/rpi/prendebaño.py');
        break;
}
```

```
case 'Apagar baño':
    echo "BAÑO APAGADO";
    exec('sudo python /var/www/html/rpi/apagabaño.py');
    break;
case 'encender comedor':
    echo "COMEDOR ENCENDIDO";
    exec('sudo python /var/www/html/rpi/predecomedor.py');
    break;
case 'Apagar comedor':
    echo "COMEDOR APAGADO";
    exec('sudo python /var/www/html/rpi/apagacomedor.py');
    break;
case 'encender calefactor':
    echo "CALEFACTOR ENCENDIDO";
    exec('sudo python /var/www/html/rpi/predecalefactor.py');
    break;
case 'Apagar calefactor':
    echo "CALEFACTOR APAGADO";
    exec('sudo python /var/www/html/rpi/apagacalefactor.py');
    break;

case 'encender ventilador':
    echo "VENTILADOR ENCENDIDO";
    exec('sudo python /var/www/html/rpi/predeventilador.py');
    break;
case 'Apagar ventilador':
    echo "VENTILADOR APAGADO";
    exec('sudo python /var/www/html/rpi/apagaventilador.py');
    break;
default:
    echo 'COMANDO NO VALIDO';
    break;
}
..

?>
```

Anexo V: Ficheros .py

```
pi@raspi: ~  
GNU nano 2.2.6 Fichero: abrepuerta.py
```

```
import RPi.GPIO as GPIO  
import time  
  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(5, GPIO.OUT)  
  
rojo = GPIO.PWM(5, 100)  
rojo.start(42)  
  
GPIO.output(5, True)  
time.sleep(1)  
  
GPIO.output(5, False)  
time.sleep(1)
```

```
pi@raspi: ~  
GNU nano 2.2.6 Fichero: cierrapuerta.py
```

```
import RPi.GPIO as GPIO  
import time  
  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(6, GPIO.OUT)  
  
rojo = GPIO.PWM(6, 100)  
rojo.start(42)  
  
GPIO.output(6, True)  
time.sleep(0.72)  
  
GPIO.output(6, False)  
time.sleep(0.72)
```

```
pi@raspi: ~  
GNU nano 2.2.6 Fichero: abreventana.py
```

```
import RPi.GPIO as GPIO  
import time  
  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(22, GPIO.OUT)  
  
rojo = GPIO.PWM(22, 100)  
rojo.start(34)  
  
GPIO.output(22, True)  
time.sleep(0.50)  
  
GPIO.output(22, False)  
time.sleep(0.50)
```

```
pi@raspi: ~  
GNU nano 2.2.6 Fichero: cierraventana.py
```

```
import RPi.GPIO as GPIO  
import time  
  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(27, GPIO.OUT)  
  
rojo = GPIO.PWM(27, 100)  
rojo.start(32)  
  
GPIO.output(27, True)  
time.sleep(0.50)  
  
GPIO.output(27, False)  
time.sleep(0.50)
```

```
pi@raspi: ~  
GNU nano 2.2.6 Fichero: prendedormitorio.py
```

```
import RPi.GPIO as GPIO  
import time  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(24, GPIO.OUT)  
GPIO.output(24, True)
```

```
pi@raspi: ~  
GNU nano 2.2.6 Fichero: apagadormitorio.py
```

```
import RPi.GPIO as GPIO  
import time  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(24, GPIO.OUT)  
GPIO.output(24, False)
```

```
pi@raspi: ~
GNU nano 2.2.6 Fichero: prendebaño.py

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(26, GPIO.OUT)
GPIO.output(26, True)
```

```
pi@raspi: ~
GNU nano 2.2.6 Fichero: apagabaño.py

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(26, GPIO.OUT)
GPIO.output(26, False)
```

```
pi@raspi: ~
GNU nano 2.2.6 Fichero: prendecomedor.py

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.OUT)
GPIO.output(23, True)
```

```
pi@raspi: ~
GNU nano 2.2.6 Fichero: apagacomedor.py

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.OUT)
GPIO.output(23, False)
```

```
pi@raspi: ~
GNU nano 2.2.6 Fichero: prendesala.py

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(25, GPIO.OUT)
GPIO.output(25, True)
```

```
pi@raspi: ~
GNU nano 2.2.6 Fichero: apagasala.py

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(25, GPIO.OUT)
GPIO.output(25, False)
```

```
pi@raspi: ~  
GNU nano 2.2.6 Fichero: predeventilador.py
```

```
import RPi.GPIO as GPIO  
import time  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(16, GPIO.OUT)  
GPIO.output(16, True)
```

```
pi@raspi: ~  
GNU nano 2.2.6 Fichero: apagaventilador.py
```

```
import RPi.GPIO as GPIO  
import time  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(16, GPIO.OUT)  
GPIO.output(16, False)
```

```
pi@raspi: ~  
GNU nano 2.2.6 Fichero: predecalefactor.py
```

```
import RPi.GPIO as GPIO  
import time  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(4, GPIO.OUT)  
GPIO.output(4, True)
```

```
pi@raspi: ~  
GNU nano 2.2.6 Fichero: apagacalefactor.py
```

```
import RPi.GPIO as GPIO  
import time  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(4, GPIO.OUT)  
GPIO.output(4, False)
```

Anexo W: Solicitud de presupuesto, para casa domótica.

habitissimo

Tu petición de presupuesto

Detalles de tu presupuesto

 Recuerda que cuanto más detallada sea la información, más fácil será que el profesional entienda qué es lo que hay que hacer y que el presupuesto estimado se acerque a la realidad.

Presupuesto sin título [Cambiar título »](#)

Trabajo solicitado el 31-01-2018 en Construcción Casas [Cambiar categoría »](#)

Trabajo a realizar	Proyecto y Construcción Casa
Fecha estimada del proyecto	Lo antes posible
Preferencia	Relación calidad/precio
Tipo de cliente	Particular
¿Superficie útil de la vivienda?	Entre 71 y 100 m2
¿Nº de plantas?	Mas de 2 Plantas
¿Cuántas habitaciones?	5 o mas

¿Cuántos baños? 5 o más

¿Qué estilos te interesan? Moderna

Galería de imágenes

Te aconsejamos que subas fotos para que el trabajo a realizar sea más claro.

[Añadir imágenes \(opcional\) »](#)

¿Necesitas ayuda?

[Guía para una experiencia positiva](#)

[Opiniones de Particulares](#)

[10 consejos para elegir un profesional](#)

[Sellos de confianza](#)

[Preguntas Frecuentes](#)

[Contacta con nosotros](#)

Anexo X: Presupuesto del prototipo.

PRESUPUESTO DEL PROTOTIPO			
Materiales	Cantidad	P. unitario	Total
Diodo Led 10mm	4	0,5	2
Resistencia 1K	6	0,1	0,6
Diodo 1N4148	6	0,25	1,5
Transistor 2N222A	6	0,3	1,8
Puente H L298N	1	8	8
Motor corriente DC 9V	3	2	6
Motor corriente DC 5V	1	2	2
Cable UTP categoría 5 (1m)	5	0,6	3
Soldadura de cables aleación estaño plomo	1	3	3
Pasta para soldar	1	2	2
Cautín pistola de madera	1	5	5
Cable Jumper macho hembra	40	0,1	4
Barra de silicona	20	0,2	4
Pistola de silicona	1	3,5	3,5
Termocontraible (1m)	5	0,75	3,75
Piñón de plástico	3	1	3
Cremallera para engranajes	2	2	4
Electrodo 2,5 mm	2	0,15	0,3
Tubo Flexible de agua	2	0,5	1
Raspberry Pi modelo 3B	1	50	50
Case Raspberry Pi modelo 3B	1	15	15
Fuente de alimentación Raspberry pi (5v 2,5A)	1	5	5
Fuente de alimentación (9v 1A)	1	5	5
Micro SD 32 GB	1	20	20
Cable HDMI	1	10	10
Teclado	1	20	20
Mouse	1	12	12
Monitor	1	100	100

Maqueta	1	150	150
Caja circuito	1	40	40
Gastos varios	1	30	30
Mano de obra	2	1000	2000
Total			2515,45