UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y TELECOMUNICACIONES
INGENIERÍA ELECTRÓNICA

TRABAJO DE GRADO PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

TÍTULO:

Implementación De Sistema De Control Remoto Mediante Una Aplicación Web A Un Proceso Domótico Sobre Un Sistema Embebido

Autor:

Oscar David Almanza Leones

Director:

Ing. José Daniel Ramírez Corzo

Co-Director:

Ing. German Arley Portilla González

PAMPLONA-COLOMBIA

JULIO de 2019

UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y TELECOMUNICACIONES
INGENIERÍA ELECTRÓNICA

TRABAJO DE GRADO PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

TÍTULO:

Implementación De Sistema De Control Remoto Mediante Una Aplicación Web A Un Proceso Domótico Sobre Un Sistema Embebido

Autor:

Oscar David Almanza Leones

Director:

Ing. José Daniel Ramírez Corzo

Co-Director:

Ing. German Arley Portilla González

JURADO CALIFICADOR:

_____

PAMPLONA-COLOMBIA

JULIO de 2019

AGRADECIMIENTOS

El agradecimiento en primera instancia a DIOS que es el que ha hecho posible este nuevo paso en mi vida, a mis padres, mis hermanos, Isis Almanza Redondo que fue mi apoyo y creyó en mi en todo momento, Luis Carlos Guzmán Almanza, Rosalbina redondo mi abuela que fue la que me encarrilo en este carrera tan linda, gustavo guerrero, Wilson noriega, Ericka Martínez, Carlos pinto, Daniel Ramírez Corzo, Wilmer Rodríguez, Juan Romario, Juan Rodríguez y Gustavo Guajada, que son mis hermanos que han estado hay en las buenas y las malas y a la familia vega Angulo que fueron parte importante en este proceso.

También quiero agradecer a todas las personas que formaron parte de este proceso, a mi director de tesis ing. Electrónico Daniel Ramírez corzo, codirector German portilla, al ramillete de profesores de la universidad de pamplona, mi familia, amigos que creyeron en mi hasta el último momento y en especial a esta ciudad pamplona que me enseño los valores de la vida y la rectitud en todos los ámbitos.

Gracias a mi Dios que ha hecho posible este sueño.

RESUMEN

En este proyecto se propone un control domótico, basado en un sistema integrado que controlará una serie de sensores y actuadores ubicados en una casa. Este sistema se comunica a Internet a través de una conexión física o inalámbrica (IoT - internet for things) para monitorear mediante un dispositivo móvil y almacenar en tiempo real las variables de control del sistema de automatización del hogar. El diseño tendrá un sistema auxiliar de energía renovable alternativo, a fin de mantener el sistema en funcionamiento en ausencia de energía eléctrica. Al final de este proyecto, se espera el logro de un sistema capaz de vincular un sistema integrado con un dispositivo móvil de bajo costo que pueda implementarse en diferentes áreas del control de la automatización del hogar.

ABSTRACT

This project proposes a domotic control, based on an integrated system that will control a series of sensors and actuators located in a house. This system communicates to the Internet through a physical or wireless connection (IoT - internet for things) to monitor by means of a mobile device and store in real time the control variables of the home automation system. The design will have an alternative renewable energy auxiliary system, in order to keep the system running in the absence of electricity. At the end of this project, it is expected to achieve a system capable of linking an integrated system with a low-cost mobile device that can be implemented in different areas of home automation control.

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

1. INTRODUCCIÓN

La domótica surge cuando aparecieron los primeros dispositivos de automatización en la década de los 70 consiguiendo así integrar dos sistemas el eléctrico y electrónico rigiéndose bajo el sistema x-10, protocolo de comunicación que opera a través del accionar de un control remoto, progresando a gran escala por el desarrollo de las redes informáticas de comunicación, ya sea por cableado o vía wi-fi supliendo así las falencia de los comienzos de la domótica y en la actualidad generando una oferta en torno a los servicios de la domótica por los nuevos protocolos que permiten un desarrollo que en principio era impensado.

En la presente investigación se refiere al tema de la domótica la cual permite la interconexión de una serie de equipos con el fin de obtener información sobre el entorno implementado, controlando así una serie de elementos electrónicos tales como detectores, actuadores, captadores, sensores, etc. Trasmitiendo la señal a una unidad central de control embebido que por medio de una determinada programación actuara sobre los determinados circuitos leyendo así unas señales.

Los cambios recogidos por la unidad central del sistema de control embebido serán suministrados por medio de una lectura a una base de dato en una nube, que posteriormente se actualizarán siendo cercanos a tiempo real en una página web que la URL será sura enlazada con app inventor una herramienta de Google para también ser monitoreada por un una Tablet o celular.

Para este tipo sistema de control domótico se trabajó con raspberry pi la cual consta de un sistema operativo propio llamado raspbian y dentro de una herramienta que se llama PAYTON por la cual se programaron los gpio para controlar el acceso de una puerta con contraseña y la iluminación de un cuarto enlazándola con una serie de comando a Firebase para tener conectividad remota, posteriormente ser enlazad para controlar y monitorear por una pagina web que esta enlazado por medio de Firebase.

1.1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

Estudios realizados para viviendas de interés social en Colombia, al investigar sobre el nivel de conocimiento en domótica por partes de los entrevistados, se encontró que solo el 40% manifestaron tener conocimiento sobre el tema y solo este considera que es factible la implementación de este tipo de tecnología. Teniendo en cuenta el porcentaje de apreciación negativa con una factibilidad del 60%, se ha considerado que unos de los factores fundamental es el costo de su instalación, creando así una barrera socioeconómica para este tipo de implementación.

Se requiere cambiar este tipo de pensamiento en los sistemas domótico y que no solamente pueden ser implementarlos en viviendas, por ejemplo, sistema domótico en una empresa para vigilar un proceso, control domótico de un vehículo, etc. Para ello la relación costo-beneficio es un factor muy importante y lo que buscamos es tener un producto que sea muy bueno y competitivo en el mercado, para que los estratos socioeconómicos bajos tengan accesibilidad a este tipo de tecnología y tener mejor calidad de vida.

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Implementar un sistema de control remoto mediante una aplicación móvil a un proceso domótico sobre un sistema embebido.

1.2.2. OBJETIVOS ESPECIFICOS

Para atender al objetivo general del presente trabajo, han sido planteados los siguientes objetivos específicos:

- Diseñar un aplicativo web que enlace el sistema de control domótico.
- Establecer los algoritmos para el control y almacenamiento de las variables del proceso domótico bajo un sistema embebido y almacenamiento Firebase.
- Seleccionar un sistema de energía alternativa que cubra la necesidad del proyecto de control.
- Validar el sistema de control domótico.

2. ESTADO DEL ARTE

Con la evolución de comunicaciones hoy en día, se ha creado un entorno de comunicación entre personas y las nuevas tecnologías que permiten controlar de forma remota dispositivos y electrónicos ya sea desde la internet, la computadora o un dispositivo móvil. Estas tecnologías creadas por el hombre se le denomina domótica y es la interacción de diferentes áreas del conocimiento como la informática, las telecomunicaciones, la electrónica y la electricidad que reunidas mejoran la calidad de vida del ser humano.

“El hogar digital automatización doméstica basada en tecnología IP”. Moreno Barajas Miguel Ángel (2006) Comillas – Universidad Pontificia. El objetivo de este proyecto fue realizar una pasarela que sirva de enlace directo entre Internet y un relé que pueda actuar sobre distintos elementos de la casa. Actualmente, la comunicación se realiza llegando desde el exterior, mediante el protocolo de Internet (IP), a una pasarela en la vivienda, cambiando en dicho punto a diversos protocolos (X10, EIB.) que permiten la comunicación con los distintos actuadores o sensores domóticos del hogar. El proyecto pretende la comunicación con dichos actuadores directamente por IP, mediante un desarrollo basado en programación de microcontroladores.

Prototipo de un Sistema de Telemetría y Control para Seguridad en vehículos, soportado en Redes Móviles. Cárdenas Valencia, A.H. Echeverry Giraldo A.F. (2012) Tesis de pregrado. Universidad Católica de Pereira. El objetivo de este proyecto fue desarrollar un sistema para controlar un vehículo, el cual le permitía al usuario a partir de mensajes de texto manejar el encendido o apagado del vehículo, bloqueo de puertas, temperatura y nivel de aceite. Este prototipo se compone de una parte de telemetría y una parte de control; en cuanto a la parte de telemetría está instalada en el vehículo por medio de un modem de telefonía celular GSM, el cual permite la comunicación por medio de un mensaje de texto que va dirigido al número celular del propietario del vehículo, suministrándole información actual sobre el estado de éste. En la parte de control, el usuario utiliza el teléfono celular para llamar al vehículo y el sistema modem contesta automáticamente y establece una comunicación para poder controlar por medio del teclado del teléfono las diferentes opciones que se mencionan anteriormente.

Diseño e implementación de una arquitectura multimedia para el hogar digital. Jiménez Suárez C.J. (2011) Gran Canaria – Universidad de Las Palmas, el objetivo de este proyecto fue la implementación de la infraestructura telemática de una red informática para el hogar digital basada en la arquitectura UPnP que está compuesta por cinco electrodomésticos. Dos de estos electrodomésticos son de multimedia y permiten la reproducción de audio y video, también se implementó un equipo de control, el cual permite la interacción entre el usuario y los servicios que se prestan los dispositivos integrados a la red.

Diseño de un sistema de control domótico basado en la plataforma Arduino”. Lledó Sánchez Emilio (2012) Valencia - Escuela Técnica Superior Ingeniería Informática Politécnica Universitaria. Este proyecto se encarga de brindar los conocimientos básicos para entender qué es y cómo funciona un sistema domótico, y cómo utilizando el hardware libre de Arduino se puede crear 16 un sistema estable con un presupuesto muy inferior al de las viviendas de alta categoría.

2.1. HISTORIA DE LA DOMOTICA

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

El automatismo se inició durante el siglo XIX con el desarrollo industrial, el cual permitía controlar y establecer secuencialmente los procesos productivos. Con el paso del tiempo y hasta la actualidad, los sistemas han sido perfeccionados hasta llegar al punto en donde las industrias basan gran parte de sus fases de producción en tareas automatizadas o temporizadas.

Los pioneros en desarrollar casas y productos inteligentes fueron estados unidos y Japón en el año 1977, bajo la influencia económica de esa época. Desde ese año se comenzaron a realizar estudios pertinentes sobre qué impacto tendría en la sociedad la automatización en el hogar y que rentabilidad en el mercado tendría esta ideología tecnológica en el sector industrial.

En Colombia la domótica ha sido una tendencia tecnológica que se ha integrado lentamente, dado que el tipo de productos de innovación en el hogar son costosos para la población colombiana su comercialización ha sido para sectores exclusivos de altos grupos sociales que tiene dominio en la economía del país, privando a la mayoría de este beneficio por no tener manufactura de producción de los nuevos productos tecnológicos.

2.1.1. Domótica

De manera general, un sistema domótico dispondrá de una red de comunicación y diálogo que permite la interconexión de una serie de equipos a fin de obtener información sobre el entorno doméstico y, basándose en ésta, realizar unas determinadas acciones sobre dicho entorno. El funcionamiento de una casa inteligente consistiría a grandes rasgos de lo siguiente: los elementos de campo (detectores, sensores, captadores, etc.), transmitirán las señales a una unidad central inteligente que tratará y elaborará la información recibida. En función de dicha información y de una determinada programación, la unidad central actuará sobre determinados circuitos de potencia relacionados con las señales recogidas por los elementos de campo correspondientes.[3]

Previamente hare una pequeña explicación de la conceptualización a tratarse dentro del contexto del proyecto. Iniciando por el significado de domótica: Es un conjunto de sistemas capaces de hacer que una vivienda realice actividades automáticamente. La funcionalidad de este sistema comprende los servicios de gestión para el ahorro de energía, bienestar, accesibilidad entre otras funciones. Estos sistemas pueden estar integrados gracias a redes interiores y exteriores de comunicación, cableadas o inalámbricas y cuyo control puede ser gestionado desde dentro y fuera de la casa mediante dispositivos móviles.

En la Figura 1 se muestra un esquema a grandes rasgos del funcionamiento para sistema domótico, en el recuadro amarrillo se describen los usuarios los cuales monitorean el estado y operan el sistema a través de aplicaciones de escritorio o móviles, estas aplicaciones permiten el control y vigilancia del proceso en tiempo real remotamente. En el recuadro naranja del centro se describen los elementos de control los cuales pueden variar dependiendo del sistema que controlen, estos elementos pueden ser sistemas electrónicos embebidos con la capacidad de recibir y enviar información a las aplicaciones que controlan y supervisan el sistema, la información recibida es interpretada y ejecutada a través de los actuadores al mismo tiempo que vigilan el estado de los sensores para enviar esta información a los usuarios.

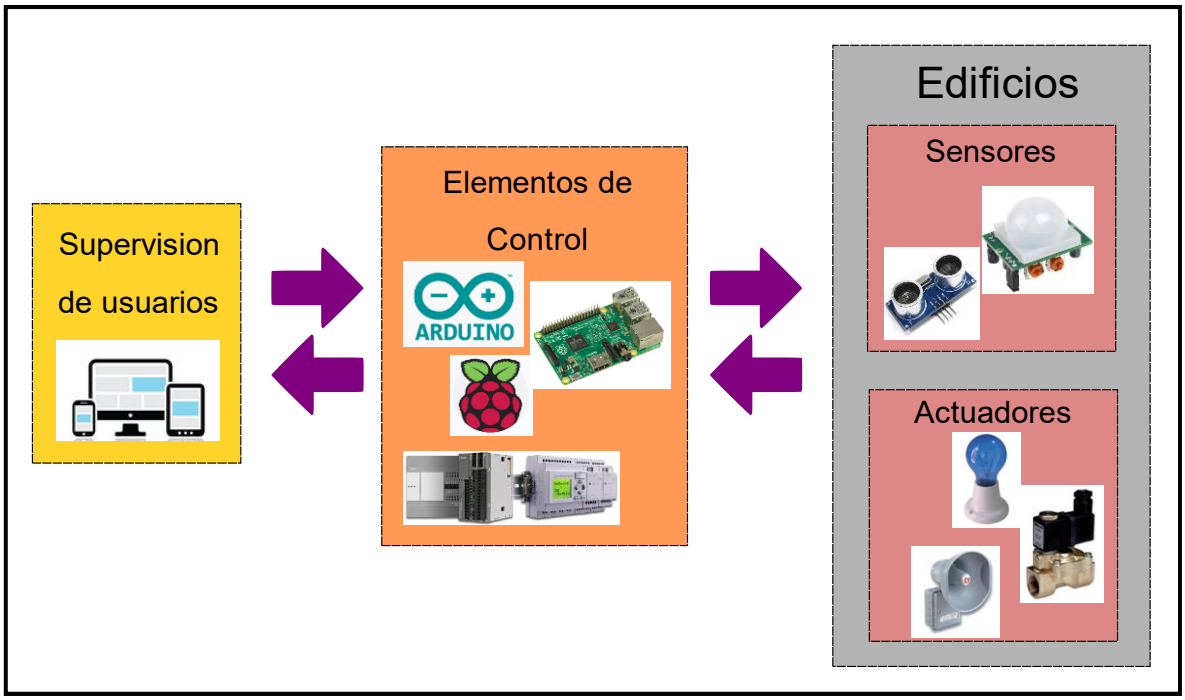


Figura 1. Sistema de control inteligente para un control domótico, está dividido en tres sectores una parte de supervisión por usuarios. Fuente: Autor.

2.1.2. Aspectos de la domótica

La domótica ofrece múltiples servicios dentro del hogar enfocados en cuatro funciones básicas que describen un sistema autónomo domestico: confort, seguridad, energía y telecomunicaciones. El costo de la inversión en la instalación a realizarse va hacer proporcional a la finalidad de la cantidad de automatismos que se quieren controlar en estos cuatros grandes funciones de despliegue del mismo sistema. La Figura 2 muestra un esquema resaltando los aspectos más importantes en la domótica como lo son la seguridad de sistema de control, el confort de la infraestructura donde se aplique el sistema y la facilidad de control por parte del usuario, para facilitar la comunicación del sistema y gestión de energía para evitar colapsos en el funcionamiento.

2.2. CLASIFICACION DE LOS SISTEMAS DOMOTICOS

En el mercado mundial los sistemas domésticos se aprecian configuraciones genéricas y clásicas que permiten atender a determinados criterios. Estos especifican el modo en que los elementos de control del sistema se van hacer ubicados en él espacio. Existen dos arquitecturas básicas:

2.2.1. Arquitectura centralizada

Es aquella en la que los elementos a controlar y supervisar (sensores, luces, válvulas, etc) han de cablearse hasta el sistema de control de la vivienda (PC o similar). El sistema de control es el 'corazón' de la vivienda y si falla, todo deja de funciona. En este tipo de sistemas, sería conveniente elegir la topología de cableado en la fase de construcción. La ampliación de este tipo de sistemas es complicada y costosa.[4]

Tiene una unidad central inteligente encargada de administrar la edificación, la cual enviara y recibirá información de distintos elementos electrónicos, la central se encarga de procesar los datos del entorno y, en función de la información y la programación que se haya hecho sobre ella, actuando sobre los determinados circuitos electrónicos encargados de cumplir funciones. Este controlara elementos y supervisara (sensores, luces, puertas válvulas, etc).

En la Figura 2 se muestra un esquema de bloques para un sistema centralizado de una arquitectura domótica, aquí todos los sensores y actuadores s se comunican a un sistema central de control.

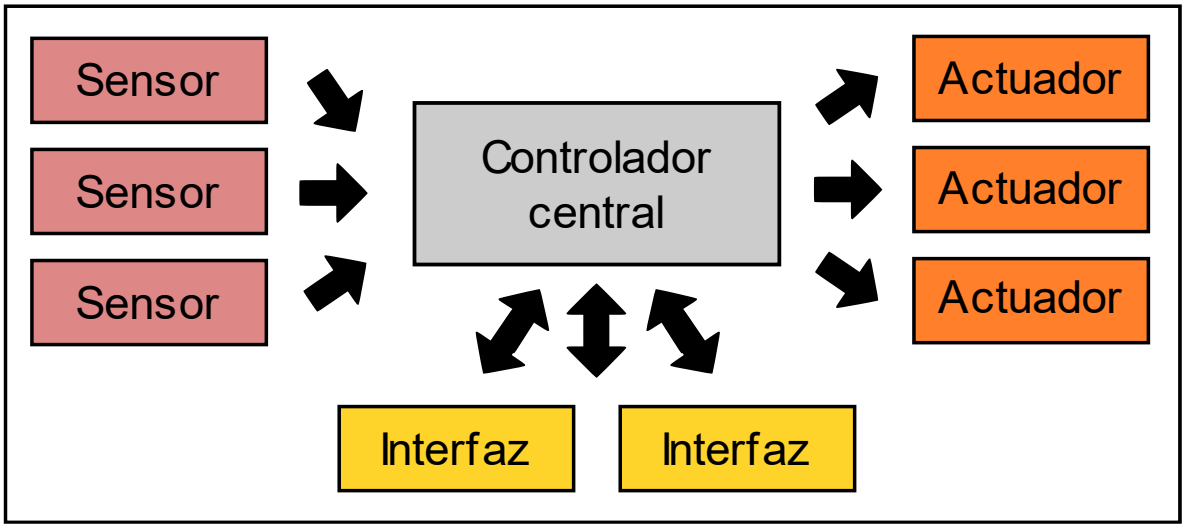


Figura 2. Esquema de bloques para una arquitectura domótica centralizada, donde controla los actuadores y sensores por medio del controlador central y controlada y monitoreada por una interfaz que esta comunicada a aplicaciones de escritorio o móviles. Fuente: Autor.

2.2.2. Sistemas distribuidos

Los sistemas distribuidos combinan las tipologías centralizada y descentralizada. La inteligencia del sistema está localizada en cada uno de los nodos de control y cada nodo tiene acceso físico directo a una serie limitada de elementos de red. Es necesario, al igual

que en el caso de los sistemas descentralizados, un protocolo de comunicaciones para que todos los módulos produzcan una acción coordinada.

La Figura 3 muestra la estructura de una arquitectura distribuida en donde se observa que los sensores, actuadores e interfaz de control utilizan un bus para la comunicación, en este sistema la comunicación es bidireccional y toda fluye mediante un solo canal de comunicación que sirve para los sensores, actuadores e interfases de control.

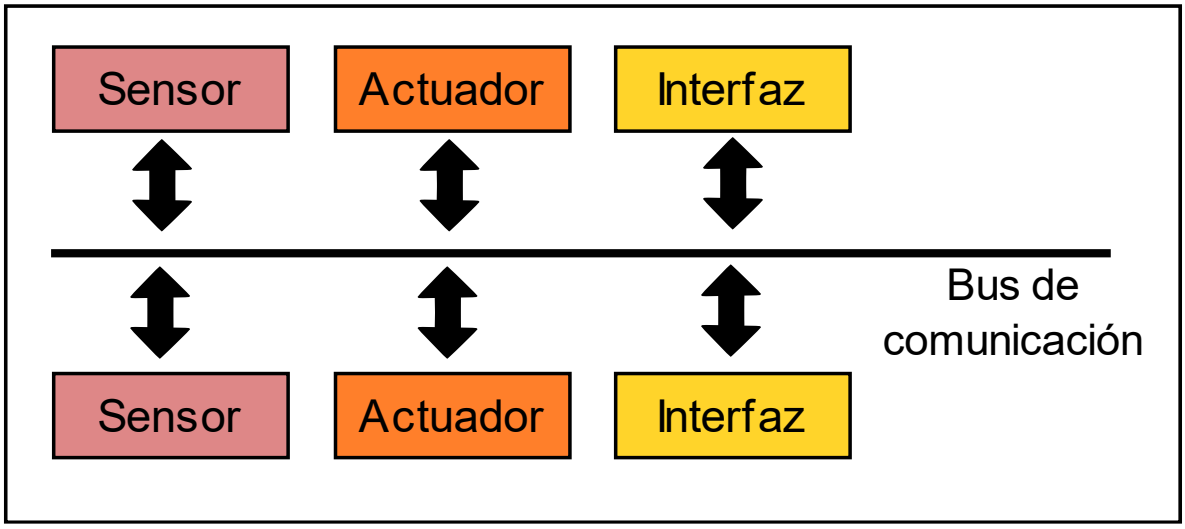


Figura 3. Esquema de bloques para una arquitectura domótica distribuida. Fuente: Autor.

La Tabla 1 muestra una comparación de las dos arquitecturas de domótica, resaltando características de las mismas, una gran ventaja de la arquitectura domótica centralizada es su costo frente a distribuida, para una arquitectura domótica centralizada se facilita el diseño de la interfaz de control ya que todos los elementos controlados envían su información a mismo punto.

TIPO DE ARQUITECTURA	VENTAJAS	INCONVENIENTES	INTERFACES HOMBRE-MAQUINA
CENTRALIZADA	Equipos más económicos.	Gran cantidad de cableado. Centralización de funciones. Complicados	No se corresponde con la filosofía de sistemas domóticos. Muchos sistemas son Autómatas programable
DISTRIBUIDA	Sistemas más robustos frente a fallos. Fácil diseño de instalaciones. Gran facilidad de uso.	Requiere programación o configuración Precio.	Necesita de una interface definida para que su control sea el adecuado.

Tabla 1. Tabla comparativa para las arquitecturas domóticas definidas.

2.3. SISTEMAS EMBEBIDOS

En este apartado se trata de analizar qué son los sistemas embebidos, cuáles son sus características básicas, así como las interfaces. La comunicación adquiere gran importancia en los sistemas embebidos, por tanto, un sistema pueda comunicarse mediante interfaces estándar de cable o inalámbricas.

Se trata de un sistema de computación diseñado para realizar una o algunas funciones dedicadas frecuentemente en un sistema de computación en tiempo real. Al contrario de lo que ocurre con los ordenadores de propósito general (como por ejemplo una computadora personal o PC) que están diseñados para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para cubrir necesidades específicas. En un sistema embebido la mayoría de los componentes se encuentran incluidos en la placa base (la tarjeta de vídeo, audio, módem, etc.) y muchas veces los dispositivos resultantes no tienen el aspecto de lo que se suele asociar a una computadora.[5]

Los sistemas embebidos se caracterizan por enfocarse y programarse directamente en el lenguaje ensamblador del microcontrolador o microprocesador incorporado sobre el mismo en montaje superficial. Este sistema utiliza compiladores específicos en lenguajes como C o C++; en algunos casos cuando su tiempo de respuesta no se necesita que seas tan rápidos y para mayor tiempo de respuesta se utilizan compiladores de lenguajes como JAVA.

2.3.1. Características de los sistemas embebidos

Muchos sistemas embebidos deben cumplir restricciones de tiempo real. Un sistema de tiempo real debe reaccionar a estímulos del objeto controlado (u operador) dentro de un intervalo definido por el ambiente.

Las características básicas de los sistemas embebidos son las siguientes:

- Deben ser confiables para que el sistema trabaje correctamente dado que está funcionando en t=0.
- La mantenibilidad para que el sistema vuelva a trabajar correctamente d unidades de tiempo después de un fallo.
- La disponibilidad para que el sistema esté funcionando en el tiempo t.
- La seguridad informática para mantener la comunicación confidencial y autentificada.
- Deben ser eficientes en cuanto a la energía, al tamaño de código, al peso y al costo.
- Están dedicados a ciertas aplicaciones.
- Interfaces de usuario dedicadas sin ratón, keyboard y pantalla.

Una característica a resaltar es que siempre se encuentran conectados a sistemas con ambientes físicos por medio de sensores y actuadores, es decir son sistemas híbridos por tener partes analógicas y digitales, los cuales están en interacción continua con su entorno y su ejecución es a ritmo determinado por su entorno.

El diseño de un producto en base a sistemas embebidos está orientado a minimizar los costos y maximizar la confiabilidad, también es importante resaltar la seguridad del sistema, funciones y protocolos de encriptado que protegen la información durante todas las fases.

2.3.2. Tipos de sistemas embebidos

En el mercado existen una gama de sistemas de embebidos dependiendo el tipo de arquitectura y para que aplicaciones se necesite, esto define la versatilidad del dispositivo en cuanto a la necesidad de trabajar lo más cercano al tiempo real.

Los sistemas embebidos más utilizados en el mercado son:

- Controladores Lógicos Programables (Plc)
- Raspberry Pi
- Arduino

2.3.3. Controladores lógicos programables

Un Controlador Lógico Programable es un computador especialmente diseñado para automatización industrial, en donde su principal característica llevar control de una maquina o proceso industrial. Carece de periféricos como teclado, mouse, pantallas inteligentes como las computadoras modernas, pero internamente si se consideran unidades de computo por sus procesadores, memoria y puertos de comunicación. También por su software con sistema operativo llamado Firmware y su programación específica para un tipo de aplicación.

La principal diferencia entre un PC y un PLC, es que el PLC contienen múltiples canales para medir distintas señales provenientes de sensores instalados en la maquina o proceso que controlan. Y también tienen canales de salida de señal que actúan sobre la maquina o proceso que controlan.[6]

Un PLC protege un proceso industrial por medio de su control maestro, posibilitando además las opciones de monitoreo y diagnóstico, mostrándolo por medio de una interfase HMI o presentándolas a una red de control superior. Una característica notable de estos sistemas de control es su respuesta en tiempo real, ya que reacciona automáticamente ante diferentes estímulos que captan los sensores del sistema. Estos controladores lógicos programables puede ser parte de un sistema de control distribuido o ser parte de un SCADA.

2.3.4. Raspberry pi

La raspberry pi es una tarjeta de adquisición de datos que se utiliza para diseño e implementación de software utilizado para diferentes aplicaciones, su versatilidad la hace una herramienta poderosa ya que es una minicomputadora de procesamiento muy potente. Las características más notables de este tipo de tarjeta son su variabilidad de puertos que ofrece al programador, siendo una herramienta de enlaces muy eficiente para las diferentes tecnologías. existen varios modelos de este tipo de tarjeta que han ido saliendo a lo largo de su desarrollo en las cuales se han introducido mejoras como la velocidad del procesador, tamaño de la memoria RAM, conexión inalámbrica entre otros.

Este tipo de tarjeta permite la ejecución de diferentes sistemas operativos, pero el más usado y recomendado es Raspbian. Este sistema operativo trae preinstalado varios entornos de programación para el desarrollo de software entre ellos están Python, Java, Sonic pi, entre otros. Las tarjetas Raspberry pi traen una serie de pines GPIO que permiten

la lectura y control de valores digitales entre 0 y 3,3 voltios. Estos pines pueden ser usados para controlar sensores, actuadores, pantallas y varios tipos de comunicación como son SPI e I2c.

2.3.5. Arduino

Arduino es una herramienta para hacer que los ordenadores puedan controlar el mundo físico a través de tu ordenador personal. Es una plataforma de desarrollo de computación física de código abierto, basada en una placa con un sencillo microcontrolador y un entorno de desarrollo para crear programas para la placa con sintaxis resumida y muy fácil de interpretar basada en C y C++.

Puedes usar Arduino para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos. Los proyectos con Arduino pueden ser autónomos o comunicarse con un programa (software) que se ejecute en tu ordenador. La placa puedes montarla tú mismo o comprarla ya lista para usar, y el software de desarrollo es abierto.[7]

2.4. ENTORNOS DE PROGRAMACIÓN

En este apartado resaltaremos los entornos de programación más utilizados en el ámbito aplicativo enfocado en la domótica aprovechando sus características de enlaces con dispositivos desarrolladores y su sintaxis fácil de leer con ventajas en activarse con una reacción muy cerca al tiempo real.

A continuación, se especificarán los entornos de programación más eficientes actualmente en el mercado:

2.4.1. Lenguaje Python

Python es un lenguaje de programación sintetizado de alto nivel en donde se realizan códigos de programas en base de un intérprete que trabaja en conjunto con el mismo, ejecutando las sentencias en un determinado tiempo de respuesta según sea la exigencia de memoria en el Raspberry pi. Este lenguaje está inmerso en el sistema operativo Raspbian igual que su interprete por el cual al ejecutar una sentencia de código creado este se puede mostrar por los puertos de salidos de la misma tarjeta Raspberry creando en si una aplicación interactiva de una situación real de la vida cotidiana del ser humano.

Dentro de la gran gama de decisiones que pueden llegar ejecutarse en Python se encuentra un sin número de librería que sintetizan el script aplicativo creado por el programador, reduciendo el número de líneas de códigos y dejando más fácil el análisis e interpretación del mismo para demás lectores y aficionados a esta lógica de programación.

Se trata de un lenguaje de programación multiparadigma, Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional.[8]

Hay una solución intermedia: podemos diseñar lenguajes de programación que, sin ser tan potentes y expresivos como los lenguajes naturales, eliminen buena parte de la complejidad propia de los lenguajes ensambladores y estén bien adaptados al tipo de problemas que podemos resolver con los computadores: los denominados lenguajes de programación de alto nivel. El calificativo de alto nivel señala su independencia de un ordenador concreto. Por contraposición, los códigos de máquina y los lenguajes ensambladores se denominan lenguajes de programación de bajo nivel.[9]

2.4.2. Lenguaje HTML

Las páginas Web se localizan en sitios específicos en la rede de internet llamados servidores Web. Por lo tanto, un documento WWW tiene la propiedad de enlazarse a otros documentos que se encuentran en el mismo servidor Web o en otros servidores Web, logrando la interconexión más cercana al tiempo real para diferentes aplicaciones.

El lenguaje estandarizado para la creación de páginas Web es el lenguaje HTML (HyperText Markup Language, Lenguaje de Marcas Hipertexto). HTML es un lenguaje muy sencillo que permite describir documentos hipertexto. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado).[10]

Por medio de páginas con programación HTML se pueden crear interfase de monitoreo en donde se extrae información proveniente de un gestor de base de datos que organiza la información clasificándola y actualizándola para que la aplicación ligada a estos sistemas siempre tenga respuestas inmediatas cercanas a tiempo real. La característica más significativa de este tipo de control por página web de monitoreo es su flexibilidad de desarrollar opciones en accionamientos de control es decir insertar botones, barras de despliegue de información, hipervínculos para enlaces con más páginas web e información misma del funcionamiento de la aplicación. Además, ofrece conexión con APP para Android para que el monitoreo se pueda hacer desde un celular o una tableta si así lo desea el usuario que quiera este servicio en su hogar. Con esta información acerca de programación HTML se quiere acaparar al lector para que en el transcurso del desarrollo de este aplicativo se tenga en cuenta este tipo de tecnología que va hacer de gran ayuda en la creación de la interfase de comunicación entre el gestor de base de datos Firebase y la tarjeta de control Raspberry pi acoplado a los módulos de accionamientos enfocado en la domótica.

2.4.3. Lenguaje CSS

CSS también conocidas como hojas de estilo es el complemento perfecto de página HTML ya que su papel es gestionar la apariencia de la página web en cuanto el diseño, posicionamiento, colores, tamaño de texto por tanto para que una sentencia CSS funcione es necesario estar ligado a una página con programación en HTML.

Se puede crear un sitio web únicamente en HTML, pero no va a quedar muy estético por la forma como aparecerá la información. Esta es la razón por la que CSS siempre lo completa. Para hacerse una idea, la figura siguiente muestra cómo se ve la misma página sin CSS y con CSS.[11]

2.4.4. Lenguaje JavaScript

De todos los servicios que ofrece INTERNET, no cabe duda de que el más popular es la WWW más conocida como red mundial. La WWW no es más que millones de páginas en formato electrónico, con los contenidos y temáticas más diversas a las que podemos acceder gracias a un ordenador una red y un navegador.

Como cualquier otro lenguaje de programación, JavaScript tiene algunas características especiales: sintaxis, modelo de objetos, etc. Claramente, cualquier cosa que diferencia un lenguaje de otro. Además, descubrirás rápidamente que JavaScript es un lenguaje relativamente especial en su acercamiento a las cosas. Esta parte es esencial para cualquier principiante de programación e incluso para aquellos que ya conocen un lenguaje de programación debido a que las diferencias con otros lenguajes de programación son numerosas.[12]

2.4.5. Formato JSON

JSON es un formato de datos muy ligero basado en un subconjunto de la sintaxis de JavaScript: literales de matrices y objetos. Como usa la sintaxis JavaScript, las definiciones JSON pueden incluirse dentro de archivos JavaScript y acceder a ellas sin ningún análisis adicional como los necesarios con lenguajes basados en XML.[13].

JSON está construido sobre dos estructuras:

- Un nombre o dato que tiene un determinado valor o estado.
- Un objeto o nodo que puede contener dentro otros nodos o datos.

En la Figura 4 se muestra un para JSON, los nodos comienzan con un “{“ y terminan con un “}”, mientras que un dato o nombre es seguido de “:” y el estado o valor en que se encuentra, los nodos y datos son separados por “,”.

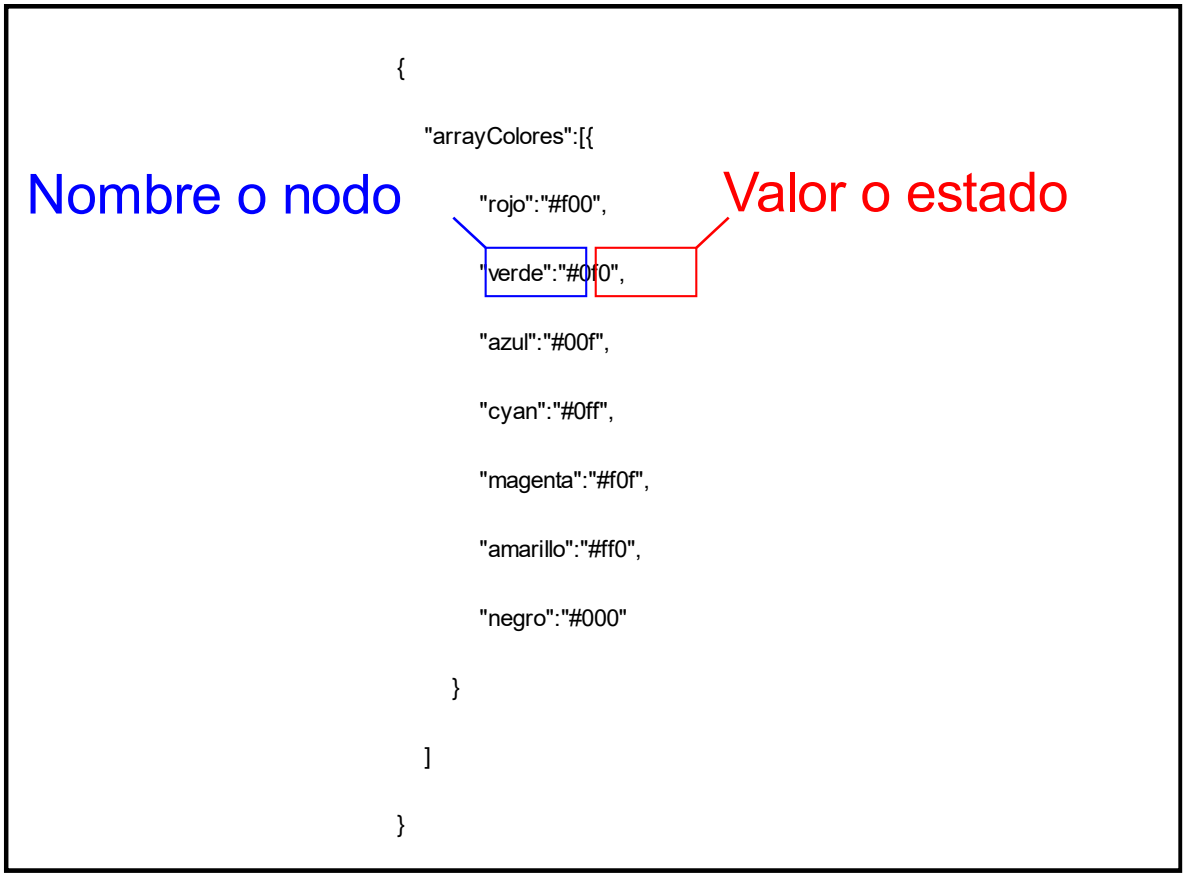


Figura 4. Formato basico de un archivo JSON. Fuente: Autor.

2.5. FIREBASE

La empresa fue fundada en 2011 por ANDREW LEE Y JAMES TAMPLIN. Producto inicial de Firebase era una base de datos en tiempo real, lo que proporciona una API que permite a los desarrolladores almacenar y sincronizar los datos a través de múltiples clientes. Con el tiempo, se ha ampliado su línea de productos para convertirse en un paquete completo para el desarrollo de aplicaciones. La compañía fue adquirida por Google en octubre de 2014 y un número significativo de nuevas características se presentaron en de mayo de 2016 a Google I / O.[8]. Firebase es una plataforma móvil que te permite desarrollar rápidamente apps de alta calidad, aumentar la base de usuarios y ganar más dinero. Firebase contiene funciones complementarias que puedes combinar y adaptar según tus necesidades.

2.5.1. Herramientas de Firebase

Firebase proporciona una solución eficaz frente no solo a problemas de desarrollo, sino también de escalabilidad a medida que la base de usuarios de la aplicación crece, ya que los servidores son proporcionados por Google. Entre sus funcionalidades se encuentra un servicio de autenticación, base de datos en tiempo real, almacenamiento de archivos,

solución de errores, funciones Backend, testeo, y medida de estadísticas recogidas de los usuarios.[14]

2.5.2. Base de datos en tiempo real

Firebase proporciona un gestor de base de datos (Raltime Database) NoSQL que almacena datos y los sincroniza en tiempo real. Este modelo organizacional de información ha sido utilizado para almacenar toda clase de datos, resultados que son necesario para que una aplicación trabaje en tiempo real sin perder información necesaria para luego actualizarse automáticamente. Esta base de datos utiliza un formato JSON para la gestión de los datos permitiendo una gran versatilidad al momento de generar las consultas y además mejora la velocidad para transacciones para las lecturas y actualizaciones de los nodos y estado de los datos.

2.5.3. Autenticación

La plataforma Firebase proporciona un método de registro e inicio de sesión que no solo incluye autenticación a través de correo, sino que también permite la autenticación a través de proveedores externos como Facebook, Twitter, Github y Google. Esto permite una gran facilidad al momento de desarrollar la parte de inicio de sesión de las aplicaciones mejorando el tiempo de desarrollo y despliegue del aplicativo.

2.5.4. Hosting

Firebase Hosting es un servicio de hosting de contenido web con nivel de producción orientado a programadores. Con un solo comando, puedes implementar aplicaciones web y entregar contenido dinámico y estático en una red de distribución de contenidos global rápidamente y a la vez sincronizar al actualizarse.

Las principales funciones que tiene Firebase hosting son:

- Entregar contenido mediante una conexión segura
- Alojar contenido dinámico y estático, además de microservicios.
- Publicar contenido con rapidez
- Implementar versiones nuevas con un comando.
- Realizar la reversión con un clic.

Firebase Hosting se creó para el programador web moderno. Los sitios web y las apps son más potentes que nunca gracias al surgimiento de marcos de trabajo de Frontend de JavaScript, como Angular, y herramientas de generación estática, como Jekyll. Ya sea que estés implementando una página de destino sencilla de la app o una app web progresiva (AWP) compleja, Hosting te ofrece la infraestructura, las funciones y las herramientas orientadas a la implementación y administración de sitios web y apps

2.6. TRABAJOS RELACIONADOS

En Colombia, el desarrollo de aplicativos ha avanzados rápidamente con el paso de los años ya que se han convertido en fuentes de ingreso para el mercado colombiano. Una de las características de este tipo de tecnología es la versatilidad de adaptación que tienen los

aplicativos webs como las bases de datos en tiempo real para comunicarse con algún tipo de sistema que mejora la calidad de vida de los usuarios.

Un ejemplo claro que soporta mi investigación relacionada a la domótica es este tema de tesis creado por un estudiante de especialización en ingeniería de software titulada "Prototipo de aplicación móvil para consulta de información referente a vuelos por parte de los usuarios de aerolíneas en Colombia basado en el Análisis de Tecnologías para Transmisión de Información en Tiempo Real" el cual relaciona el control y tráfico de información guardada en un gestor de base de datos en tiempo real para que las respuesta de este sistema sean lo más eficiente posible. En su justificación el deja claro su investigación en este apartado:

Este proyecto busca implementar un prototipo de aplicación móvil para consulta de información referente a vuelos por parte de los usuarios de aerolíneas en Colombia aplicando el resultado obtenido del análisis de tecnologías para transmisión de información en tiempo real. Dicho análisis pretende explorar nuevas tecnologías basadas en la innovación que permitan mantener informado al usuario de aerolíneas en tiempo real de las eventualidades que puedan ocurrir sobre los cambios en la información de los vuelos, brindando así un puente de comunicación en tiempo real entre la aerolínea y el usuario del vuelo.[15]

3. METODOLOGÍA

En este trabajo se implementó un sistema domótico centralizado, se escogió este tipo de sistema debido a sus ventajas sobre otros sistemas como se muestra en la Tabla 1. De esta tabla se resaltan el bajo costo para el desarrollo de un sistema de este tipo, el costo es un papel muy importante al momento de montar un sistema de control domótico ya que puede ser el punto de flexión para que una persona opte por hacer una implementación domótica. La implementación de una arquitectura domótica centralizada se basa un controlador que hace el papel de cerebro o corazón del sistema, el controlador es donde llegan todas las señales de los sensores que monitorean la casa y controlan los actuadores.

El diseño de este sistema domótico se basa en el control y monitoreo de una bombilla y una puerta. para la bombilla se controla el estado de encendido o apagado, este estado es visualizado en la aplicación de control desarrolla para el usuario, para la puerta se controla el seguro que permite su apertura y se detecta el estado de abierto o cerrado para ser visualizado también en la aplicación del usuario.

3.1. ARQUITECTURA DEL SISTEMA DOMÓTICO PROPUESTO

En la Figura 5 se muestra un diagrama para el sistema domótico centralizado propuesto en este trabajo, este diagrama se divide en cuatro partes: Interfaz de control y supervisión, Servidor en la nube, Controlador, Dispositivos de acción y control. En los ítems siguientes se explican más a detalle cada uno de estos bloques.

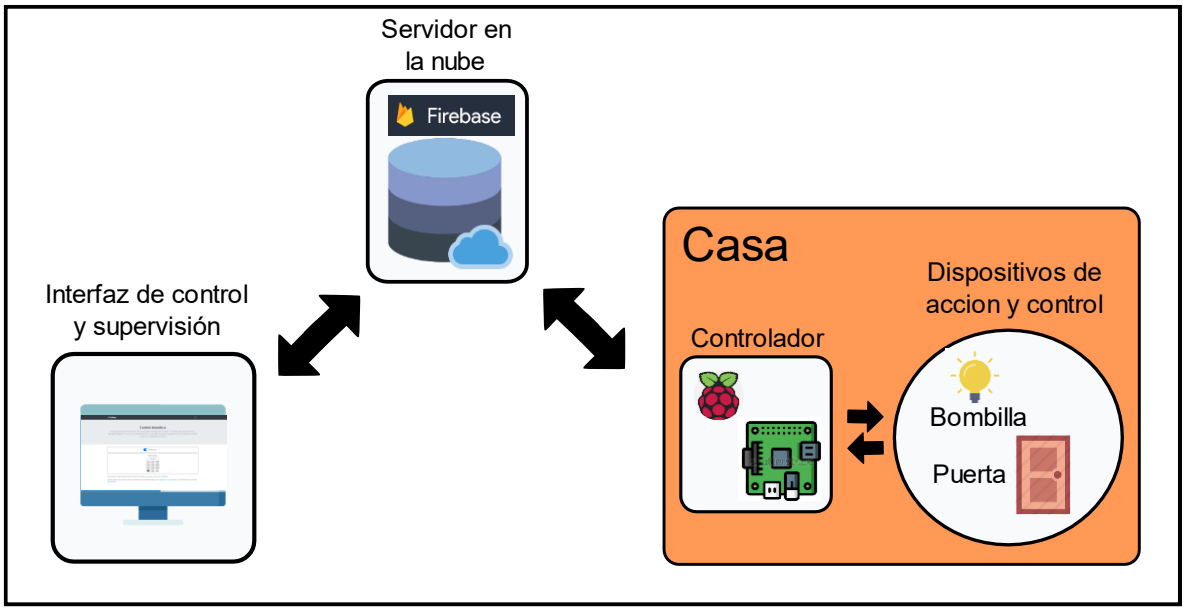


Figura 5. Diagrama de bloques para el sistema de control propuesto. Fuente: Autor.

3.2. INTERFAZ DE CONTROL Y SUPERVISIÓN

La interfaz para controlar y supervisar el sistema domótico es una aplicación web desarrollado en HTML y JavaScript, esta aplicación proporciona que se encuentra alojada en un servidor y puede ser consultada desde un sitio web, esta aplicación proporciona control solo los actuadores de la casa y muestra el estado de los sensores en esta. la interfaz se comunica con a través de una base de datos alojada en la nube cargando los datos en esta para que el controlador detecte estos cambios y realice las acciones necesarias.

3.3. SERVIDOR EN LA NUBE

El servidor es parte esencial para la comunicación entre la aplicación de control y supervisión con el controlar, en el servidor también se aloja la aplicación que se sirve al usuario a treves de una dirección web o URL. El cliente excede a la aplicación la cual carga las credenciales que permiten conectarse con la base de datos, en la base de datos se almacenan los estados y valor de los dispositivos de acción y control que se encuentran en la casa, el estado de estos dispositivos es consultado constantemente por el controlador para llevar a cabo las acciones necesarias y escribir el estado para los sensores de la bombilla y la puerta.

3.4. CONTROLADOR

El controlador vigila los cambios en la base de datos sabre los valores de los actuadores que controla, también lee el estado de los sensores para la bombilla y puerta y escribe estos valores en la base de datos constantemente, la aplicación que escucha esto cambios se actualiza y muestra información en tiempo real al usuario sobre el estado de la casa. El controlador que se utilizo fue la Raspberry pi modelo b+ v1.2, este controlador es un sistema embebido con pines digitales de entrada y salida.

3.5. DISPOSITIVOS DE ACCION Y CONTROL

Estos dispositivos de acción son los encargados de llevar a cabo las acciones en la casa que el usuario selecciona desde la aplicación, el control de la bombilla y la el seguro de la puerta de la habitación son accionado finalmente por estos dispositivos. Los dispositivos de control son los sensores, los sensores están leyendo el estado de la puerta y la bombilla enviando esta información al controlador el cual efectúa los cambios en la base de datos y puedan mostrarse en la aplicación web del usuario.

4. DESARROLLO DEL SISTEMA PROPUESTO

Firebase en una plataforma de Google para desarrollar aplicaciones en sistemas Andorid, IOS y Web, proporciona un servidor que permite manejar los datos de las aplicaciones en tiempo real. Para este trabajo se utilizaron las herramientas de Firebase *Realtime Database* para gestionar la base de datos y *Hosting* para el alojamiento del aplicativo web, que controla las variables y monitorea los eventos de la casa.

4.1. BASE DE DATOS EN TIEMPO REAL

Realtime Database es una base de datos NoSQL que permite almacenar y sincronizar datos en tiempo real en formato JSON (JavaScript Object Notation), Firebase proporciona un SDK (Software Development Kit) para desarrollar las aplicaciones sin necesidad de utilizar servidores, además permite manejar los eventos de lado del *Backend* en la aplicación activados por la base de datos. A continuación, se muestran los pasos usados para crear la base de datos de donde se monitorean las variables de la casa.

- **CREAR LA CUENTA:** Firebase es una herramienta de Google por lo que es necesario tener una cuenta de correo de Gmail para ingresar. En la página principal de la página esta un botón *Ir a la consola,* al dar clic en este botón se accede al panel de control de la cuenta.
- **CREAR EL PROYECTO:** Firebase permite crear varios proyectos lo que permite mejor organización, aquí se ingresa el nombre del proyecto, nen este trabajo se le dio el nombre de *Firerasp,* ya que se trabajara Firebase y el controlador Raspberry. En la Figura 6 y se muestra el panel de control de firerasp y la opción de agregar proyecto.

Figura 6. Panel de control de Firebase para crear un proyecto. Fuente: Autor.

- **CREAR LA BASE DE DATOS:** En el panel de control del proyecto, en la sección *DataBase, se creó una base de datos Realtime con la estructura JSON mostrada en la figura 7.* El nodo *home* representa la casa y dentro de este nodo se encuentran

los nodos secundarios, *dooors* > *statusDoor1* este nodo guarda el estado del sensor que detecta si la puerta se encuentra cerrada o abierta*, lights* > *light1* este nodo guarda el estado que controla la bombilla*, locks* > *doorLock1* este nodo controla el seguro de la puerta, *passwords* > *passDoor1* este nodo guarda la contraseña para permitir desactivar el seguro de la puerta. El nodo *rpio* representa la casa la tarjeta controladora Raspberry pi y dentro de este se encuentra un solo nodo llamado *state* para vigilar si la tarjeta se encuentra en línea.

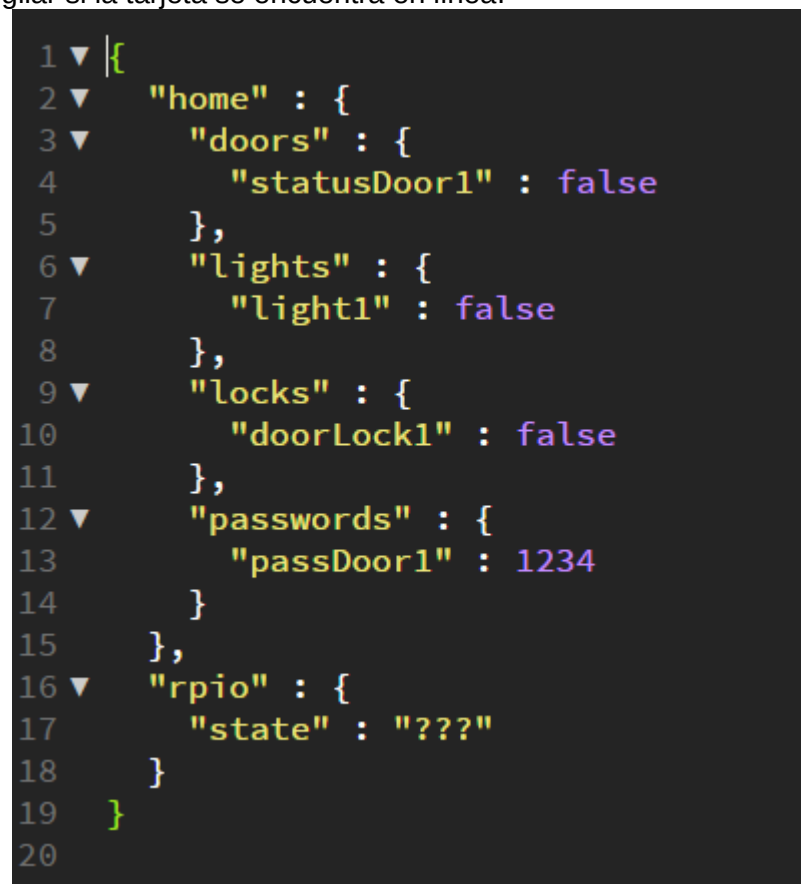


Figura 7. Estructura JSON para la base de datos. Fuente: Autor.

4.2. SERVIDOR LOCAL FIREBASE

Un servidor web local proporciona herramientas para que las partes de la aplicación que utiliza el SDK de Firebase JavaScript que necesitan estar asociados a un servidor y permite que funcione correctamente. Firebase CLI (Command Line Interface) proporciona una serie de herramientas para administras proyectos de Firebase, las herramientas utilizadas para este trabajo fueron: generar un servidor local para pruebas de la aplicación y el despliegue de la aplicación en el hosting del proyecto Firebase que se creó.

Para instalar Firebase CLI se debe ejecutar el siguiente código desde el CMD(Command) de la computadora:

- npm install -g firebase-tools

Terminado el proceso de instalación, están disponibles una serie de comandos de Firebase desde el CMD del equipo. Para poder tener acceso a los proyectos y enlazar el equipo local con Firebase, se inició sesión con el siguiente comando:

- firebase login

Con la sesión iniciada se sincronizo el proyecto de Firebase con el directorio local del equipo donde se trabajó el aplicativo web, primero se inició el CMD en el directorio local del proyecto con el siguiente comando:

- cd /ruta/proyecto/local

una vez se ubicó el CMD en el directorio del proyecto, se ejecutó el siguiente comando para sincronizar el directorio con el proyecto de Firebase:

- firebase init

Este comando permitió seleccionar la configurar del proyecto para las herramientas de Firebase que se usaron, en este caso *Realtime Database y Hosting.* Terminada la configuración se creó un archivo un archivo llamado *firebase.json* con la configuración del proyecto predeterminado al directorio local, este archivo se muestra en la Figura 8.

.

Figura 8. Archivo Firebase de configuración para el proyecto sincronizado. Fuente: Autor.

Para arrancar el servido local en el directorio del proyecto se ejecutó el siguiente comando:

- *firebase serve*

Este comando abre un puerto en la dirección IP (Internet Protocol) local del proyecto para servir los archivos que se encuentren allí utilizando un entorno de servidor local.

4.3. REGLAS PARA LA BASE DE DATOS

Las reglas de la base de datos permiten la conexión a la base de datos para poder escribir y leer los datos. Las reglas pueden ser muy específicas o generales dependiendo de la aplicación, estas reglas una sintaxis parecida a JavasScript con un formato JSON, estas reglas se encuentran en el archivo llamado *database.rules.json* en el directorio local del proyecto sincronizado. Para este trabajo de fin académico se especificaron de acceso libre a todas las aplicaciones que cuenten con las credenciales del proyecto Firebase. En la Figura 9 se muestra la estructura de las reglas cargadas para el proyecto de este trabajo.

.

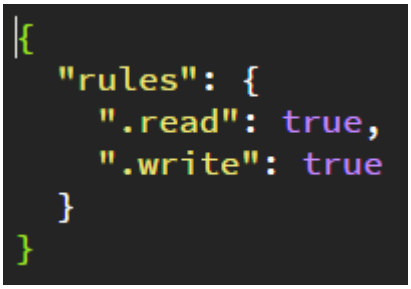


Figura 9. Configuración de las reglas para el proyecto. Fuente: Autor.

Por defecto al crear el proyecto de Firebase este no permite la lectura ni escritura de datos, por esto es necesario actualizar el archivo con las reglas que se encuentra en el proyecto Firebase. Para subir los archivos del directorio local al servidor remoto del proyecto en Firebase se ejecutó el siguiente comando:

- firebase deploy

Este comando verifica que los archivos de configuración para el proyecto y las reglas estén bien formados y los sobrescribe, también toma los archivos que se encuentren en la ruta */public* del directorio local y los subiendolos al *Hosting.*

4.4. DESARROLLO DEL APLICATIVO WEB

El aplicativo web se desarrolló mediante HTML y CSS para la interfaz gráfica que se le muestra al usuario y JavaScript para parte del *Backend* encargado de controla y mostrar los eventos sobre la base de datos. Para la programación se utilizó el editor de texto *Brackets,* este editor proporciona un IDE (*Integrated Development Environment)* de desarrollo ligero y poderoso para creación de proyectos web.

La interfaz del aplicativo web permite encender y apagar una bobilla y a su misma ves monitorear el estado de esta en caso que sea encendida manualmente desde la casa, también vigila el estado de la puerta y controla el seguro de esta. En la Figura 10 se muestra una imagen de la interfaz gráfica para el aplicativo web.

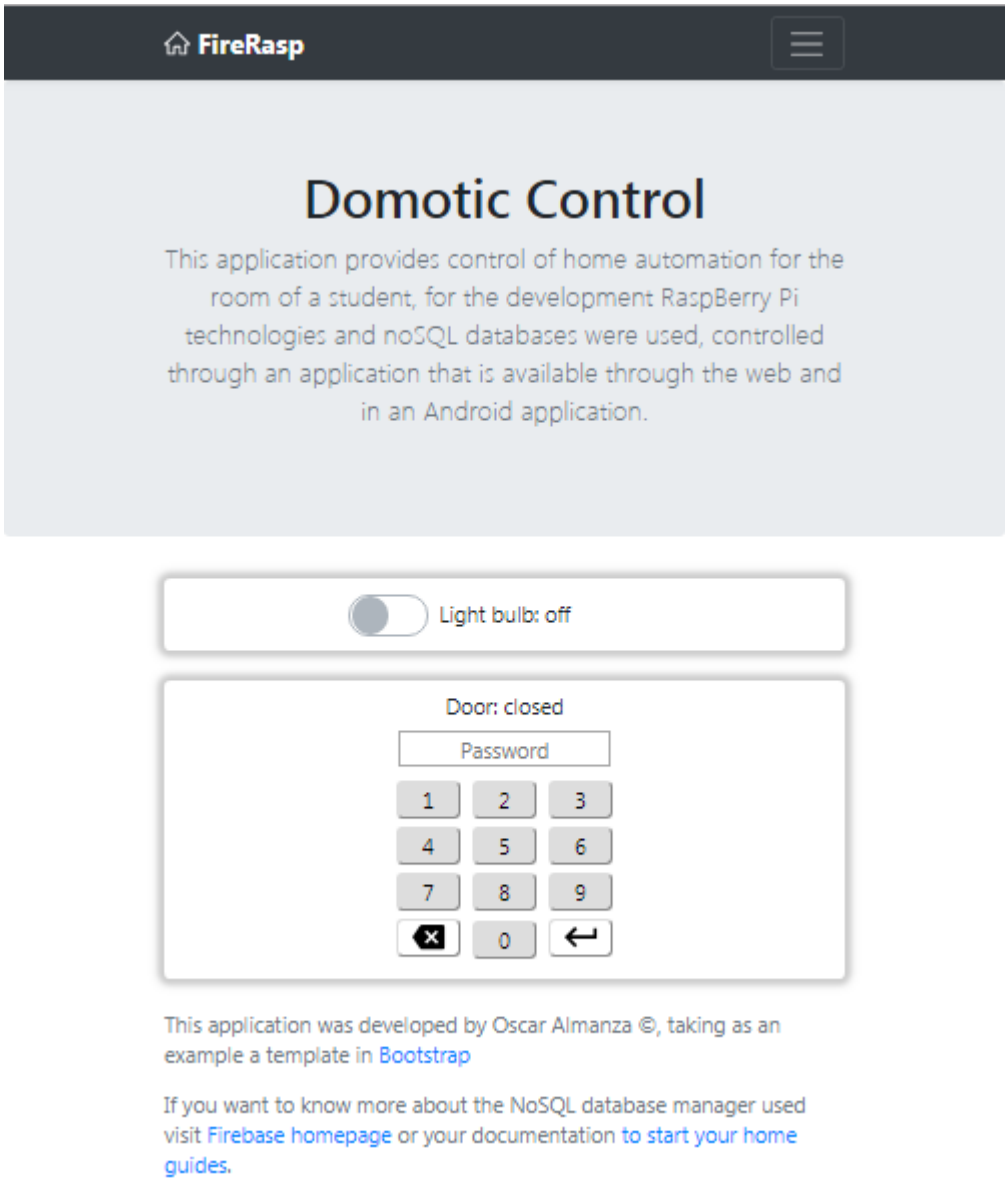


Figura 10. Interfaz gráfica para el aplicativo web..Fuente : Autor.

Un agente MTConnect es el encargado de recibir los datos del adaptador y formatearlos en un documento XML para set entregados a los clientes web por medio del protocolo HTTP. Para este trabajo se utilizó el MTConnect C++ Agent en la versión 1.4.0 el cual funciona como un servicio en y provee un desarrollo completo del servidor HTTP que solicita el estándar MTConnect, instalado el agente es necesario solamente especificar la descripción XML del dispositivo y la dirección IP del dispositivo donde solicitara los datos.

4.4.1. Configuración del aplicativo web con Firebase

El SDK Firebase se importó mediante el CDN (Content Delivery Network) al archivo *index.html* del aplicativo para poder utilizar los métodos de JavaScript para la escritura y lectura de la base de datos, para conectarse a la base de datos primero se inicializo un objeto de configuración de Firebase con las credenciales del proyecto y una variable la cual contiene un objeto de escucha asíncrono *firebase.database.ref()* apuntado a la referencia del nodo *home*, esta parte se encuentra en el archivo *startFirebase.js* y se muestra en la Figura 11.

```
// Credenciales de Firebase
var firebaseConfig = {
    apiKey: "AIzaSyDBmm-DIh7gmtvYgC6VC-4VYNeUVD4ZeSY",
    authDomain: "firerasp-domotic.firebaseapp.com",
    databaseURL: "https://firerasp-domotic.firebaseio.com",
    projectId: "firerasp-domotic",
    storageBucket: "firerasp-domotic.appspot.com",
    messagingSenderId: "568808424875",
    appId: "1:568808424875:web:08da5ffb96e46b40"
};
// Incio del objeto Firebase
firebase.initializeApp(firebaseConfig);

var db = firebase.database().ref('home');

```

Figura 11. Archivo startFirebase.js con las credenciales de Firebase. Fuente: Autor.

4.4.2. Eventos de cambio en la base de datos

Firebase proporciona un escucha para los cambios en la base de datos, se utilizó este método para actualizar las etiquetas en la interfaz del aplicativo que muestran los estados para la bombilla y la puerta.

El parámetro *data* devuelto por la función *on()* tiene los valores de la base de datos que cambiaron y se obtienen con el método *val()* y los apuntadores a los nodos que se van a evaluar, el script para esta función se muestra en la Figura 12 y se encuentra en el archivo *connectionFunctionFB.js*

```
// último estado
db.on('value', function (data) {
    //bombilla
    let light1 = data.val().lights.light1;
    document.getElementById("customSwitch1").checked = light1;
    let doorState = document.getElementById("doorState");
    if (light1) {
        doorState.innerHTML = "on";
    } else {
        doorState.innerHTML = "off";
    }
    //puerta
    let statusDoor1 = data.val().doors.statusDoor1;
    let pageDoorStatus = document.getElementById("doorValue");
    if (statusDoor1 == true) {
        pageDoorStatus.innerHTML = "open";
    } else {
        pageDoorStatus.innerHTML = "closed";
    }
});
```

Figura 12. Script para evaluar los cambios en la base de datos. Fuente: Autor.

4.4.3. Actualizar estados en la base de datos

Para escribir en la base de datos desde la aplicación web se utilizaron los métodos de Firebase *update,* a este método se le paso la estructura del nodo que con el valor actualizar. En la Figura 13, en la parte superior se muestra el script que se encuentra en el archivo *connectionFunctionsFB.js* utilizado para actualizar el nodo de la bombilla de pendiendo del valor de la etiqueta con el id *customSwicht* en la interfaz, de esta misma forma se actualiza el valor del seguro de la puerta en el archivo *keyboard.js* en la parte inferior de la imagen.

```
// setear valores
function setLight1() { // funcion para la bombilla
    let stateLights = document.getElementById("customSwitch1").checked;
    db.update({
        lights: {
            light1: stateLights
        }
    });
}
```

connectionsFunctionsFB.js

```
db.update({
    locks: {
        doorLock1: true
    }
});
```

keyboard.js

Figura 13. Script para actualizar el estado de la bombilla y seguro de la puerta n la base de datos. Fuente: Autor.

4.4.4. Lectura de estados de la base datos

La lectura de la contraseña en base de datos se realizó utilizando el método *once* de Firebase para JavaScript, este método permite leer el estado de un dato sin la necesidad de detectar cambios para poder activarse y solo se ejecuta una sola vez luego de ser utilizado. En la Figura 14 se muestra el utilizado en el archivo *keyboard.js* que permite leer el valor del dato *passDoor1* en la base de datos y compararla con la introducida por el

usuario, en el caso de ser iguales desactivar el seguro de la puerta en la base de datos, también a su vez inicia un contador de tiempo por medio del método *setInterval de javaScript* que activa el seguro de la puerta luego de 10s nuevamente, el script para este contador se muestra en la Figura 15.

```
                if (textField.value == passwordDoor1) {
                    db.update({
                        locks: {
                            doorLock1: true
                        }
                    });
                    let timeDoor = document.getElementById('timeDoor');
                    timeDoor.style.display = 'block';
                    timeDoor.innerHTML = 10;
                    timeDoorOpen();
                } else {
                    alert('Incorrect password');
                }
```

Figura 14. Lectura del valor para la contraseña de la puerta en la base de datos y comparación con la introducida por el usuario en la interfaz del aplicativo. Fuente: Autor.

```
function timeDoor() {
    let seg = document.getElementById('timeDoor').textContent;
    if (seg > 0) {
        document.getElementById('timeDoor').innerHTML = seg - 1;
        db.once('value').then(function (snapshot) {
            let statusDoor1 = snapshot.val().doors.statusDoor1;
            if (statusDoor1 == true) {
                document.getElementById('passwordField').value = "";
                document.getElementById('doorValue').innerHTML = "open";
                clearInterval(timeCloseDoor);
                document.getElementById('timeDoor').style.display = "none";
            }
        });
    } else {
        db.update({
            locks: {
                doorLock1: false
            }
        });
        document.getElementById('timeDoor').style.display = "none";
    }
}

```

Figura 15. Script para el contador que activa el seguro de la puerta nuevamente. Fuente: Autor.

4.5. FUNCIONAMIENTO DE LA INTERFAZ

La interfaz diseñada permite controlar y monitorear el estado de la bombilla y la puerta de la casa, a continuación, se explica en detalle los paneles de control en la aplicación que permiten este proceso.

- **PANEL DE CONTROL PARA LA BOMBILLA:** El panel de la bombilla se muestra en la Figura 16, permite controlar el estado de la bombilla en un estado On-Off a través de la etiqueta de tipo caja de chequeo en forma de punto deslizante, el cual ejecuta la función mostrada en la Figura 8 parte superior, actualizando el estado de esta en base de datos dependiendo del valor de la etiqueta HTML, el monitorea

miento del estado de la bombilla se lleva con esta misma etiqueta mostrando su valor en la parte derecha del panel. Ya sea manualmente o desde la aplicación esta etiqueta siempre toma el valor del estado del dato en la Firebase.

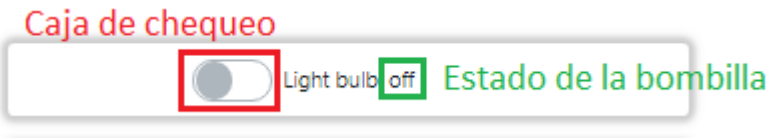


Figura 16. Panel de control y monitoreo para la bombilla en la interfaz. Fuente : Autor.

- **PANEL DE CONTROL PARA LA PUERTA:** En la Figura 17 se muestra el panel de control para la puerta de la casa, el panel del control para la puerta se divide en tres partes. La mostrada en color rojo que monitorea el estado de la puerta y es susceptible a cualquier cambio del dato *statusDoor1* de la base de datos que corresponde al sensor de la puerta, vigilando si esta se encuentra abierta o cerrada. La mostrada en color verde que corresponde al campo de texto que captura la contraseña para ser comparada con la almacenada en la base de datos como se muestra en la Figura 14. Por último, está la mostrada en color azul que es el teclado para digitar la contraseña que captura el campo de texto.

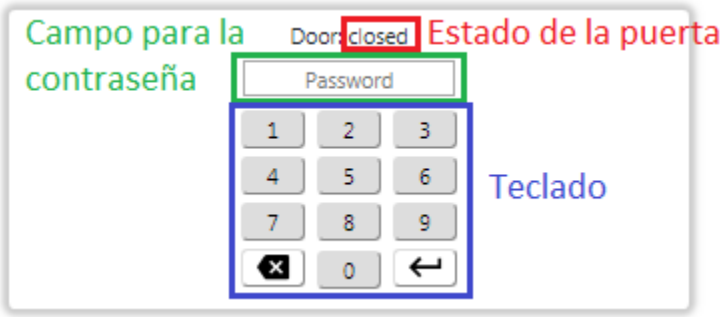


Figura 17. Panel de control y monitoreo para la puerta en la interfaz. Fuente: Autor.

4.6. DESPLIEGUE DE LA APLICACIÓN

El despliegue de la aplicación web significa subir los archivos de esta al servidor remoto para poder conectarse a esta desde cualquier parte a través de internet. El CLI de Firebase proporciona un comando deploy que permite ejecutar esta tarea de forma rápida y sencilla, los archivos guardados en la ruta del directorio local /public se suben como archivos al Hosting de Firebase.

- firebase deploy

La herramienta Hosting de Firebase proporciona un alojamiento en la nube para la aplicación web, proporciona un subdominio dentro de Firebase que apunta a la aplicación del proyecto, este dominio puede cambiarse por un dominio privado si se desea. En la Figura 18 se muestra el despliegue de la aplicación en la el Hosting Firebase.

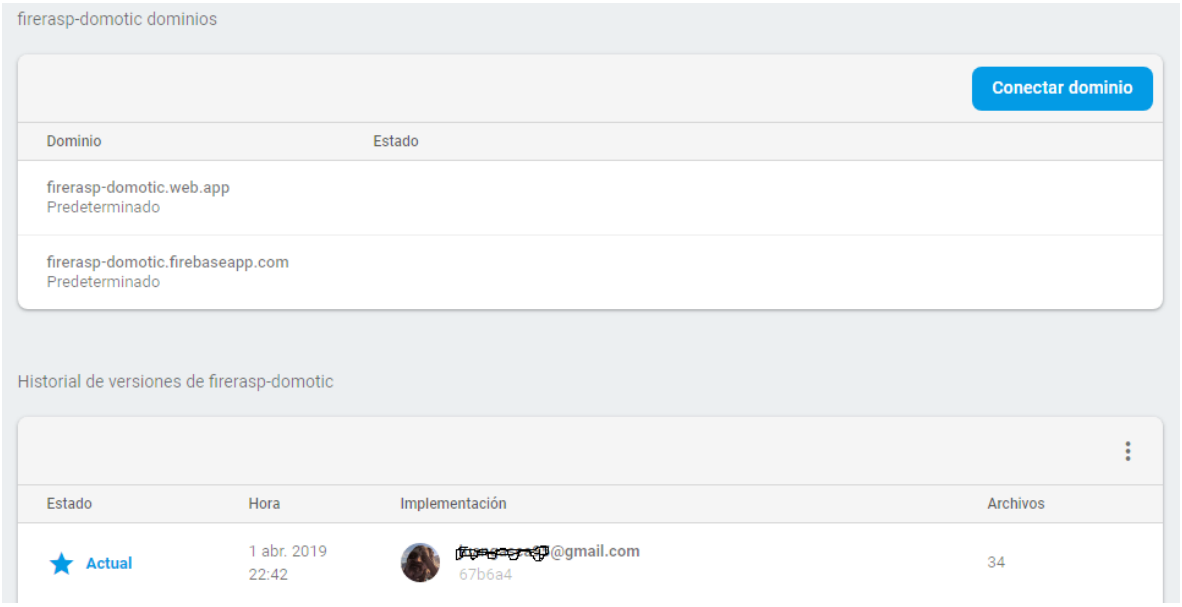


Figura 18. Despliegue de la aplicación en el servidor web. Fuente: Autor.

4.7. CIRCUITOS ELECTRÓNICOS DE ACOPLE

Se hizo necesario el desarrollo de un circuito electrónico de acople para el control de los sensores y actuadores de la casa, este circuito utiliza un auto enclave para el encendido y apagado de la bombilla para que esta se pueda apagar y encender manualmente desde la casa como por medio del controlador Raspberry pi, el auto enclave funciona por utilizando dos relés de estado sólido, el relé 1 encienda la bombilla al recibir el pulso en el pin de entrada, un cargador en paralelo con la bombilla generando la señal de auto enclave para el relé 1. El circuito de la bombilla se ilustra en la Figura 19.

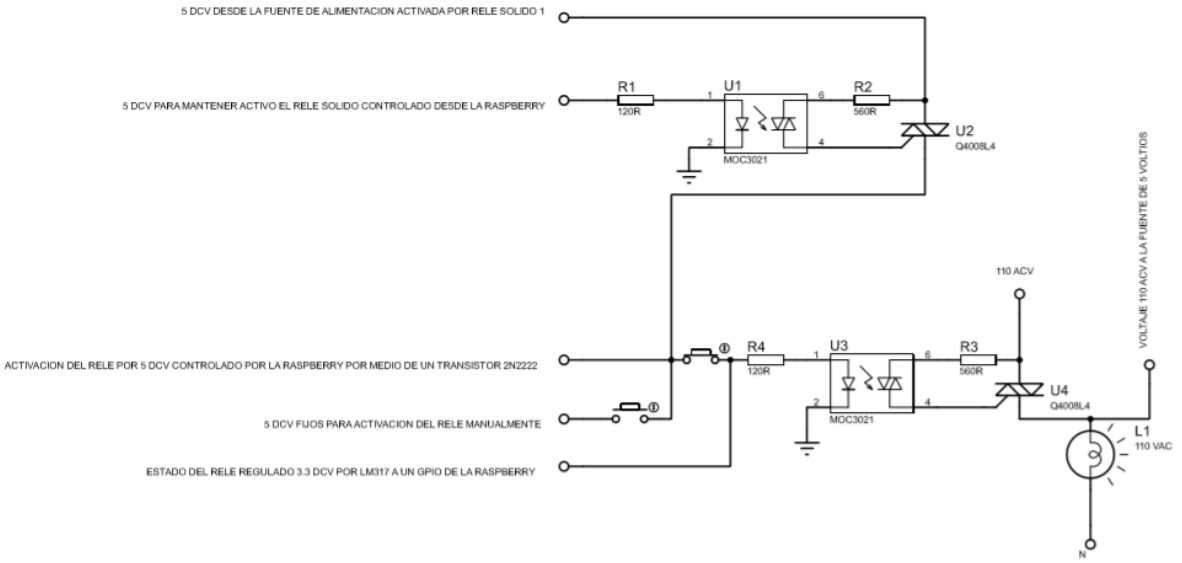


Figura 19. Circuito de acople para el control de la bombilla. Fuente: Autor.

El control de la puerta se realizó mediante el circuito de acople mostrado en la Figura 20, este circuito funciona utilizando un relé de estado sólido, este relé acciona la bobina activando o desactivando el seguro de la puerta. La detección del estado de la puerta (abierta-cerrada) se utilizó un sensor magnético, el sensor detecta una pieza metálica de la puerta al entrar en el marco detectando el estado.

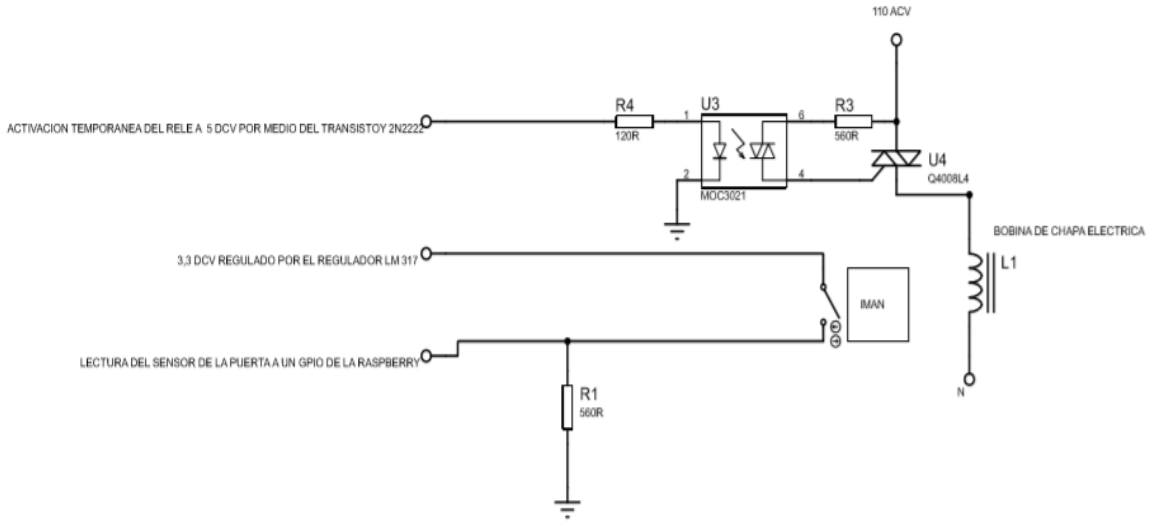
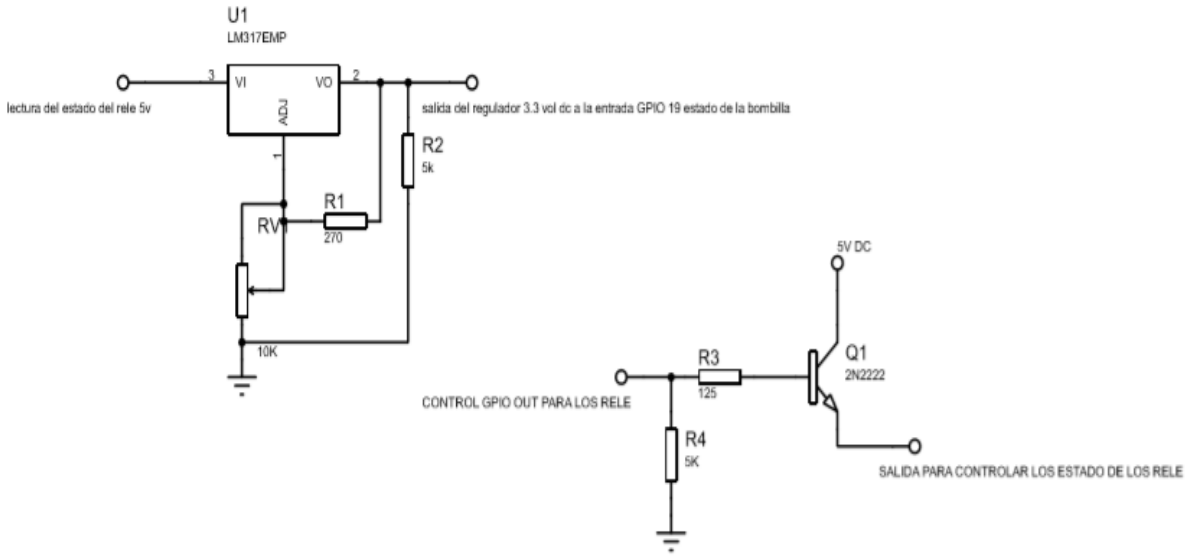


Figura 20. Circuito de acople para el control de la de la puerta. Fuente: Autor.



4.8. PROGRAMACIÓN DEL SISTEMA EMBEBIDO

El sistema embebido utilizado para controla los sensores y actuadores de la casa fue la Rasberry pi modelo B+, esta tarjeta cuenta con un procesador ARM Cortex-A53 con 1GB de memoria RAM, cuenta con puerto RJ45 para una conexión LAN a una red ethernet. El sistema operativo que se instaló en esta tarjeta fue el Raspbian en su versión 2018-03-20,

Raspbian es el sistema operativo oficial para las tarjetas Raspberry, el sistema operativo trae preinstalado con Python.

4.8.1. Script de control en Python

El algoritmo o script encargado de leer y accionar los sensores y actuadores de la casa se desarrolló utilizando el lenguaje Python, este lenguaje es de alto nivel y de propósito general para programación caracterizándose por la facilidad de legibilidad, es un lenguaje orientado a objetos haciendo excelente para el desarrollo de proyectos a pequeña y gran escala. El SDK de Firebase se puede implementar utilizando Python lo que llevo a la selección de este como lenguaje para el control de la electrónica en la casa.

El código en Python desarrollado se ejecuta al iniciar la Raspberry en forma de servicio lo que asegura que a la falta energía para la alimentación de la tarjeta o después de un reinicio la lectura y control de la electrónica funcione de nuevo, internamente el servicio lee y escribe en los pines GPIO de la tarjeta estipulados permitiendo el control de la electrónica asociada constantemente debido a un ciclo que se repite, con el fin de detectar los cambios en la base de y actualizar las salidas en los pines o viceversa, leer las entrada de los pines que cambiaron para actualizar el estado en la base de datos.

En la Tabla 3 se muestra los comandos utilizados para instalar las herramientas necesarias de Firebase SDK y conectarse a la base de datos, estos comandos se ejecutaron desde el CMD de las de Rasbian en la Raspberry.Tabla 3. Comandos para la instalación de los paquetes utilizados en Python

COMANDO	DESCRIPCIÓN
$ sudo wget https://bootstrap.pypa.io/get-pip.py $ sudo python get-pip.py	Instalar PIP, para la administración de paquetes en python
$ sudo pip install firebase-admin	Instalar el paquete de Firebase para python
$ sudo apt install python-gpiozero	Instalar paquete para control de los pines GPIO
$ sudo nano test_fire.py	Testear la instalación de Firebase

Tabla 2. Comandos CMD para instalar SDK Firebase para Python.

4.8.2. Librerías en Python

En la Figura 21 se muestra las importaciones utilizadas para el código de control en Python. la librería firebase_admin es la que carga todas las herramientas de Firebase SDK para la conexión con la base de datos, RPi.GPIO se utilizó para manejar los pines digitales de entrada y salida de la tarjeta Raspberry que controlan la electrónica.

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
import RPi.GPIO as GPIO
from time import sleep
```

Figura 21. Librerías importadas en el código Python. Fuente: Autor.

4.8.3. Credenciales de Firebase para Python

Las credenciales a igual que en la aplicación web necesitan ser cargadas en el objeto Firebase para que permita la conexión con la base de datos. En el script Python la variable cred importa las credenciales que se encuentran en un archivo externo en la ruta donde se encuentra el script, están credenciales se obtienen de Firebase en la siguiente ruta Configuración>Usuarios y permisos>Cuentas de servicio>Generar clave privada. El archivo con las claves privadas tiene la apariencia mostrada en la Figura 22


```
{
  "type": "service_account",
  "project_id": "firerasp-domotic",
  "private_key_id": "4b6db6d88e4806feec4ffd6341f5d4e52cd2700b",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEugIBADANBgkqhkiG9w0BAQ
  "client_email": "firebase-adminsdk-4z5t2@firerasp-domotic.iam.gservicea
  "client_id": "112292029360003897294",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/ce
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x
}
```


Figura 22. Archivo de credenciales para la conexión desde Python. Fuente: Autor.

La carga de las credenciales se muestra en la Figura 23. donde la variable cred carga las credenciales del archivo externo donde se guardaron, luego se inició el objeto Firebase fire_app pasando la variable cred y el URL de la base de datos

```
# cargar las credenciales de firebase
cred = credentials.Certificate("serviceAccountKey.json")

# Initialize the app with a service account, granting admin privileges
fire_app = firebase_admin.initialize_app(cred, {
        'databaseURL':'https://firerasp-domotic.firebaseio.com/'
    })
```

Figura 23. Cargar las credenciales e iniciar el objeto Firebase en Python. Fuente: Autor

4.8.4. Configuración de los pines GPIO

Los pines GPIO de la tarjeta se configuraron como se muestra en la Figura 31, el pin GPIO-24 se configuro como salida y controla el pulso para encender la bombilla, el pin GPIO-20 funciona también como salida uy controla el pulso para apagar la bombilla, el pin GPIO-19 se utiliza como entrada para detectar si la bombilla esta encendida o apagada, el pin GPIO-6 controla es una salida para manejar el seguro de la puerta y para detectar el estado de la puerta se configuro el pin GPIO-12 como entrada

```
# Congiguracion de los pines GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(26, GPIO.OUT) # pin para la bombilla ON
GPIO.setup(20, GPIO.OUT) # pin para la bombilla OFF
GPIO.setup(19, GPIO.IN)  # pin para el estado de la bombilla
GPIO.setup(6, GPIO.OUT)  # pin para el seguro de la puerta
GPIO.setup(12, GPIO.IN)  # pin para el sensor de la puerta
```

Figura 24. Configuración para los pines GPIO utilizado. Fuente: Autor.

4.8.5. Control de eventos para base de datos en la Raspberry

Los eventos detectados en la base de datos y la tarjeta son manejados por un bucle repetitivo que evalúa constantemente los cambios en las entradas GPIO de la tarjeta para actualizar su valor en la base de datos, a su vez también pregunta por los cambios que ocurren n la base de datos y que afectan el estado de las salidas GPIO de la tarjeta. En la Figura 25, se muestra el código interno del bucle repetitivo encargado de hacer lo anteriormente mencionado.

```
# estado anterior de la bombilla
temp_statusLight1 = GPIO.input(19)

# bucle repetitivo
while True:
    # lectura del estado para la bombilla
    light1 = db.reference('home/lights/light1').get()
    statusLight1 = GPIO.input(19)

    #cambiar estado de la bombilla si cambia en firebase
    if light1 == True and statusLight1 == False:
        print('Light 1 ON:')
        GPIO.output(26,1)
        sleep(0.05)
        GPIO.output(26,0)
    elif light1 == False and statusLight1 == True:
        print('light 1 OFF')
        GPIO.output(20,1)
        sleep(0.05)
        GPIO.output(20,0)

    #cambiar estado de la bombilla si cambia manualmente
    if temp_statusLight1 == False and statusLight1 == True:
        print('Light 1 ON:')
        db.reference('home/lights/light1').set(True)
        temp_statusLight1 == statusLight1
    elif temp_statusLight1 == True and statusLight1 == False:
        print('Light 1 OFF:')
        db.reference('home/lights/light1').set(False)
        temp_statusLight1 == statusLight1

    # lectura del estado para el seguro y el sensor de la puerta
    lockDoor1 = db.reference('home/locks/lockDoor1').get()
    statusDoor1 = GPIO.input(6)

    # estado del seguro de la puerta
    if lockDoor1 == True:
        GPIO.output(12,1)
    else:
        GPIO.output(12,0)

    # estado del sensor de la puerta
    if statusDoor1 == True:
        db.reference('home/locks/lockDoor1').set(True)
    else:
        db.reference('home/locks/lockDoor1').set(False)

    # pausa minima por cada ciclo
    sleep(0.1)
```

Figura 25. Script para el bucle repetitivo controla los eventos de cambio. Fuente: Autor.

creo una variable llamada temp_statusLight1 para detectar el cambio en el estado de la bombilla si este cambia manualmente y no desde la aplicación web, el bucle infinito vigila el estado del dato light1 en la Firebase evaluando esta condición para enviar el pulso de apagado o encendido dependiendo del estado actual de la bombilla en la casa, si el estado de la bombilla cambia manualmente se actualiza el estado en la base datos. La puerta se maneja de la misma forma, en cada ciclo se verifica el estado del seguro de la puerta para activar o desactivar la salida que controla este, el sensor magnético de la puesta al cambia de estado cambia en la base de datos que vigila este campo para que se muestre el estado en tiempo real en la aplicación.

4.9. CONFIGURAR SCRIPT COMO SERVICIO

Un servicio es un programa de computadora o script que se ejecuta en segundo plano al arrancar el sistema operativo. Se configuro el programa que controlar los dispositivos de acción y control de la casa para que inicie como un servicio al arrancar la Raspberry, a continuación, se muestran los pasos hechos para esta configuración en el sistema operativo Raspbian que se ejecuta en la Raspberry.

- Desde la ventana CMD de Rasbian ejecutar el siguiente código *contrap -e*
- En una línea vacia escribir el siguiente comando @reboot /home/pi/Desktop/FireRasp.py

La ruta corresponde a la ubicación del script en Python que se dearrollo para el control de la casa y la comunicación con la Firebase.

4.10. SISTEMA AUXILIAR DE ENERGIA

Se selecciono un sistema auxiliar de energía UPS (uninterruptible power supply), este sistema esta conectado a la red de energía alterna domestico de la casa y consta con un circuito de carga para una celda de baterías en su interior, este sistema auxiliar detecta cuando hay un fallo en la red domestica y proporciona un sistema de anergia de respaldo a Raspberry. En la Figura 26 se muestra la UPS selecciona como sistema de respaldo de energía y sus características eléctricas.

Figura 26. UPS para el sistema auxiliar de energía. Fuente: Autor.

5. RESULTADOS: EXPERIMENTOS Y VALIDACIÓN

Los resultados mostrados a continuación se tomados de las validaciones llevadas a cabo al sistema domótico anteriormente descrito y fueron llevadas a cabo en una casa domestica donde se montaron los dispositivos electrónicos de acción y control junto con el sistema embebido de control. Se implemento un sistema domótico centralizado, gestionado por un sistema embebido Raspberry pi y controlado remotamente a través de una aplicación web en la nube mediante Firebase. En la figura 28 se muestra una imagen con el plano arquitectónico casa haciendo énfasis en la habitación donde se llevó a cabo la instalación del sistema propuesto y la distribución del de los sensores junto con el cableado.

Como se muestra en la Figura *27*. Plano arquitectónico con la distribución de los equipos. Fuente: Autor.Figura 26Figura *1*. Sistema de control inteligente para un control domótico, está dividido en tres sectores una parte de supervisión por usuarios. Fuente: Autor. La línea roja muestra la conexión del circuito electrónico de acople para los sensores de puerta y bombilla, la línea verde indica la conexión de la Raspberry con el punto de conexión wifi para conectarse a internet y gestionar los cambios en la base de datos.

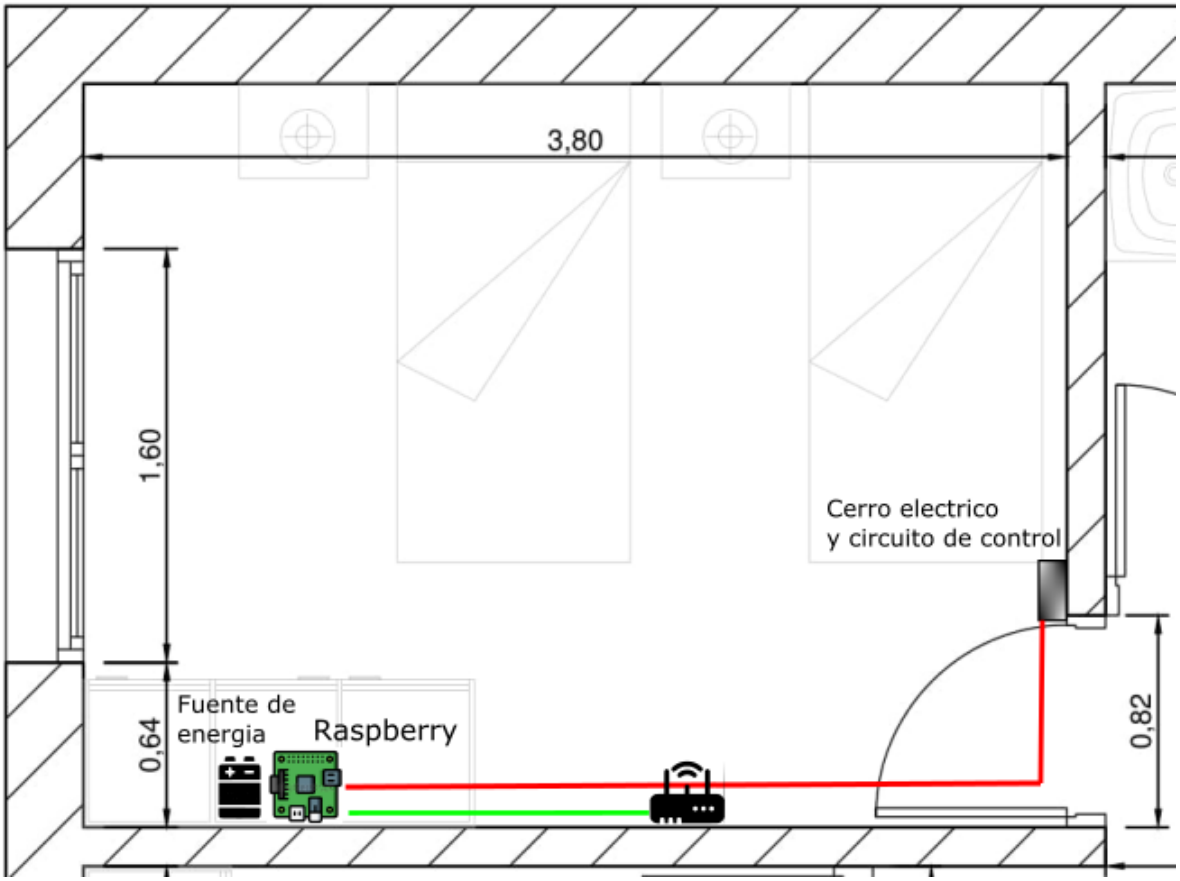


Figura 27. Plano arquitectónico con la distribución de los equipos. Fuente: Autor.

5.1. CONEXIÓN DE LA RASPBERRY A INTERNER

La conexión de la Raspberry a internet se realizó mediante el puerto ethernet R-j45que trae la tarjeta. Este cable se conectó a un punto con acceso a internet. Para comprobar la conexión mediante DHCP de la tarjeta a red de forma automática se analiza la red desde un equipo portátil conectado a esta misma red de forma inalámbrica, en la Figura 27 se muestra una captura de análisis a la red de conexión de la Raspberry.

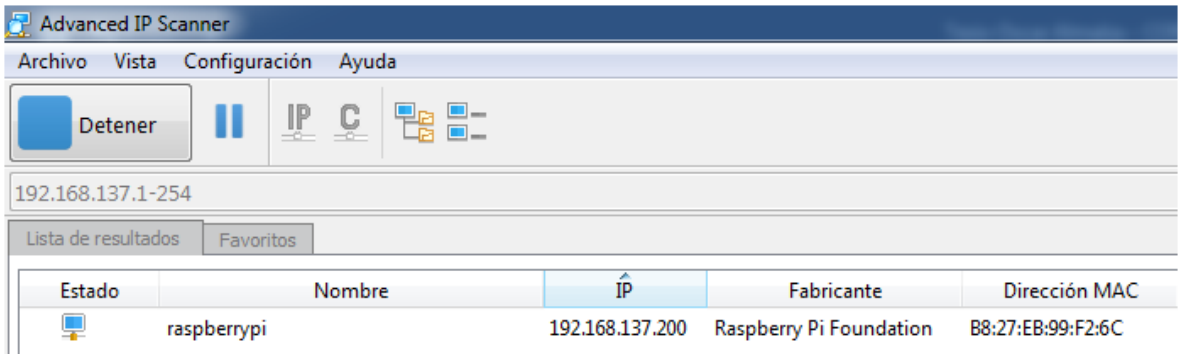


Figura 28. Escaneo de la red que verificar conexión de la Raspberry. Fuente: Autor.

La conexión física de la Raspberry se muestra en la Figura 28. El cable UTP para conexión a internet se conectó en un extremo a la Raspberry y el otro al puerto rj45 del modem con acceso a internet.

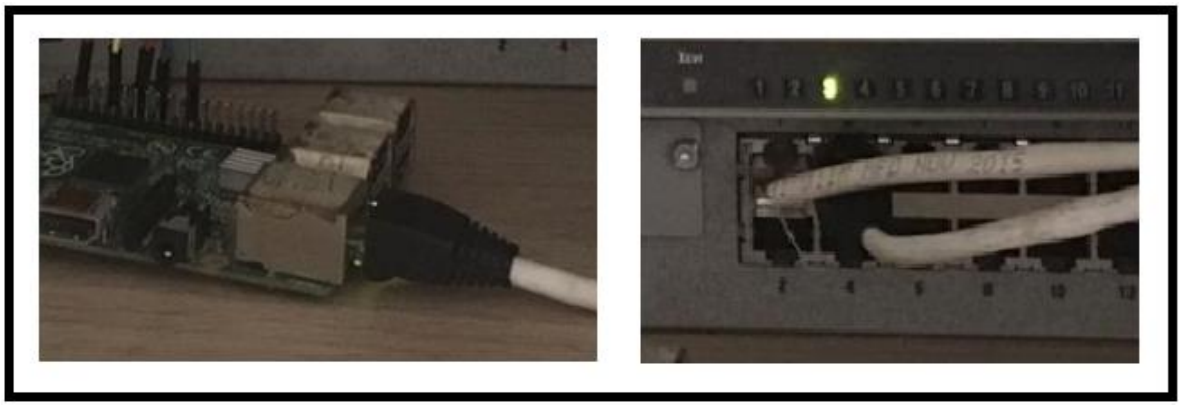


Figura 29. Conexión física de la Raspberry a punto con acceso a internet. Fuente: Autor.

5.2. PRUEBAS DE VELOCIDAD

Los test de velocidad de conexión se hicieron por medio de la página https://www.testdevelocidad.es/. La velocidad de subida de la red es de 2.28 Mbps para enviar datos a Firebase y 3.43 Mbps para recibir datos a través de la conexión a internet. Se recomienda una conexión con una velocidad no menor a 1Mbps, la velocidad de

conexión puede desestabilizar el servicio en Raspberry y que el sistema operativo Rasbian bloquee lo bloquee o desactive.

Figura 30. figura de prueba para la velocidad de conexión de la Raspbery. Fuente: Autor.

5.3. CONEXIÓN DE LOS SENSORES Y ACTUADORES

La conexión de los sensores se realizó mediante un cable UTP que conecta el circuito de acople de la electrónica con la Raspberry. El circuito de acople se instalo en la puerta junto al sensor magnético y la señal que indica si el bombillo este encendido o no, de aquí es conecto mediante el cable UTP por la canaleta hacia los pines GPIO de la Raspberry. En la Figura 30 se muestra la conexión saliente de la caja que contenedora para el circuito electrónico, este cable está conectado a los pines GPIO de la tarjeta que se encuentra en un estante como se indico en la Figura 26.

Figura 31. Conexión del circuito de acople hacia la Raspberry. Fuente: Autor.

En la figura 30 se muestran dos pulsadores que sirven para encender y apagar la bombilla de forma manual en el caso de que no se cuente con la aplicación en el momento, los cambios manualmente también son leídos por el controlador y actualizados en la base de datos para ser mostradas en la aplicación.

En la Figura 31 se muestra el sensor magnético utilizado este sensor envía la señal al controlador Raspberry mediante un cable UTP. La señal enviada por este sensor es digital enviando un uno lógico en el caso que se la puerta se encuentre cerrada y un cero lógico para el estado abierta.

Figura 32. Sensor magnético para el estado de la puerta. Fuente: Autor.

5.4. SALIDA SERIAL DE PRUEBA PARA LOS CAMBIOS EN LA RASPBERRY

En la figura 32 se muestra la salida serial del script para los cambios efectuados desde la aplicación, la salida serial se obtiene al ejecutar el script por medio del idle de Python que trae Rasbian, los cambios en la base de datos son vigilados y mostrados como impresiones, se muestra un de encendido de bombilla desde Firebase y apagado manualmente, también un cambio de seguro realizado desde la aplicación a través de la base de datos Firebase.

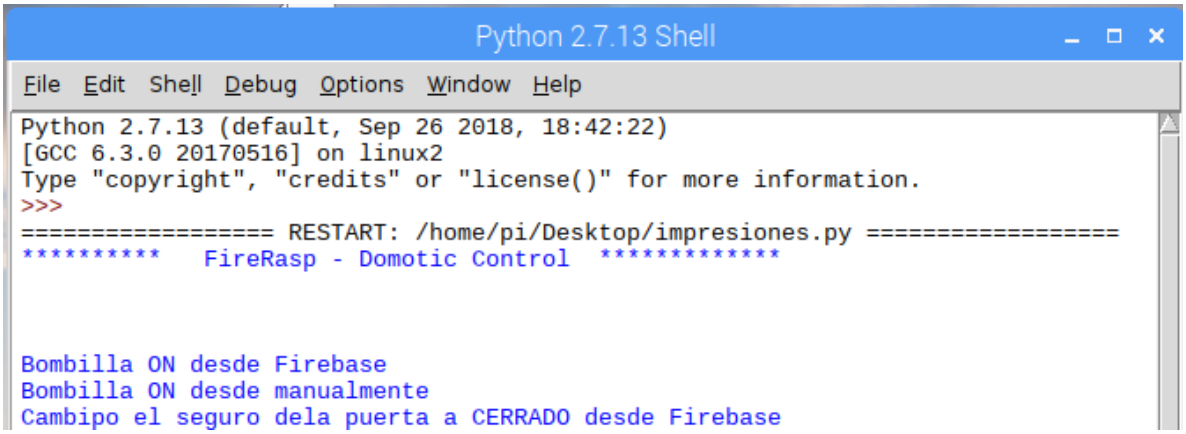


Figura 33. Salida serial para las pruebas del script de control Python. Fuente: Autor.

5.5. APLICACION Y FIREBASE

En la Figura 33 a) se muestra la aplicación web vista desde un navegador web, se muestran las opciones de control que afectan la base de datos vista en la Figura 33 b), los cambios que se efectúan son resaltados por la aplicación Firebase en amarrillo al momento del cambio.

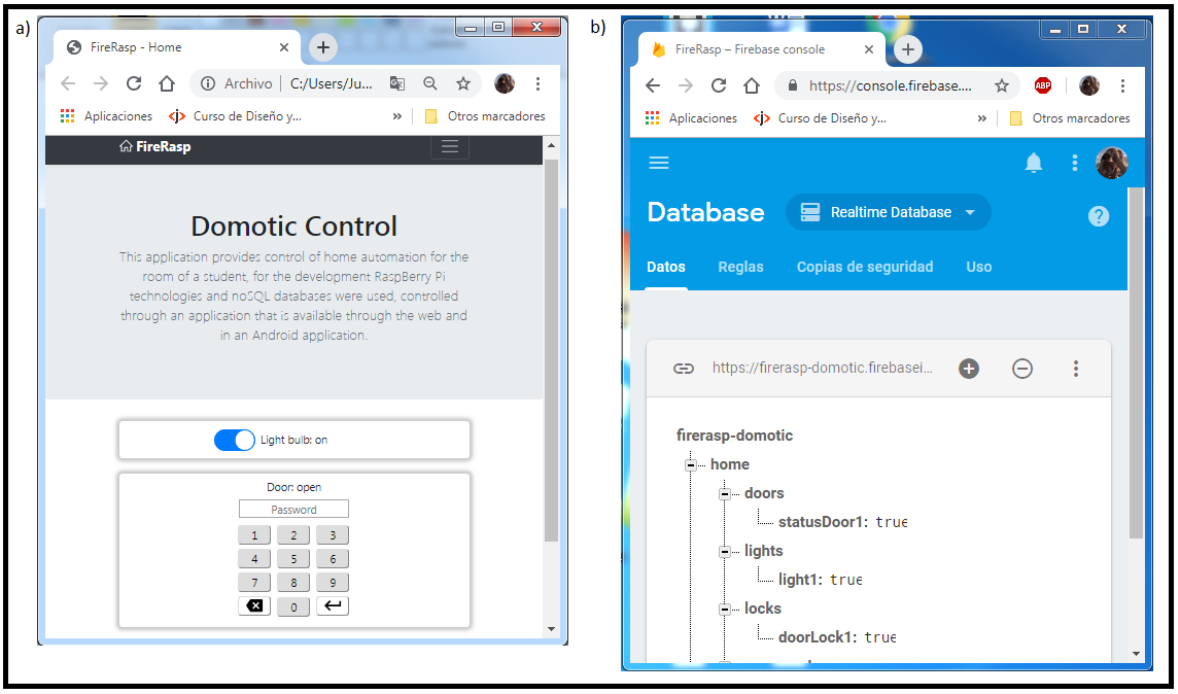


Figura 34. a) Aplicativo web ejecutada desde un navegador web. b) Pagina de Firebase mostranado parte de la estructura de la base de datos

6. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se implementó un sistema de monitoreo y control remoto para un proceso domótico, usando un sistema embebido Raspberry pi que permite la gestión de los actuadores y lectura de la sensórica mediante un script en Python. El aplicativo web desarrollado en HTML y JavaScript se comunica con el sistema embebido a través de una base de datos en la nube gestionada con la plataforma de Google Firebase, esta plataforma proporciona un completo kit de desarrollo de software que facilitan la creación y despliegues de aplicaciones tanto web como Android y IOS.

El diseño del aplicativo web se realizó utilizando las herramientas de Firebase para JavaScript, la implementación de estas herramientas facilito el desarrollo de la aplicación en la parte de la conexión con el servidor donde se encuentra la base de datos. La actualización de la aplicación según los cambios en la base de datos se ejecuta de forma rápida permitiendo un monitoreo preciso de sistema domótico controla, la velocidad de actualización es afectada por la conexión que se tenga tanto el usuario con l aplicación con la de la tarjeta Raspberry a internet. La gestión de los datos para el aplicativo web y la base de datos se realizó utilizando JavaScript y las herramientas del SDK de Firebase. Estas herramientas proporcionaron una facilidad para lo conexión con el servidor, su implementación en el script de control del aplicativo se hizo importando el CDN de Firebase para el uso de la RealTime Database. Para la parte del controlador se desarrollo un script en Python, la librería Firebase admin proporciono la conexión con la base de datos y las funciones necesarias para la gestión de los datos.

Se selección un sistema alternativo UPS (uninterruptible power supply) de energía con las características adecuadas para el funcionamiento del sistema domótico, ya que suministra la energía en caso de fallas de la red eléctrica para el funcionamiento del sistema domótico. La UPS seleccionada 250 vatios por ser el valor más comercial que cuenta con una entrada de voltaje 110 voltios que cuenta con un sistema de carga para baterías y un sistema invierte que permite generar una tensión 110 voltios al momento de la ausencia de energía de la red eléctrica normal. Se selecciono por la facilidad de adquisición de este tipo de sistemas en comparación con un sistema solar.

La validación del sistema de control se realizado operando remotamente desde un navegador web en una computadora. Detectando los cambios ocurridos en la casa, el control y supervisión del sistema por parte del aplicativo funciono completamente. Se puede decir que este tipo de sistema propuesto puede ser una buena opción para instalación en hogares pequeños con familias de bajos recursos ya que su costo es bajo. Como limitación se puede decir que el controlador implementado no es un sistema robusto por o que puede ser vulnerable ante cambios abruptos de tensión y ocasionar fallos en todo el sistema provocando que se pierda la conexión con la base de datos.

REFERENCIAS BIBLIOGRÁFICAS

[1] R. J. C. TENORIO, *DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN PARA DISPOSITIVOS MÓVILES, CON SISTEMAS OPERATIVO ANDROID, PARA EL CONTROL DE LUMINARIAS Y MONITOREO DE CONSUMO DE ENERGÍA ELÉCTRICA DE UNA VIVIENDA*, no. PROYECTO DE FACTIBILIDAD TÉCNICA, ECONÓMICA Y FINANCIERA DEL CULTIVO DE OSTRA DEL PACÍFICO EN LA PARROQUIA MANGLARALTO, CANTÓN SANTA ELENA, PROVINCIA DE SANTA ELENA. 2016.

[2] O. Rayo, C. E. Gonzalez, angelusdie@hotmail.com, and angelusdie@gmail.com, “Diseño e Implementación de una Interfaz Sobre C++ Para Control Inalámbrico de la Iluminación de un Edificio a través de Internet 2 Utilizando un Sistema Embebido,” 2016.

[3] S. Dom, “INTRODUCCIÓN A LA DOMÓTICA.”

[4] Lifelong Learning, “Ingenieria de los sistemas embebidos.,” pp. 1–19, 2011.

[5] A. Marzal and I. Gracia, “Introducción a la Programación controladores lógicos,” 2012, pp. 1–43.

[6] Universidad de Cádiz, “Practica en arduino.” 2012.

[7] JULIO ENRIQUE BARÓN ALARCÓN DIEGO FERNANDO PÉREZ OROZCO, *INVERNADERO DE RIEGO Y TEMPERATURA CONTROLADO REMOTAMENTE DESDE LA WEB BASADO EN TECNOLOGÍAS MÓVILES*, no. December. 2017.

[8] A. Marzal Varó, I. Gracia Luengo, and P. García Sevilla, *Introducción a la programación con Python 3*. 2014.

[9] Universidad de Murcia, *Manual básico de creación de páginas Web*. 2016.

[10] Rafael Menéndez-Barzanallana Asensio, “Lenguajes de programación HTML y CSS,” pp. 1–48.

[11] R. M. B. Asensio, “JavaScript,” no. 1, pp. 1–46.

[12] L. D. E. Matriz, “JSON.” pp. 1–12, 2008.

[13] M. Castellote Garcia, “Desarrollo de una aplicación Android de apuestas utilizando Firebase para la sincronización de datos,” *Univ. JAUME I*, no. 1, pp. 1–92, 2017.

ANEXSOS

Script de control en Python A.

En este anexo se muestra el código realizado en Python y que controla y monitorea el estado de sensores y actuadores en la Raspberry pi.

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
import RPi.GPIO as GPIO
from time import sleep

# cargar las credenciales de firebase
cred = credentials.Certificate("serviceAccountKey.json")

# Initialize the app with a service account, granting admin privileges
fire_app = firebase_admin.initialize_app(cred, {
        'databaseURL':'https://firerasp-domotic.firebaseio.com/'
    })

# Congiguracion de los pines GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(26, GPIO.OUT) # pin para la bombilla ON
GPIO.setup(20, GPIO.OUT) # pin para la bombilla OFF
GPIO.setup(19, GPIO.IN)  # pin para el estado de la bombilla
GPIO.setup(6, GPIO.OUT)  # pin para el seguro de la puerta
GPIO.setup(12, GPIO.IN)  # pin para el sensor de la puerta

print ("**********   FireRasp - Domotic Control  *************")
print('\n\n')

# estado anterior de la bombilla
temp_statusLight1 = GPIO.input(19)

# bucle repetitivo
while True:
    # lectura del estado para la bombilla
    light1 = db.reference('home/lights/light1').get()
    statusLight1 = GPIO.input(19)

    #cambiar estado de la bombilla si cambia en firebase
    if light1 == True and statusLight1 == False:
        #print('Light 1 ON:')
        GPIO.output(26,1)
        sleep(0.05)
        GPIO.output(26,0)
        print('Bombilla ON desde Firebase')
    elif light1 == False and statusLight1 == True:
        #print('light 1 OFF')
        GPIO.output(20,1)
```

```
        sleep(0.05)
        GPIO.output(20,0)
        print('Bombilla OFF desde Firebase')

    #cambiar estado de la bombilla si manualmente
    if temp_statusLight1 == False and statusLight1 == True:
        #print('Light 1 ON:')
        db.reference('home/lights/light1').set(True)
        temp_statusLight1 == statusLight1
        print('Bombilla ON desde manualmente')
    elif temp_statusLight1 == True and statusLight1 == False:
        #print('Light 1 OFF:')
        db.reference('home/lights/light1').set(False)
        temp_statusLight1 == statusLight1
        print('Bombilla ONFF desde manualmente')
    # lectura del estado para el seguro y el sensor de la puerta
    lockDoor1 = db.reference('home/locks/lockDoor1').get()
    statusDoor1 = GPIO.input(6)

    # estado del seguro de la puerta
    if lockDoor1 == True:
        GPIO.output(12,1)
        print('Cambipo el seguro dela puerta a CERRADO desde
Firebase')
    else:
        GPIO.output(12,0)
        print('Cambipo el seguro dela puerta a ABIERTO desde
Firebase')

    # estado del sensor de la puerta
    if statusDoor1 == True:
        db.reference('home/locks/lockDoor1').set(True)
        print('El sensor de la puerta cambio a CERRADA')
    else:
        db.reference('home/locks/lockDoor1').set(False)
        print('El sensor de la puerta cambio a ABIERTA')

    # pausa minima por cada ciclo
    sleep(0.1)


firebase_admin.delete_app(firebase_admin.get_app())
print ('Mensaje: Se finalizo la aplicacion')
```

Script de control en Python A.

Debido al tamaño de los scripts del aplicativo web se creó un repositorio en Google Driver donde se subieron y pueden ser consultados a través de este link:

https://drive.google.com/open?id=1blnwL1DOHM7X1fk49RfulMKD2RW2Q7Po