



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE SISTEMAS DE LA COMPUTACIÓN

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO
DE TECNOLOGÍA EN COMPUTACIÓN**

TEMA:

**“IMPLEMENTACIÓN DE UN CONTROL DOMÓTICO UTILIZANDO
RASPBERRY PI Y UNA BASE DE DATOS NoSQL”.**

AUTORES:

**JÁCOME LLERENA, BYRON WILMER
VELASCO BONIFAZ, FABIÁN EDMUNDO**

SANGOLQUÍ

2017



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍA EN COMPUTACIÓN

CERTIFICACIÓN Certifico que el trabajo de titulación, "IMPLEMENTACIÓN DE UN CONTROL DOMÓTICA UTILIZANDO RASPBERRY PI Y UNA BASE DE DATOS NoSQL " realizado por los señores BYRON WILMER JÁCOME LLERENA y FABIÁN EDMUNDO VELASCO BONIFAZ, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a los señores BYRON WILMER JÁCOME LLERENA y FABIÁN EDMUNDO VELASCO BONIFAZ, para que lo sustente públicamente.

Ciudad, 24 de Abril del 2017.



MARIO ALMACHE CUEVA





DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE TECNOLOGÍA EN COMPUTACIÓN

AUTORÍA DE RESPONSABILIDAD

Yo, BYRON W. JÁCOME LLERENA, con cédula de identidad N° 1714905955, declaro que este trabajo de titulación "IMPLEMENTACIÓN DE UN CONTROL DOMÓTICO UTILIZANDO RASPBERRY PI Y UNA BASE DE DATOS NoSQL", ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Quito, 28 de Marzo del 2017.

BYRON W. JÁCOME LLERENA.
1714905955



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE TECNOLOGÍA EN COMPUTACIÓN

AUTORÍA DE RESPONSABILIDAD

Yo, FABIAN EDMUNDO VELASCO BONIFAZ, con cédula de identidad N° 1709062135, declaro que este trabajo de titulación "IMPLEMENTACIÓN DE UN CONTROL DOMÓTICO UTILIZANDO RASPBERRY PI Y UNA BASE DE DATOS NoSQL", ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Quito, 18 de Marzo del 2017.

FABIAN EDMUNDO VELASCO BONIFAZ
1709062135



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE TECNOLOGÍA EN COMPUTACIÓN

AUTORIZACIÓN

Yo, BYRON W. JÁCOME LLERENA, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación "IMPLEMENTACIÓN DE UN CONTROL DOMÓTICO UTILIZANDO RASPBERRY PI Y UNA BASE DE DATOS NoSQL", cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Quito, 28 de Marzo del 2017.

BYRON W. JÁCOME LLERENA.
1714905955



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE TECNOLOGÍA EN COMPUTACIÓN

AUTORIZACIÓN

Yo, FABIAN EDMUNDO VELASCO BONIFAZ, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación "IMPLEMENTACIÓN DE UN CONTROL DOMÓTICO UTILIZANDO RASPBERRY PI Y UNA BASE DE DATOS NoSQL", cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Quito, 18 de Marzo del 2017.

.....
FABIAN EDMUNDO VELASCO BONIFAZ
1709062135

DEDICATORIA

A mi DIOS, mis padres, mis tíos, mis hermanos, mis profesores y a toda mi familia que siempre confió en mí, y que con su apoyo, siempre me empujó hacia adelante.

Especialmente, a la memoria de mi pequeño hermanito.

Byron W. Jácome Llerena.

Para Adrián y todos los que nunca se rinden.

Fabián E. Velasco Bonifaz.

AGRADECIMIENTOS

A mi DIOS por permitir que mis padres, hermanos y tíos, se hayan mantenido juntos guiándome y alentándome en mi carrera universitaria.

A mis padres por apoyarme, en todos los aspectos, y haberme inculcado valores de lealtad y de respeto.

A mi prima, por estar a mi lado en los primeros años de mi carrera universitaria.

A mis profesores, todos sin excepción, en especial a mi Director de Carrera, Ing. Germán Ñacato, que siempre confió en mí y que con su apoyo y conocimientos me supo guiar desde mis primeros años de estudio, convirtiéndose en un amigo y en un referente.

Byron W. Jácome Llerena.

Para los excelentes profesores que han influido positivamente en mi desarrollo personal: Mario Almache, amigo de siempre, Jenny Ruiz, mentora de este trabajo, Gilma Toaza y Myriam Noury.

Muchas gracias.

Fabián E. Velasco Bonifaz.

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDO

DEDICATORIA	vi
AGRADECIMIENTOS.....	vii
RESUMEN.....	xv
ABSTRACT	xvi

CAPITULO 1 INTRODUCCIÓN

GENERALIDADES.	1
1.2 JUSTIFICACIÓN E IMPORTANCIA	3
1.3 OBJETIVO GENERAL.....	6
1.4 OBJETIVOS ESPECÍFICOS.....	6
1.5 ALCANCE.....	7

CAPITULO 2 MARCO TEÓRICO

CAPITULO II. MARCO TEÓRICO	9
2.1 GENERALIDADES.....	9
2.1.1 INTRODUCCIÓN.....	9
2.1.2 SITIO O PORTAL WEB	11
2.1.3 FUNCIONALIDADES DE UN SITIO O PORTAL WEB.....	12
2.1.4 TIPOS DE PORTAL WEB.....	13
2.2 INTERNET.....	13
2.2.1 INTERNET CONCEPTO.....	13

2.3 BASE DE DATOS.....	15
2.3.1 CONCEPTO DE BASE DE DATOS.....	15
2.3.2 CARACTERÍSTICAS DE LA BASE DE DATOS.....	16
2.3.3 BASES DE DATOS NoSQL.....	17
2.3.4 FIREBASE.....	17
2.4 REDES.....	18
2.4.1 CONCEPTO DE REDES.....	18
2.4.2 CARACTERÍSTICAS DE LAS REDES.....	20
2.5 SERVIDORES WEB.....	20
2.5.1 CONCEPTO DE SERVIDORES WEB.....	20
2.5.2 CONCEPTO SERVIDOR APACHE.....	21
2.6 INGENIERÍA DE SOFTWARE.....	22
2.6.1. INTRODUCCIÓN.....	22
2.6.2 CONCEPTO DE INGENIERÍA DE SOFTWARE.....	22
2.7 HERRAMIENTAS DE DESARROLLO WEB.....	22
2.8 SOFTWARE LIBRE.....	23
2.8.1 CONCEPTO DE SOFTWARE LIBRE.....	23
2.9 HARDWARE LIBRE.....	24
2.9.1 CONCEPTO DE HARDWARE LIBRE.....	24
2.10 SINGLE BOARD COMPUTER.....	25
2.10.1 RASPBERRY Pi.....	25
2.11 DOMÓTICA.....	26
2.12 METODOLOGÍA.....	27
2.12.1 UWE INTRODUCCIÓN.....	27
2.12.1.1 FASE DE LA METODOLOGÍA UWE.....	29
• Análisis de requerimientos.....	29
• Modelo Conceptual.....	29
• Modelo Navegacional.....	30
• Modelo de Presentación.....	30
2.12.2 ARTEFACTOS.....	30
2.12.2.2 DIAGRAMA DE CLASES.....	30
2.12.2.3 DIAGRAMAS DE SECUENCIA.....	31
2.12.2.4 DIAGRAMAS DE ESTADO.....	31

2.12.2.5 DIAGRAMAS DE DESPLIEGUE	32
2.12.2.6 DIAGRAMAS DE ACTIVIDADES	32
2.12.3 LENGUAJE DE MODELAMIENTO UNIFICADO (UML)	32

CAPITULO 3

DESARROLLO DE LA SOLUCIÓN

CAPITULO III. DESARROLLO DE LA SOLUCIÓN	33
3.1 MODELAMIENTO DE CASOS DE USO.....	33
3.1.1 MODELO DE REQUERIMIENTOS.....	33
3.2 Lista de Actores	34
3.3 DIAGRAMAS DE CASO DE USO	34
3.4 CASOS DE USO DEL MÓDULO DE CONFIGURACIÓN	35
3.5 CASOS DE USO DEL MÓDULO ACCESOS.....	35
3.6 DESCRIPCIÓN DE CASOS DE USO.....	36
3.6.1 MÓDULO DE CONFIGURACIÓN	36
3.6.2 MÓDULO DE ACCESO	37
3.7 MODELO DE CONTENIDO.....	37
3.8 DIAGRAMA DE MODELO DE NAVEGACIÓN.....	38
3.9 MODELAMIENTO DE LA FIREBASE (Clave-Valor).....	40
3.10 MODELAMIENTO ELECTRÓNICO.....	40

CAPITULO 4

IMPLEMENTACIÓN

CAPITULO IV. IMPLEMENTACIÓN.....	42
4. CONFIGURACIÓN DEL HARDWARE.	42
➤ Paso1: Conexión y configuración de la Raspberry Pi.....	42
➤ Paso2: Instalación de librerías necesarias para control de los pines GPIO	43
➤ Paso3: Instalación de servidor apache	45

➤ Paso4: Instalar la versión ligera de Firefox para Raspberry Pi llamada “iceweasel”	46
➤ Paso5: Redirección de Archivos	48
➤ Paso6: Cuenta en Firebase	50
4.2 IMPLEMENTACIÓN DEL APLICATIVO WEB.	52
4.3 REQUISITOS PARA LA IMPLEMENTACIÓN	54

CAPÍTULO 5

PRUEBAS DEL APLICATIVO EN HARDWARE/SOFTWARE

CAPÍTULO V. PRUEBAS DEL APLICATIVO EN HARDWARE/SOFTWARE.....	55
5. PRUEBAS DE LA APLICACIÓN.....	55
5.1 PRUEBAS DE CONECTIVIDAD.....	56
5.2 PRUEBAS DE CONCURRENCIA	60
5.3 PRUEBAS DE COMPATIBILIDAD	61
5.4 PRUEBAS DE RENDIMIENTO.....	61
5.5 PRUEBAS DE CAJA NEGRA	63

CAPÍTULO 6

CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES	68
6.1. CONCLUSIONES	68
6.2. RECOMENDACIONES	69
BIBLIOGRAFÍA.....	73
NETGRAFÍA.....	76

ÍNDICE DE FIGURAS

Figura 1 Gráfico Estadístico de los hogares ecuatorianos con acceso al Internet.	9
Figura 2 Funcionamiento de la Internet.	14
Figura 3 Tipos de Bases de Datos.	16
Figura 4 Bases de Datos Clave-Valor grafos.	17
Figura 5 Tipos y componentes de una RED.	19
Figura 6 Estructura y tipos de Servidores Web.	21
Figura 7 Clasificación del hardware Libre.	24
Figura 8 Hogar digital.	26
Figura. 9 Etapas y Modelos UWE.	28
Figura. 10 Diagrama de Casos de uso para el Módulo de Configuración.	35
Figura. 11 Diagrama de Casos de uso para del Módulo Accesos.	35
Figura. 12 Diagrama de Clases Módulo Accesos.	38
Figura. 13 Modelo de Navegación para gestión de luces.	39
Figura. 14 Modelamiento de la Base de Datos FireBase.	40
Figura. 15 Modelamiento Eléctrico/electrónico.	41
Figura. 16 Modelamiento Eléctrico/electrónico, esquema gráfico.	41
Figura. 17 Escritorio Raspberry Pi.	42
Figura. 18 Instalación de librerías necesarias para control de los pines GPIO.	43
Figura. 19 Instalación de librerías necesarias para control de los pines GPIO.	44
Figura. 20 Instalar demjson.	45
Figura. 21 Instalación de servidor apache.	46
Figura. 22 versión ligera de Firefox para Raspberry Pi llamada “iceweasel”	47
Figura. 23 Servidor Apache para Raspberry Pi.	48
Figura. 24 Modificación de Archivos.	49

Figura. 25 Modificación de Archivos.....	50
Figura. 26 Cuenta en Firebase.	51
Figura. 27 Aplicativo Web.	52
Figura. 28 Aplicativo Web.	53
Figura. 29 Dos usuarios conectados al mismo tiempo al Aplicativo Web.	53
Figura. 30 Ping de Ubuntu a Raspberry.	56
Figura. 31 Ping desde Raspberry a PC Ubuntu.....	57
Figura. 32 Pruebas de conectividad.	58
Figura. 33 Pruebas de conectividad.	58
Figura. 34 Usando Fing Aplicación Iphone.	59
Figura. 35 Usando Network Scanner Samsung.	59
Figura. 36 Pruebas de rendimiento.....	62
Figura. 37 Pruebas de Caja Negra.	64
Figura. 38 Pruebas de Caja Negra.	65
Figura. 39 Pruebas de Caja Negra.	66
Figura. 40 Pruebas de Caja Negra.	67

ÍNDICE DE TABLAS

Tabla 1	5
Tabla 2 Comparación entre las tecnologías RASPBERRY PI MODELO B y CUBIEBOARD2.	25
Tabla 3 Actores del aplicativo web.	34
Tabla 4 Descripción Caso de Uso Gestionar Usuario.	34
Tabla 5 Descripción Caso de Uso Acceso.	37
Tabla 6 Notación UWE para los elementos del modelo de Navegación.....	38
Tabla 7 Pruebas de concurrencia.	60
Tabla 8 Pruebas de compatibilidad.....	61
Tabla 9 Pruebas de Caja Negra.	63
Tabla 10 Pruebas de Caja Negra.	64
Tabla 11 Pruebas de Caja Negra.	65
Tabla 12 Pruebas de Caja Negra.	66

RESUMEN

En este proyecto, se obtendrá como resultado final un sistema domótico, cuya interfaz web permitirá el control automatizado del sistema de iluminación eléctrico de una vivienda. Los requerimientos actuales de la tecnología computacional hacen necesario el control domótico para facilitar las actividades y contar con mandos inalámbricos para tareas como el encendido o apagado de luces, involucrando los conocimientos de informática y electrónica. Se consideró el uso de una Raspberry Pi, en la cual se desarrolle una aplicación Web, que tenga un control en tiempo real, pudiendo ser la solución al problema planteado. La base de datos Firebase de Google que ofrece el control de los datos para aplicaciones en tiempo real, así como una API para conectarse con la Raspberry Pi. El sistema web, ha sido probado con los navegadores más comerciales, tanto de PC como de smartpho`n`s u otros periféricos de última generación, de allí, que el usuario no tenga inconvenientes a la hora de ejecutar el aplicativo web. El resultado final, ha sido un sistema de hardware y software estable y ágil en lo que respecta a su conexión y ejecución.

PALABRAS CLAVES

- **SISTEMA DOMÓTICO.**
- **APLICATIVO WEB.**
- **RASPBERRY PI.**
- **FIREBASE.**

ABSTRACT

In this project, the final result will lead to a home automation system, whose web interface will allow to an automated control of the electrical lighting system of a house. The current requirements of computer technology make it necessary for domotic controls to simplify activities and the usage of wireless controls for tasks such as turning on or turning off lights, all of these involving computer and electronic knowledge. The use of a Raspberry Pi was considered, in which a web application is supported and gets the control on real time application; it will be able to be the solution to the problem proposed. The Google database called Firebase is the one which provides data control for real-time applications as well as an API to connect to the Raspberry Pi. The web system has been tested with the most commercial browsers designed for both PCs and smartphones and any other peripherals device of last generation, therefore, the user does not have problems when running the web application. The final result has been a stable and dynamic hardware and software system in terms of its connection and execution.

KEYWORDS

- **DOMOTIC SYSTEM.**
- **WEB APPLICATION.**
- **RASPBERRY PI.**
- **FIREBASE.**

CAPITULO I. INTRODUCCIÓN

GENERALIDADES.

El presente proyecto, tiene como finalidad el control domótico de luces, enfatizando en la actualización de los estados de los controles de encendido y apagado de la aplicación web, de todos los usuarios de forma automática y en tiempo real.

Comprende, el desarrollo de un sistema web que permite controlar en tiempo real, el encendido y apagado de dos luces: una de sala y otra de comedor, mediante el uso de teléfonos inteligentes (Tablet's, notebook's, laptop's etc.) funcionando con cualquier sistema operativo. Las luces son encendidas o apagadas por medio de las salidas GPIO de una Raspberry Pi, que con ayuda de una interface de relés, permiten la alimentación de 120VAC requeridos por las mismas.

Para el control en tiempo real se utilizarán las características que ofrece la Base de Datos "Firebase", que provee una API para guardar y sincronizar datos en la nube en tiempo real, evitando la generación de código adicional.

En la Raspberry Pi se instalará un servidor Web Apache, el mismo que proveerá el aplicativo que controlarán los teléfonos inteligentes.

Quedan fuera del alcance de este proyecto, los controles manuales de encendido y apagado que en una instalación real deberían estar implementados en paralelo a los relés de salida.

Este proyecto tiene como fin el utilizar software y hardware disponible en el mercado para cualquier usuario. Aunque es cierto que la Raspberry Pi lleva en el mercado mundial desde 29 de febrero de 2012, en Sudamérica y en particular, en el Ecuador, ha tardado en llegar, ya sea por el desconocimiento del hardware o por la indisposición a reemplazar la ya extendida placa controladora de Arduino, y aunque ambas placas electrónicas pueden acoplarse a otros hardware's y ser programadas para controlarse desde una PC, la funcionalidad de la Raspberry Pi está por encima de cualquier versión de Arduino, ya que la Raspberry Pi en realidad es un computador miniaturizado y, que al mismo tiempo, puede programarse y acoplarse a otros dispositivos eléctricos y/o electrónicos sin ninguna dificultad, puesto que maneja un número ilimitado de lenguajes de programación, e incluso tiene su propio sistema operativo. Por otra parte, Arduino no deja de ser una tarjeta controladora y por tanto carece de sistema operativo. Además, la Raspberry Pi tiene su propio puerto de comunicación Ethernet, cosa de la carece que Arduino, necesitando Shield para acceder a internet. Otra de las diferencias, y tal vez la más llamativa es que la Raspberry Pi posee puertos HDMI, USB, RCA, Audio 3,5mm y 40 pines GPIO (en las versiones estándar), todo lo contrario del Arduino, que apenas si tiene puertos USB y 19 pines GPIO (en las versiones estándar).

Por otro parte, la aplicación Web que se utilizará, ha sido desarrollada de tal forma que puede accederse desde cualquier dispositivo como un Smartphone, una tablet o una Pc portátil o de escritorio.

La utilización de una Base de Datos NoSQL, permite una mayor velocidad de la Raspberry Pi ya que aunque sea una computadora, no hay que olvidar que utiliza una CPU de cuatro-núcleos ARM Cortex-A53 a 1,2 GHz (procesador soporte de 64-bit, en la versión más moderna), he ahí la necesidad de evitar que la Raspberry Pi utilice su microprocesador en procesos evitables. Así pues, la base de datos NoSQL, almacena los datos en la nube, actualizando la información en microsegundos y descartando que el aplicativo web comparta los datos desde un servidor, y además, que cualquier usuario pueda acceder a estos datos en tiempo real y de forma sincronizada desde cualquier dispositivo web o móvil.

1.2 JUSTIFICACIÓN E IMPORTANCIA

En los últimos años, la domótica ha ido instaurándose en los hogares, llegando a condicionar el funcionamiento de los mismos, de tal manera que, en la construcción de las actuales viviendas, se han comenzado a diseñar con sistemas de gestión automatizados, utilizando redes cableadas o inalámbricas como enlace entre la integración tecnológica y la estructura física de la vivienda, buscando un mejor estilo de vida en comodidad, seguridad y gestión energética.

Sin embargo, la utilización de este tipo de tecnología, aún resulta cara, puesto que la arquitectura actual no ha cambiado desde el Congreso Internacional de Arquitectura Moderna (1928-1959), en la que se delineó la funcionalidad de las viviendas, si bien los actuales diseños arquitectónicos de las estructuras buscan

sustentabilidad y sostenibilidad, estos, no han priorizado en la automatización de la vivienda, porque la integración tecnológica encarece los costos y delimita el mercado a sectores de poder adquisitivo alto.

En la actualidad existen muchos proyectos domóticos domésticos, que utilizan la plataforma de prototipos electrónicos de código abierto (open-source), conocida como en Arduino. Sin embargo, el desarrollo de aplicaciones e interfaces para Arduino requieren de conocimientos técnicos, de ahí que utilizar Arduino como plataforma domótica domestica conlleva contratar personal especializado, tanto en software como en hardware.

La domótica doméstica, terminará integrándose en todos los hogares, en mayor o menor medida, esto dependerá no sólo de las necesidades, sino también de los costos de fabricación e integración. Existen viviendas en las que la necesidad se impone al presupuesto, aunque éste resultase muy elevado. Véase el caso, por tomar un ejemplo, de un persona con algún tipo de discapacidad, que le imposibilitase abrir una puerta, encender una luz o cerrar un grifo de agua de forma tradicional, en estos casos, la necesidad se impone no sólo a los costos, sino que además resulta imprescindible, y por qué no, obligatoria.

La seguridad de los hogares es uno de los problemas que mejor se ha adaptado a la domótica doméstica, puesto que las nuevas tecnologías permiten controlar cámaras con la utilización de sus IP, abrir/cerrar ventanas o puertas utilizando aplicativos web o incluso detectar fugas de agua o de gas con App's telefónicas.

Por estos motivos, la domótica doméstica ya es parte habitual de las viviendas y la utilización de Raspberry Pi facilita esta integración, puesto que al ser una

computadora del tamaño de una tarjeta de crédito, y aunque existe otros dispositivos similares como Gooseberry o Cubieboard, la Raspberry Pi es la más barata y eficiente, además de ser la más extendida en el mercado, ya que la gran mayoría de productores de Sistemas Operativos han lanzado OS adaptables a esta plataforma. Por otra parte, es un producto con propiedad registrada pero de uso libre, lo cual abarata los costos permitiendo el uso de ésta tecnología no sólo por sus precios, sino por la poca especialización que se necesita para desarrollar aplicaciones o interfaces para Raspberry Pi.

Tabla.1

Comparativo de precios referenciales a Raspberry PI.

Board name	Price	Version	Licence	Community	Docs	Processor	Clock Speed	SoC	GPU	RAM	Memory	Max Memory
Arduino Uno	\$27 20€	Rev 3				ATmega328	16Mhz	/	/	2KB	32KB	32KB
Arduino Due	\$53 39€	Rev 1				AT91SAM3X8E	85MHz	/	/	96KB	512KB	512KB
Raspberry Pi, model B	\$35 26€	Rev 2				ARM11	700MHz	Broadcom BCM2835	VideoCore IV	512MB	None	32GB via SD
CubieBoard	\$49 36€	REV 1				ARM Cortex-A8	1GHz	Allwinner A10	ARM Mali-400	1GB	4GB NAND flash	32GB via SD
Gooseberry	\$63 46€	Rev 1				ARM Cortex-A8	1GHz	Allwinner A10	ARM Mali-400	1GB	4GB NAND flash	32GB via SD
APC Rock	\$79 59€	Rev 1				ARM Cortex-A9	800MHz	Wondemedia Prizm WM9950	ARM Mali-400	512MB	4GB NAND flash	32GB via SD
A13 OLinuXino Wifi	\$74 55€	Rev. E				ARM Cortex-A8	1GHz	Allwinner A13	ARM Mali-400	512MB	4GB NAND flash	32GB via SD
A10 OlinuXino	 	TBA				ARM Cortex-A8	1GHz	Allwinner A10	ARM Mali-400	1GB	4GB NAND flash	32GB via SD
Hackberry A10	\$65 48€	Rev 1				ARM Cortex-A8	1.2GHz	Allwinner A10	ARM Mali-400	1GB	4GB NAND	32GB via SD

Fuente: (Techwatch, 2016)

1.3 OBJETIVO GENERAL

Desarrollar e Implementar un sistema integrado de software y hardware, que permita controlar un grupo independiente de luces para un número ilimitado de usuarios.

1.4 OBJETIVOS ESPECÍFICOS

- Diseñar y Desarrollar el sistema Domótico doméstico de Software y Hardware para un Aplicativo Web útil en una PC y en dispositivos móviles (Smartphone o tablet's), y que además, sea capaz de interactuar con el usuario por medio de una conexión cableada o inalámbrica de internet, de tal forma que el uso de las luces de la casa puedan ser manipuladas por varios usuarios al mismo tiempo.
- Identificar los requerimientos funcionales y no funcionales del Sistema de automatización de una vivienda, de acuerdo a las adaptaciones del sistema Domótico Domestico instalado en la vivienda.
- Utilizar una metodología ágil para el desarrollo del sistema (software), con el fin de obtener entregables periódicamente, los mismos que faciliten la implementación del sistema integrado de software y hardware.
- Implantar la aplicación para PC y dispositivos móviles (Smartphone o tablet's), y su eficacia entre varios usuarios.

1.5 ALCANCE

El proyecto contempla el Desarrollo e implantación de un aplicativo web, también, adaptable a un teléfono celular (aplicación móvil).

Este sistema se ha desarrollado por la búsqueda del abaratamiento de costos en la domótica doméstica y en la innovación de nuevas plataformas de hardware y software para el control automatizado de toda la vivienda, puesto que aunque en este proyecto se utilice para el control de luces, los aplicativos de hardware y software que se ha desarrollado, en este proyecto, se pueden adaptar a toda la estructura interna y externa de una vivienda: luces, agua, gas, electrodomésticos, etc.

La utilización de una base de datos No SQL (Firebase), es completamente nueva en la domótica doméstica, ya que lo que se ha hecho es vincularla a un aplicativo web y una Raspberry Pi para poder actualizar el estado de la aplicación en tiempo real, de tal forma que puedan acceder una cantidad ilimitada de usuarios, que en este caso, bien podrían ser los miembros de una familia.

La estructura completa del sistema está distribuida en 3 grupos principales:

1. SOFTWARE: La configuración de la Raspberry Pi, necesita la instalación de varias librerías (incluyendo el levantamiento de los GPIO de su placa base), además de la instalación de la Base de Datos No SQL (Firebase), así como también el enlace entre el Sistema Operativo de la Raspberry Pi (Debian), la Base

de Datos No SQL (Firebase) y el aplicativo Web que se ha desarrollado como interfaz principal.

2. HARDWARE: La Raspberry Pi es un producto con propiedad registrada pero de uso libre, conocida como la computadora más pequeña del mundo, que a diferencia del Arduino que es una placa controladora, la Raspberry Pi, no necesita de programación interna puesto que cuenta con su propio Sistema Operativo y cumple todas las funcionalidades de Hardware que cumple un computador de escritorio. La Raspberry Pi, es en realidad el cerebro del proyecto.

3. DISPOSITIVOS ELÉCTRICOS Y ELECTRÓNICOS: Aunque la Raspberry Pi esté en condiciones de controlar cualquier dispositivo eléctrico o electrónico, su voltaje de salida resulta ser muy pequeño (5 DC), por ello se requiere de rele's integrados en un circuito electrónico, además serán necesarios otros circuitos eléctricos/electrónicos independientes que puedan conectarse a la Raspberry Pi.

CAPITULO II. MARCO TEÓRICO

2.1 GENERALIDADES

2.1.1 INTRODUCCIÓN

En la actualidad, más del 40% de los hogares ecuatorianos, a nivel urbano, tiene acceso ilimitado a internet (Fig. 1)¹, ello quiere decir que: casi la mitad de los hogares de las zonas urbanas están en condiciones de domotizar su hogar utilizando Internet como enlace de conexión para transmitir información, y en el caso de este proyecto, para convertir esa información en pulsos eléctricos con la finalidad de controlar el sistema de iluminación eléctrico de la vivienda.

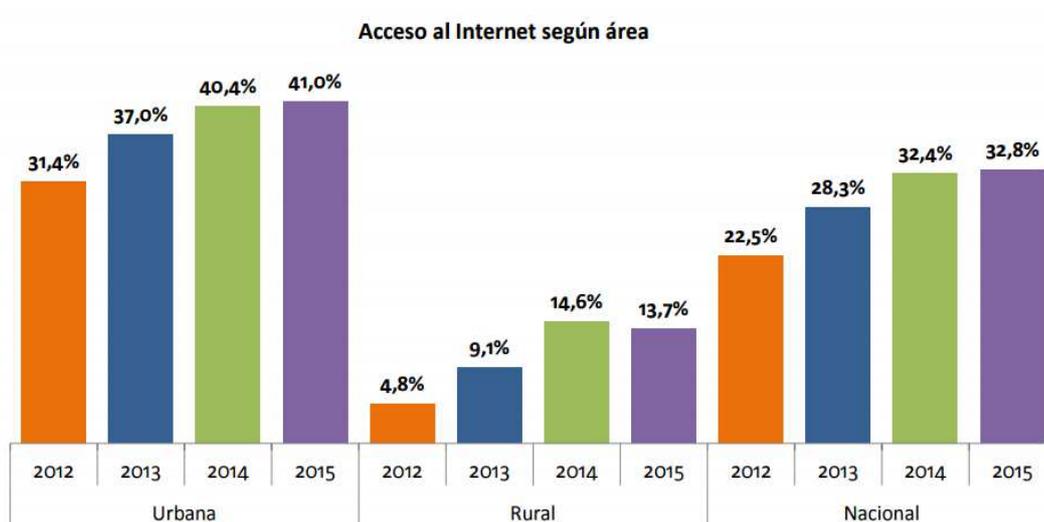


Figura 1 Gráfico Estadístico de los hogares ecuatorianos con acceso al Internet.
Fuente: (ENEMDU, INEC, 2016, pág. 7)

¹ http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2016/170125.Presentacion_Tics_2016.pdf

Aunque la “FireBase”, necesita de internet para poder actualizar la información en tiempo real, ésta también ha sido optimizada para poder usarla sin conexión a la red, ya que si el usuario pierde la conexión, el SDK (teniendo en cuenta que la FireBase posee SDK para Android, iOS y JavaScript) de la base de datos, usará la caché local del dispositivo para almacenar los cambios y cuando el usuario vuelva a conectarse, todos los datos locales automáticamente se volverán a sincronizar. Sin embargo, el aplicativo deberá siempre estar conectado a internet.

Entonces, existen las condiciones necesarias no sólo para desarrollar el sistema de CONTROL DOMÓTICO UTILIZANDO RASPBERRY PI y una BASE de DATOS NoSQL, sino que además se da el entorno adecuado para incorporarlo en los hogares ecuatorianos.

El sistema de CONTROL DOMÓTICO UTILIZANDO RASPBERRY PI y una BASE de DATOS NoSQL, controla de forma automatizada y sincronizada el uso del sistema eléctrico de iluminación, de la vivienda, utilizando dispositivos móviles, Smartphone`s, o PC`s de cualquier tipo, con el único requisito de poseer un navegador web instalado, que le permita acceder al aplicativo web, quien será el encargado, por medio de su interfaz, de controlar y mostrar el estado on/off del sistema de iluminación eléctrico de la vivienda.

Como dato relevante, ha de entenderse que la funcionalidad del sistema de CONTROL DOMÓTICO UTILIZANDO RASPBERRY PI y una BASE de DATOS NoSQL, no sólo serviría para controlar el sistema eléctrico de iluminación, sino que también podría aplicarse al control automatizado del sistema eléctrico de fuerza o,

en algún caso particular, cualquier electrodoméstico electrónico o eléctrico que existiese en la vivienda.

2.1.2 SITIO O PORTAL WEB

Un Sitio o Portal Web es una aplicación que, por medio de una interfaz simple, provee a sus usuarios (consumidores de datos) acceso a diferentes fuentes de datos o proveedores. (CALERO, 2010, pág. 568). Para el caso de este proyecto, en particular, la aplicación web proveerá a los usuarios del estado on/off de las lámparas de la vivienda.

La idea es emplear estos portales para localizar la información y los sitios que interesan y de ahí comenzar con la actividad en Internet. Un Sitio Web no alcanza el rango de portal sólo por tratarse de un sitio robusto o por contener información relevante. Un portal es más bien una plataforma de despegue para la navegación en la Web (MILENIUM, 2016).

Así pues, según CALERO y ESCOBAR, el sitio o portal web, no es otra cosa que una puerta de entrada de datos, en una interfaz capaz de modificar la información según los requerimientos de un determinado usuario, en el caso de este proyecto, el usuario buscará actualizar la información que recibe de la interfaz web.

2.1.3 FUNCIONALIDADES DE UN SITIO O PORTAL WEB

- Información en tiempo real.

Un sitio o aplicación web, deberá permitir al usuario requerir de una determinada información externa, antes publicada, permitiendo que éste usuario, actualice dicha información en tiempo real, de tal forma que, esa información, se modifique de forma interna.

- Clasificación automática de los contenidos.

Hay que entender que un aplicativo web, no es un ente estático, ya que éste debe tener la capacidad de gestionar, de forma dinámica, el flujo de información, de tal forma que los directorios (de la FireBase) existentes se modifiquen en tiempo real. Así pues, la información mostrada en la interfaz quedará actualizada.

Un sitio o portal web, deberá tener muchas más funcionalidades, como un directorio de navegación o el enlace a sitios web internos, sin embargo, para el caso de estudio de este proyecto, sólo se han presentado dos funcionalidades básicas, pues el aplicativo web desarrollado, es una interfaz de actualización de contenido del estado de la máquina Raspberry Pi.

2.1.4 TIPOS DE PORTAL WEB.

Existen básicamente dos tipos de portales web:(CASTRO, 2016).

- Portales verticales.- Son páginas especializadas que funcionan como punto de entrada a un mercado, un tema o un interés en específico.
- Portales horizontales.- Son páginas especializadas que funcionan como punto de entrada a diversos temas. Su intención es abarcar el mayor número de usuarios posibles.

Para este proyecto, el portal requerido es un Portal Vertical, pues la información requerida apunta a un interés específico.

2.2 INTERNET

2.2.1 INTERNET CONCEPTO.

No es una simple red de ordenadores, sino una red de redes, es decir, un conjunto de redes interconectadas a escala mundial con la particularidad de que cada una de ellas es independiente y autónoma.

Hoy en día, el Internet es un medio de comunicación pública, cooperativa y autosuficiente en términos económicos, accesible a cientos de millones de personas en el mundo entero. Físicamente, el Internet usa parte del total de recursos actualmente existentes en las redes de telecomunicaciones. Técnicamente, lo que distingue al Internet es el uso del protocolo de comunicación

llamado TCP/IP (Transmission Control Protocol/Internet Protocol). (UNIVERSIDAD FRANCISCO GAVIDIA, 2017, pág. 2).

Los componentes básicos del Internet, se basa en tres bases fundamentales: (UNIVERSIDAD DE LA PUNTA, 2017, pág.1).

- Servidores.
- Líneas de comunicaciones primarias entre nodos de Internet.
- Clientes.
- Programas o software específico.



Figura. 2 Funcionamiento de la Internet,
Fuente: (UNIVERSIDAD DE LA PUNTA, 2017, pág.1).

2.3 BASE DE DATOS.

2.3.1 CONCEPTO DE BASE DE DATOS.

Una base de datos, es un conjunto de datos relacionados entre sí, organizados y estructurados, con información referente a algo. Podremos utilizar una base de datos para cosas tan sencillas como mantener un registro de nuestra agenda personal de teléfonos, o tan complicadas como llevar toda la gestión de una gran empresa u organización. (UNIVERSIDAD CONTINENTAL, 2015, pág. 57).

En el caso de este proyecto, servirá para que la información almacenada en la Base de Datos actualice el estado real de la interfaz, mostrando el momento on/off de alguna de las lámparas del sistema eléctrico de iluminación.

Así entonces, se puede entender una Base de Datos como: Es un fondo común de información almacenada en una computadora para que cualquier persona o programa autorizado pueda acceder a ella, independientemente de su procedencia y del uso que haga. (PONS, 2009, pág. 6).

Si bien, RAMOS y PONS, conceptualizan lo que es una Base de Datos, hay que entender, además, que una Base de Datos contiene un esquema estructural lógico organizado, que permite procesar y generar información cada vez que el usuario acceda a esta colección de datos.

2.3.2 CARACTERÍSTICAS DE LA BASE DE DATOS.

Entre las principales características de los sistemas de base de datos podemos mencionar: (REDIRIS, 2017).

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

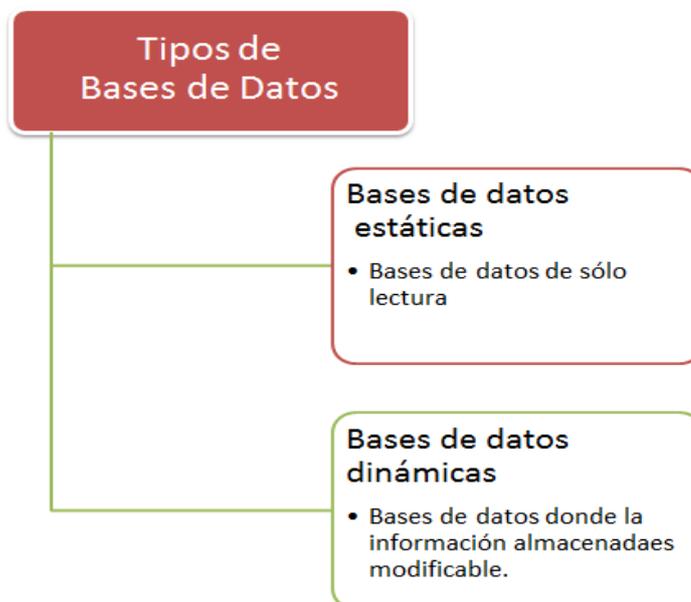


Figura. 3 Tipos de Bases de Datos
Fuente: (DATE, 2007, Séptima Edición).

2.3.3 BASES DE DATOS NoSQL.

Las bases de datos NoSQL son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación. Tampoco utilizan una estructura de datos en forma de tabla donde se van almacenando los datos sino que para el almacenamiento hacen uso de otros formatos como clave-valor, mapeo de columnas o grafos (TELEFÓNICA, 2017).

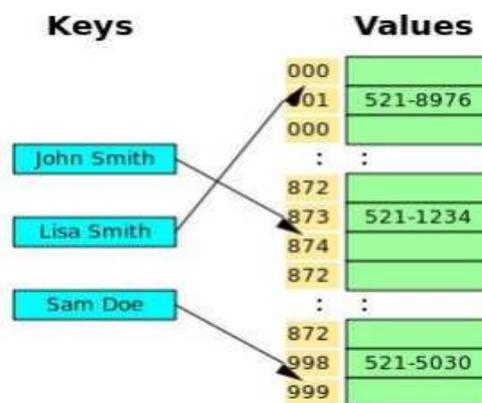


Figura. 4 Bases de Datos Clave-Valor grafos.
Fuente: (TELEFÓNICA, 2017)

2.3.4 FIREBASE

Firebase es una tecnología que permite hacer aplicaciones web sin necesidad de programación. Con Firebase, no existe la necesidad de insistir en el sobreaprovisionamiento de servidores o la construcción de las API REST. Es compatible con la web, iOS, OS X, y Android. Las aplicaciones que utilizan Firebase pueden utilizar y controlar los datos, sin tener que pensar en la forma en que se almacenan

los datos, y se sincronizan a través de varios ejemplos de la aplicación en tiempo real. No hay necesidad de escribir código del lado del servidor (QUORA, 2016).

Lo destacado de utilizar una FIREBASE, es que para acceder al servicio web, no es necesario acudir a un servidor ya que una API guarda y sincroniza datos en la nube en tiempo real.

2.4 REDES

2.4.1 CONCEPTO DE REDES

Es un conjunto de máquinas que se interconectan entre sí por algún medio físico (permanente o no) y cuyo cometido es facilitar el intercambio de información entre diferentes emisores y receptores. (DE PALOS, 2011).

Si bien DE PALOS, conceptualiza lo que es una red, ha de entenderse, también, que una red no sólo permite el intercambio de información, sino que además, como es en el presente caso, permitirá el control automatizado de un determinado periférico, utilizando la red como una herramienta de conexión.

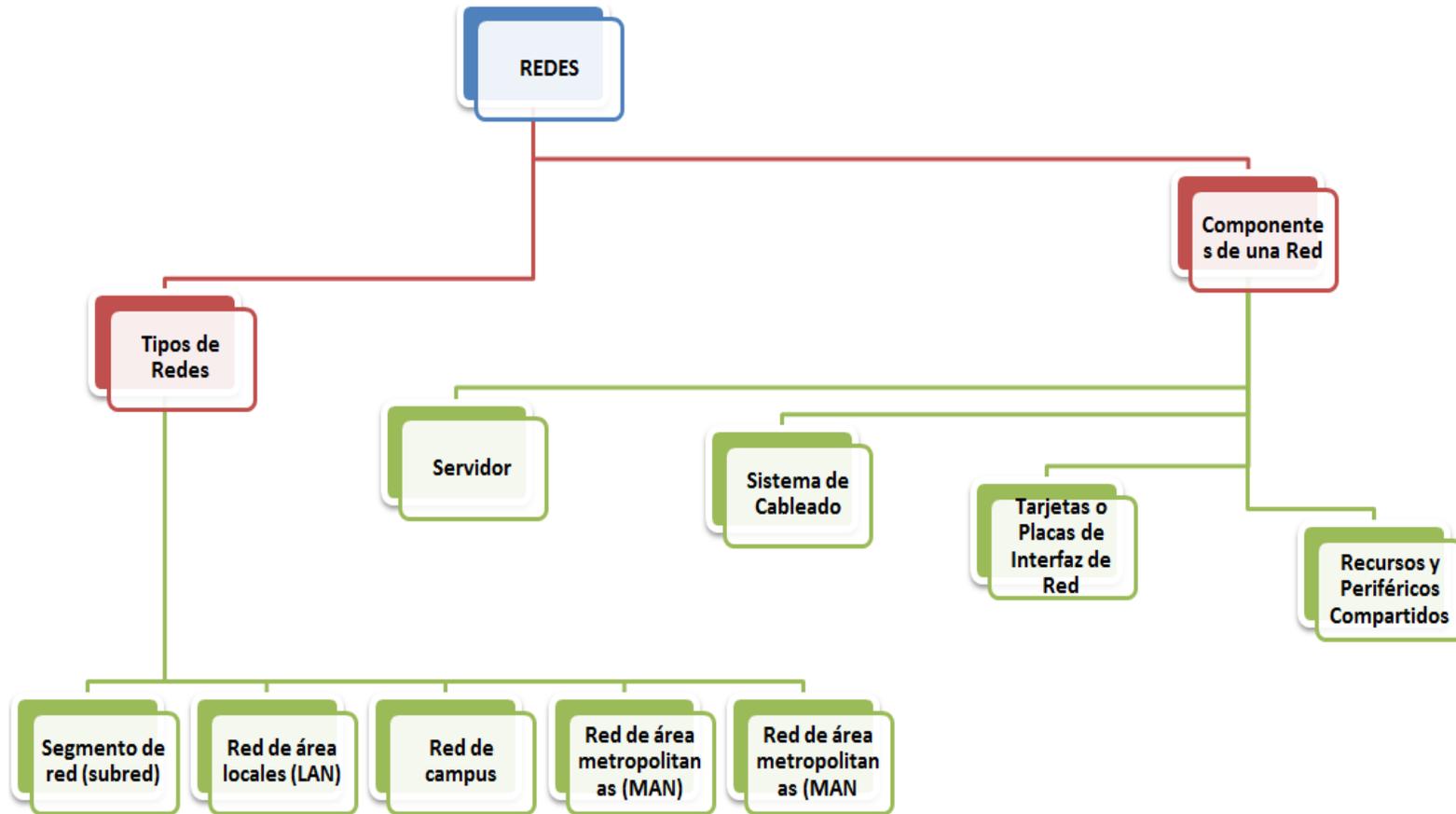


Figura. 5 Tipos y componentes de una RED.
Fuente: (BARBA, 1999)

2.4.2 CARACTERÍSTICAS DE LAS REDES

Los sistemas operativos sofisticados de red local como el Netware Novell ofrecen un amplio rango de servicios. A continuación, se cita algunas características principales: (Google Docs, 2017).

- Servicios de archivos.
- Compartir recursos.
- SFT (Sistema de tolerancia a fallas).
- Sistema de Control de Transacciones.
- Seguridad.
- Acceso Remoto.
- Conectividad entre Redes.
- Comunicaciones entre usuarios.
- Servidores de impresoras.
- Colas de impresión.

2.5 SERVIDORES WEB

2.5.1 CONCEPTO DE SERVIDORES WEB

Son ordenadores que ofrecen sus servicios al resto de equipos conectados. Suelen tener una presencia estable en la red, lo que se concreta en tener asignadas direcciones IP permanentes. En ellos es donde están alojadas, por ejemplo, las páginas web (COBO, 2005, pág. 6).

2.5.2 CONCEPTO SERVIDOR APACHE.

Actualmente más del 60 por 100 de los administradores de toda la Web utilizan Apache. Se trata de la plataforma de servidores Web de código fuente abierto más poderosa del mundo. Día a día aumenta el número de corporaciones que aceptan este maravilloso código fuente abierto en su infraestructura IT. Son muchas las grandes compañías, como IBM, que ofrecen Apache entre sus productos. El futuro de Apache parece muy prometedor (KABIR, 2003).

Para el desarrollo de este proyecto se ha utilizado este servidor puesto que APACHE 2 puede utilizar archivos HTML a través de HTTP, y además puede ejecutar páginas web dinámicas utilizando lenguajes de programación como PHP. Ténganse en cuenta que la utilización de este servidor se debe a que la máquina Raspberry Pi, tiene como Sistema Operativo Debian, y el servidor APACHE 2 se ejecuta y combina perfectamente con este sistema operativo.

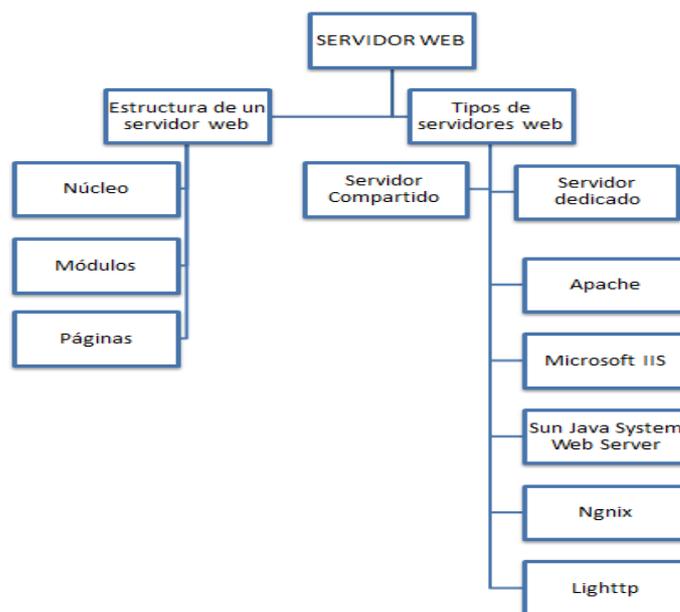


Figura. 6 Estructura y tipos de Servidores Web.

Fuente: (TIMOTHY, 2010).

2.6 INGENIERÍA DE SOFTWARE

2.6.1. INTRODUCCIÓN

En la actualidad, el mundo del software ha visto un floreciente crecimiento, de allí es pues que los usuarios y las empresas, han buscado formas de automatizar sus procesos, mejorando su productividad y/o comodidad, para adaptarse a los cambios tecnológicos de esta era. Sin embargo, para que esa automatización sea eficaz y eficiente, se necesita de personal especializado y tecnificado, siendo el recurso más valioso el talento humano y las técnicas para poder aplicar estas tecnologías en la automatización y organización de procesos.

2.6.2 CONCEPTO DE INGENIERÍA DE SOFTWARE.

La aplicación de una aproximación sistemática, disciplinada y cuantificable, al desarrollo, las operaciones y al mantenimiento del software; Esto es básicamente la aplicación de la Ingeniería al software (SOMMERVILLE, 2006).

2.7 HERRAMIENTAS DE DESARROLLO WEB.

El mercado actual, ofrece un sin número de herramientas de desarrollo web (herramientas con licencia u Open Source), que los programadores pueden utilizar según sus necesidades y conocimientos, puesto que, en la gran mayoría de estas herramientas, existe la particularidad de un determinado lenguaje de programación.

La utilización de estas herramientas de desarrollo web ha facilitado el trabajo de los programadores, mejorando la organización de los procesos y reduciendo el tiempo de desarrollo.

2.8 SOFTWARE LIBRE

2.8.1 CONCEPTO DE SOFTWARE LIBRE

Según la Licencia Pública General, para que se considere un software libre han de reunirse las siguientes libertades: (MENA, 2007)

1. La libertad de usar el programa, con cualquier propósito.
2. La libertad de estudiar cómo funciona el programa y adaptando a tus necesidades. El acceso al código fuente es una condición previa para esto.
3. La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
4. La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

Ha de tenerse, precaución en no confundirse software libre con "software de dominio público".

2.9 HARDWARE LIBRE

2.9.1 CONCEPTO DE HARDWARE LIBRE

El hardware abierto o libre toma estas mismas ideas (libertades del Software Libre) para aplicarlas en su campo. Es una propuesta casi tan antigua como la del software libre, sin embargo su empleo no es tan directo. Compartir diseños hardware es más complicado. No hay una definición exacta (se pueden encontrar referencias a distintos). Al no existir una definición clara de hardware abierto, cada autor lo interpreta a su manera. Se han creado licencias, algunas de las cuales están todavía en desarrollo (GONZÁLES, 2003, pág. 1).



Figura. 7 Clasificación del hardware Libre, según su naturaleza.

Fuente: (GONZÁLES, 2003).

2.10 SINGLE BOARD COMPUTER.

2.10.1 RASPBERRY Pi.

Los computadores de placa reducida o SBC (Single Board Computer) son computadores completos en una sola placa de tamaño reducido, que son lo suficientemente potentes para ejecutar un sistema operativo real, además estos son de bajo costo y versátiles. Por lo cual son dispositivos útiles para la ejecución de diversos proyectos ya sean educativos o de investigación científica (GIULIANELLI, 2017).

Tabla. 1

Comparación entre las tecnologías RASPBERRY PI MODELO B y CUBIEBOARD2

Especificaciones Técnicas	Raspberry Pi modelo B	Cubieboard2
Cpu	ARM 1176JZF-S a 700 MHz	ARM Cortex -A7 Dual- Core (1GHz)
Gpu	Broadcom VideoCore IV	Mali 400 MP2
Ram	512 MB (Compartidos con la GPU)	1GB DDR3 @480M
Almacenamiento interno	N/A	4GB interna NAND flash
Almacenamiento SD	SD/MMC/ Ranura para SDIO	64GB Slot microSD (con soporte SDHC)
Conectividad	1 Puerto USB 2.0	2 USB , 1 SATA, 1 infrarrojos
Salida de video	Conector RCA (Pal y Ntsc), HDMI (Rev 1.3 y 1.4)	HDMI 1080p
Salida de audio	Conector de 3.5mm, puerto HDMI	Por HDMI o Salida jack 3.5 mm
Sistema operativo	Linux vía Micro SD	Android preinstalado, Linux vía Micro SD
Red	Puerto RJ45 Ethernet 10/100	10/100 RJ45 ethernet
Pines GPIO	8 x GPIO, SPI,UART	2x48 pines con posibilidad de: I2C, SPI, RGBN,VGA o touch- panel
Alimentación	5V vía Micro USB o GPIO header	5VDC input 2A or USB otg input
Botones	N/A	1 botón de encendido/apagado y 1 botón para entrar en modo FEL (para flashear la memoria interna)
Sistema operativo	Linux vía Micro SD	Android preinstalado, Linux vía Micro SD
Red	Puerto RJ45 Ethernet 10/100	10/100 RJ45 ethernet

Fuente: (ENGI Revista Electrónica De La Facultad De Ingeniería Vol. 3 No. 1 3 ISSN 2256-5612., 2017)

Como se puede apreciar en la **Tabla. 2**, existen varias tecnologías que utilizan el concepto de Single Board Computer. Para el desarrollo de este proyecto se ha utilizado la tecnología Raspberry Pi, basada en el microcontrolador Atmel ATmega644 y fundada en Caldecote, en mayo de 2009, con la denominación de “Fundación Raspberry Pi”, la cual tuvo como objetivo un fin caritativo.

2.11 DOMÓTICA.

Desde hace bastantes años se están desarrollando numerosas soluciones para la mayor integración entre todos los sistemas y equipos domésticos. La integración tecnológica de los sistemas electrotécnicos en el hogar se ha venido denominando en muchas ocasiones como Domótica. Sin embargo, es importante conocer que, de forma estricta, se define la vivienda domótica como: “aquella vivienda en la que existen agrupaciones automatizadas de equipos, normalmente asociados por funciones, que disponen de la capacidad de comunicarse interactivamente entre ellas a través de un bus doméstico multimedia que las integra” (JUNESTRAND, 2004).

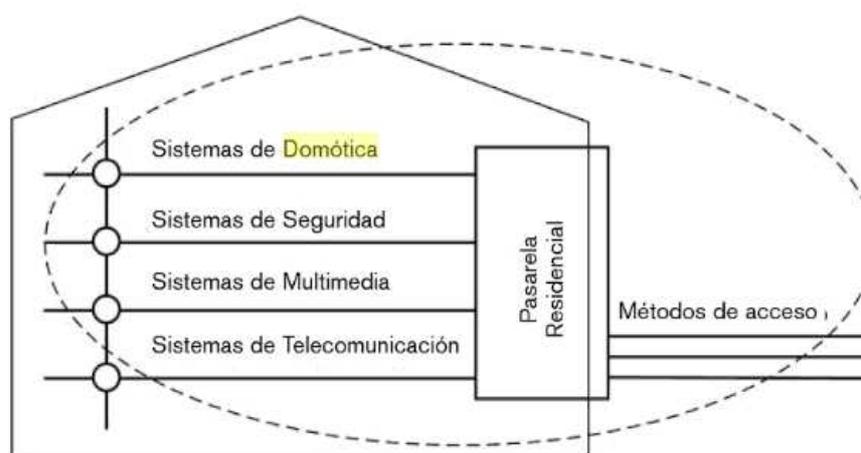


Figura. 8 Hogar digital.
Fuente: (JUNESTRAND, 2004).

2.12 METODOLOGÍA

Para el desarrollo del aplicativo se ha seleccionado la metodología UWE (UML Web Engineering), ya que incorpora elementos básicos en el desarrollo Web y además, ésta metodología aprovecha la notación estándar del UML.

2.12.1 UWE INTRODUCCIÓN.

Casi el ciento por ciento, de los aplicativos web, es ejecutado en la World Wide Web.

Mediante la utilización de páginas estáticas, la evolución de la web ha mejorado sus procesos de concurrencia, seguridad, organización y optimización, dentro de lo conocido como ingeniería web.

La existencia de metodologías web, y sus procesos de desarrollo han demostrado que la metodología UWE es la más destacada, puesto que permite que un aplicativo web sea más detallado, sistematizado y organizado, basándose en la notación estándar del uso de UML (Unified Modeling Language), para ello utiliza varias etapas y modelos de implementación.

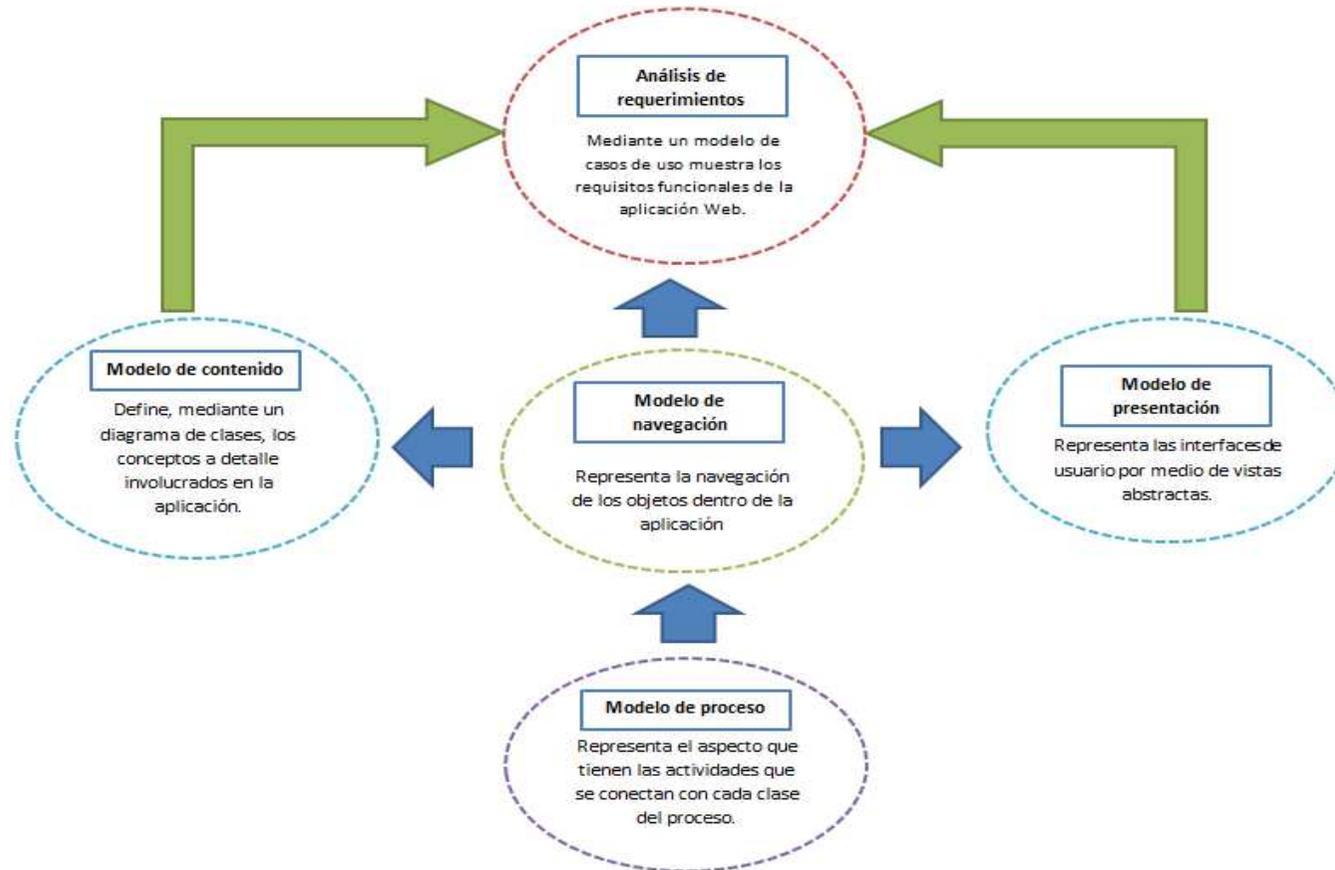


Figura. 9 Etapas y Modelos UWE.
Fuente: (ESTRELLA, 2017).

Para el análisis de requerimientos en la metodología UWE, se utiliza los modelamientos de casos de uso, es decir la estructura: Actor ↔ Casos de Usos, en donde el actor es el usuario del aplicativo web.

Téngase en cuenta que el Modelo de Contenido (representado con diagramas de clase) viene a ser el Modelo Conceptual del aplicativo web, claro está, directamente relacionado con los requerimientos establecidos en los Casos de Uso. Del resultado del análisis de requisitos junto con el Modelo de Contenido se obtendrá el Modelo de Navegación (representado con diagramas de clase). Finalmente, el modelo de Navegación permitirá obtener el modelo de Presentación.

2.12.1.1 FASE DE LA METODOLOGÍA UWE.

La notación UML deriva las siguientes fases para la metodología UWE:

- **Análisis de requerimientos:** El objetivo principal de este modelo es estructurar el alcance y el proceso del sistema, entendiendo que los requerimientos se han de clasificar en funcionales y no funcionales.
- **Modelo Conceptual:** En este modelo se presentará la relación de los contenidos del aplicativo web, utilizando los diagramas de clase para determinar la estructura de los datos hospedados en el aplicativo web.

- **Modelo Navegacional:** En este modelo representa la interacción que tendrá el usuario con la interfaz del aplicativo web, mostrando las rutas estructuradas que el usuario utilizará para navegar por la interfaz.
- **Modelo de Presentación:** En este modelo se especificarán las clases del aplicativo web, mostrando la interfaz final que a la que el usuario tendrá acceso.

2.12.2 ARTEFACTOS

2.12.2.1 CASO DE USO DIAGRAMAS.

Un diagrama de casos de uso es una colección de situaciones que se documentan a partir del punto de vista del usuario. Por lo tanto los casos de uso es una de las herramientas más importantes, en la conversión de los requisitos vistos desde el lado del usuario en requisitos a ser implementados por los profesionales del desarrollo de Software (LÓPEZ, 2006).

2.12.2.2 DIAGRAMA DE CLASES

Este diagrama es la representación gráfica de la estructura del sistema, ya que los diagramas de clases son los utilizados durante el diseño y análisis del aplicativo web, utilizando relaciones entre los objetos que forman parte del aplicativo web, así pues, éstas relaciones pueden ser:

- Herencia
- Composición

- Agregación
- Asociación

2.12.2.3 DIAGRAMAS DE SECUENCIA

La funcionalidad de cada caso de uso estará detallada en la estructura secuencial de éste diagrama, mostrando por medio de módulos la intervención del usuario con la interfaz del aplicativo web, conteniendo los siguientes elementos:

- Objetos
- Mensajes
- Métodos

2.12.2.4 DIAGRAMAS DE ESTADO

En este diagrama se mostrara los sucesos temporales variables de un determinado objeto, es decir, mostrará la actividad del aplicativo web. Estos diagramas, se usan para identificar las vías que escogen los flujos de información después de realizar un proceso. Sus elementos principales son:

- Acciones
- Actividades
- Transiciones
- Estados

2.12.2.5 DIAGRAMAS DE DESPLIEGUE

Hace referencia al enlace real entre componentes, archivos y elementos de hardware que forman parte del aplicativo web.

2.12.2.6 DIAGRAMAS DE ACTIVIDADES

El diagrama de actividades permite ver el comportamiento de cada proceso de la aplicación web y las interfaces que permiten manejar dichos procesos (DEITEL, 2004).

2.12.3 LENGUAJE DE MODELAMIENTO UNIFICADO (UML)

Dentro de las tareas del modelamiento UML están las de documentar, identificar y construir un sistema web, albergando funcionalidades como la conexión a bases de datos. Ha de entenderse que, UML es un lenguaje de modelado que documenta el desarrollo metodológico del aplicativo web, creando diagramas estructurales del funcionamiento del sistema, permitiendo, al usuario, entender la disposición del aplicativo por medio de los casos de uso.

El modelamiento, por medio de diagramas UML permite identificar las clases, objetos, atributos, operaciones y relaciones, mostrando los cambios de estado que un determinado objeto realiza durante la ejecución del aplicativo web.

CAPITULO III. DESARROLLO DE LA SOLUCIÓN

El presente proyecto responde a la necesidad de desarrollar aplicativos domóticos dentro de los hogares, ya que en el Ecuador, este tipo de aplicativos se han visto marginados debido a su alto costo y/o a la falta de personal técnico para realizar estos sistemas.

3.1 MODELAMIENTO DE CASOS DE USO

Para el desarrollo del aplicativo web que controlará el sistema eléctrico de la vivienda se hace necesario identificar los requerimientos en base a la metodología UWE, utilizando modelamientos diagramales de casos de casos de uso, los cuales permiten conceptualizar la funcionalidad usuarios/interfaz.

3.1.1 MODELO DE REQUERIMIENTOS

Se realiza en base a la siguiente estructura:

- Especificación de Requerimientos de Software según el estándar IEEE830-1998A.
- Lista de Actores.
- Diagramas de Casos de Uso y
- Descripción de Casos de Uso.

3.2 Lista de Actores

En la Tabla. 3 se identifica y conceptualiza, de forma general, el listado de los actores del aplicativo web:

Tabla 2

Actores del aplicativo web.

ACTORES DEL APLICATIVO WEB		
1	Administrador del Sistema	Es el encargado de la configuración y mantenimiento del Aplicativo Web y quien monitorizará Usuarios, Perfiles de Usuario. Etc.
2	Usuario	Son los usuarios que navegan por el Aplicativo Web (Habitantes/ocupantes de la vivienda), para controlar de forma remota y desde cualquier periférico inteligente el sistema eléctrico de la vivienda.

3.3 DIAGRAMAS DE CASO DE USO

Para el desarrollo de diagramas de este aplicativo web se dividido en dos grupos generales:

1. Configuración
2. Acceso

3.4 CASOS DE USO DEL MÓDULO DE CONFIGURACIÓN

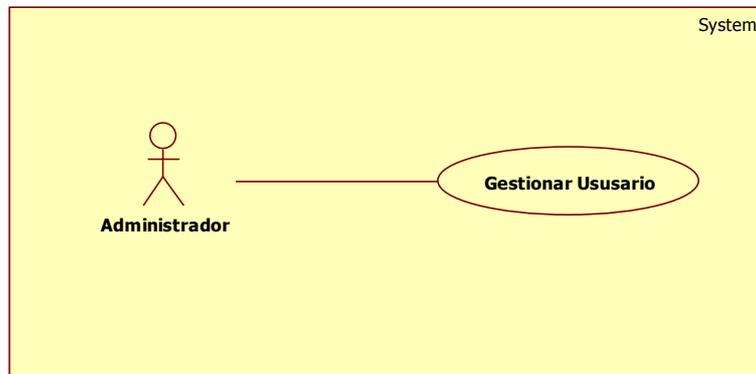


Figura. 10 Diagrama de Casos de uso para el Módulo de Configuración.

3.5 CASOS DE USO DEL MÓDULO ACCESOS

En el siguiente diagrama se puede ver como el usuario accede al sistema de iluminación, validando el aplicativo web desde su periférico inteligente (Dirección_IP/tiempo_real_sc/).



Figura. 11 Diagrama de Casos de uso para del Módulo Accesos.

3.6 DESCRIPCIÓN DE CASOS DE USO

La descripción de los Casos de Uso (CU) se realizará mediante una tabla descriptiva, en la cual se podrá apreciar:

- Nombre
- Descripción
- Actores
- Pre-condiciones
- Actividades.

3.6.1 MÓDULO DE CONFIGURACIÓN

Tabla 3

Descripción Caso de Uso Gestionar Usuario.

Caso de uso: Asignar Usuario	
1	Descripción: Permite Gestionar (ingresar, eliminar actualizar) un usuario que esté conectado a la red, por medio de la seguridad en la FireBase.
2	Actores: Administrador del Sistema.
3	Pre-condiciones: El usuario debe haber ingresado con perfil de Administrador, al aplicativo de la FireBase.
4	Actividades: Escenario principal (Autorizar a un usuario) : <ol style="list-style-type: none"> 1. El Administrador verifica y/o selecciona los usuarios que pueden acceder al aplicativo web. 2. El Administrador modifica las seguridades establecidas en la FireBase. 3. El Administrador guarda en el aplicativo de la FireBase las nuevas restricciones. 4. El sistema de la FireBase informa al Administrador, que los datos han sido actualizados con éxito.
5	Escenarios alternos (Ingresar o Eliminar Perfil de Usuario) : <ol style="list-style-type: none"> a) Ingresar nuevo Usuario <ol style="list-style-type: none"> 1. El Administrador permite el acceso de un nuevo usuario al aplicativo web. 2. El Administrador guarda en el sistema los datos ingresados. 3. El sistema permite de forma automática el acceso de un usuario nuevo al aplicativo. b) Eliminar un Usuario existente <ol style="list-style-type: none"> 1. El Administrador, por medio de las seguridades del aplicativo de la FireBase, restringe el acceso de un usuario existente. 2. El aplicativo web, de forma automática, invalida el acceso del usuario. 3. El aplicativo web no se puede ejecutar desde el periférico invalidado.

3.6.2 MÓDULO DE ACCESO

Éste módulo, es el que permitirá la inacción del usuario con la interfaz, permitiéndole controlar el sistema eléctrico de la vivienda.

Tabla 4

Descripción Caso de Uso Acceso.

Caso de uso: Acceso al aplicativo web.	
Descripción:	
1	Permite que un usuario acceda al aplicativo web, y por medio de la interfaz, controle el sistema eléctrico de la vivienda.
2	Actores: Usuario.
3	Pre-condiciones: El usuario debe haber ingresado a la red privada de la vivienda.
4	Actividades: Escenario principal (Navegar por el aplicativo web): <ol style="list-style-type: none"> 1. El usuario visualiza en la interfaz los interruptores de ON/OFF de cada una de las luces de la vivienda. 2. Cada vez que el usuario activa o desactiva una lámpara del sistema eléctrico, la FireBase actualiza, en todos los periféricos existente, el estado real ON/OFF de las lámparas. Es decir, si un usuario ha encendido la lámpara de la sala, en el periférico de los otros usuarios se mostrará el estado ON del interruptor, y viceversa si se ha apagado. 3. Cada vez que el usuario se desconecte de la red, a la que está enlazada la máquina Raspberry Pi, el aplicativo dejará de funcionar.

3.7 MODELO DE CONTENIDO

En este modelo se muestran las entidades del aplicativo web, utilizando un Diagrama de Clases, para los Módulos de, entiéndase que dentro del Diagrama de Clase se muestran únicamente los atributos y no las operaciones, ya que no está orientada a objetos.

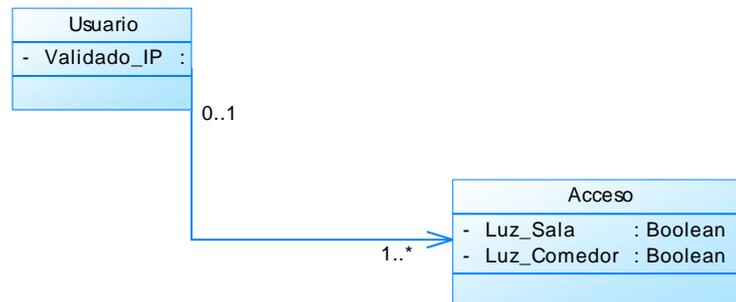


Figura. 12 Diagrama de Clases Módulo Accesos.

3.8 DIAGRAMA DE MODELO DE NAVEGACIÓN

Cuando se realiza un aplicativo web, es necesario saber cómo están enlazadas las diferentes páginas, en el caso de presente proyecto es necesario conocer el funcionamiento de cada uno de los elementos de la interfaz web, que mediante un diagrama de contenidos (nodos navegables) y enlaces se especifica la relación de contenidos mediante los menús que direccionan las rutas de navegación.

Tabla 5

Notación UWE para los elementos del modelo de Navegación.

Notación UWE para los elementos del modelo de Navegación			
Nº	Elemento	Estereotipo UWE	Símbolo
1	Interruptor en estado ON.	«clase de proceso»	
2	Interruptor en estado OFF.	«clase de proceso»	

El diagrama de navegación del aplicativo permite ver en forma de nodos navegables, la estructura de funcionalidad del aplicativo web. Es decir, en el caso del usuario, permitirá conocer las rutas de acceso que tiene el usuario para poder controlar el sistema domótico, controlando las luces de la vivienda, dependiendo de la dirección a la que se ha asignado el nodo, para este proyecto, sólo se ha asignado dos rutas:

- Luz Sala y,
- Luz Comedor.

Además, cómo se puede apreciar en la fig. 13, referente al diagrama de navegación, sólo existe un módulo de navegación.

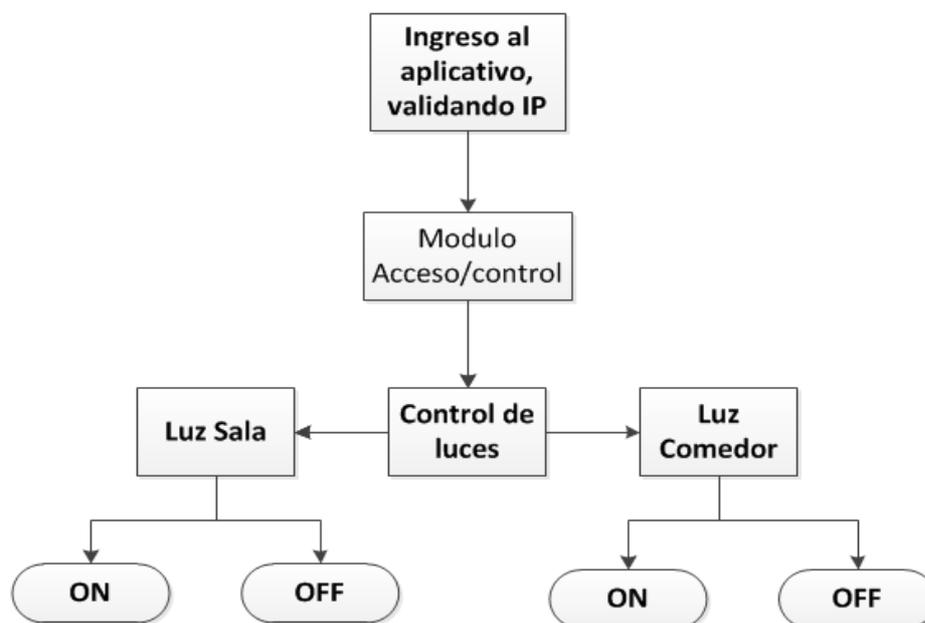


Figura. 13 Modelo de Navegación para gestión de luces.

3.9 MODELAMIENTO DE LA FIREBASE (Clave-Valor).

Este modelo se muestra los datos de la Firebase Realtime Database que se almacenarán como objetos JSON, véase esta Base de Datos como un árbol JSON, enlazado mediante nodos y cuyos valores se alojan en la nube, ya que las FireBase no utiliza ni tablas ni registro.

En éste proyecto, sólo se utilizará un sólo nodo, llamado luces con tres claves/valores.

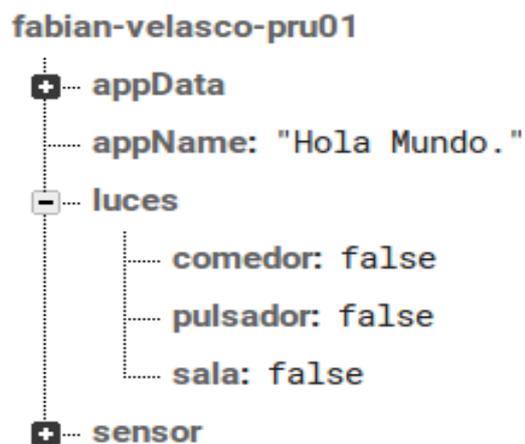


Figura. 14 Modelamiento de la Base de Datos FireBase.

3.10 MODELAMIENTO ELECTRÓNICO.

El Modelo Electrónico, es la representación gráfica de los componentes de instalación y conexión de los dispositivos electrónicos/eléctricos que se han utilizado para el desarrollo de este proyecto. En la figura 15, se puede apreciar las

salidas GPIO de la máquina Raspberry Pi conectadas a los relés que accionan las luces.

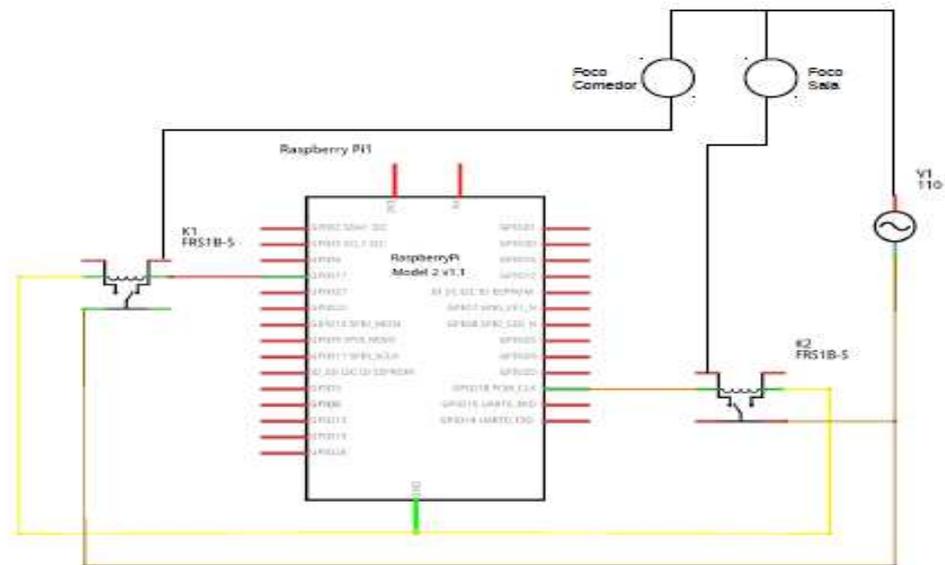


Figura. 15 Modelamiento Eléctrico/electrónico.

En el siguiente gráfico (Figura. 16), se puede ver el modelamiento real de la conexión eléctrica, utilizando el programa, Frizing.

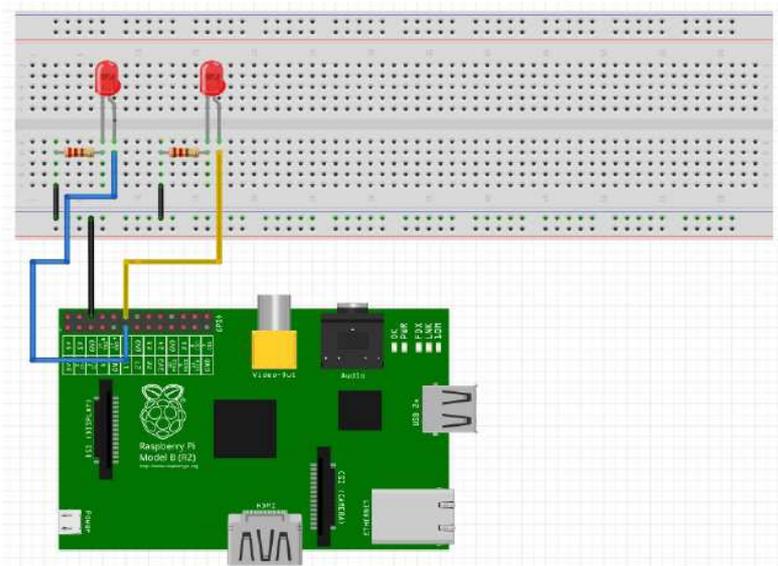


Figura. 16 Modelamiento Eléctrico/electrónico, esquema gráfico.

CAPITULO IV. IMPLEMENTACIÓN

4. CONFIGURACIÓN DEL HARDWARE.

Después de haber analizado los casos de uso y los modelamientos UWE, se concluye que la implementación debe hacerse en base a los siguientes pasos:

➤ **Paso1: Conexión y configuración de la Raspberry Pi**

Una vez, que se pueda establecer la conexión con el escritorio de la Raspberry Pi, se debería tener una imagen como la siguiente (Figura. 17):

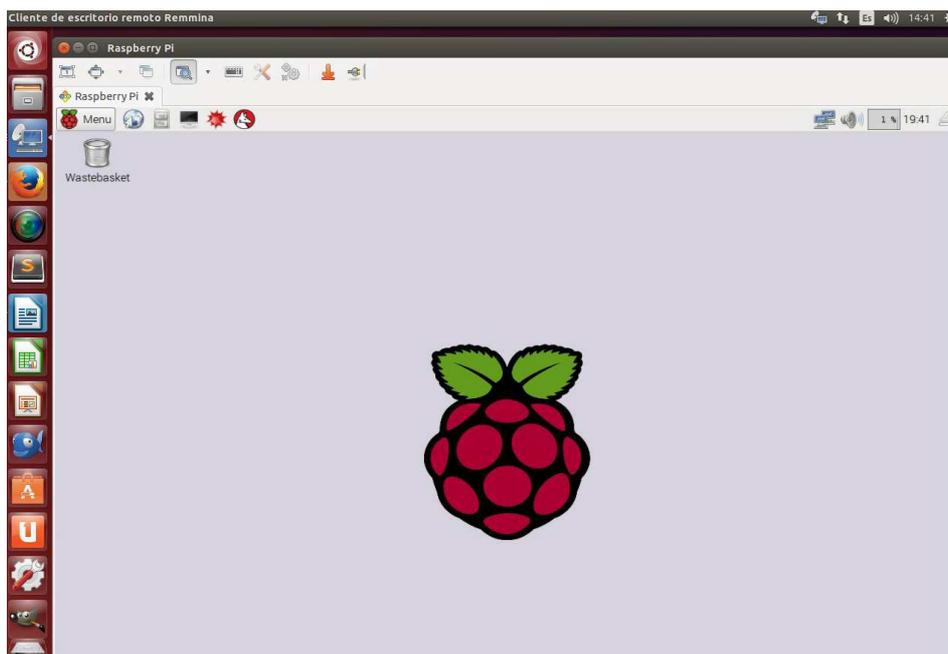


Figura. 17 Escritorio Raspberry Pi.

➤ **Paso2: Instalación de librerías necesarias para control de los pines GPIO**

Con el siguiente comando, se actualiza las librerías de la Raspberry Pi (Figura. 18).

→ `sudo apt-get update`

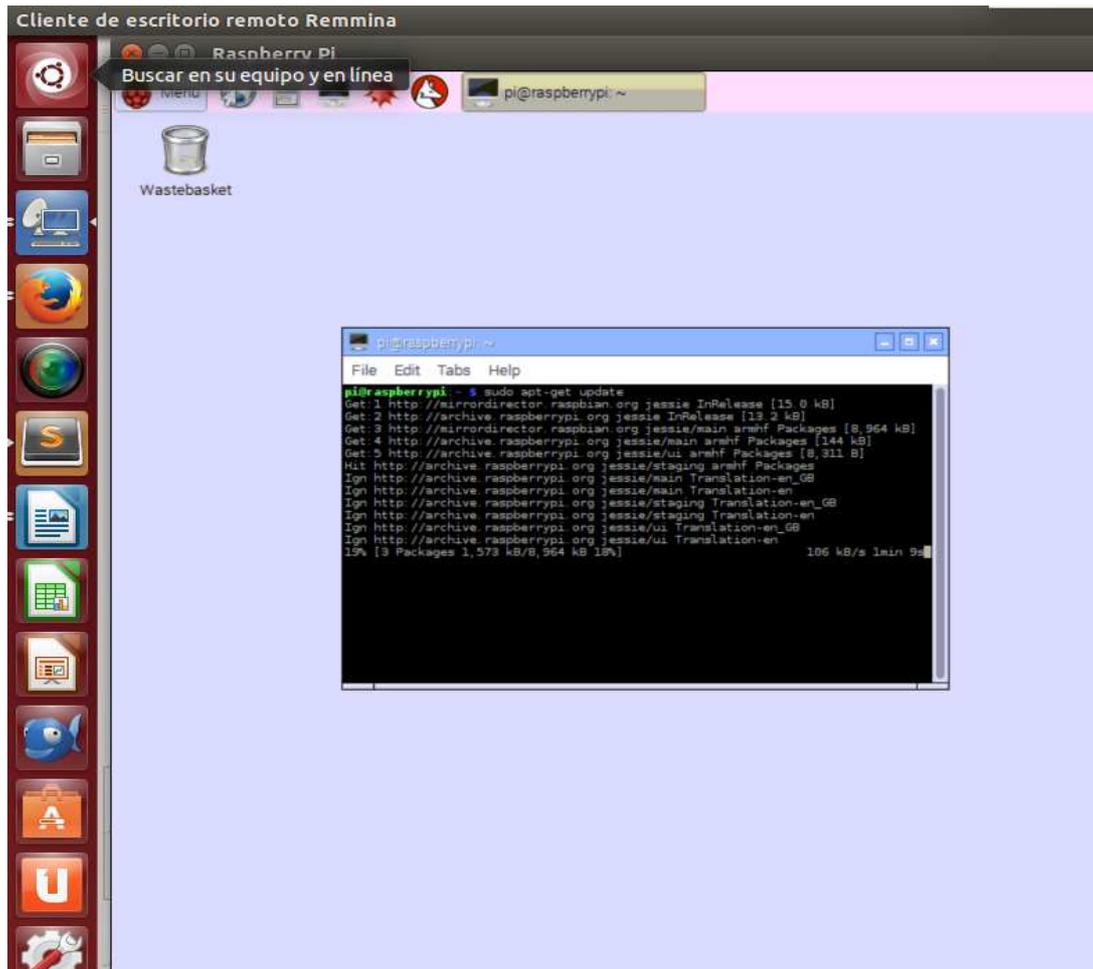


Figura. 18 Instalación de librerías necesarias para control de los pines GPIO.

Con el siguiente comando, se instala el repositorio python-dev (Figura. 19).

→ `sudo apt-get install python-dev`

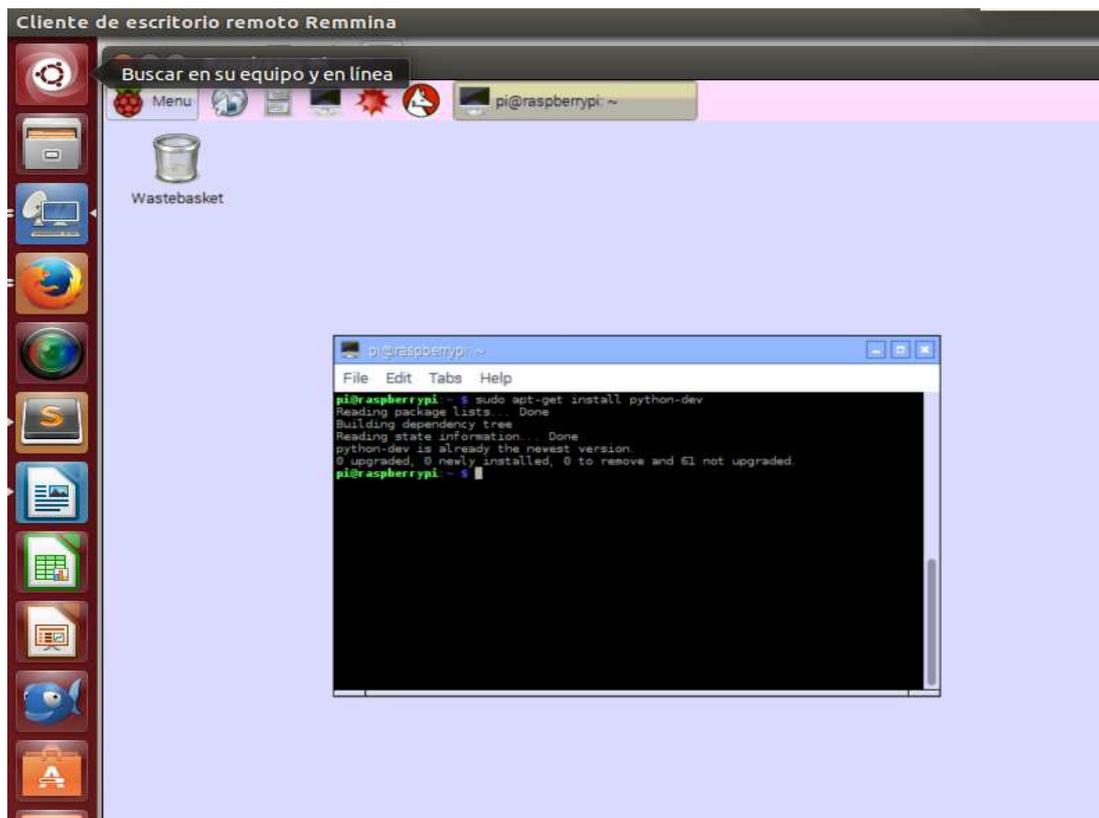


Figura. 19 Instalación de librerías necesarias para control de los pines GPIO.

Al igual que en las imágenes anteriores, se debe seguir instalando las siguientes librerías:

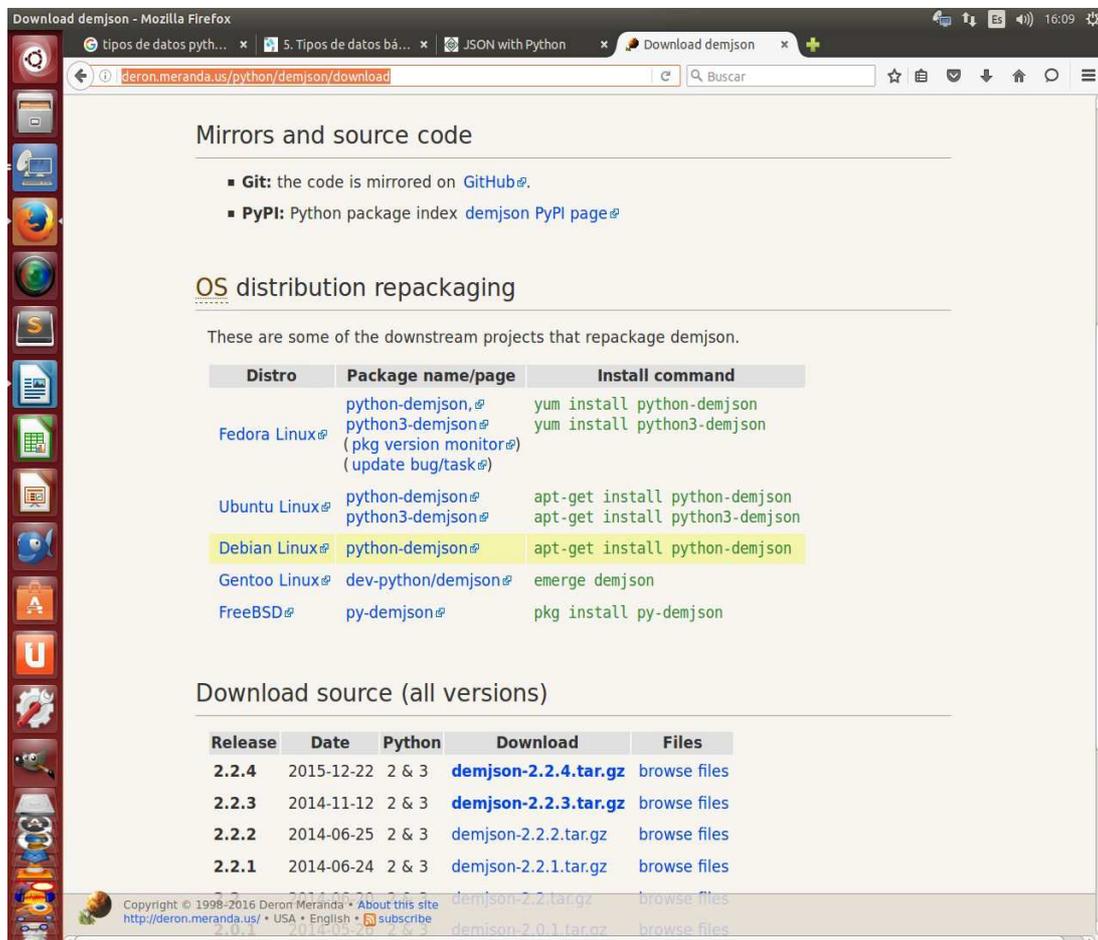
- sudo apt-get install git
- sudo apt-get install python-gpiozero
- sudo apt-get install python-pip
- sudo pip install requests==1.1.0
- sudo pip install python-firebase

Instalar demjson:

Ejecutándose como root el comando:

→ apt-get install python-demjson

Esta librería permite la transformación en tipo de datos json (Figura. 20).



The screenshot shows the website for downloading demjson. It includes sections for mirrors and source code, OS distribution repackaging, and a download source table.

Mirrors and source code

- **Git:** the code is mirrored on [GitHub](#).
- **PyPI:** Python package index [demjson PyPI page](#)

OS distribution repackaging

These are some of the downstream projects that repackage demjson.

Distro	Package name/page	Install command
Fedora Linux	python-demjson , python3-demjson (pkg version monitor) (update bug/task)	yum install python-demjson yum install python3-demjson
Ubuntu Linux	python-demjson python3-demjson	apt-get install python-demjson apt-get install python3-demjson
Debian Linux	python-demjson	apt-get install python-demjson
Gentoo Linux	dev-python/demjson	emerge demjson
FreeBSD	py-demjson	pkg install py-demjson

Download source (all versions)

Release	Date	Python	Download	Files
2.2.4	2015-12-22	2 & 3	demjson-2.2.4.tar.gz	browse files
2.2.3	2014-11-12	2 & 3	demjson-2.2.3.tar.gz	browse files
2.2.2	2014-06-25	2 & 3	demjson-2.2.2.tar.gz	browse files
2.2.1	2014-06-24	2 & 3	demjson-2.2.1.tar.gz	browse files

Figura. 20 Instalar demjson.

➤ Paso3: Instalación de servidor apache

Con los siguientes comandos, se instala el servidor Apache (Figura. 21).

→ sudo apt-get install apache2 php5 libapache2-mod-php5

→ sudo service apache2 restart

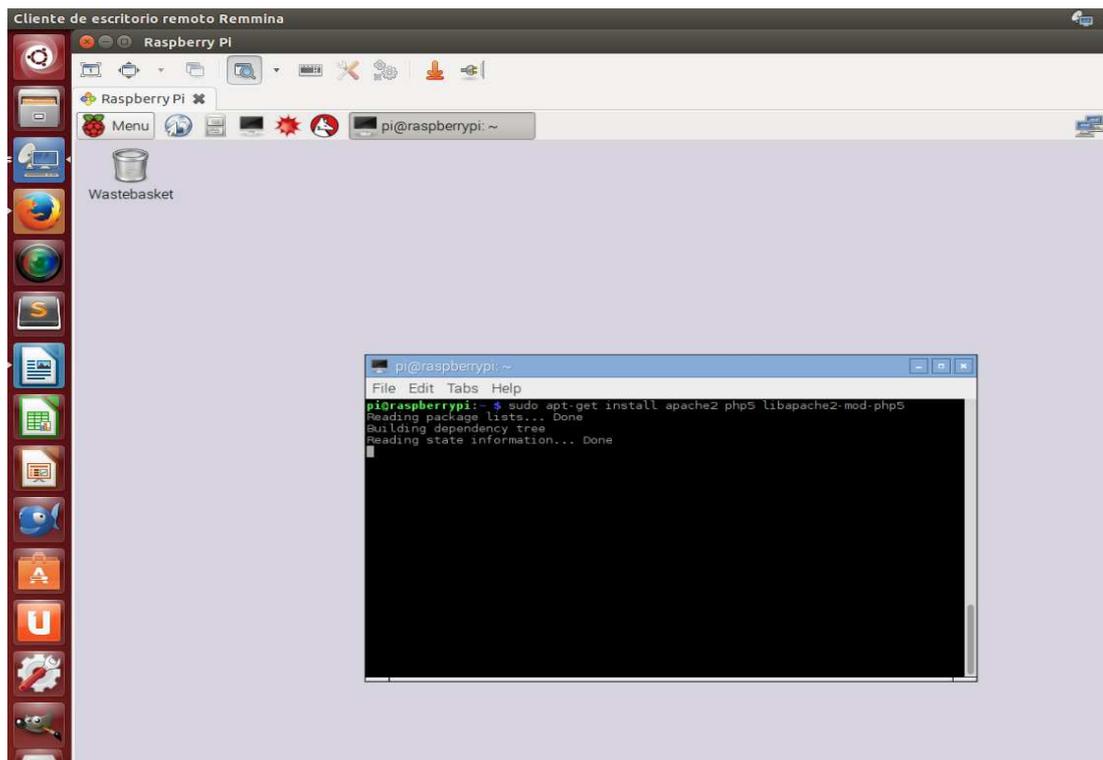


Figura. 21 Instalación de servidor apache.

➤ **Paso4: Instalar la versión ligera de Firefox para Raspberry Pi llamada “iceweasel”**

Se procede a instalar la versión ligera (Figura. 22), de Firefox para Raspberry Pi llamada “iceweasel” con el siguiente comando en la terminal:

→ `sudo apt-get install iceweasel`

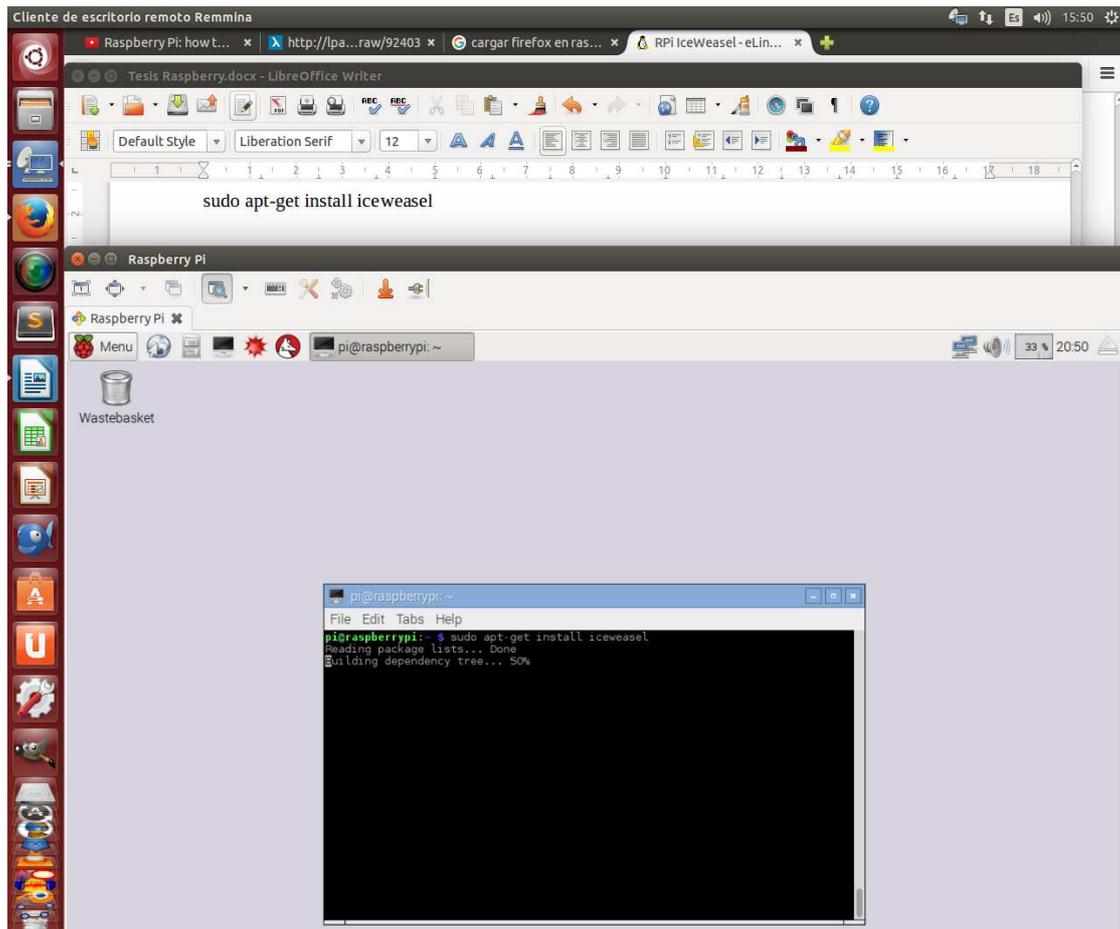


Figura. 22 versión ligera de Firefox para Raspberry Pi llamada “iceweasel”.

Una vez instalado se puede verificar el funcionamiento del servidor Apache conectándose al localhost (Figura. 23):

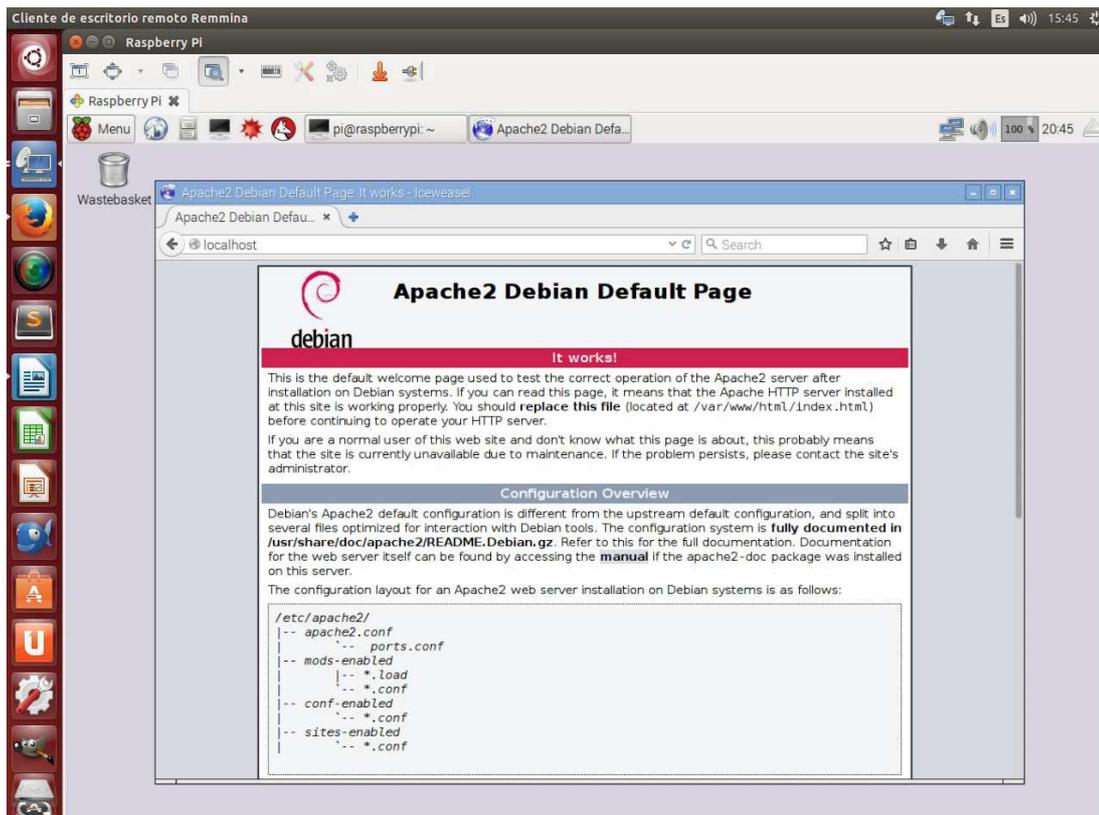


Figura. 23 Servidor Apache para Raspberry PI.

➤ Paso5: Redirección de Archivos

Posteriormente es necesario cambiar el usuario y el grupo al directorio con el comando:

→ Sudo chown www-data: www-data /var/www/html

La ubicación de los archivos en los cuales se realizará a realizar las modificaciones son:

`/var/www/html`

Por lo que se deben dar permisos para que se permitan copiar, crear y editar archivos o carpetas en esta dirección con el comando (Figura. 25):

→ `sudo chmod 775 /var/www/html`

Como el usuario por defecto de la Raspberry Pi es pi, se añadirá éste, al grupo **www-data** que se crea al instalar el servidor Apache, con el comando:

→ `sudo usermod -a -G www-data pi`

El comando se ejecutará con:

→ `sudo visudo`

Se añade al final del archivo:

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

```

file Edit Tabs Help
GNU nano 2.2.6 File: /etc/sudoers.tmp Modified

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

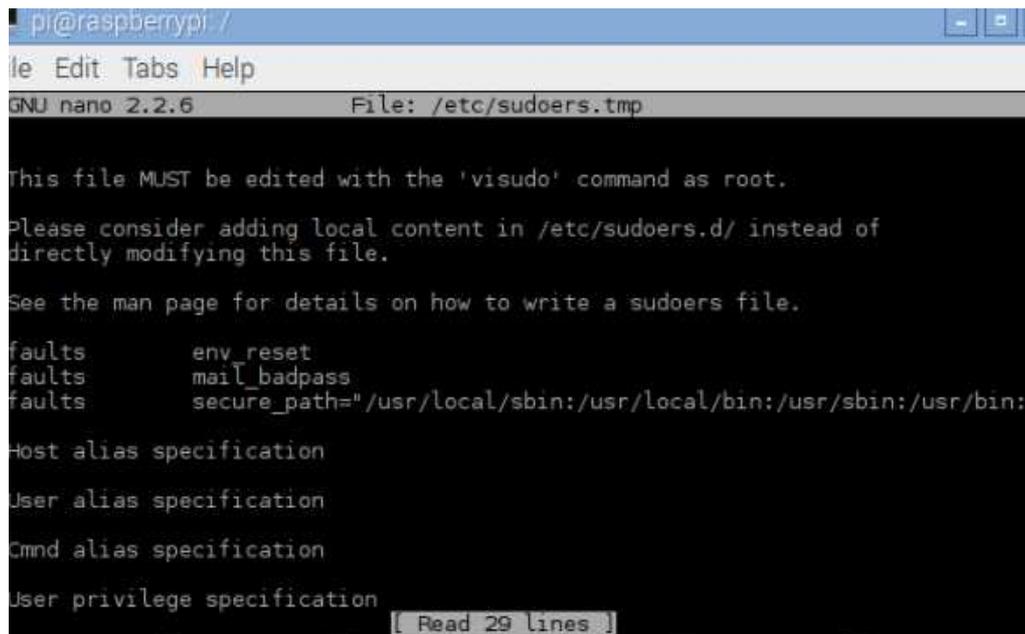
ALL=(ALL) NOPASSWD: ALL

www-data ALL=(root) NOPASSWD: ALL

```

Figura. 24 Modificación de Archivos.

Luego de escribir la instrucción el archivo queda de la siguiente forma (Figura. 24):



```
pi@raspberrypi: /
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/sudoers.tmp

This file MUST be edited with the 'visudo' command as root.

Please consider adding local content in /etc/sudoers.d/ instead of
directly modifying this file.

See the man page for details on how to write a sudoers file.

faults      env_reset
faults      mail_badpass
faults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:

Host alias specification

User alias specification

Cmd alias specification

User privilege specification
[ Read 29 lines ]
```

Figura. 25 Modificación de Archivos.

➤ Paso6: Cuenta en Firebase

La obtención de una cuenta en Firebase es muy simple y gratuita en el modo de desarrollo y lo que se interesa es la URL que le sea asignada a la aplicación (Figura. 26).

En el caso de este proyecto se me ha asignado la dirección:

'https://fabian-velasco-pru01.firebaseio.com' que va a ser incluida en el script de control posteriormente.

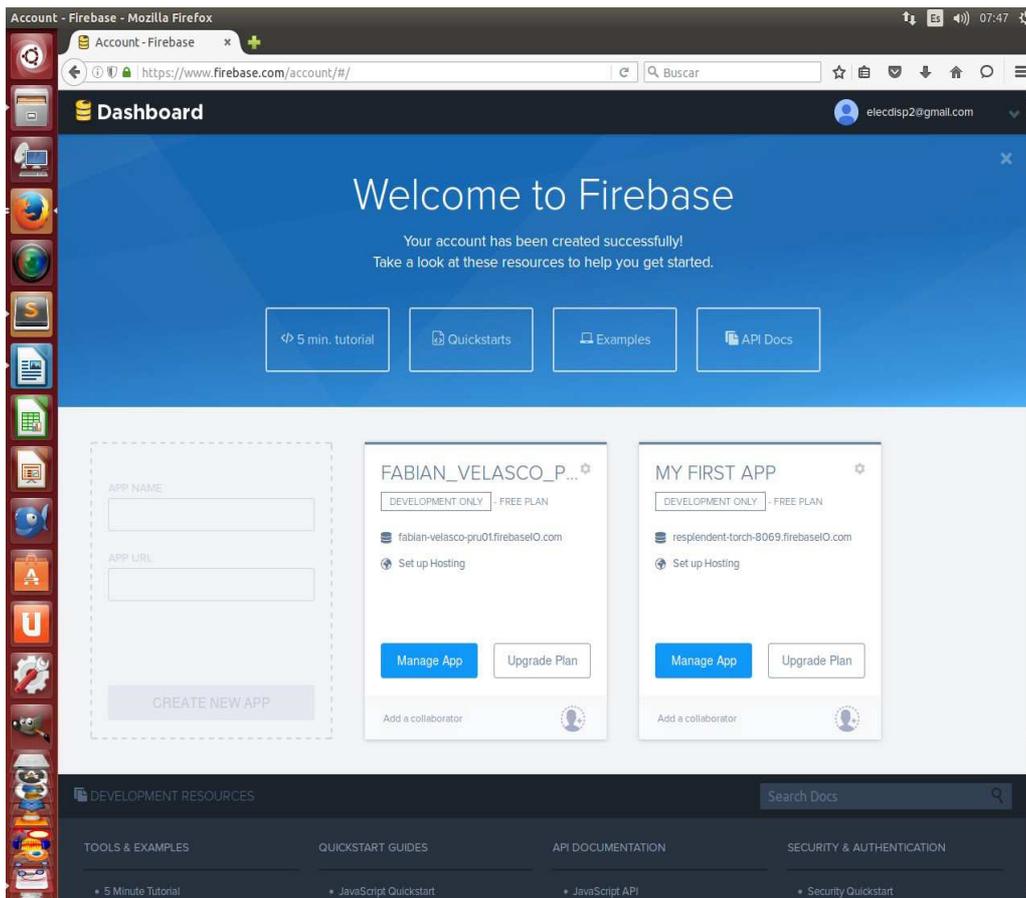
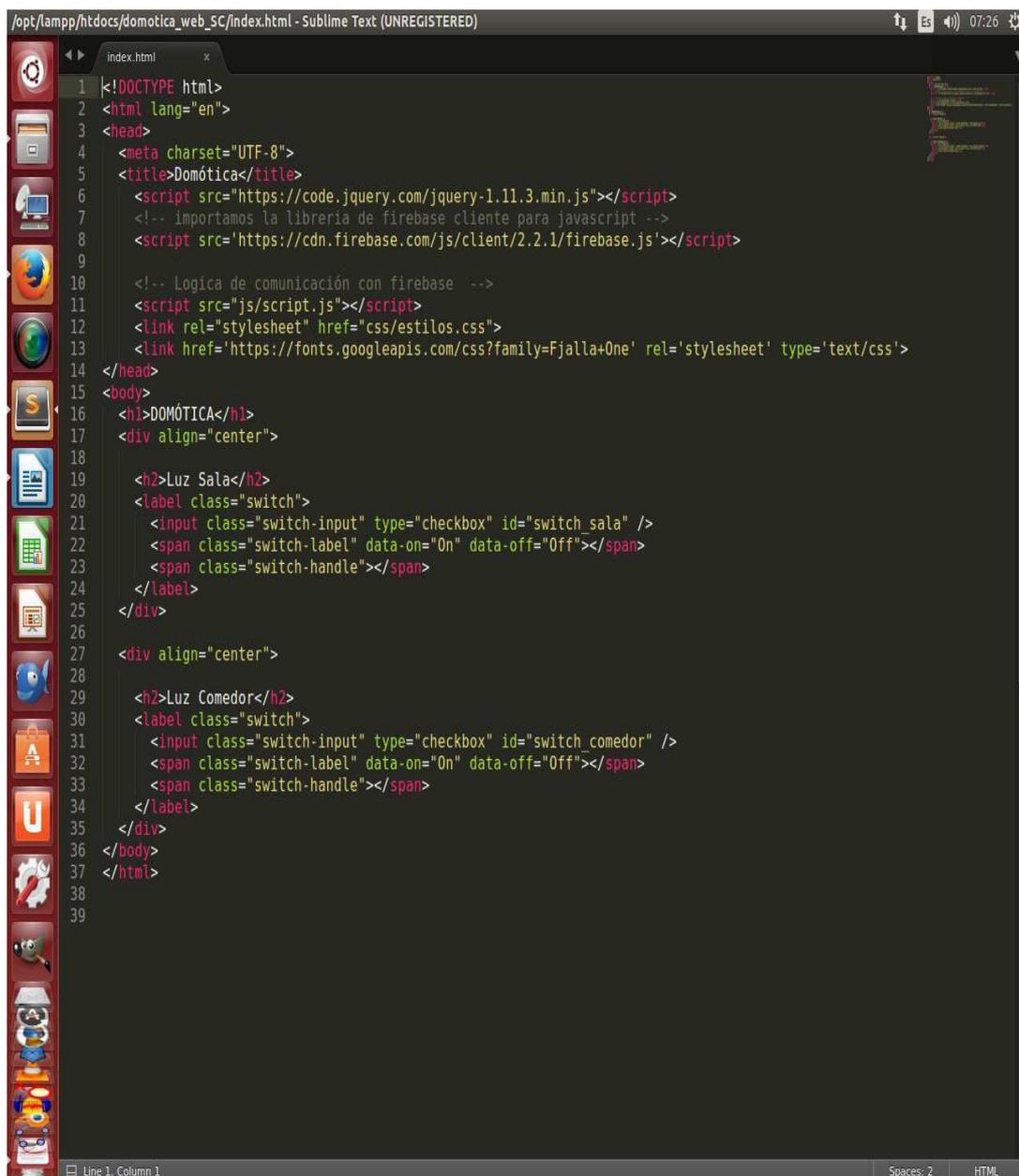


Figura. 26 Cuenta en Firebase.

4.2 IMPLEMENTACIÓN DEL APLICATIVO WEB.

Con la aplicación Sublime Text, se ha desarrollado el archivo index.html, que es el archivo de inicio de la aplicación Web del proyecto, con las siguientes características (Figura. 27):



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Domótica</title>
6 <script src="https://code.jquery.com/jquery-1.11.3.min.js"></script>
7 <!-- importamos la libreria de firebase cliente para javascript -->
8 <script src='https://cdn.firebase.com/js/client/2.2.1/firebase.js'></script>
9
10 <!-- Logica de comunicación con firebase -->
11 <script src="js/script.js"></script>
12 <link rel="stylesheet" href="css/estilos.css">
13 <link href='https://fonts.googleapis.com/css?family=Fjalla+One' rel='stylesheet' type='text/css'>
14 </head>
15 <body>
16 <h1>DOMÓTICA</h1>
17 <div align="center">
18
19 <h2>Luz Sala</h2>
20 <label class="switch">
21 <input class="switch-input" type="checkbox" id="switch_sala" />
22 <span class="switch-label" data-on="On" data-off="Off"></span>
23 <span class="switch-handle"></span>
24 </label>
25 </div>
26
27 <div align="center">
28
29 <h2>Luz Comedor</h2>
30 <label class="switch">
31 <input class="switch-input" type="checkbox" id="switch_comedor" />
32 <span class="switch-label" data-on="On" data-off="Off"></span>
33 <span class="switch-handle"></span>
34 </label>
35 </div>
36 </body>
37 </html>
38
39
```

Figura. 2724 Aplicativo Web.

El código anterior genera la siguiente interfaz Web (Figura. 27):



Figura. 28 Aplicativo Web.

Se procede a probar el aplicativo para dos usuarios conectados al mismo tiempo.

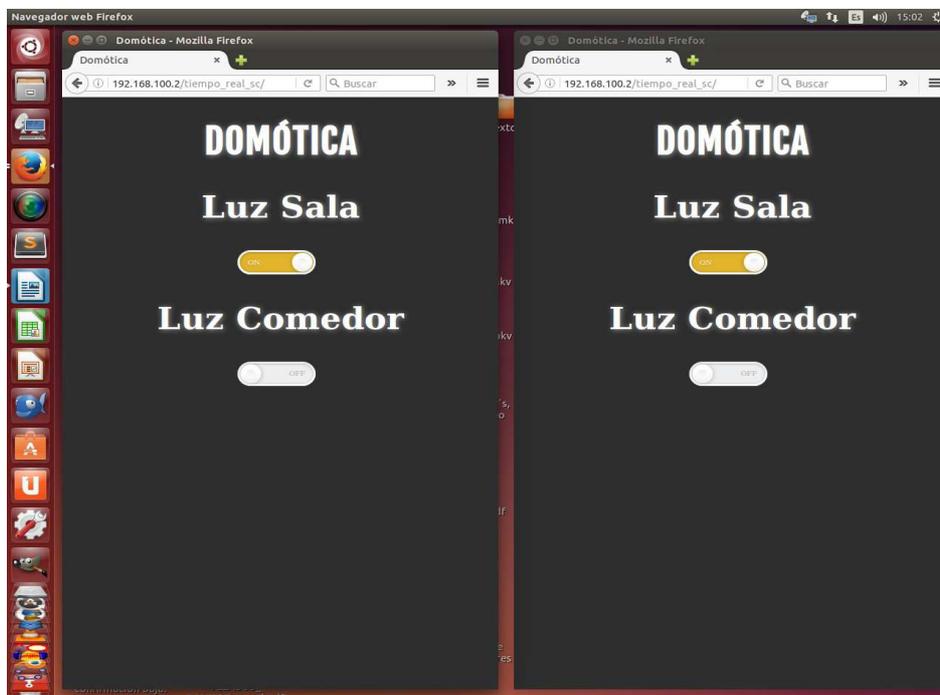


Figura. 29 Dos usuarios conectados al mismo tiempo al Aplicativo Web.

4.3 REQUISITOS PARA LA IMPLEMENTACIÓN

En el desarrollo del aplicativo se instaló y utilizó en las máquinas de los desarrolladores las herramientas que a continuación se especifican:

- Sistema Operativo: Windows 7.

En la máquina Raspberry Pi, se instaló el siguiente software:

- Sistema Operativo RASPBIAN JESSIE WITH PIXEL Image with PIXEL desktop based on Debian Jessie, Kernel version: 4.4.
- Aplicación de Servidores Apache 2.0 Handler.
- versión ligera de Firefox para Raspberry Pi “iceweasel”
- Raspberry Pi® (Broadcom BCM2835 SoC/ Kernel Linux):
 - Procesador ARM1176JZF-S de 700 MHz.
 - VideoCore IV GPU y 512 Megabytes de RAM.
 - Tarjeta SD (“Disco duro” de la Raspberry Pi).

CAPÍTULO V. PRUEBAS DEL APLICATIVO EN HARDWARE/SOFTWARE

5. PRUEBAS DE LA APLICACIÓN

En este capítulo, se analizará el rendimiento del sistema, tanto en hardware como en software, utilizando herramientas de software que permitan observar y medir el rendimiento del proyecto, de tal forma que llegue a determinarse la calidad del producto. Existen niveles de pruebas tales como:

- Pruebas de desarrollador,
- Pruebas funcionales y
- Pruebas de usuario final.

Sin embargo, hay que tener en cuenta que no todos las pruebas de software mencionadas sirven para todos los aplicativos, de allí que se utilizarán las pruebas que se considere necesarias según los requerimientos especificados.

Para análisis de calidad del presente proyecto, se tomarán en cuenta las siguientes pruebas:

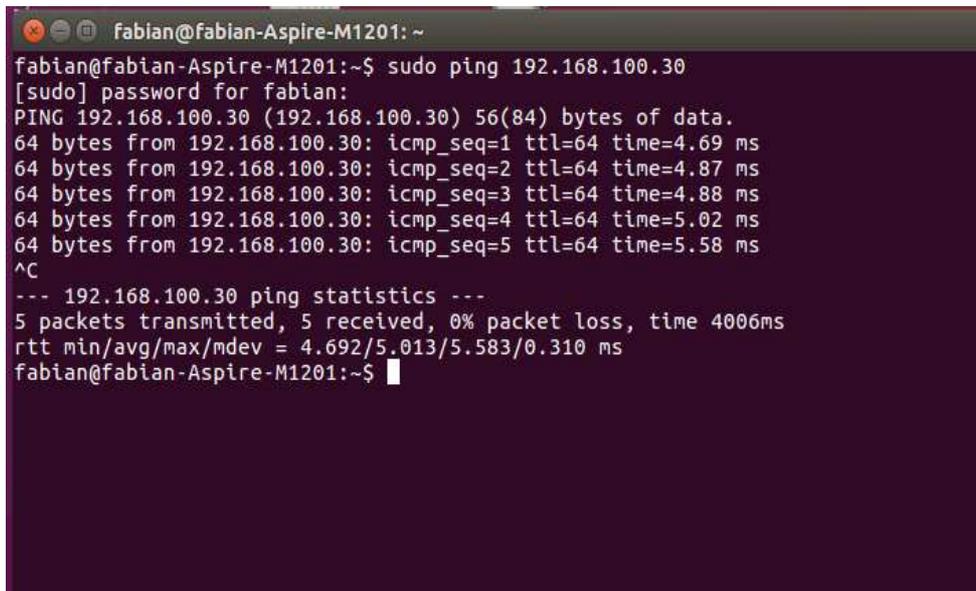
- Pruebas de conectividad,
- Pruebas de concurrencia,
- Pruebas de compatibilidad,
- Pruebas de rendimiento y
- Pruebas de caja negra.
- Para el servidor web se realizarán:
 - Pruebas de conectividad,
 - Pruebas de funcionamiento,

- Pruebas de carga y
- Pruebas de concurrencia de acceso al aplicativo Web (localhost e internet.).

5.1 PRUEBAS DE CONECTIVIDAD

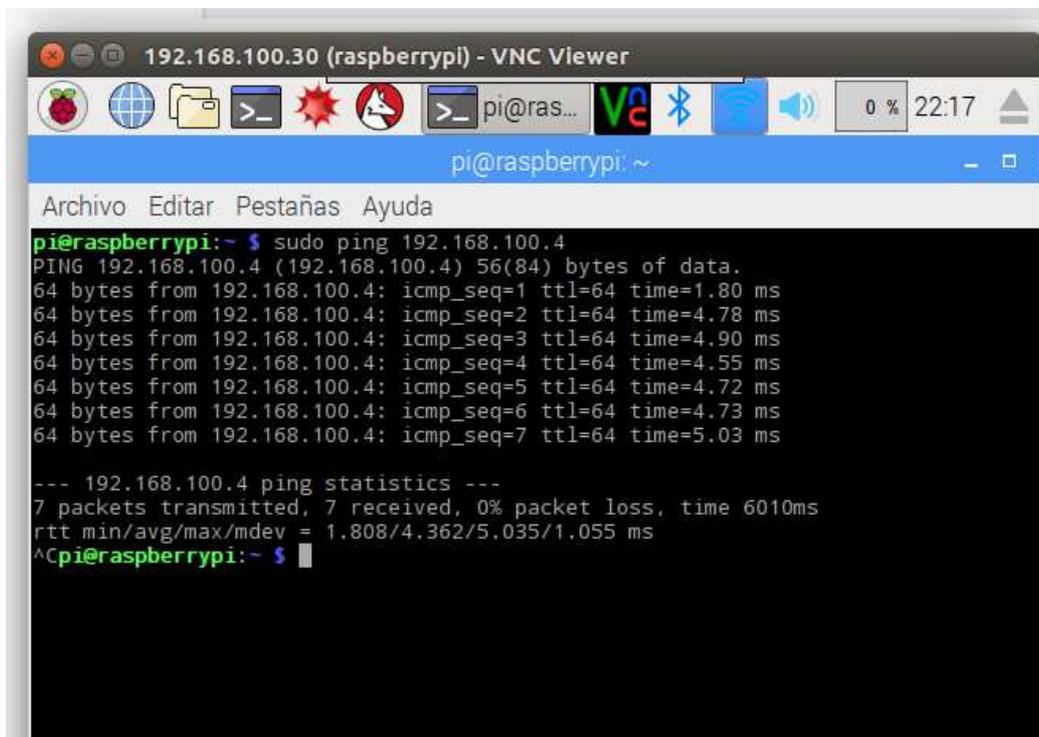
Para las pruebas de conectividad se utilizará el comando “ping” desde una máquina de Windows, éste mostrará la conexión de la máquina (host local) en la misma red, lo que en realidad hace este comando, es mostrar los paquetes de datos enviados y devueltos en un determinado tiempo.

Como se puede ver en la Fig. 30 y Fig. 31, la conexión resulta exitosa.

A terminal window with a dark purple background. The prompt is 'fabian@fabian-Aspire-M1201: ~'. The user enters 'sudo ping 192.168.100.30'. The terminal shows the password prompt, then the output of the ping command: 'PING 192.168.100.30 (192.168.100.30) 56(84) bytes of data.' followed by five lines of ping results showing 64 bytes from 192.168.100.30 with increasing sequence numbers and times. The user presses Ctrl-C (^C). The terminal shows '--- 192.168.100.30 ping statistics ---' and summary statistics: '5 packets transmitted, 5 received, 0% packet loss, time 4006ms rtt min/avg/max/mdev = 4.692/5.013/5.583/0.310 ms'. The prompt returns to 'fabian@fabian-Aspire-M1201:~\$'.

```
fabian@fabian-Aspire-M1201: ~
fabian@fabian-Aspire-M1201:~$ sudo ping 192.168.100.30
[sudo] password for fabian:
PING 192.168.100.30 (192.168.100.30) 56(84) bytes of data.
64 bytes from 192.168.100.30: icmp_seq=1 ttl=64 time=4.69 ms
64 bytes from 192.168.100.30: icmp_seq=2 ttl=64 time=4.87 ms
64 bytes from 192.168.100.30: icmp_seq=3 ttl=64 time=4.88 ms
64 bytes from 192.168.100.30: icmp_seq=4 ttl=64 time=5.02 ms
64 bytes from 192.168.100.30: icmp_seq=5 ttl=64 time=5.58 ms
^C
--- 192.168.100.30 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 4.692/5.013/5.583/0.310 ms
fabian@fabian-Aspire-M1201:~$
```

Figura. 30 Ping de Ubuntu a Raspberry.



```
192.168.100.30 (raspberrypi) - VNC Viewer
pi@rasberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ sudo ping 192.168.100.4
PING 192.168.100.4 (192.168.100.4) 56(84) bytes of data:
64 bytes from 192.168.100.4: icmp_seq=1 ttl=64 time=1.80 ms
64 bytes from 192.168.100.4: icmp_seq=2 ttl=64 time=4.78 ms
64 bytes from 192.168.100.4: icmp_seq=3 ttl=64 time=4.90 ms
64 bytes from 192.168.100.4: icmp_seq=4 ttl=64 time=4.55 ms
64 bytes from 192.168.100.4: icmp_seq=5 ttl=64 time=4.72 ms
64 bytes from 192.168.100.4: icmp_seq=6 ttl=64 time=4.73 ms
64 bytes from 192.168.100.4: icmp_seq=7 ttl=64 time=5.03 ms

--- 192.168.100.4 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 1.808/4.362/5.035/1.055 ms
^Cpi@raspberrypi:~$
```

Figura. 31 Ping desde Raspberry a PC Ubuntu.

Para verificar la IP de la máquina Raspberry Pi se utilizará el comando “ifconfig” puesto que hemos utilizado una máquina con un Sistema Operativo de Distribución Linux (Fig. 32), Por otro lado, para encontrar la dirección IP de la Raspberry Pi (Fig. 33), se utilizará el comando:

→ sudo nmap -sP 192.168.100.0/24

```

fabian@fabian-Aspire-M1201: ~
fabian@fabian-Aspire-M1201:~$ ifconfig
eth0      Link encap:Ethernet  direcciónHW 00:1f:e2:58:e3:77
          Direc. inet:192.168.100.2  Difus.:192.168.100.255  Másc:255.255.255
          Dirección inet6: fe80::21f:e2ff:fe58:e377/64  Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:75 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:86 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:11439 (11.4 KB)  TX bytes:13558 (13.5 KB)
          Interrupción:17

lo        Link encap:Bucle local
          Direc. inet:127.0.0.1  Másc:255.0.0.0
          Dirección inet6: ::1/128  Alcance:Anfitrión
          ACTIVO BUCLE FUNCIONANDO  MTU:65536  Métrica:1
          Paquetes RX:267 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:267 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:0
          Bytes RX:20091 (20.0 KB)  TX bytes:20091 (20.0 KB)

fabian@fabian-Aspire-M1201:~$ █

```

Figura. 32 Pruebas de conectividad.

```

fabian@fabian-Aspire-M1201: ~
fabian@fabian-Aspire-M1201:~$ sudo nmap -sP 192.168.100.0/24

Starting Nmap 6.40 ( http://nmap.org ) at 2016-03-20 11:04 ECT
Nmap scan report for 192.168.100.1
Host is up (0.0025s latency).
MAC Address: 7C:A2:3E:36:15:F8 (Unknown)
Nmap scan report for 192.168.100.3
Host is up (0.041s latency).
MAC Address: 2C:29:97:02:18:F0 (Unknown)
Nmap scan report for 192.168.100.4
Host is up (0.00086s latency).
MAC Address: B8:27:EB:DD:E3:70 (Raspberry Pi Foundation)
Nmap scan report for 192.168.100.2
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.09 seconds
fabian@fabian-Aspire-M1201:~$ █

```

Figura. 33 Pruebas de conectividad.

Búsqueda de la dirección IP, de la máquina Raspberry Pi, puesto que la IP de la máquina Raspberry Pi es dinámica y se necesita identificar su dirección y conectividad desde teléfonos inteligentes, para ello se ha utilizado las siguientes herramientas:

Resultados:

1. La máquina Raspberry Pi, se conecta de forma inmediata a la red en una media de 6010ms, sin presentar problemas de inestabilidad, el momento de su conexión.
2. Como se puede apreciar en las Figura. 34 y Figura. 35, una vez que la máquina Raspberry Pi ha establecido conexión con la red local, cualquier dispositivo inteligente puede detectar su dirección IP, siendo exitosa la conexión que ha establecido la máquina Raspberry Pi con la red.



Figura. 34 Usando Fing Aplicación Iphone



Figura. 35 Usando Network Scanner Samsung.

5.2 PRUEBAS DE CONCURRENCIA

Las pruebas de concurrencia se basaron en el ingreso simultáneo, al aplicativo web, desde varios navegadores. Al mismo tiempo, se procedió a activar los botones ON/OFF de la interfaz, para comprobar la actualización del aplicativo desde la base de datos, con diferentes usuarios y en diferentes navegadores.

Resultados:

1. La máquina Raspberry pi y todos los periféricos inteligentes (Smartphone, Tablet`s, notebook`s, etc.), pueden acceder de forma ágil e inmediata a los navegadores web más comerciales (Tabla 7).
2. Cuando uno o más usuarios acceden al mismo tiempo la conexión se mantiene estable. Téngase en cuenta, que la velocidad de la conexión dependerá del operador y del servicio contratado.

Tabla 6

Pruebas de concurrencia.

Pruebas de concurrencia				
CIUDAD	BROWSER	FECHA INGRESO	FECHA SALIDA	RESULTADO
Quito	Mozila Firefox	05 de Enero 2017 16:06	05 de Enero 2017 17:11	Navegación correcta
Quito	Safari iOS Mac	05 de Enero 2017 18:22	05 de Enero 2017 18:49	Navegación correcta
Quito	Google Chrome	05 de Enero 2017 19:34	05 de Enero 2017 20:06	Navegación correcta
Quito	Internet Explorer 9	05 de Enero 2017 22:09	05 de Enero 2017 22:53	Navegación correcta

5.3 PRUEBAS DE COMPATIBILIDAD

Para la realización de las pruebas de compatibilidad se utilizaron los browser más comerciales, verificando la compatibilidad y funcionalidad del aplicativo web.

Resultados:

Como se puede apreciar en la Tabla 8, el aplicativo web responde perfectamente a los navegadores comerciales, sin presentar inconvenientes de incompatibilidad, en ninguno de los cinco casos estudiados.

Tabla 7

Pruebas de compatibilidad.

PRUEBAS DE COMPATIBILIDAD			
BROWSER	NAVEGACIÓN	COMPATIBILIDAD	PLUGIN
Internet Explorer 9	OK	OK	OK
Google Chrome	OK	OK	OK
Safari iOS Mac	OK	OK	OK
Opera 29.0	OK	OK	OK
Mozilla Firefox	OK	OK	OK

5.4 PRUEBAS DE RENDIMIENTO

La herramienta utilizada, para esta prueba es “Apache Benchmark”, que permite medir el rendimiento óptimo del aplicativo y de la máquina Raspberri Py, ya que funciona como un servidor web (Figura. 36).

```

feroz@feroz-Satellite-L515: ~
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.100.30 (be patient).....done

Server Software:      Apache/2.4.10
Server Hostname:     192.168.100.30
Server Port:         80

Document Path:       /tiempo_real_sc/
Document Length:     1180 bytes

Concurrency Level:   20
Time taken for tests: 0.266 seconds
Complete requests:   100
Failed requests:     0
Total transferred:   145400 bytes
HTML transferred:    118000 bytes
Requests per second: 376.40 [#/sec] (mean)
Time per request:    53.135 [ms] (mean)
Time per request:    2.657 [ms] (mean, across all concurrent requests)
Transfer rate:       534.46 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:  10   13  2.5   13   20
Processing: 13   36  7.5   36   55
Waiting:   12   36  7.6   36   55
Total:     23   49  8.5   50   71

Percentage of the requests served within a certain time (ms)
 50%    50
 66%    51
 75%    52
 80%    53
 90%    60
 95%    64
 98%    69
 99%    71
100%    71 (longest request)
feroz@feroz-Satellite-L515:~$

```

Figura. 36 Pruebas de rendimiento.

Como podemos apreciar se obtiene una velocidad aceptable de carga del Portal Web como se muestra:

Resultados:

1. Como se puede ver en la Figura. 36, respuesta de rendimiento es del 100%, teniendo un tiempo promedio de respuesta de 53.135ms y un ratio de transferencia de 534,46 kbytes/sec. Es decir, el rendimiento del aplicativo web es óptimo.

5.5 PRUEBAS DE CAJA NEGRA

Las pruebas de Caja Negra permiten determinar las condiciones de entrada/salida de los requisitos funcionales del aplicativo web. Mediante éstas pruebas se puede identificar errores de:

- Funciones incorrectas o ausentes del aplicativo web.
- Errores que pudiere tener la interfaz.
- Errores que pudiesen existir en las estructuras de datos, o en accesos que se realiza a las Bases de Datos externas, en este caso a la FireBase.
- Posibles errores de rendimiento en el aplicativo web.
- E incluso, errores de inicialización y terminación del aplicativo web.

Para las pruebas de caja negra se ha utilizado luces Led, en el hardware, para poder detectar los errores y asumir su corrección.

Primera Prueba:

- Luz de la sala encendida y luz comedor apagada.
- Led derecho encendido y led de izquierda apagado.

Tabla 8

Pruebas de Caja Negra.

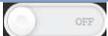
PRUEBAS DE CAJA NEGRA				
	INTERFAZ	ESTADO INTERRUPTOR	FireBase	RESULTADO
Luz Sala		ON	True	EXITOSO
Luz Comedor		OFF	false	EXITOSO



Figura. 37 Pruebas de Caja Negra.

Segunda Prueba:

- Luz del comedor encendido y luz sala apagada.
- Led derecho apagado y led de izquierda encendido.

Tabla 9

Pruebas de Caja Negra.

PRUEBAS DE CAJA NEGRA				
	INTERFAZ	ESTADO INTERRUPTOR	FireBase	RESULTADO
Luz Sala		OFF	false	EXITOSO
Luz Comedor		ON	True	EXITOSO



Figura. 38 Pruebas de Caja Negra.

Tercera Prueba:

- Luz de la sala y comedor encendidas al mismo tiempo.
- Led derecho y led de izquierda encendidos.

Tabla 10

Pruebas de Caja Negra.

PRUEBAS DE CAJA NEGRA				
	INTERFAZ	ESTADO INTERRUPTOR	FireBase	RESULTADO
Luz Sala		ON	True	EXITOSO
Luz Comedor		ON	True	EXITOSO

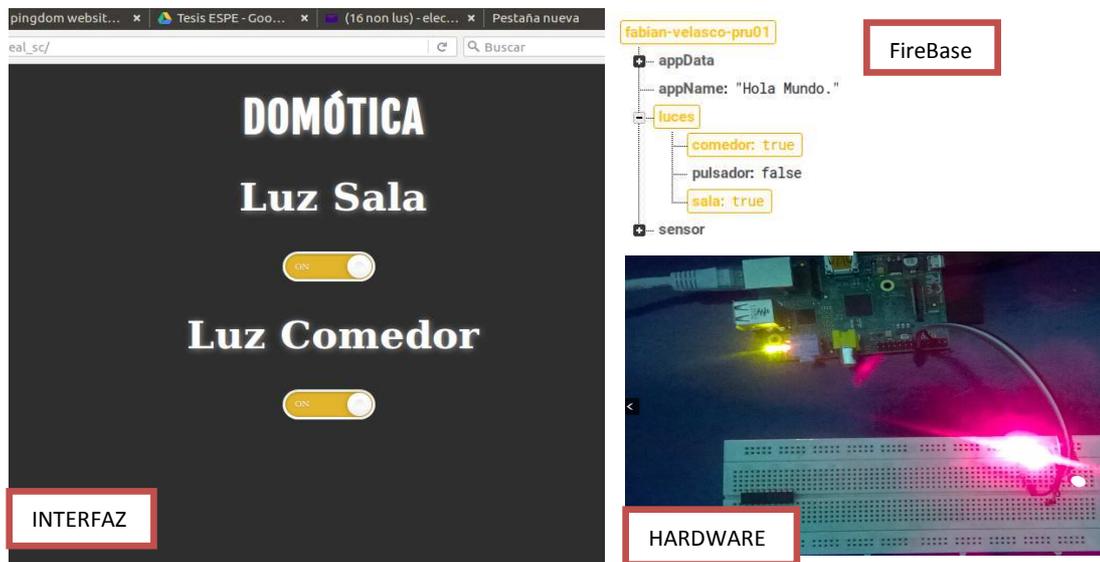


Figura. 39 Pruebas de Caja Negra.

Cuarta Prueba:

- Luz de la sala y comedor apagadas al mismo tiempo.
- Led derecho y led de izquierda apagados.

Tabla 11

Pruebas de Caja Negra.

PRUEBAS DE CAJA NEGRA				
	INTERFAZ	ESTADO INTERRUPTOR	FireBase	RESULTADO
Luz Sala		OFF	False	EXITOSO
Luz Comedor		OFF	False	EXITOSO

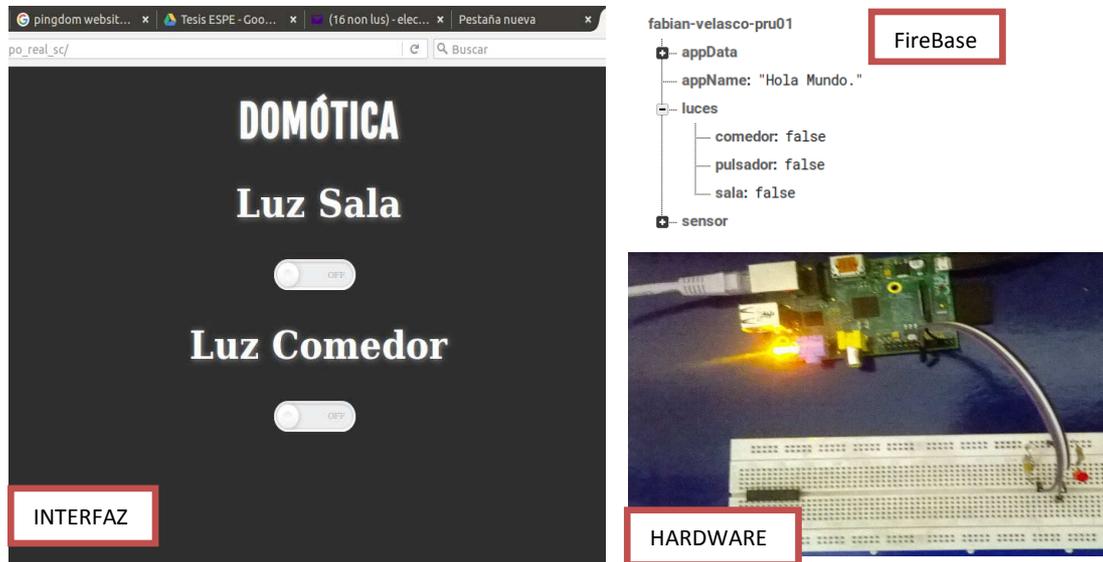


Figura. 40 Pruebas de Caja Negra.

CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES

A lo largo del desarrollo del proyecto, y del análisis de los capítulos anteriores, se ha podido llegar a las siguientes conclusiones y recomendaciones:

6.1. CONCLUSIONES

- La utilización de la metodología ágil UWE (basada en UML), ha resultado ser eficaz y eficiente para el del diseño e implementación de un control domótico utilizando Raspberry Pi y una base de datos NoSQL, puesto que ha permitido conceptualizar de forma visual los requerimientos del proyecto.
- Esta metodología, promueve en cada una de sus fases, que el desarrollador abarque, en su gran mayoría, los requerimientos del usuario y en contra posición, permite que el usuario se adapte, de forma fácil, al uso y funcionamiento del aplicativo.
- Los usuarios, pueden acceder en cualquier momento, en cualquier lugar y desde cualquier periférico inteligente al aplicativo web, y tener el control del sistema eléctrico de la vivienda, sin necesidad de trasladarse a los interruptores/pulsadores físicos de la vivienda.
- El uso de la FireBase, permite actualizar la información en tiempo real, de tal forma que un número ilimitado de usuarios puedan acceder al aplicativo web en el mismo instante, de tal forma que todos los periféricos que tengan el aplicativo muestren en su interfaz el estado real de las luces de la vivienda, siendo la FireBase, actualmente, la mejor opción para el desarrollo de este tipo de aplicativos.

- El desarrollo de la interfaz, ha resultado amigable para el usuario. De forma intuitiva el usuario controla el sistema sin necesidad de socializaciones o algún tipo de inducción.
- La domótica se ha ido instalando en los hogares de casi todo el mundo, y el desarrollo de este tipo de proyectos permite que los usuarios ecuatorianos, pueden acceder a este tipo de sistemas a bajo costo y con los recursos tecnológicos más avanzados.
- La utilización de Raspberry Pi como unidad de control del sistema ha facilitado enormemente el desarrollo del presente proyecto, ya que mediante el uso de unos pocos comandos Linux se pueden tener instalados los servicios requeridos, tanto para el servidor Web mediante Apache o el control físico de las luces que funciona en Python. Es importante también la posibilidad de conectarse a internet así como el control de escritorio remoto mediante una conexión VNC o bien el paso de archivos mediante SSH.
- Este proyecto es una primera aproximación al uso de Firebase en el control de procesos de tiempo real. Ha buscado, una respuesta a un error bastante común en las aplicaciones de domótica, realizadas por técnicos en electrónica, los cuales pasan por alto la actualización de los estados en tiempo real.

6.2. RECOMENDACIONES

- La metodología UWE, debe convertirse, de forma sistemática, en el proceso elegido por los desarrolladores, puesto que los requerimientos funcionales son percibidos de mejor manera y permite un desarrollo más conciso.

- El soporte del proyecto, debe hacerse por usuarios con algún tipo de conocimiento en Bases de Datos, redes y desarrollo de aplicativos web, de tal forma que tanto la máquina Raspberry Pi, como el aplicativo web puedan alcanzar su funcionalidad óptima, en el caso de alguna avería.
- Se recomienda tener siempre conectada la máquina Raspberry Pi, a internet puesto que la interfaz recibe la información actualizada de la Firebase y aunque el sistema completo puede fusionar por separado, necesariamente debe actualizar su clave-valor desde el internet.
- Como punto particular, se recomienda realizar el estudio de un proyecto, en base a esta monografía, en el que se pueda controlar los electrodomésticos del hogar, así como también la parte de servicio de agua potable de la vivienda.
- Se recomienda complementar y actualizar las ideas proporcionadas en este trabajo, ya que se dejaron de lado los controles manuales que siempre deben estar presentes en las instalaciones reales. Adicionalmente el rápido desarrollo que ha tenido esta tecnología podría dejar de lado el uso de las librerías utilizadas al reemplazarlas por otras más simples de usar.
- No se recomienda el uso de este aplicativo en sitios de alto riesgo, ya que se requiere contar siempre con alimentación eléctrica para la Raspberry Pi y los relés de control. Adicionalmente, la activación de las salidas de la Raspberry Pi se realizan de manera manual cuando el sistema se resetea, por medio de una terminal. Una versión sin control en tiempo real está disponible en el CD adjunto.
- Finalmente, para una siguiente versión de este producto, se recomendaría utilizar algún tipo de automatización, o por qué no, de inteligencia artificial, que le permita al sistema tomar decisiones por sí solo, es decir, que si el usuario

olvidase apagar/encender una luz o cualquier electrodomésticos (Con una versión adaptada), el sistema deberá ser capaz de decidir qué acción tomar, sin necesidad de la intervención física del usuario.

GLOSARIO DE TÉRMINOS.

Firestore:	Plataforma (Base de Datos) de desarrollo móvil en la nube de Google.
SDK:	(Siglas en inglés de software development kit), conjunto de herramientas de desarrollo de software.
Smartphone:	Término smartphone pertenece a la lengua inglesa y hace referencia a aquello que, en nuestro idioma, conocemos como teléfono inteligente.
Raspberry Pi:	Placa computadora (SBC) desarrollado en el Reino Unido por la Fundación Raspberry Pi (Universidad de Cambridge) en 2011.
GPIO:	(General Purpose Input/Output, Entrada/Salida de Propósito General) es un pin genérico en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución.
API:	Application Programming Interface (cuya traducción es Interfaz de Programación de Aplicaciones).
NoSQL:	Sistemas de gestión de bases de datos que difieren del modelo clásico de SGBDR (Sistema de Gestión de Bases de Datos Relacionales, en inglés, RDBMS).
RCA:	(RCA Jack o CINCH/AV connector). Tipo de conector eléctrico, empleado comúnmente en el mercado de audio y video.
UWE UML:	UWE UML (UML-Based Web Engineering) es una herramienta para modelar aplicaciones web. UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web.
CU:	Casos de Uso.
JSON:	Acrónimo de JavaScript Object Notation. JSON es un subconjunto de la notación literal de objetos de JavaScript.
VNC:	Siglas en inglés de Virtual Network Computing (Computación Virtual en Red).
SSH:	(Secure Shell = intérprete de órdenes seguro), sirve para acceder a máquinas remotas a través de una red.

BIBLIOGRAFÍA

- UNIVERSIDAD CONTINENTAL*. (2015). Obtenido de http://www.academia.edu/19539200/MATERIAL_DE_TRABAJO_INFORMATICA_II_FINAL
- MILENIUM*. (26 de Diciembre de 2016). Obtenido de <http://www.informaticamilenium.com.mx/es/temas/que-es-un-portal.html>
- QUORA*. (2016). Obtenido de <https://www.quora.com/What-is-firebase>
- Techwatch*. (12 de Diciembre de 2016). Obtenido de <http://techwatch.keeward.com/geeks-and-nerds/arduino-vs-raspberry-pi-vs-cubieboard-vs-gooseberry-vs-apc-rock-vs-olinuxino-vs-hackberry-a10/>
- ENGI Revista Electrónica De La Facultad De Ingeniería Vol. 3 No. 1 3 ISSN 2256-5612. (2017). *ENGI* .
- Google Docs*. (22 de Enero de 2017). Obtenido de https://docs.google.com/document/preview?hgd=1&id=1aeTWJ86TjRWKz2zPm7ivqIJ_BplqzifdxtF6JCta_M
- REDIRIS*. (16 de ENERO de 2017). Obtenido de <http://elies.rediris.es/elies9/4-1-2.htm>
- UNIVERSIDAD DE LA PUNTA*. (10 de ENERO de 2017). Obtenido de http://contenidosdigitales.ulp.edu.ar/exe/computacion/componentes_de_internet.html
- Universidad Francisco Gavidia*. (07 de ENERO de 2017). Obtenido de <http://ri.ufg.edu.sv/jspui/bitstream/11592/7743/3/004.738-F634a-CAPITULO%20II.pdf>
- BARBA, A. (1999). *GESTIÓN DE RED*. Barcelona : Edicions UPC.
- Business Intelligence. (s.f.). *Business Intelligence*. Recuperado el 03 de 2011, de <http://sites.google.com/site/inteligenciadenegocio/>
- Calero, C. (2010). *Calidad del Producto y Proceso Software* . Madrid: RA-MA Editorial.
- CASTRO, L. (17 de MARZO de 2016). *ABOUT*. Obtenido de <http://aprenderinternet.about.com/od/buscadores/g/Que-Es-Un-Portal-De-Internet.htm>
- COBO, Á. (2005). *PHP y MySQL TECNOLOGÍAS PARA EL DESARROLLO DE APLICACIONES WEB*. Madrid: Ediciones Díaz de Santos .
- DATE, C. J. (2007). *INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS*. California: Pearson.

- David, C. E. (2010). *Desarrollo de una solución OLAP para el soporte a la toma de decisiones gerenciales de la empresa Fermagri S.A.* Sangolquí: ESPE.
- DE PALOS, C. (2011). *Organización y transformación de los sistemas de información de la empresa.* Madrid: ESIC EDITORIAL.
- DEITEL, H. (2004). *CÓMO PROGRAMAR EN JAVA.* México: Pearson Educación.
- Díaz, M. S. (s.f.). *Metodologías para el Data Warehousing.* Obtenido de <http://www.docstoc.com/docs/52595481/Metodolog%C3%ADas-para-el-Data-Warehousing>
- ENEMDU. (22 de 12 de 2016). *INEC.* Obtenido de <http://www.ecuadorencifras.gob.ec/institucional/home/>
- ENEMDU. (21 de 12 de 2016). *INEC.* Obtenido de http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2016/170125.Presentacion_Tics_2016.pdf
- ESTRELLA, P. (27 de Enero de 2017). *ESPE.* Obtenido de <http://repositorio.espe.edu.ec/bitstream/21000/9962/1/T-ESPEL-SII-0133.pdf>
- GIULIANELLI, D. i. (25 de Enero de 2017). *unlp.* Obtenido de http://sedici.unlp.edu.ar/bitstream/handle/10915/53038/Documento_completo.pdf-PDFA.pdf?sequence=1
- GONZÁLES, I. (06 de Septiembre de 2003). *nluug.* Obtenido de <http://ftp1.nluug.nl/ftp/pub/ftp/os/Linux/doc/LuCaS/Presentaciones/200309hispalinux/8/8.pdf>
- Infante, I. D. (s.f.). *Pasos Para crear Cubos en Mondrian Workbench.* Comunidad Open Business Intelligence.
- Ing Lilliam Vega Torres, I. L. (11 de 2008). *La Inteligencia de Negocios. Su implementación mediante la plataforma Pentaho.* Recuperado el 03 de 2011, de www.redciencia.info.ve/memorias/ProyProsp/trabajos/l3.doc
- JUNESTRAND, S. (2004). *DOMÓTICA y HOGAR DIGITAL.* Madrid: Thomson Ediciones Spain.
- KABIR, M. (2003). *SERVIDOR APACHE 2.* Madrid: ANAYA MULTIMEDIA.
- LÓPEZ, E. (2006). *semanticscholar.* Obtenido de <https://pdfs.semanticscholar.org/a97d/9124ac9665839f1195f84542827d986e60f8.pdf>
- MENA, N. (Noviembre de 2007). *SciELO - Scientific Electronic Library Online.* Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352007001100003
- Mike, T. (25 de 03 de 2008). *Evaluating Pentaho Analysis.* Recuperado el 03 de 2011, de <http://wiki.pentaho.com/display/COM/Evaluating+Pentaho+Analysis>

- PONS, O. (2009). *INTRODUCCIÓN A LAS BASES DE DATOS*. Madrid: Thomson Editors Spain.
- SOMMERVILLE, I. (2006). *INGENIERÍA DEL SOFTWARE*. Madrid: PEARSON ADDISON WESLEY.
- Soto, J. A. (06 de 2005). *Gestiopolis*. Recuperado el 03 de 2011, de <http://www.gestiopolis.com/recursos5/docs/ger/buconce.htm>
- TELEFÓNICA. (18 de ENERO de 2017). *ACENS*. Obtenido de <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>
- TIMOTHY, B. (2010). *Desarrollo Web con PHP 6, Apache y MySQL*. Madrid: Anaya Multimedia.
- www.islavisual.com. (06 de Noviembre de 2016). *www.islavisual.com*. Obtenido de [www.islavisual.com: http://www.islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php](http://www.islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php)
- www.islavisual.com. (s.f.). *www.islavisual.com*. Recuperado el 06 de Noviembre de 2016, de [www.islavisual.com: http://www.islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php](http://www.islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php)
- xprograming. (1 de 1 de 2012). *xprograming*. Recuperado el 5 de 7 de 2014, de <http://xprogramming.com/xpmag/whatisxp>
- Yupa, E. E. (2009). *Propuesta Metodológica para aplicar Business Intelligence Caso Práctico "COHERVI S.A"*. Riobamba: ESPOCH.

NETGRAFÍA

HISTORIA, EVOLUCIÓN Y ESTADO ACTUAL E LA ARQUITECTURA. (2016-09-05). Obtenido de <https://infogr.am/historia-evolucion-y-estado-actual-e-la-arquitectura> [Último acceso: 01/02/2017].

Definición de arquitectura sostenible. (2-11-2013). Obtenido de <http://blog.deltoroantunez.com/2013/11/definicion-arquitectura-sostenible.html> [Último acceso: 01/02/2017].

PROYECTO DE INSTALACIÓN ELÉCTRICA Y DOMÓTICA EN UNA VIVIENDA UNIFAMILIAR. (12-01-2011). Obtenido de <http://upcommons.upc.edu/bitstream/handle/2099.1/11132/PFC.pdf> [Último acceso: 01/02/2017].

Raspberry Pi. (2017). Obtenido de <https://3dnews.ru/658701> [Último acceso: 01/02/2017].

ANÁLISIS, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA ORIENTADO A LA WEB PARA GESTIÓN ACADÉMICA. CASO PRÁCTICO: JHONN OESTEEN CHRISTIAN ACADEMY. (2014). Obtenido de <http://repositorio.espe.edu.ec/bitstream/21000/5661/1/AC-SISTEMA-ESPE-033734.pdf> [Último acceso: 01/02/2017].

TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES. (2015). Obtenido de http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2015/Presentacion_TIC_2015.pdf [Último acceso: 01/02/2017].