

**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO**

TEMA:

Desarrollo de interfaces gráficas en Matlab y Android para el control de sistemas domóticos utilizando el dispositivo de comunicación inalámbrica NodeMcu Esp8266

AUTORA:

Vivar Estrella, Ana Nicole

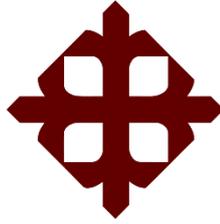
Trabajo de titulación previo a la obtención del título de
INGENIERA ELECTRÓNICA EN CONTROL Y AUTOMATISMO

TUTOR:

Palacios Meléndez, Edwin Fernando

Guayaquil, Ecuador

13 de marzo del 2018



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por **Vivar Estrella, Ana Nicole** como requerimiento para la obtención del título de **Ingeniera Electrónica en Control y Automatismo**.

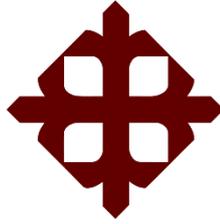
TUTOR

Palacios Meléndez, Edwin Fernando

DIRECTOR DE CARRERA

Heras Sánchez, Miguel Armando

Guayaquil, a los 13 días del mes de marzo del año 2018



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Vivar Estrella, Ana Nicole**

DECLARO QUE:

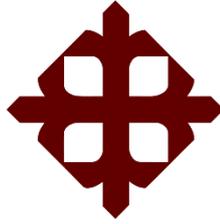
El trabajo de titulación “**Desarrollo de interfaces gráficas en Matlab y Android para el control de sistemas domóticos utilizando el dispositivo de comunicación inalámbrica NodeMcu ESP8266**” previo a la obtención del Título de **Ingeniera Electrónica en Control y Automatismo**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 13 días del mes de marzo del año 2017

AUTORA

Vivar Estrella, Ana Nicole



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO

AUTORIZACIÓN

Yo, **Vivar Estrella, Ana Nicole**

Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: “**Desarrollo de interfaces gráficas en Matlab y Android para el control de sistemas domóticos utilizando el dispositivo de comunicación inalámbrica NodeMcu ESP8266**”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 13 días del mes de marzo del año 2018

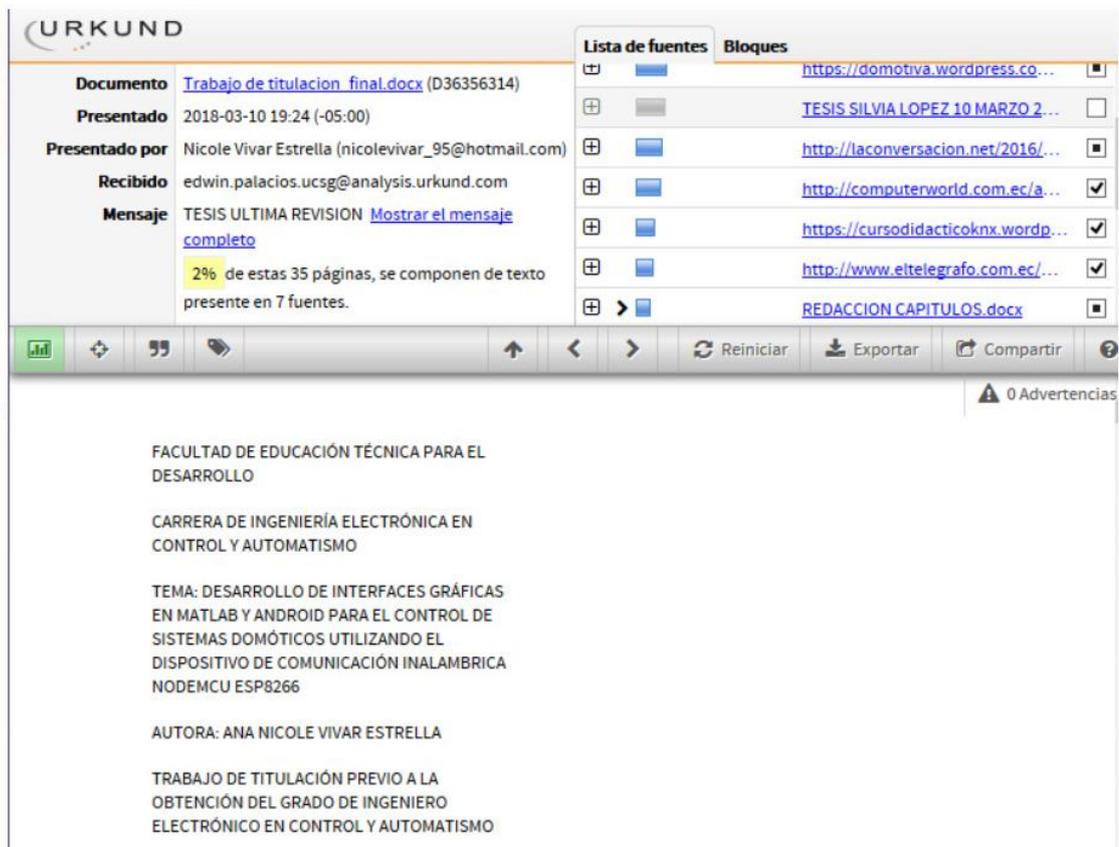
AUTORA

Vivar Estrella, Ana Nicole

REPORTE DE URKUND

Título: Desarrollo de interfaces gráficas en Matlab y Android para el control de sistemas domóticos utilizando el dispositivo de comunicación inalámbrica NodeMcu ESP8266

Autor: Ana Nicole Vivar Estrella



The screenshot displays the URKUND software interface. On the left, a sidebar shows document details: 'Documento: Trabajo de titulación final.docx (D36356314)', 'Presentado: 2018-03-10 19:24 (-05:00)', 'Presentado por: Nicole Vivar Estrella (nicolevivar_95@hotmail.com)', 'Recibido: edwin.palacios.ucsg@analysis.orkund.com', and 'Mensaje: TESIS ULTIMA REVISION [Mostrar el mensaje completo](#)'. Below this, a yellow box indicates '2% de estas 35 páginas, se componen de texto presente en 7 fuentes.' The main area on the right is titled 'Lista de fuentes' and 'Bloques', showing a list of sources with checkboxes. The sources include: 'https://domotiva.wordpress.co...', 'TESIS SILVIA LOPEZ 10 MARZO 2...', 'http://laconversacion.net/2016/...', 'http://computerworld.com.ec/a...', 'https://cursodidacticoknx.wordp...', 'http://www.eltelegrafo.com.ec/...', and 'REDACCION CAPITULOS.docx'. The bottom of the interface shows a toolbar with icons for 'Reiniciar', 'Exportar', and 'Compartir', and a notification area indicating '0 Advertencias'. The main content area displays the following text:

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y AUTOMATISMO

TEMA: DESARROLLO DE INTERFACES GRÁFICAS EN MATLAB Y ANDROID PARA EL CONTROL DE SISTEMAS DOMÓTICOS UTILIZANDO EL DISPOSITIVO DE COMUNICACIÓN INALAMBRICA NODEMCU ESP8266

AUTORA: ANA NICOLE VIVAR ESTRELLA

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO

El trabajo de titulación de la Srta. Ana Nicole Vivar Estrella, presenta un porcentaje del 2% de coincidencias con otros documentos encontrados en internet.

PALACIOS MELENDEZ, EDWIN FERNANDO
COORDINADOR DE TITULACIÓN

AGRADECIMIENTO

Quiero mostrar mi profundo agradecimiento a Dios, quién es el que me guía día a día hasta mi meta, quién me da la fuerza para perseverar en mi sueño, y me ayuda a convertirme en una excelente profesional.

Así mismo agradezco a mis padres Jorge Vivar y Ana Estrella, y a mi hermana Janise Vivar, quienes han sido mi guía constante, mi hombro, y mi soporte, quienes me educaron y me enseñaron que las cosas que quiero primero están en mi mente, que debo desearlas desde lo más profundo de mi ser y esforzarme y luchar persistentemente para hacerlas realidad, sintiendo que son mías, antes de que lo sean. Ellos quienes me enseñaron que la clave del éxito es la constancia y que el fracaso solo es una derrota temporal, una oportunidad de aprendizaje y un peldaño a subir en la escalera del éxito. Agradezco a mis tíos, Guillermo Vivar y Katty Estrella por ayudarme en el desarrollo de mi tesis. Sin ellos no hubiese sido posible.

Agradezco a mi esposo Leonardo Chávez, y a mi hijo Jorgito Chávez, quienes han sido mi motor, mi impulso para seguir adelante y alcanzar mí tan anhelado sueño del que cada día estoy más cerca.

Agradezco a todos mis compañeros, en especial a Pablo Fuentes y Joseph Peñafiel por haber sido dos ángeles en el camino, me cuidaron, me ayudaron y me acompañaron en todo momento de mi carrera, siempre estuvieron ahí.

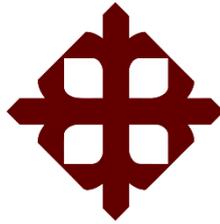
Agradezco a mis profesores, quienes con esfuerzo y dedicación se mostraron dispuestos a enseñarnos todo lo que ellos sabían, e inculcaron en nosotros el deseo de convertirnos en profesionales exitosos.

Nicole Vivar

DEDICATORIA

Dedico mi trabajo a mis padres Jorge Vivar y Ana Estrella, a mi hermana Janise Vivar, a mi hijo Jorge Chávez, a mi sobrina Liah Torres. Me apoyaron en todo el proceso de desarrollo de mi trabajo y fueron los que hicieron posible que alcanzaré este logro.

Nicole Vivar



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
AUTOMATISMO**

TRIBUNAL DE SUSTENTACIÓN

ROMERO PAZ, MANUEL DE JESÚS
DECANO

PALACIOS MELENDEZ, EDWIN FERNANDO
COORDINADOR DE TITULACIÓN

BASTIDAS, THOMAS GASPAR
OPONENTE

ÍNDICE GENERAL

ÍNDICE DE FIGURAS.....	XI
ÍNDICE DE TABLAS	XII
INTRODUCCION	2
CAPÍTULO 1: GENERALIDADES DEL PROYECTO DE TITULACIÓN.....	3
1.1 Antecedentes del Problema.....	3
1.2 Definición del Problema.....	6
1.3 Justificación del problema.....	6
1.4 Objetivos.....	7
1.4.1 Objetivo General.....	7
1.4.2 Objetivos Específicos.....	7
1.5 Hipótesis.....	8
1.6 Metodología.....	8
CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA.....	9
2.1. Domótica.....	9
2.1.1. Casa Inteligente.....	10
2.2 El Internet de las cosas.....	12
2.3 M2M.....	13
2.4 Redes de comunicación inalámbrica.....	14
2.4.1 WPAN.....	15
2.4.2 WLAN.....	16
2.4.3 WMAN.....	17
2.4.4 WWAN.....	18
2.5 Protocolos y estándares.....	19
2.5.1 Protocolo de internet TCP/IP.....	19
2.5.2 Protocolos de la domótica.....	20
2.5.2.3 KNX.....	21
2.5.2.4 X10.....	22
2.5.2.5 LonWorks.....	23
2.5.2.6 UPB.....	24
2.5.2.7 Z – Wave.....	24
2.5.2.8 Insteon.....	26
2.5.2.9 Zig Bee.....	27
2.5.2.10 Wi – Fi.....	27

2.5.2.11 Bluetooth.....	28
2.5.2.12 Thread.....	29
2.5.2.13 Apple “Home Kit”.....	30
2.6 Servidor Web.....	30
2.7 Firebase.....	32
2.8 Android Studio.....	33
2.9 Matlab.....	34
2.10 NodeMcu ESP8266.....	35
2.10.1 MCU.....	37
2.10.2 SRAM y ROM Interna.....	38
CAPÍTULO 3: DESARROLLO DEL PROYECTO.....	39
3.1 FUNCIONAMIENTO.....	39
3.2 Diseño del Hardware.....	40
3.2.2 Diseño de la maqueta.....	42
3.2.3 Diseño del circuito.....	42
3.3 Diseño del software.....	46
3.3.1 Diseño del software en Android Studio.....	47
3.4 Implementación.....	68
3.5 Pruebas y resultados.....	69
3.5.1 Pruebas y resultados de software.....	69
3.5.2 Pruebas y resultados de hardware.....	70
CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.....	71
4.1 CONCLUSIONES.....	71
4.2 RECOMENDACIONES.....	72
BIBLIOGRAFIA:.....	73
ANEXOS.....	78
Anexo A. Datasheet de los parámetros del ESP8266.....	78
Anexo B. Datasheet del transistor 2N2222.....	81
Anexo C. Datasheet de puente L293D.....	86
Anexo D. Procesador del ESP-8266.....	89
Anexo E. Imágenes de la PCB.....	90
Anexo F. Imágenes de la maqueta.....	91

ÍNDICE DE FIGURAS

Figura 1. Sistema domótico.	10
Figura 2. Imagen de Javier Pierini sobre casas inteligentes.	11
Figura 3. Ilustración del Internet de las cosas.	13
Figura 4. Clasificación de las redes inalámbricas.	15
Figura 5. Diagrama de redes WLAN en hogares	16
Figura 6. Diagrama de redes WLAN en hogares	17
Figura 7. Redes satelitales y de celular	19
Figura 8. Logo estándar KNX.....	22
Figura 9. Logo estándar LonWorks	23
Figura 10. Logo compañía Z-wave	25
Figura 11. Logo compañía INSTE-ON.....	26
Figura 12. Logo compañía ZigBee	27
Figura 13. Logo compañía Wi-Fi.....	28
Figura 14. Logo compañía Bluetooth	29
Figura 15. Logo compañía THREAD	29
Figura 16. Diagrama del funcionamiento de un servidor.....	31
Figura 17. Logo Android Studio.....	34
Figura 18. Logo de MATLAB	35
Figura 19. NodeMCU ESP8266.....	36
Figura 20. Diagrama de Bloque del esp8266.....	36
Figura 21 . Diagrama del funcionamiento del sistema.....	40
Figura 22. Diseño de la distribución en Sketch Up de la casa a escala.....	42
Figura 23. Esquemático del circuito	43
Figura 24. Diseño de la PCB.....	43
Figura 25. Circuito de iluminación	44
Figura 26. Circuito de climatización y temperatura.....	45
Figura 27. Circuito de servomotor para persianas y motor dc para puerta	46
Figura 28. Ventana de bienvenida, inicio de sesión y registro de la aplicación	47
Figura 29. Mensajes de error.....	48
Figura 30. Menú de la aplicación.....	48
Figura 31. Ventanas de las variables.....	49
Figura 32. Interacción entre Android Studio y Firebase	50
Figura 33. Interface de Matlab.....	56
Figura 34. Interface de Matlab, después de iniciar sesión.	57

ÍNDICE DE TABLAS

Tabla 1. Capas del protocolo tcp/ip	20
---	----

RESUMEN

El siguiente trabajo de titulación está ideado en el desarrollo de interfaces gráficas en Matlab y Android para el control de una vivienda inteligente donde se utiliza el dispositivo de comunicación inalámbrica Nodemcu ESP8266, donde se busca un mayor impacto en el ámbito de la domótica a nivel estudiantil en la UCSG. La idea se presenta por medio de una maqueta de una vivienda a escala, la cual permite observar la interacción entre el usuario a través de las interfaces con diferentes variables de la vivienda, como luces, puertas, persianas, y temperatura, lo novedoso es que se puede tener un control en tiempo real y además sin importar el lugar en el que se encuentre debido a que la comunicación se realiza por medio de Wi-Fi, fue necesario utilizar el servidor Firebase, el cual trabaja con una base de datos y permite enviar y recibir información desde las interfaces al sistema. En conclusión, se realizó con éxito la implementación y además abre cabida a la posibilidad de no solo realizar este tipo de controles para sistemas domóticos, sino también para distintos procesos industriales.

Palabras clave: Interfaz, Matlab, Android, NodeMcuESP 8266, domótica, Wi-Fi, Firebase.

ABSTRACT

The next accreditation project is based on the graphic interface development in Matlab and Android for the control of a smart housing where a NodeMCU ESP8266 wireless communication device is utilised, where a greater impact of home automation to a student level is wanted. The idea is presented via a scale model of a house, where we can observe first hand the interaction between the user throughout several variables of the house, such as lighting, doors, blinds, and temperature, what's new is that you can have control in real time regardless of the location you are due to the Wi-Fi based communications, it was necessary to use the Firebase server, which works with a data base and allows to send and receive information from the system interfaces. To conclude, the implementation was performed successfully and this also opens up doors to the possibility to not only perform this type of control to home automation systems, but also for different industrial processes.

Key Words: *Matlab, Android, NodeMCU ESP8266, home automation, Wi-Fi, Firebase.*

INTRODUCCION

El siguiente trabajo de investigación está orientado hacia el diseño de interfaces gráficas, las mismas que tendrían lugar mediante el uso de Matlab y un sistema operativo Android, utilizando un módulo ESP8266, brindándole al usuario una forma más cómoda de poder manejar y controlar distintas variables de su vivienda por medio de una red WI-FI.

La domótica en los últimos años ha tenido un gran avance a nivel mundial, por lo que a medida que pasa el tiempo es más probable que escuchemos acerca de tecnologías que están revolucionando la forma en que vivimos. En la publicación del periódico Heraldo, (2017) *“se estima que en 2020 habrá 50 millones de dispositivos dentro de las casas que funcionen conectados a internet”*. Además, según la red de internet Netlife, (2015) asegura que siendo el Internet la vía que conecta las operaciones de los diferentes usuarios y las acciones que los sistemas de domótica generan, sería básico mantener una *“conexión a Internet confiable, rápida y estable en un escenario de automatización de hogares”*.

Entonces de ser así, el futuro apunta hacia las casas inteligentes. Gracias a la ayuda de los avances tecnológicos en el área de la domótica existe ya un sin número de formas de conectar una vivienda a internet haciéndolo uno mismo. Entre los motivos por que los clientes prefieren hogares inteligentes es debido a la necesidad de mejorar la seguridad de la vivienda, mantenerse alerta a posibles ‘hacks’ de información, además otro de los factores importantes es el ahorro económico y energético, así como también hacer el hogar más cómodo y confortable controlando su temperatura. (Heraldo, 2017)

Se ha dividido el trabajo en dos partes generales que son hardware y software. El hardware tendrá las siguientes variables: bombillas, motor DC con reductor de velocidad, servomotores, sensor de temperatura, un ventilador DC, y un módulo wifi ESP8266. La estructura es de una maqueta de una casa a escala 1:20 simulando una vivienda real. El software se lo ha diseñado en las plataformas de Matlab y Arduino.

Capítulo 1: Generalidades del proyecto de titulación.

1.1 Antecedentes del Problema.

El termino domótica es la unión de la palabra “Domo”, proveniente del latín “domus” que significa casa, y el sufijo “tica” que hace referencia a automática, lo que significa que ‘funciona por si sola’. La domótica tiene sus inicios a partir de los años ’70, la investigación de la domótica se encontraba apenas en sus primeros pasos cuando aparecieron los productos de serie X, donde la escocesa Pico Electronics Co, se denominó como una de las empresas pionera en productos de automatización, en primera instancia, la tecnología se basaba prácticamente en brindar acciones de automatización básicas, y elementales a edificios de oficinas de trabajo. La tecnología de control automatizado para edificios se le conoce como inmótica, Estados Unidos y Japón fueron los primeros países que disfrutaron de estos sistemas ya comercializados, sin embargo los mismos solo eran capaces de controlar la temperatura de los ambientes. (Reimón, 2013)

Para los años ’80 los circuitos integrados empezaron a ser comercializados, haciendo posible más tarde el desarrollo de sistemas integrados en el ámbito doméstico de los hogares de la ciudad. Es en este momento cuando los sistemas de domótica lograron unificar dos grandes sistemas siendo estos los: eléctricos y electrónicos, comunicando distintos dispositivos de una vivienda. (Arkiplus, 2013)

Para la década de los ’90 los sistemas automatizados se expandieron a las viviendas, la domótica se encarga de todos los aspectos inteligentes de la misma, como la seguridad del hogar, la energía eléctrica, temperatura, comunicación, y otras variables. El ser humano desde sus inicios ha buscado dar comodidad a sus tareas diarias, la domótica brinda la posibilidad de satisfacer las tareas y necesidades de los seres humanos a un mínimo esfuerzo. En el año de 1998, la mujer y socia de Ad Van Berlo, Corien Van Berlo, empezó con la instalación de ‘casas inteligentes’. Corien y Ad tenían como objetivo promocionar el uso de la domótica, haciendo posible ejecutar proyectos que demostraran su uso, y empezar su estandarización. En los países bajos se desarrolló una serie de casas domóticas, solo para su demostración y observar el impacto que tendrían en la sociedad.

Todos estos proyectos culminaron entre el año 2000 y 2001. Las personas mayores se convirtieron en el mayor mercado de los sistemas de domótica, ya que ofrecía soluciones y mejoras económicas a sus necesidades. En los países bajos, en el 2001, fue tal el impacto de los sistemas de domótica que ofrecían los Van Berlo que empezaron a contratar personal sumamente calificado para que sean ellos quienes construyan las primeras casas domóticas en dicho país. (Cincodays, 2015)

En 1999, Kevin Ashton en el Auto – ID Center de MIT, hablo por primera vez sobre el Internet de las cosas, que significa básicamente la interconexión entre cosas u objetos, abriendo paso a la posibilidad de conectar todos los dispositivos de nuestro hogar a internet, obteniendo un control sobre ellos. Actualmente encontramos un sinnúmero de dispositivos en nuestro hogar conectados a internet. En el mío por ejemplo, televisores, teléfonos, cámaras de seguridad, pero se podría llegar aún más lejos. La empresa Cisco, (2013) ha creado una especie de ‘contador de objetos’ que permite saber cuántas cosas se han conectado a internet desde el 2013 hasta el 2020 alrededor del mundo.

En el año 2005 aproximadamente para el señor Rodrigo Chancusig, quien es el encargado de la coordinación de la escuela de tecnologías de la Universidad de las Américas, llegó por primera vez al Ecuador el término domótica e inmótica, junto a profesionales con postgrados obtenidos en Europa. Así con el paso del tiempo tomaron fuerza estos conceptos, y llegaron a los hogares de los ecuatorianos. Dice el señor Rodrigo a la revista EKOS, (2017) *“en la actualidad hay empresas medianas encargadas de desarrollar e instalar entornos domóticos, y forman parte de un mercado emergente, con altas probabilidades de seguir creciendo.”* (EKOS, 2017)

La empresa SayCont incursionó en el mundo de la domótica, y trajo al país increíbles creaciones de ambientes inteligentes, por ejemplo Kruger Labs, del señor Marco Velásquez de la empresa ya antes mencionada. Según Velásquez para EKOS, (2017) menciona que este laboratorio fue ideado para un entorno de inmótica. Desde el diseño del edificio hasta su desarrollo se esperó encontrar distintas soluciones y que las mismas estén enfocadas al control del edificio, que puede ser controlado desde un celular o Tablet. Se puede tener acceso a la iluminación y al sistema de seguridad en tiempo real, y asegura

Velásquez, (2017) *“el próximo paso es realizar simulaciones de render, con las cuales la aplicación tiene un mayor grado de personalización y adaptabilidad al usuario.”*

Después, se realizó una búsqueda minuciosa en el repositorio de la Universidad Católica de Santiago de Guayaquil, de todos los proyectos de titulación que se encuentran en esta base de datos, y se han encontrado 2 trabajos de titulación relacionados con el tema de los sistemas domóticos. A continuación, se especifican los dos trabajos:

1) En el trabajo de titulación de la Carrera de Ingeniería Electrónica en Control y Automatismo de la UCSG realizado por Haz, (2016) denominado “Estudio y diseño de un ambiente inteligente aplicando inmótica en el auditorio Dr. Leónidas Ortega Moreira de la Universidad Católica de Santiago de Guayaquil”. En este trabajo se realizó el estudio y diseño para lograr un posible ambiente automatizado para el auditorio Dr. Leónidas Ortega Moreira de la UCSG, haciendo controlables desde una misma plataforma lo que son iluminación, climatización, gestión energética, seguridad. En este trabajo, plantean y describen los equipos posibles a utilizar y se establece cada una de las áreas del auditorio donde se podrían instalar. Obteniendo finalmente, el valor que se obtendría para su automatización.

2) En el trabajo de titulación de la UCSG realizado por Cevallos, (2016) denominado “Aplicación de la plataforma hardware y software raspberry pi ii y el módulo de conectividad de red inalámbrica photon wi-fly, para el diseño de aplicaciones domóticas basadas en tecnología wi-fi”. En este trabajo de titulación se ha realizado una interfaz domótica haciendo uso de la tecnología Wi – Fi, utilizando los dispositivos Rasperry pi II y photon wi – fly. Se ha diseñado un sistema de domótica por medio de PCB’s que controlan y permiten la visualización del estado de los subsistemas desarrollados en la aplicación. El usuario es capaz de controlar todo el sistema mediante a la interfaz realizada, teniendo como interfaz una página web, de este modo cualquier dispositivo con conexión a internet sería capaz de controlar el sistema. (Cevallos, 2016)

1.2 Definición del Problema.

Según Fritz, (2017) las personas buscan automatizar sus hogares para realizar tareas que naturalmente se realizan en el día a día por si solas y cuando el usuario los desee, por ejemplo al ver una película, que las luces se ajusten para tener una mejor apreciación de la imagen, así como también reducir los gastos de los servicios públicos y así apagar luces y regular climatización cuando estas no se estén utilizando evitando gastos innecesarios. La seguridad del hogar en cuanto a accidentes y robos es otra cuestión, por un lado tenemos los accidentes que a diario existen alrededor del mundo debido a la poca visibilidad, con la automatización del hogar se podría encender o apagar las luces en los lugares que vayamos a necesitarla. Robos e intrusos en nuestros hogares sería casi imposible mediante la automatización pues así la policía podría estar 24/7 en nuestros hogares solo activando un simple botón.

Existen muchos motivos por los que en el 2017 las personas buscarían y optarían cada vez más por hogares automatizados, sin embargo ¿por qué aún no se ha logrado distribuir de forma importante en el Ecuador las casas automatizadas? Al parecer el desconocimiento de las personas es uno de los factores, pues prefieren optar por lo que ya conocen. Por ejemplo, según el Telégrafo, (2016) más de 403.000 cocinas de inducción se han vendido hasta el momento, desde que inició el programa en el año 2014, esto significa que más de 313.000 hogares se pasaron a las cocinas de inducción.

Las cifras revelan que en Ecuador existen 14'483.499 habitantes y 4'654.054 viviendas particulares. (Telégrafo, 2011), lo cual significa que tan solo el 6.72% de las viviendas del país en lo que va desde el 2014 se han cambiado a las cocinas de inducción. Es relativamente bajo para las mejoras que ofrece este servicio. Es válida la comparación de la domótica con las cocinas de inducción debido a que ambas ofrecen comodidad, mejoras y beneficios al usuario, además han tenido un impacto relativamente bajo debido aparentemente al desconocimiento de las personas.

1.3 Justificación del problema.

El siguiente trabajo busca abrir las posibilidades de la implementación de sistemas de domótica en el país. Como se lo estableció, uno de los mayores problemas de que las cosas no generen un mayor impacto de ventas, sería el que las personas carecen de información por lo que prefieren quedarse con sus objetos que ya utilizan.

Partiendo desde aquí, el trabajo va enfocado en realizar una pequeña simulación de un control básico de lo que sería iluminación, persianas, puertas y climatización de un hogar dando punto de partida a los sistemas de domótica en el país. Haciendo una maqueta vistosa, abriendo cabida a la posibilidad de empezar a automatizar los hogares.

Según el Telégrafo, (2016) “*En la provincia de Guayas se vendieron las cocinas de inducción en un 33%, seguido por Pichincha con 22%, Manabí 9%, Los Ríos 8%, El Oro y Santo Domingo de los Tsáchilas 4%, Esmeraldas y Santa Elena 3%, Imbabura y Tungurahua 2%*”. Debido a que Guayas fue la provincia donde más se cambió a las cocinas de inducción, y al ser la de mayor población sería ideal iniciar el proceso de enseñanza sobre las casas inteligentes en esta provincia, al tener apertura y posibilidad de realizarlo a través de un trabajo de titulación en la UCSG, el punto de partida iniciaría a través de la misma para los estudiantes de la Facultad de Educación Técnica para el Desarrollo.

1.4 Objetivos.

1.4.1 Objetivo General.

Desarrollar interfaces gráficas en Matlab y Android para el control de sistemas domóticos utilizando el dispositivo de comunicación inalámbrica NodeMcu ESP8266, con el fin de proporcionar información y lograr un mayor impacto acerca de sistemas de domótica.

1.4.2 Objetivos Específicos.

- Describir la fundamentación teórica y sus protocolos para la elaboración de sistemas domóticos.
- Diseñar interfaces gráficas en Matlab y Android para el sistema domótico.
- Realizar el algoritmo de programación que permita la comunicación inalámbrica con el dispositivo ESP8266.
- Validar el sistema domótico diseñado en Matlab, Android y Arduino.

1.5 Hipótesis.

Contribuir a la comunidad estudiantil de la UCSG mediante el desarrollo de las interfaces y simulación de un sistema domótico, con la ayuda de esta herramienta práctica, la misma sería capaz de ampliar el conocimiento e información con el que cuentan los estudiantes de la carrera de ingeniería electrónica en control y automatismo.

1.6 Metodología.

El método utilizado para el desarrollo del trabajo es de una investigación empírica debido a que se trata de un tema experimental, uno de los motivos que llevó a trabajar con el tema en cuestión fue la escasa información que hay sobre sistemas de domótica en Ecuador. Se cree necesario realizar un aporte al asunto, por medio del desarrollo del proyecto. Acompañado de un enfoque cuantitativo, el desarrollo del siguiente proyecto se apoya en las pruebas y simulaciones donde se obtuvieron resultados exactos. Los mismos que nos permitirán establecer mejoras en el aprendizaje de los estudiantes de la Carrera de Ingeniería Electrónica en Control y Automatismo de la Universidad Católica de Santiago de Guayaquil.

Capítulo 2: Fundamentación Teórica.

2.1. Domótica.

La domótica o automatización de las casas, es un término del cual se habla desde los años '70, siendo recién en la década de los '90 donde se concreta esta idea y pasa de ser una investigación a una realidad, pasando de circuito y de dispositivos interconectados, vía internet, al control de viviendas desde una misma plataforma. A la automatización de los hogares se le conoce por el término el “Internet de las cosas”, lo cual significa la conexión a internet de dispositivos.

La domótica, es una industria relativamente nueva, es por esto que el término domótica ha tomado varias formas de decirle, como hogares inteligentes, casas automatizadas, sin embargo se termina en la idea central que es domótica. Según Wood, (2012) la palabra 'domótica' proviene del latín 'domus' que significa hogar y robótica, entonces significa 'robótica doméstica' en realidad. Sin embargo esto no da una idea completa de lo que se pretende con la palabra domótica debido a que no se venden robots que planchen o arreglen la cama. La palabra robot proviene del checo 'robota' que significa 'trabajo pesado' o 'trabajo forzado'. Así que la domótica es una solución que puede hacer el trabajo pesado a su hogar, y hacerlo más fácil.

La domótica busca y está enfocado en facilitar las tareas diarias que se realizan, las mismas que en cierto modo ya son realizadas por inercia, salir apresurado de casa y olvidar si se cerró la ventana, se apagaron las luces, y se desconectó la plancha, por mencionar algunos, se vuelve un problema con el que se tiene que lidiar en el día a día, pero si se tuviera control de este tipo de cosas sin importar estar o no en casa, se ahorraría muchísimo tiempo y preocupaciones haciéndolo ideal en la actualidad para toda persona. Entonces en un futuro la domótica será una realidad para todos, y se convertirá en una herramienta infaltable en los hogares.

Un sistema de domótica, por lo tanto, consiste en la integración de los diversos productos, sistemas y servicios que permiten la automatización de las funciones domésticas, incluso con equipos (aparatos electrodomésticos, calderas, lámparas, dispositivos de audio / vídeo) que ya tiene un lugar tradicional en el hogar. La mayoría

ve con frecuencia en los hogares de gama alta, en los últimos años también se ha comenzado a ganar más de un punto de apoyo en edificios compartidos, debido principalmente a la estandarización de los componentes utilizados. (International, 2017)

En el Ecuador, la domótica es un campo que no está muy explotado aun, una de sus causas puede ser el costo, por lo cual su mercado se centra exclusivamente para personas de un nivel socio-económico elevado. Además que las personas tienden a confundir este tipo de sistemas con luces que se encienden al sentir movimientos, y no se puede resumir la domótica en tan pobre significado, (Flores, 2016) afirma que *“la domótica propiamente dicha es cuando los sistemas están integrados, cuando todo se puede controlar desde un lugar o varios lugares sin generar esfuerzos; si no hay integración de algunos sistemas no se podría decir que es domótica, incluso este concepto solo se aplica en la parte del hogar”*.



Figura 1. Sistema domótico.
Fuente: (Flores, 2016)

2.1.1. Casa Inteligente.

Una casa inteligente es un hogar que cuenta con sistemas automatizados altamente avanzados para controlar y monitorear cualquier función de iluminación de una casa, control de temperatura, multimedia, seguridad, operaciones de puertas y ventanas, calidad del aire o cualquier otra tarea de necesidad o comodidad, realizado por un residente de la casa. Con el aumento de la informatización inalámbrica, los dispositivos controlados por control remoto se vuelven inteligentes justo a tiempo. Hoy, es posible fijar un chip programado en cualquier ocupante y hacer que los

sistemas se ajusten a medida que una persona pasa por y a través de una casa inteligente. (Craven, 2017)

La domótica puede ser especialmente útil para personas mayores, personas con discapacidades físicas o cognitivas y personas con discapacidad que desean vivir de forma independiente. La tecnología hogareña es el juguete de los súper ricos, como la casa de Bill y Melinda Gates en el estado de Washington. Llamada Xanadu 2.0, la casa de los Gates es tan de alta tecnología que permite a los visitantes elegir la música ambiental para cada habitación que visitan. (Craven, 2017)

Según (Brodsky, 2016) se necesitan dos cosas para que las casas sean verdaderamente "inteligentes". *“Primero, hay sensores, actuadores y dispositivos que obedecen órdenes y brindan información de estado. Ya hay cientos, sino miles, de productos para el hogar inteligente en el mercado . Estos han evolucionado en los últimos años más allá de simples sensores de puerta e interruptores de luz para termostatos inteligentes como Nest y dispositivos de comando de voz como Amazon Echo”*. *“En segundo lugar, hay protocolos y herramientas que permiten que todos estos dispositivos, independientemente del proveedor, se comuniquen entre ellos. Sin embargo, esta es una gran empresa y no sucederá de la noche a la mañana. Mientras tanto, las aplicaciones para teléfonos inteligentes, los centros de comunicación y los servicios basados en la nube están permitiendo soluciones prácticas que se pueden implementar en este momento”*, dice Brodsky.



Figura 2. Imagen de Javier Pierini sobre casas inteligentes.
Fuente: (Craven, 2017)

2.2 El Internet de las cosas.

El "Internet de las cosas" (IoT) según Morgan, (2014) se está convirtiendo en un tema de conversación cada vez más creciente tanto en el lugar de trabajo como fuera de él. Es un concepto que no solo tiene el potencial de afectar la forma en que vivimos sino también cómo trabajamos. Hay muchas complejidades en torno al "Internet de las cosas", pero quiero seguir con lo básico. Se están teniendo muchas conversaciones técnicas y relacionadas con políticas, pero muchas personas todavía están tratando de captar las bases de qué diablos son estas conversaciones.

El acceso a Internet de banda ancha está cada vez más disponible, el costo de conexión está disminuyendo, se están creando más dispositivos con capacidades de Wi-Fi y sensores incorporados, los costos de tecnología están bajando y la penetración de teléfonos inteligentes está aumentando vertiginosamente. Todas estas cosas están creando una "*tormenta perfecta*" para el IoT, el concepto de básicamente es conectar cualquier dispositivo con un interruptor de encendido y apagado a Internet (y / o entre sí). Esto incluye todo, desde teléfonos celulares, cafeteros, lavadores, auriculares, lámparas, dispositivos portátiles y casi cualquier otra cosa que se te ocurra. Esto también se aplica a los componentes de las máquinas, por ejemplo, un motor a reacción de un avión o el taladro de una plataforma petrolera. Como mencioné, si tiene un interruptor de encendido y apagado, entonces es posible que forme parte del IoT. La firma analista Gartner dice que para 2020 habrá más de 26 mil millones de dispositivos conectados. Son muchas las conexiones (algunos incluso estiman que esta cifra es mucho más alta, más de 100 mil millones). El IoT es una red gigante de "cosas" conectadas (que también incluye personas). La relación será entre personas: personas, personas, cosas y cosas, cosas. (Morgan, 2014)

El internet de las cosas se define como una red que intercomunica distintos dispositivos físicos, tangibles por medio de internet, dispositivos que antes se conectaban mediante circuitos cerrados como intercomunicadores, cámaras, sensores y otros, en la actualidad estos dispositivos se valen de sistemas embebidos para comunicarse entre sí, estos hardware especializados no solo les permite la conexión a Internet sino que también se pueden programar eventos específicos, en función de las tareas que se le dicten de forma remota.

2.4 Redes de comunicación inalámbrica.

Las redes de comunicación inalámbricas son las que permiten que dos dispositivos o equipos puedan conectarse entre sí, de modo que no se requiera cableado. Este tipo de red transmite los datos de forma eficaz a través de un enlace usando ondas electromagnéticas.

Los dispositivos comúnmente utilizados para redes inalámbricas incluyen computadoras portátiles, computadoras de escritorio, computadoras de mano, asistentes digitales personales (PDA), teléfonos celulares, computadoras basadas en lapiceras y buscapersoas. Las redes inalámbricas funcionan de manera similar a las redes cableadas, sin embargo, las redes inalámbricas deben convertir las señales de información en una forma adecuada para la transmisión a través del medio aéreo. Las redes inalámbricas tienen varios propósitos. En algunos casos se usan como un tipo de reemplazo para los cables, mientras que en otros casos se utilizan para proporcionar acceso a datos corporativos desde ubicaciones remotas.

La infraestructura inalámbrica se puede construir por un costo muy bajo en comparación con el cableado tradicional pero construir redes inalámbricas solo se trata en parte de ahorrar dinero para brindar a las personas de su comunidad un acceso más económico y fácil a la información, se beneficiarán directamente de lo que Internet ofrece. El tiempo y el esfuerzo ahorrado al tener acceso a la red global de información se traduce en riqueza a escala local, ya que se puede hacer más trabajo en menos tiempo y con menos esfuerzo.

Las redes inalámbricas permiten que los dispositivos remotos se conecten sin dificultad, independientemente, estos dispositivos están a pocos metros o varios kilómetros de distancia y no es necesario romper paredes para pasar cables o instalar conectores. Esto ha hecho que el uso de esta popular tecnología, se extienda rápidamente. Hay varias tecnologías que difieren en la frecuencia de transmisión, velocidad y rango de las mismas, por otro lado, hay algunos problemas relacionados con la regulación legal del espectro electromagnético. Las ondas electromagnéticas se transmiten a través de muchos dispositivos, pero son propensos a la interferencia. Por esta razón, todos los países necesitan regulaciones que definen los rangos de frecuencia y la potencia de transmisión para cada tecnología es permitida. (Salazar, 2017)

Además, las ondas electromagnéticas no se confinan fácilmente a un área geográfica restringida, por esto, un hacker puede escuchar fácilmente una red si los datos transmiten no están codificados. Por lo tanto, se deben tomar todos los pasos necesarios para garantizar la privacidad de datos transmitidos a través de redes inalámbricas.

Existen varios tipos de redes inalámbricas, según el tamaño o la cobertura que la red utiliza, esta se puede dividir en dos segmentos: corto y largo alcance. Los de corto alcance son para áreas limitadas, y los de largo alcance son usados para grandes áreas como un estado o provincia o un país entero.

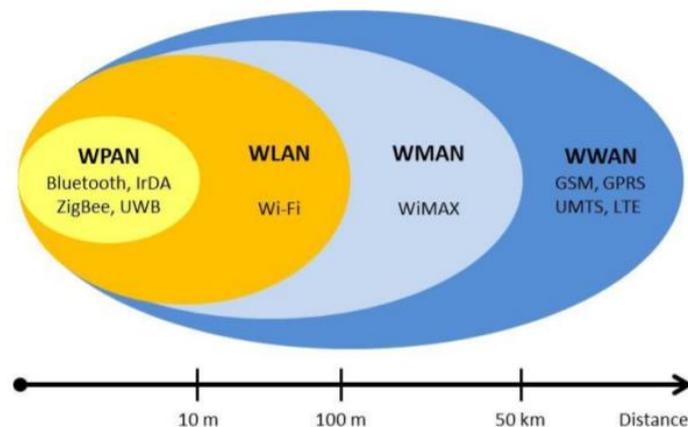


Figura 4. Clasificación de las redes inalámbricas.

Fuente: (Salazar, 2017)

2.4.1 WPAN.

Las redes de área personal inalámbricas se basan en IEEE 802.15. Ellos permiten una comunicación en un rango muy corto, de aproximadamente 10 metros. A diferencia de otras redes inalámbricas, una conexión hecha a través de un WPAN implica poca o ninguna infraestructura o conectividad directa al mundo fuera del enlace. Esto permite una poca potencia eficiente, soluciones de bajo costo para ser implementadas para una amplia gama de dispositivos tales como un teléfono inteligente y un PDA.

Estas redes se caracterizan por bajas demandas de potencia y una baja tasa de bits. Este tipo de redes de retransmisión en tecnologías como Bluetooth, IrDA, ZigBee o UWB, desde el punto de vista de una aplicación, Bluetooth está destinado a un mouse inalámbrico, teclado y auriculares manos libres, IrDA está destinado a enlaces punto a

punto entre dos dispositivos para transferencias de datos simples y sincronización de archivos, ZigBee está diseñado para redes confiables de monitoreo y control en red inalámbrica, y UWB es orientado a enlaces multimedia de gran ancho de banda. (Salazar, 2017)

2.4.2 WLAN.

Las redes inalámbricas de área local están diseñadas para proporcionar acceso inalámbrico en áreas con un rango típico de hasta 100 metros y, se usan principalmente en el hogar, la escuela, laboratorio de cómputo o entornos de oficina. Esto da a los usuarios la capacidad para moverse dentro de un área de cobertura local y aún estar conectado a la red. Las WLAN se basan en estándares IEEE 802.11, comercializados bajo la marca de Wi-Fi.

IEEE 802.11 fue más simple de implementar y lo hizo más rápido para el mercado. Esta es una familia de diferentes estándares para redes inalámbricas de área local. El IEEE 802.11b fue el primer estándar aceptado que admite hasta 11 Mbps en banda de espectro 2.4GHz sin licencia. Luego, se diseñó el estándar IEEE 802.11g como un sucesor de mayor ancho de banda para la IEEE 802.11b. Un punto de acceso IEEE 802.11g admite clientes 802.11b y 802.11g. Del mismo modo, una computadora portátil con una tarjeta IEEE 802.11g podrá acceder a los puntos de acceso 802.11b existentes, así como a los nuevos puntos de acceso 802.11g. Eso es porque las LAN inalámbricas basadas en 802.11g usarán la misma banda de 2,4 GHz que utiliza 802.11b. La tasa de transferencia máxima para el enlace inalámbrico IEEE 802.11g es de 54 Mbps, pero disminuirá automáticamente cuando la señal de radio sea débil o cuando se detecta interferencia. (Salazar, 2017)



Figura 5. Diagrama de redes WLAN en hogares

Fuente: (Salazar, 2017)

2.4.3 WMAN.

Las redes inalámbricas de área metropolitana son el tercer grupo de redes inalámbricas. Los WMAN se basan en el estándar IEEE 802.16, que a menudo se denomina WiMAX (Interoperabilidad mundial para el acceso por microondas). WiMAX es una tecnología de comunicaciones que admite arquitectura punto a multipunto donde su objetivo es suministrar datos de forma inalámbrica de alta velocidad en redes de área metropolitana. Esto permite LAN inalámbricas más pequeñas para ser conectadas entre sí por WiMAX creando una gran red WMAN. Por lo tanto, la creación de redes entre las ciudades se puede lograr sin la necesidad de costoso cableado.

WiMAX es similar a Wi-Fi, pero proporciona cobertura a mayores distancias. Mientras Wi-Fi está destinado a proporcionar cobertura en áreas relativamente pequeñas, como en oficinas o puntos calientes, WiMAX opera en dos bandas de frecuencia, una mezcla de banda con y sin licencia, de 2 GHz a 11 GHz y de 10 GHz a 66 GHz, y puede transferir alrededor de 70 Mbps en una distancia de 50 km a miles de usuarios de una misma base. Como puede operar en dos frecuencias, las bandas WiMAX pueden trabajar por línea de visión y sin línea de visión. El rango de frecuencia a los 2 a 11 GHz funciona sin línea de visión, donde una computadora dentro de un edificio se comunica con una antena fuera del edificio. Las transmisiones de baja frecuencia no se interrumpen fácilmente por obstrucciones físicas. Las transmisiones de alta frecuencia se usan para el servicio de línea de visión. Esto permite a las antenas comunicarse entre sí a una mayor distancia. (Salazar, 2017)



Figura 6. Diagrama de redes WLAN en hogares
Fuente: (Salazar, 2017)

2.4.4 WWAN.

Según Salazar, (2017) las redes de área amplia inalámbricas se extienden más allá de los 50 kilómetros y normalmente utilizan frecuencias con licencia. Este tipo de redes se puede mantener en grandes áreas, como ciudades o países, a través de múltiples sistemas de satélites o sitios de antena después por un proveedor de servicios de internet. Existen principalmente dos tecnologías disponibles:

- **Telefonía celular digital y satelital:**

En el sistema celular, el área de cobertura se divide en celdas. Un transmisor celular, en centro de la célula, está diseñado para servir a una célula individual. Todos los transmisores son conectados a una estación de base y estos últimos a un centro de telecomunicaciones que conecta la red de telefonía móvil y celular. Los sistemas buscan hacer un uso eficiente de los canales disponibles mediante el uso de transmisores de baja potencia para permitir la reutilización de frecuencias a distancias mucho más pequeñas.

La primera generación, 1G, era analógica y concebida y diseñada exclusivamente para llamadas de voz casi sin consideración de servicios de datos, con una velocidad de hasta 2.4 kbps. La segunda generación, 2G, se basó en tecnología digital e infraestructura de red (GSM), habilitando mensajes de texto, y con una velocidad de datos de hasta 64 Kbps. La generación 2.5G estaba entre el segundo y el tercero. A veces ha sido referido como 2G + GPRS, es una versión mejorada de 2G, con una velocidad de hasta 144 Kbps. La generación 3G se introdujo en el año 2000, con una velocidad de datos de hasta 2 Mbps. La generación 3.5G es una versión mejorada de 3G que usa HSDPA para acelerar las transferencias de datos a 14 Mbps. Finalmente, la cuarta generación, 4G, es capaz de proporcionar hasta 1 Gbps velocidad y cualquier tipo de servicio en cualquier momento según los requisitos del usuario, en cualquier lugar. La generación 5G se espera para el año 2020.

- **Satélite:**

Las comunicaciones inalámbricas también pueden desarrollarse vía satélite. Debido a su alto altitud, las transmisiones por satélite pueden cubrir una amplia área sobre la superficie de la tierra. Esto puede ser muy útil para los usuarios que se encuentran en áreas remotas o islas donde no hay cables submarinos en servicio. En

estos casos, se necesitan teléfonos satelitales. Cada satélite está equipado con varios transpondedores que consisten en un transceptor y una antena. La señal entrante se amplifica y luego se retransmite en un canal diferente frecuencia.

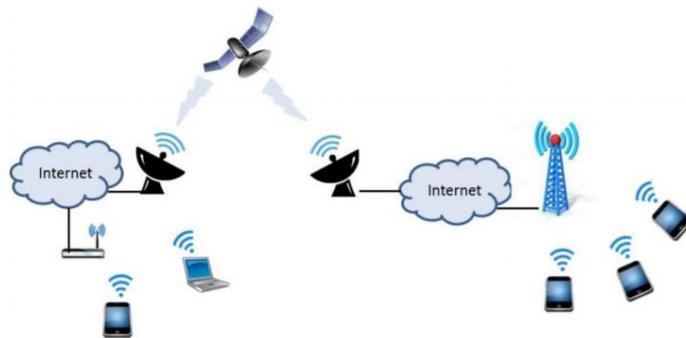


Figura 7. Redes satelitales y de celular
Fuente: (Salazar, 2017)

2.5 Protocolos y estándares.

Los estándares y protocolos son un tipo de instrucciones implícitas. Los estándares con frecuencia son las normas o reglas que se deben cumplir en una profesión o trabajo, y los protocolos suelen ser las instrucciones y herramientas, es por eso que van de la mano. Los estándares se crean como un conjunto de normas o reglas que varias empresas utilizan con el fin de que los dispositivos de desarrollo de domótica de distintas marcas y fabricantes puedan ser compatibles entre ellas, en otras palabras el estándar es el acuerdo establecido para la realización o medición de algo, en este caso el desarrollo de un sistema de domótica. Por otro lado, los protocolos son todas las reglas que gobiernan la comunicación entre un dispositivo con otro, o de una aplicación con otra, es decir las instrucciones principales para intercambiar la información.

2.5.1 Protocolo de internet TCP/IP.

TCP / IP es una familia de protocolos de comunicación utilizados para conectar sistemas informáticos en una red. Lleva el nombre de dos de los protocolos de la familia: Protocolo de control de transmisión (TCP) y Protocolo de Internet (IP). El protocolo de transferencia de hipertexto (HTTP) es un miembro de la familia TCP / IP. (IBM Knowledge Center, 2018)

Los protocolos en la familia TCP / IP corresponden, en muchos casos, a las capas del modelo de Interconexión de Sistemas Abiertos (OSI). La Tabla 1 muestra HTTP y las capas subyacentes de la familia TCP / IP en términos del modelo OSI. También se muestran las capas de arquitectura de red de sistemas (SNA), que coinciden aproximadamente con las capas de OSI.

TABLA 1. CAPAS DEL PROTOCOLO TCP/IP

Layer	OSI	SNA	TCP/IP
7	Application	Application	HTTP
6	Presentation	Presentation	(empty)
5	Session	Data flow	(empty)
4	Transport	Transmission	TCP
3	Network	Path control	IP
2	Data link	Data link	Subnetwork
1	Physical	Physical	Subnetwork

Autor: (IBM Knowledge Center, 2018)

2.5.2 Protocolos de la domótica.

El protocolo de comunicación de la domótica se refiere al lenguaje que utilizan los dispositivos de una vivienda para comunicarse entre ellos. Siendo la comunicación una parte vital para poder conocer hoy en día un poco más acerca de los sistemas de domótica resultaría pertinente entender cómo funcionan cada uno de ellos. Existen alrededor de diez protocolos distintos de domótica disponibles. Cada uno tiene su propio lenguaje. Cada dispositivo habla su propio idioma, el cual les indica a los diversos dispositivos conectados que función realizar.

Elegir un protocolo hogareño inteligente puede ser complicado. Obviamente, desea una que admita una gran cantidad de dispositivos, así como una que ofrezca la mejor interoperabilidad posible entre dispositivos (la capacidad de los dispositivos para comunicarse entre ellos). Pero también hay otros factores a considerar, como el consumo de energía, el ancho de banda y, por supuesto, el costo. (EH Contributor, 2016)

Estos protocolos se dividen en dos categorías básicas, que serían: cableada e inalámbrica. Entre los protocolos cableados, tenemos UPB y X10, estos usan el cableado existente del hogar para comunicarse. Estos tipos de protocolos resultan confiables, pero pueden ser lentos, y complicados para encriptar. Los protocolos inalámbricos suelen ser más rápidos y más compatibles con otros dispositivos. Los protocolos inalámbricos de domótica incluyen Z-Wave, ZigBee, Wi-Fi, Thread y Bluetooth. Se conectan y se comunican sin depender de las líneas eléctricas, lo que las hace más fáciles de asegurar. (Edwards, 2017)

2.5.2.1 Propietarios o cerrados.

Este tipo de protocolo lo utilizan las empresas que optan por que los dispositivos de sus marcas o fabricantes sean solo compatibles entre sí, de este modo solo el fabricante puede realizar modificaciones, mejoras y crear más dispositivos que puedan comunicarse solo entre ellos.

Estos tipos de protocolos poseen sus pros y sus contras, siendo los pros la protección de plagio de sus productos y sus derechos, y entre sus contras el que limita la producción de evoluciones de los sistemas de domótica, otro inconveniente es la vida útil, debido a que, si la empresa desaparece, eventualmente los dispositivos también lo harán, ya que el mismo depende de la vida de la empresa en gran medida. (IECOR, 2017)

2.5.2.2 Estándar o abiertos.

Estos protocolos son desarrollados por varias empresas con el fin de unificar criterios, en un estándar, si una de las empresas desaparece no habría mayor problema porque sus dispositivos serian compatibles con dispositivos de otras, y se sobrellevaría de buena forma esta situación. Estos tipos de protocolos estándar en la actualidad son los siguientes: KNX, LonWorks y X10. Desde que inicio la domótica los fabricantes buscaron crear estándares para lograr una compatibilidad entre sí. (IECOR, 2017)

2.5.2.3 KNX.

KNX es un estándar creado por los europeos como iniciativa para crear un único estándar europeo para los sistemas de domótica e Inmótica, los europeos buscaban un

estándar que pudiera competir en calidad, prestaciones y precios con otros sistemas norteamericanos como son LonWorks, y CEBus. (IECOR, 2017)



Figura 8. Logo estándar KNX
Fuente: (García Pumarino, 2015)

En abril, nueve compañías europeas se unieron y establecieron Konnex para trabajar en el desarrollo de un nuevo estándar resultado de la unión de Batibus, European Installation Bus, y European Home System. Este estándar es capaz de usar varios medios físicos; por trenzado, línea eléctrica, cableado Ethernet o radio frecuencia, por lo que es fácil de acoplarse a construcciones de casas que ya existen o casa que se encuentran en construcción. (García Pumarino, 2015)

2.5.2.4 X10.

X10 es de los estándares más antiguos que se utilizan en las aplicaciones domóticas. Su diseño se dio en Escocia entre 1976 y 1978 teniendo como objetivo principal transmitir información por líneas de baja tensión a muy poca velocidad y muy bajos costos. Al utilizar las líneas eléctricas ya existentes de las viviendas, no era necesario poner cables nuevos para conectar estos dispositivos. (IECOR, 2017)

El estándar X10, puede ser utilizado por cualquier fabricante, es decir, cualquier compañía puede trabajar con sistemas X10 o crear dispositivos X10 y venderlos bajo su propia marca, sin embargo, se necesita utilizar los circuitos creados por el fabricante escocés. Aunque, sucede lo opuesto con Echelon y su Neuron Chip que usa LonWorks, ya que los circuitos X10 tienen un royalty muy bajo. (IECOR, 2017)

Este estándar es el más extendido para las aplicaciones domóticas, debido a que son económicos, abiertos, por lo que existen muchos dispositivos de estos en el mercado y que son compatibles, además que son fáciles de instalar y configurar. Gracias a que utiliza las líneas eléctricas ya existentes de la vivienda se lo conoce como “tecnología de corrientes portadoras”, y es por medio de esta que transmite la comunicación entre los elementos domóticos. Sus módulos funcionan en base de “1 y

0”, donde si existe el impulso eléctrico significa “1” y la ausencia del mismo significa “0”. (García Pumarino, 2015)

Su configuración es sumamente sencilla pues solo se asigna a cada uno de los dispositivos un código de vivienda y de unidad, con lo que hace posible un total de 256 distintas combinaciones. Los códigos se escogen de forma manual en cada dispositivo y forma la dirección lógica. Para añadir un nuevo sistema, solo se debe conectarlo, y asignarle esta dirección lógica de forma manual. (García Pumarino, 2015)

2.5.2.5 LonWorks.

El estándar LonWorks es un tipo de tecnología de control domótico, de la cual es dueña la compañía norte-americana Echelon Corp, esta tecnología se basa en el protocolo LonTalk, el cual nos brinda todas las herramientas posibles para implementar redes de control distribuidas. Tal cual como KNX, LonWorks utiliza distintos medios de transmisión, pueden ser: aire, coaxial, red eléctrica, par trenzado o fibra. Entre sus requisitos se encuentra la instalación de “nodos” a través de la red que permitan gestionar distintos sensores y actuadores. Para su instalación y configuración se requieren profesionales capaces de utilizar herramientas informáticas correctas. Entre sus pros, se puede decir que LonWorks es una tecnología que cuenta con robustez, además de ser fiable, por lo que se especializa en automatizaciones industriales. Es más utilizada en Estados Unidos, que en el continente europeo. Los dispositivos LonWorks, están basados en un micro controlador llamado Neuron Chip. (García Pumarino, 2015)



Figura 9. Logo estándar LonWorks
Fuente: (García Pumarino, 2015)

Según, (Peñalver, 2012) *“Todos los dispositivos presentes en una red Lonworks® precisan de un chip Neuron. El Neuron está constituido internamente como tres microprocesadores en uno. Dos de los microprocesadores están optimizados para ejecutar el protocolo de comunicaciones, mientras que el tercero*

está dedicado a ejecutar el programa de control del nodo". Existen entonces dos procesadores, uno está encargado de la comunicación, y el otro está encargado de la aplicación.

2.5.2.6 UPB.

UPB es una tecnología cableada, Universal Powerline Bus, aunque similar a X10, fue pensada para ser un reemplazo X10, dada su fiabilidad superior (menos susceptible al ruido de la línea de alta tensión y mayor alcance (puede transmitir más de una milla). La tecnología utiliza la potencia existente de una casa líneas, lo que reduce los costos un poco, para enviar señales que controlarán los dispositivos tanto dentro como fuera del hogar. Sin embargo, los productos UPB y X10 no son compatibles. Por lo tanto, si ya tiene productos X10 y desea agregar productos UPB a la mezcla, se requerirá un controlador especial que hablará con ambos.

Si bien UPB es un sistema de línea de alta tensión confiable, es difícil combinarlo con las tecnologías inalámbricas más nuevas como Wi-Fi, teléfonos inteligentes, etc. Además, aunque la plataforma ofrece un factor de fiabilidad del 99%, ofrece un ancho de banda relativamente bajo, por lo el rendimiento puede ser lento. Tampoco proporciona cifrado, lo que significa que no es tan seguro como el inalámbrico. La complejidad técnica del sistema hace que la experiencia de configuración del usuario sea difícil. Y la gran desventaja: hay muchos menos dispositivos compatibles con UPB que para otras tecnologías. (EH Contributor, 2016)

2.5.2.7 Z – Wave.

Z-Wave es un protocolo de automatización hogareña que utiliza frecuencias de radio para comunicarse con dispositivos domóticos. Fue desarrollado en Dinamarca y se ha convertido rápidamente en uno de los protocolos de domótica más populares, ya que cuenta con el respaldo de más de 450 empresas en todo el mundo. (Edwards, 2017)

La domótica Z-Wave usa la banda de frecuencia de 908.42 MHz para comunicarse en los Estados Unidos. El protocolo Z-Wave usa un controlador central (concentrador) para configurar y administrar la red de automatización del hogar. Una

vez que la red está configurada, puede agregar dispositivos domésticos inteligentes, que luego pueden controlarse utilizando el protocolo Z-Wave.

Existen más de 1.700 dispositivos certificados compatibles con Z-Wave que están disponibles en todo el mundo. Esta extensa selección le brinda muchas opciones para personalizar la automatización de su hogar. Los dispositivos Z-Wave son conocidos por ser fáciles de configurar y usar. Además, el protocolo Z-Wave tiende a consumir menos energía, por lo que puede usar toda la automatización que desee sin hacer que sus facturas de energía se disparen.

Los beneficios que presenta este protocolo son debido a que la frecuencia Z-Wave es significativamente menor que la frecuencia utilizada para la mayoría de los demás dispositivos inalámbricos, tiene menos posibilidades de interferencia. Esto significa que menos dispositivos compiten por la misma frecuencia, lo que hace que la comunicación sea eficiente y rápida.

Una de las mejores cosas de la automatización doméstica Z-Wave es que todos los dispositivos compatibles con Z-Wave se pueden comunicar, independientemente de su marca, tipo o versión. Incluso a medida que los productos evolucionan con el tiempo, la red Z-Wave seguirá funcionando con las versiones más antiguas y nuevas de los productos, lo que significa que no tiene que comenzar de nuevo con un nuevo concentrador cada vez que se actualice algo. Esto es especialmente importante en un mundo donde la tecnología cambia rápidamente y donde las actualizaciones y las nuevas versiones es solo cuestión de tiempo. (Edwards, 2017)



Figura 10. Logo compañía Z-wave
Fuente: (EH Contributor, 2016)

2.5.2.8 Insteon.

El protocolo Insteon es un híbrido de tecnologías inalámbricas y por cable, lo que lo convierte en una adición única al campo de los jugadores de domótica. La domótica de Insteon opera en una red patentada de doble malla que utiliza comunicaciones inalámbricas y cableadas para superar los problemas comunes que cada tipo de red encuentra por sí solo. Un concentrador Insteon se conecta con dispositivos compatibles con Insteon, lo que le permite controlar su hogar a través de su teléfono inteligente, tableta o computadora.

Existen más de 200 dispositivos compatibles con Insteon disponibles para ayudar a automatizar el hogar. Insteon confía más en su propia línea de productos de automatización del hogar que muchos otros protocolos. Esto significa que existe una compatibilidad limitada con dispositivos inteligentes fabricados por otros fabricantes. Sin embargo, los dispositivos compatibles con Insteon son compatibles tanto hacia adelante como hacia atrás, lo que significa que la automatización doméstica de Insteon siempre funcionará con dispositivos más antiguos y nuevos.

Los beneficios de Insteon Home Automation son que no importa su nivel de destreza técnica, podrá instalar la automatización doméstica de Insteon sin problemas. Siempre que sepa cómo usar un teléfono inteligente, administrar el protocolo Insteon es fácil. Lo mejor de todo es que los dispositivos compatibles con Insteon se agregan automáticamente a la red tan pronto como los enciende, lo que hace que la configuración sea rápida y sin molestias adicionales para usted. (Edwards, 2017)



Figura 11. Logo compañía INSTE-ON
Fuente: (EH Contributor, 2016)

2.5.2.9 Zig Bee.

Hay innumerables similitudes entre Z-Wave y ZigBee. Al igual que Z-Wave, ZigBee es exclusivamente un protocolo inalámbrico de automatización del hogar. Aunque muchos entusiastas de la automatización del hogar reclaman, su aceptación total está limitada por la falta de interoperabilidad entre los dispositivos ZigBee, que a menudo tienen dificultades para comunicarse con los de diferentes fabricantes. Como resultado, ZigBee no es necesariamente una opción ideal para cualquier persona que recién comienza a utilizar la automatización del hogar, a menos que, por supuesto, use dispositivos de un solo fabricante. Además, hay diferentes versiones de ZigBee que no necesariamente hablan a la perfección entre sí.

Al igual que Z-Wave, ZigBee es una tecnología de bajo costo y bajo consumo de energía, lo que significa que los dispositivos que funcionan con batería en una red ZigBee disfrutarán de una larga vida útil. Al funcionar con el estándar de comunicación inalámbrica 802.15.4, ZigBee también utiliza una estructura de red en malla que proporciona un excelente alcance y crea una comunicación rápida entre dispositivos ZigBee. La última versión, ZigBee 3.0, promete una mejor interoperabilidad entre dispositivos y versiones. Eso aún está por verse. (EH Contributor, 2016)



Figura 12. Logo compañía ZigBee
Fuente: (EH Contributor, 2016)

2.5.2.10 Wi – Fi.

Con un gran ancho de banda, el Wi-Fi ya está prácticamente en todas partes, por lo que muchos fabricantes están haciendo con entusiasmo dispositivos domésticos inteligentes para trabajar con él. Una gran cantidad de hogares en los Estados Unidos ya tienen enrutadores inalámbricos (que funcionan con el protocolo Wi-Fi), por lo que obviamente ya tienen un centro central en el que se pueden conectar dispositivos compatibles con Wi-Fi.

Sin embargo, esto tiene un inconveniente clave: problemas de interferencia y ancho de banda. Si su casa está llena de dispositivos conectados a Wi-Fi (televisores, consolas de videojuegos, computadoras portátiles, tabletas, etc.), entonces sus dispositivos inteligentes tendrán que competir por el ancho de banda y pueden ser más lentos en responder. Wi-Fi también está hambriento de poder; en consecuencia, los dispositivos inteligentes que funcionan con batería, como cerraduras y sensores, se agotan mucho antes que en otros entornos inalámbricos. (EH Contributor, 2016)

Este es el protocolo de comunicación con el que trabaja el sistema que se ha desarrollado, debido a que la tarjeta ESP8266 utiliza el mismo.



Figura 13. Logo compañía Wi-Fi
Fuente: (EH Contributor, 2016)

2.5.2.11 Bluetooth.

Bluetooth está en el centro de cientos de productos, desde bombillas hasta bases de altavoces y cerraduras (Kwikset presentó la primera cerradura Bluetooth, Kevo, en 2013). Tiene un ancho de banda de datos mayor que ZigBee y Z-Wave (aunque menor que el Wi-Fi) pero, para su ventaja, absorbe mucha menos energía que el Wi-Fi.

Por el contrario, también tiene un rango bastante limitado, por lo que para los dispositivos que requieren conexión constante-piense en sensores de movimiento, sistemas de seguridad, etc.-puede que no sea la plataforma ideal. Sin embargo, se ha informado de que la versión más reciente de Bluetooth (Bluetooth Low Energy o BLE) será capaz de formar redes de malla, ampliando enormemente su alcance y dando ZigBee y Z-Wave una carrera por su dinero. Y sin necesidad de un concentrador central, el factor de conveniencia no puede pasarse por alto. (EH Contributor, 2016)

ABI Research dice que la cuota de mercado para aplicaciones de hogares inteligentes como iluminación, seguridad y administración de energía basada en Bluetooth (que se hace muy segura a través de encriptación de nivel gubernamental) se prevé que aumente a un ritmo más rápido que cualquier tecnología inalámbrica en los próximos cinco años.



Figura 14. Logo compañía Bluetooth
Fuente: (EH Contributor, 2016)

2.5.2.12 Thread.

Thread es un nuevo protocolo inalámbrico para dispositivos domésticos inteligentes. The Thread Group se formó en julio de 2014 por siete miembros fundadores, incluidos los laboratorios Nest de Google y Samsung Electronics. Se pueden conectar más de 250 dispositivos en una red Thread y, como la mayoría de los dispositivos destinados a conectarse a la red funcionan con baterías, es muy frugal en cuanto a potencia.

Utilizando la misma frecuencia y los mismos chips de radio que ZigBee, Thread está destinado a proporcionar una red segura, fiable y de baja potencia que facilita la conexión entre más de 250 dispositivos en el hogar. Además, pueden conectarse a la nube para tener acceso ubicuo. Nest Learning Thermostat y Nest Protect ya están usando una versión de Thread, y se supone que más productos entrarán al mercado este año. (EH Contributor, 2016)



Figura 15. Logo compañía THREAD
Fuente: (EH Contributor, 2016)

2.5.2.13 Apple “Home Kit”.

Si bien no es un protocolo de comunicaciones, Apple HomeKit debe mencionarse solo por la atención de los medios que atrae. Anunciado por Apple en su Conferencia de Desarrolladores en 2014, HomeKit es en realidad un marco de software que permite a los desarrolladores construir dispositivos domésticos inteligentes que se conectarán directamente al iPhone y al iPad y serán controlados por una aplicación dedicada.

En realidad, HomeKit va a ser un "conducto" de tecnologías inalámbricas. Ya se sabe que los dispositivos aprobados por HomeKit usarán Wi-Fi y Bluetooth como sus interfaces principales, y un puente Z-Wave / ZigBee está en camino. Apple también aprobó un puente Insteon-HomeKit, y Lutron ofrece un concentrador compatible con HomeKit para su sistema Caseta Wireless.

Muchos fabricantes de dispositivos están trabajando febrilmente para integrar HomeKit en sus productos, aunque, a partir de ahora, solo hay unos pocos en el mercado, incluido el popular August Smart Lock. Esta es una tecnología que está en sus comienzos pero que sin duda soporta la observación.

2.6 Servidor Web.

El significado de “servidor web” puede referirse a hardware o software, o ambos trabajando juntos.

- Por el lado del hardware, un servidor web es una computadora que almacena los archivos componentes de un sitio web (por ejemplo, documentos HTML, imágenes, hojas de estilo CSS y archivos JavaScript) y los entrega al dispositivo del usuario final. Está conectado a Internet y se puede acceder a través de un nombre de dominio como mozilla.org.
- Por el lado del software, un servidor web incluye varias partes que controlan cómo los usuarios de la web acceden a los archivos alojados, como mínimo un servidor HTTP. Un servidor HTTP es una pieza de software que comprende las direcciones URL (direcciones web) y HTTP (el protocolo que usa su navegador para ver páginas web).

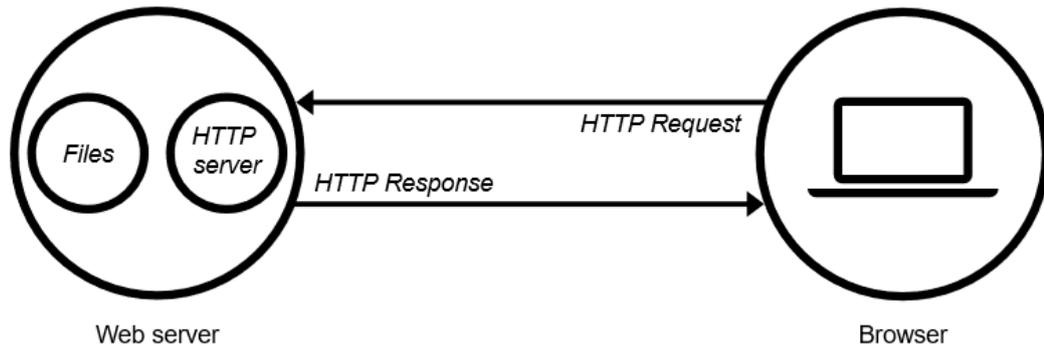


Figura 16. Diagrama del funcionamiento de un servidor

Fuente: (Hierro Duarte & Mills, 2017)

En el nivel más básico, cada vez que un navegador necesita un archivo alojado en un servidor web, el navegador solicita el archivo a través de HTTP. Cuando la solicitud llega al servidor web correcto (hardware), el *servidor HTTP* (software) devuelve el documento solicitado, también a través de HTTP. Para que un sitio web pueda publicarse, se necesita un servidor web estático o dinámico.

Un **servidor web estático**, o pila, consta de una computadora (hardware) con un servidor HTTP (software). Lo llamamos "estático" porque el servidor envía sus archivos alojados "tal cual" a su navegador.

Un **servidor web dinámico** consta de un servidor web estático más software adicional, más comúnmente un *servidor de aplicaciones* y una *base de datos*. Lo llamamos "dinámico" porque el servidor de aplicaciones actualiza los archivos alojados antes de enviarlos a su navegador a través del servidor HTTP.

Por ejemplo, para producir las páginas web finales que ve en el navegador, el servidor de aplicaciones puede llenar una plantilla HTML con contenido de una base de datos. Los sitios como MDN o Wikipedia tienen muchos miles de páginas web, pero no son documentos HTML reales, solo unas pocas plantillas HTML y una base de datos gigante. Esta configuración facilita y agiliza el mantenimiento y la entrega del contenido. (Hierro Duarte & Mills, 2017)

2.7 Firebase.

Firebase es un Backend-as-a-Service (BaaS) que significa, una forma de vincular la información de alguna aplicación o desarrolladores web a la nube, Firebase comenzó como un inicio de YC11 y se convirtió en una plataforma de desarrollo de aplicaciones de próxima generación en Google Cloud Platform. Firebase libera a los desarrolladores para centrarse en la creación de fantásticas experiencias de usuario. No necesita administrar servidores. No necesita escribir API. Firebase es un servidor, API y almacén de datos, todo escrito de manera genérica que puede modificarlo para satisfacer la mayoría de las necesidades. Sí, ocasionalmente necesitarás usar otros bits de Google Cloud para tus aplicaciones avanzadas. Firebase no puede ser todo para todos. Pero se pone bastante cerca. (Esplin, 2016)

Firebase es una base de datos en tiempo real, la mayoría de las bases de datos que existen requieren llamadas HTTP para lograr guardar y sincronizar datos, además en su gran mayoría solamente brindan información cuando se la solicita. Cuando una aplicación es conectada a Firebase, no se usa el HTTP normal, sino es conectada a través de un WebSocket, siendo este más rápido que HTTP. No tiene que hacer llamadas WebSocket individuales, porque una conexión de socket es suficiente. Todos sus datos se sincronizan automáticamente a través de ese único WebSocket tan rápido como la red de su cliente puede llevarlo. Firebase le envía datos nuevos tan pronto como se actualicen. Cuando su cliente guarda un cambio en los datos, todos los clientes conectados reciben los datos actualizados casi al instante.

Entre las cosas que ofrece Firebase se encuentra también, Firebase Storage proporciona una forma sencilla de guardar archivos binarios, generalmente imágenes, pero podría ser cualquier cosa, en Google Cloud Storage directamente del cliente. Además, Firebase Storage tiene su propio sistema de reglas de seguridad para proteger su cubo GCloud de las masas, al tiempo que otorga privilegios de escritura detallados a sus clientes autenticados.

Firebase auth tiene incorporado un sistema de autenticación de correo electrónico / contraseña. También es compatible con OAuth2 para Google, Facebook, Twitter y GitHub. Nos centraremos en la autenticación de correo electrónico / contraseña en su mayor parte. El sistema OAuth2 de Firebase está bien documentado

y en gran parte es copiar / pegar. Firebase Auth se integra directamente en la base de datos de Firebase, por lo que puedes usarlo para controlar el acceso a tus datos.

Firebase incluye un servicio de alojamiento fácil de usar para todos los archivos estáticos. Les sirve desde un CDN global con HTTP / 2, y para que su desarrollo sea especialmente sencillo, Firebase Hosting utiliza Superstatic, que puede ejecutar localmente para todas sus pruebas.

Pros

- Correo electrónico y contraseña, Google, Facebook y autenticación Github
- Datos en tiempo real
- Api confeccionado
- Seguridad incorporada en el nivel de nodo de datos
- Almacenamiento de archivos respaldado por Google Cloud Storage
- Alojamiento de archivos estáticos
- Trate los datos como flujos para crear aplicaciones altamente escalables
- No te preocupes por tu infraestructura

Contras

- Habilidades de consulta limitadas debido al modelo de flujo de datos de Firebase
- Los modelos tradicionales de datos relacionales no son aplicables a NoSQL; por lo tanto, sus chuletas de SQL no se transferirán
- Sin instalación en el lugar

2.8 Android Studio.

Android Studio es una herramienta que se define como el entorno de desarrollo integrado que usa Android como plataforma oficial. En el 2013, en la conferencia Google I/O se anunció el reemplazo de Eclipse como el IDE oficial por Android Studio para las aplicaciones de Android. El software de Android Studio está basado en IntelliJ IDEA de JetBrains y se lo puede descargar de forma gratuita a través de Apache 2.0. Se encuentra disponible para Microsoft Windows, Linux y macOS. Entre sus tantas características, tiene renderizado en tiempo real, integración de proguard y funciones de firma de aplicaciones, consejos de optimización, ayuda para traducir, estadísticas

de uso, cuenta con un emulador de Android en el que se puede ejecutar la aplicación y probarla, tiene varias plantillas, de las que se puede partir para elaborar una aplicación deseada, además cuenta con apoyo para la construcción en Gradle, herramientas Lint que logra detectar problemas de rendimiento, uso, versiones compatibles, entre otros.

Android Studio se puede usar para Windows 2003, Vista, 7, 8 , y 10 en plataformas de 32 y 64 bits, para Linux con GNOME o KDE y como mínimo memoria RAM de 2GB, y para macOS a partir desde 10.8.5. Entre sus requerimientos para Windows se necesita que su memoria RAM sea de más de 8GB para que funcione con normalidad, 3GB como mínimo y 16GB que es el recomendado, sin embargo, menor a 8GB funciona con bastante lentitud. Previo a su uso se debe descargar Java Development Kit (JDK), y contar como mínimo 500 de espacio en el disco para Android Studio, y por lo menos 1.5 para Android SDK, imágenes de sistema emulador y cachés.



Figura 17. Logo Android Studio
Fuente: (Android Studio, 2017)

2.9 Matlab.

Matlab es una herramienta que sirve como un entorno de desarrollo integrado de software matemáticos, utiliza un lenguaje de programación propio. MATLAB es una palabra compuesta que en realidad proviene de MATrix y LABoratory, que significa laboratorio de matrices. Entre las prestaciones básicas que brinda Matlab encontramos la posibilidad de manipular matrices, representar datos, funciones, desarrollar algoritmos, la creación de interfaces de usuario (GUI) y comunicación con otros programas, en sus propios lenguajes y además hacer uso de otros dispositivos hardware.

Matlab ofrece en su paquete dos herramientas que aumentan sus prestaciones, las cuales son Simulink, la cual es una plataforma que permite simulaciones multi-

dominio mediante diagrama de bloques, y GUIDE que permite editar las interfaces del usuario, es decir las GUI. También es posible ampliar sus utilidades mediante las cajas de herramientas y Simulink con paquetes de bloques.

Es uno de los softwares más utilizados en el mundo por universidades y centros de investigación, ha logrado en sus últimos años aumentar el número de prestaciones con las que cuenta, es decir, sus capacidades, por ejemplo, es posible programar de forma directa procesadores digitales de señal o incluso crear un código VHDL.

El lenguaje de programación que usa MATLAB es propio, este lenguaje cuenta como la posibilidad de interpretarse y ejecutarse en el entorno interactivo, como por medio de un archivo scprit.

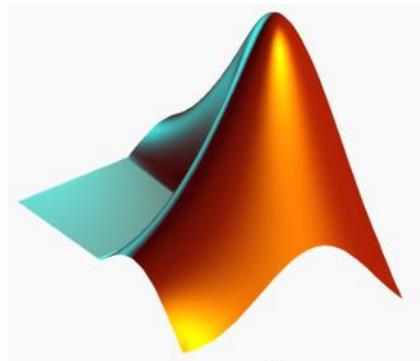


Figura 18. Logo de MATLAB
Fuente: (MathWorks, 2017)

2.10 NodeMcu ESP8266.

El NodeMcu ESP8266 es un dispositivo nuevo que permite la conexión a Internet por medio de Wi-Fi, es una pequeña placa, muy parecida a Arduino, se la puede programar en lenguaje Arduino por medio de la interfaz Arduino IDE y posee Wi-Fi embebido. En los últimos años han aparecido un sinnúmero de nuevos dispositivos y placas de desarrollo muy similares a Arduino pero que poseen otras capacidades, como son Wi-Fi, o bluetooth.

Actualmente en el mercado existe una nueva línea de software de código abierto denominada NodeMcu. Existen dispositivos de diferentes modelos en el mercado que basan su tecnología en el ESP8266, variando solamente los números de pines de entradas y salidas, y la cantidad de memoria interna. En el caso del NodeMcu,

este dispositivo fue diseñado para programarse mediante LUA, sin embargo es compatible y se la puede programar también median Microphyton o con Arduino.(Rodrigo, 2016)

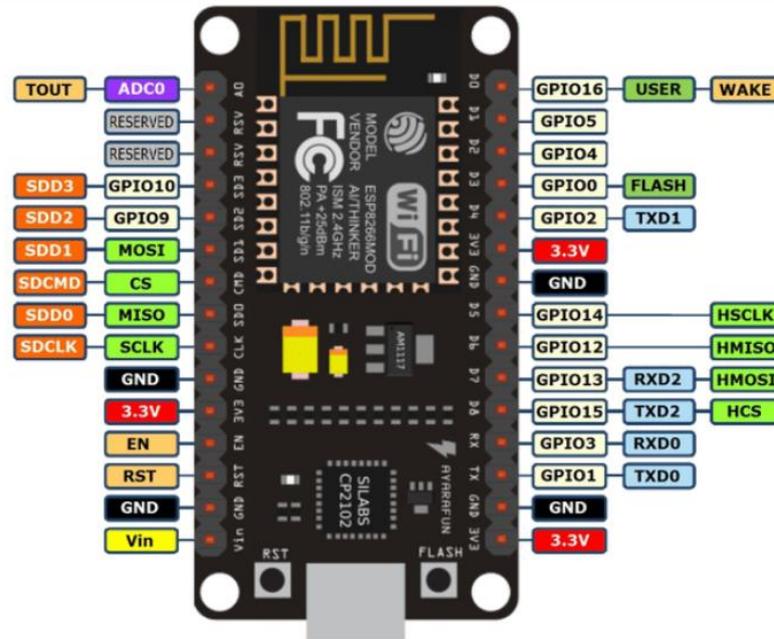


Figura 19. NodeMCU ESP8266
Fuente: (Rodrigo, 2016)

La plataforma de conectividad inteligente de Espressif Systems (ESCP) es un conjunto de alto rendimiento y alta integración SOC inalámbricos, diseñados para diseñadores de plataformas móviles con limitaciones de espacio y energía. Proporciona habilidad insuperable para integrar capacidades de Wi-Fi dentro de otros sistemas, o para funcionar como un programa de aplicación independiente, con el menor costo y el requisito de espacio mínimo. (Espressif Systems, 2015)

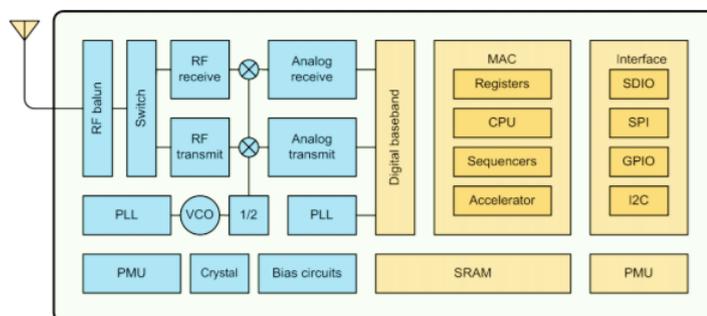


Figura 20. Diagrama de Bloque del esp8266
Fuente: (Espressif Systems, 2015)

ESP8266EX ofrece una solución de red Wi-Fi completa e independiente; se puede usar para alojar la aplicación o para descargar funciones de red Wi-Fi de otro procesador de aplicaciones.

Cuando ESP8266EX sube la aplicación, se inicia directamente desde un flash externo. Donde se ha integrado caché para mejorar el rendimiento del sistema en dichas aplicaciones. De forma alternativa, sirve como un adaptador Wi-Fi, donde puede acceder de forma inalámbrica a Internet cualquier microcontrolador basado en diseño con conectividad simple (interfaz SPI / SDIO o I2C / UART).

ESP8266EX es uno de los chips Wi-Fi más integrados en la industria; integra la antena, interruptores, balun de RF, amplificador de potencia, amplificador de recepción de bajo ruido, filtros, módulos de administración de energía, requiere un circuito externo mínimo, y la solución completa, incluido el módulo frontal, está diseñada para ocupar el área mínima de PCB.

ESP8266EX también integra una versión mejorada del procesador de 32 bits de la serie L106 Diamond de Tensilica, con SRAM en chip, además de las funcionalidades WiFi. ESP8266EX a menudo se integra con dispositivos externos sensores y otros dispositivos específicos de aplicaciones a través de sus GPIO; códigos de muestra para tales aplicaciones se proporcionan en el kit de desarrollo de software (SDK). (Espressif Systems, 2015)

2.10.1 MCU.

ESP8266EX está integrado con el microcontrolador Tensilica L106 de 32 bits (MCU), que cuenta con un nivel extra de bajo consumo de energía y RSIC de 16 bits. La velocidad del reloj de la CPU es de 80MHz. También puede alcanzar un máximo valor de 160MHz. El sistema de operación en tiempo real (RTOS) está habilitado. Actualmente, solo el 20% de MIPS ha sido ocupado por la pila Wi - Fi, el resto puede usarse para la programación de aplicaciones de usuario y desarrollo. (Espressif Systems, 2015)

Las siguientes interfaces se pueden usar para conectarse a la MCU integrada en el ESP8266EX:

- Interfaces RAM / ROM programables (iBus), que se pueden conectar con el controlador de memoria, y también se puede utilizar para visitar flash externo;
- Interfaz de RAM de datos (dBus), que puede conectarse con el controlador de memoria
- Interfaz AHB, se puede utilizar para visitar el registro.

2.10.2 SRAM y ROM Interna.

ESP8266EX WiFi SoC está integrado con un controlador de memoria, que incluye SRAM y ROM. MCU puede visitar las unidades de memoria a través de las interfaces iBus, dBus y AHB. Todas las unidades de memoria pueden visitarse por solicitudes, mientras que un árbitro de memoria decidirá la secuencia de ejecución de acuerdo con el tiempo cuando las solicitudes son recibidas por el procesador.

De acuerdo con nuestra versión actual de SDK provista, se asigna espacio SRAM que está disponible para los usuarios como a continuación:

- Tamaño de RAM <36kB, es decir, cuando ESP8266EX está trabajando bajo el modo de estación y es conectado al enrutador, el espacio programable accesible para el usuario en la sección de almacenamiento y datos es alrededor de 36kB).
- No hay ROM programable en el SoC, por lo tanto, el programa de usuario debe almacenarse en un flash SPI externo.

Capítulo 3: Desarrollo del proyecto.

3.1 Funcionamiento.

El siguiente capítulo del trabajo de titulación está ideado para desarrollar una simulación e implementación de un sistema domótico, es decir una casa inteligente a escala, logrando controlar las luces, temperatura, puerta del garaje y persianas de toda la casa por medio de una interfaz desarrollada en Matlab y Android Studio para smartphones.

El diseño de la interfaz para smartphones, cuenta con nombre de usuario, y contraseña para acceder a tu vivienda específicamente, y una interfaz en Matlab, tipo panel de control, se pueden observar y controlar todas las variables, pudiendo acceder a su control con un solo click, se puede encender o apagar luces, abrir o cerrar puertas y persianas, e incluso controlar la temperatura encendiendo o apagando el sistema de climatización de forma manual, o de forma autónoma donde el sistema es capaz de tomar esta decisión por sí mismo. Para la forma autónoma, el funcionamiento consiste en ingresar la temperatura deseada, para lo cual, el sistema de climatización se mantendrá encendido hasta que se alcance esta temperatura (trabaja a través de rangos de temperatura).

La interacción entre las interfaces de Matlab y Android con el sistema se puede realizar desde cualquier parte del mundo, debido a que el medio de comunicación del sistema es WI-FI por cualquiera de los protocolos TCP/IP y basta con que las interfaces tengan acceso a internet 3G, 4G, o WI-FI para poder controlar el sistema. Se ha trabajado con una base de datos en FireBase, la misma que permite almacenar los datos, y enviar y recibir información.

Las interfaces envían el dato al servidor y el servidor transmite este dato al NodeMcu Esp8266 y el mismo controla el sistema, esta acción está ocurriendo internamente de forma repetitiva. El servidor logra escribir y leer datos, por ejemplo en el caso de la temperatura, la misma puede ser visualizada a través de las interfaces.

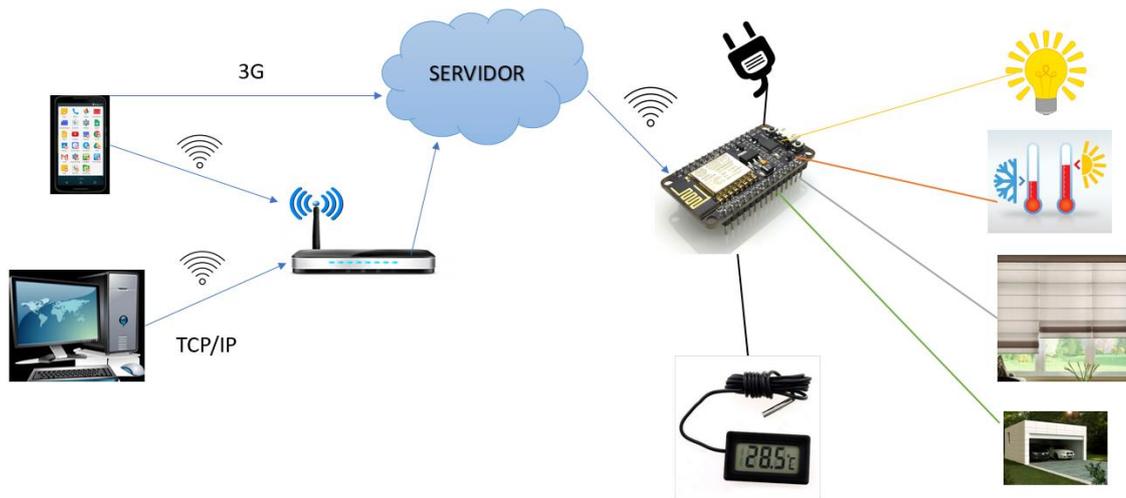


Figura 21 . Diagrama del funcionamiento del sistema
Elaborado por: Autora

3.2 Diseño del Hardware.

Para el diseño de hardware se realizó una maqueta de escala 1:20 de una casa, que cuenta con dos habitaciones, una master y una más pequeña, sala, comedor, cocina, garaje, y un patio frontal. El circuito instalado pasa su cableado a través de las paredes así se mantiene una estética en la maqueta. Todas las variables controladas, es decir, luces, persianas, temperatura y garaje van conectadas al NodeMcu ESP8266. El mismo está situado dentro la casa, de forma oculta. Se han dividido el diseño tanto de la maqueta como de la electrónica, para desarrollar una explicación más elaborada de cada uno.

3.2.1 Materiales.

Se ha trabajado para el desarrollo del hardware con los siguientes materiales:

- Dispositivo NodeMcu ESP8266
- Leds de alta potencia 1W
- Motor DC
- Reductor de velocidad
- Sensores digitales DS18B20
- Ventiladores DC
- Servomotor
- Sensor analógico LM35
- Resistencias 1k Ω

- Transistores 2N2222
- Puente L293D

Se han instalado 6 leds de alta potencia de la siguiente manera:

- En el master se ha conectado en el pin D0, en el dormitorio se ha conectado en el pin D1, los tres siguientes se han colocado uno en la sala, uno en el comedor y uno en la cocina, estos tres conectados en paralelo, por lo cual se encienden desde un mismo pin que es el D2, y los leds restantes también han sido conectados en serie al pin D3, y simulan la iluminación del exterior frontal de la casa.

El motor DC se lo ha utilizado para simular la puerta del garaje, este motor se encuentra conectado a un puente L293D, el cual se utiliza para cambiar la polaridad del motor, para que permita la simulación de apertura y cierre de la puerta, una de las salidas al motor se ha conectado al pin D4, y la otra al pin D5.

Los dos sensores digitales se han conectado de forma conjunta, en el pin D6, gracias a la programación se logra que ambos lean por separado siendo el primero el sensor del master, y el segundo el sensor del dormitorio.

Por otro lado tenemos 3 ventiladores, los cuales se han conectado de la siguiente manera:

- En la sala se ha conectado un ventilador grande, el cual ha sido conectado de forma independiente al pin D7. En el master se ha colocado un ventilador pequeño, el cual ha sido conectado en paralelo a otro, de las mismas características, en el dormitorio. Ambos ventiladores se han conectado al pin D8, lo que indica que estos últimos trabajaran de forma simultánea, lo que significa que si uno se enciende, el otro también lo hará.

Las persianas son controladas por un servomotor, el cual ha sido conectado en el pin D9. El sensor analógico se conecta en el pin A0, y se lo utiliza para medir la temperatura de la sala.

3.2.2 Diseño de la maqueta.

Debido a varios factores, entre ellos el costo de implementación en una residencia en construcción o construida, se elaboró una maqueta de una casa a escala 1:20, la que cuenta con dos cuartos, sala, comedor, cocina, garaje y exteriores. Se ha diseñado la casa en SketchUp y posterior a esto se realizó la maqueta real, la parte frontal de la casa se la ha ideado con vidrios y ventanas grandes para poder apreciar las variaciones de las luces dentro de la casa, es importante poder visualizar todo lo que ocurre en su interior. Las paredes para que la casa tenga una mejor resistencia se la ha elaborado con lamina foam, y se la ha recubierto con cartulina plegable. Lo mismo en la parte del techo, para colocar las luces leds, se definió el lugar y se realizó un orificio para cada led. En el caso de los ventiladores se siguió el mismo procedimiento, logrando que los tres ventiladores queden en el techo y se puedan observar desde la parte superior de la maqueta.



Figura 22. Diseño de la distribución en Sketch Up de la casa a escala

Elaborado por: Autora

3.2.3 Diseño del circuito.

El diseño del circuito que se utilizó para la simulación de un sistema de domótica, se lo ha realizado en FRITZING para posteriormente fabricarlo en una PCB, logrando una mejor estética del mismo. Se ha trabajado con una fuente de alimentación de 12 voltios, 5 voltios. Además, para una mejor explicación de cada parte de nuestro sistema se lo ha subdividido en tres circuitos que son: la iluminación, ventilación y

temperatura, y por último la parte de los servo-motores y motor DC, sin embargo, los tres están conectados a la misma tarjeta NodeMcu ESP8266, es por esto que se ha colocado en primer lugar la distribución y el diseño de la PCB de los tres circuitos convergidos en una sola placa.

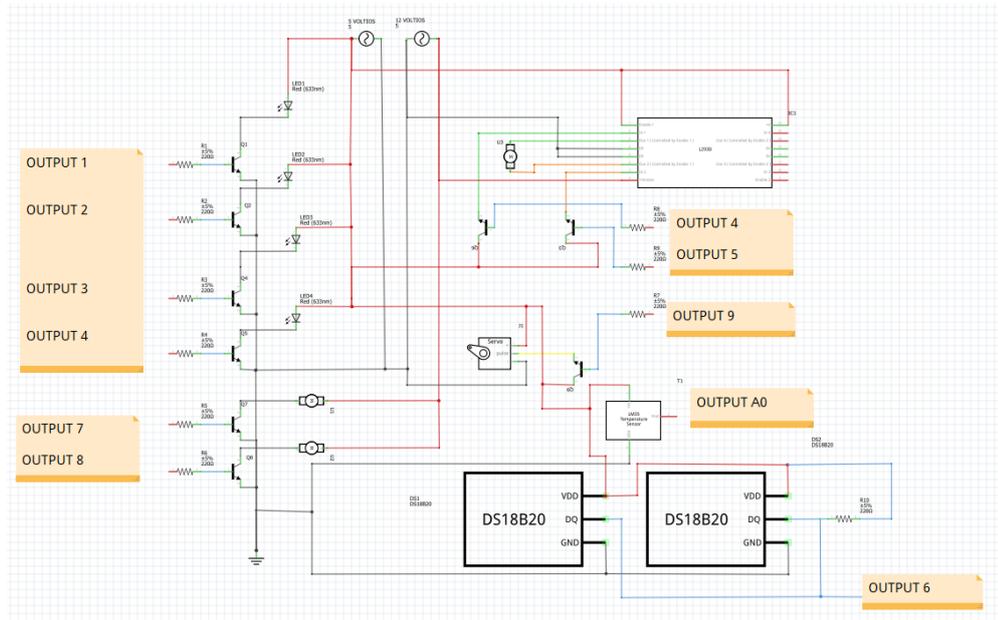


Figura 23. Esquemático del circuito
Elaborado por: Autora

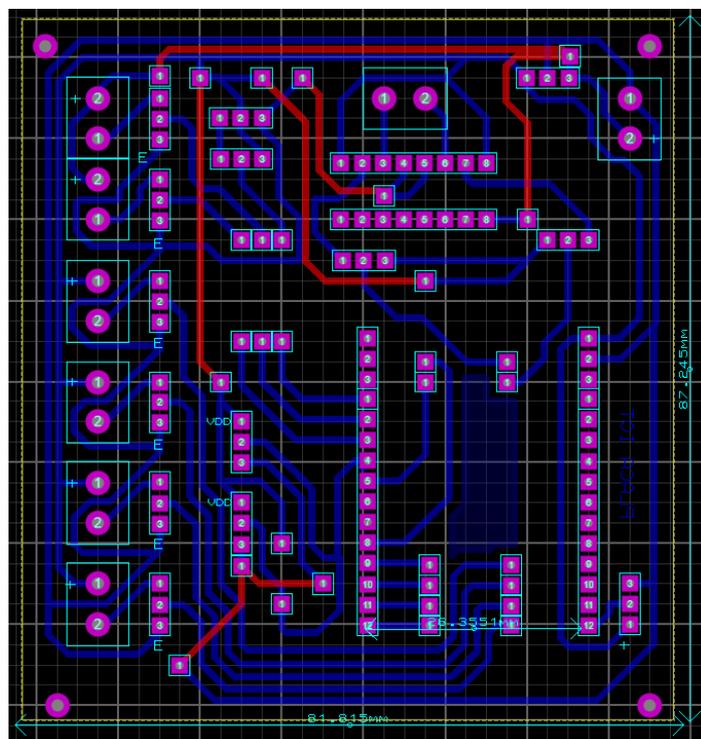


Figura 24. Diseño de la PCB
Elaborado por: Autora

3.2.3.1 Iluminación.

El desarrollo de este circuito se inició estableciendo los pines que se utilizarían, para el master es el pin D0, para el dormitorio es el pin D1, para la sala, comedor y cocina es el pin D2, y por ultimo para el exterior frontal es el pin D3. Cabe recalcar que solo para el master y el dormitorio se ha utilizado leds independientes, en los dos restantes se han colocado 3 en paralelo y 2 en paralelo respectivamente. Para la conexión, se utilizó un transistor 2N2222, donde el emisor se conecta al led, la base se conecta a una resistencia de $1k \Omega$ y aquí se conecta el pin respectivo del NodeMcu esp8266, y el colector se conecta a una fuente de 5V.

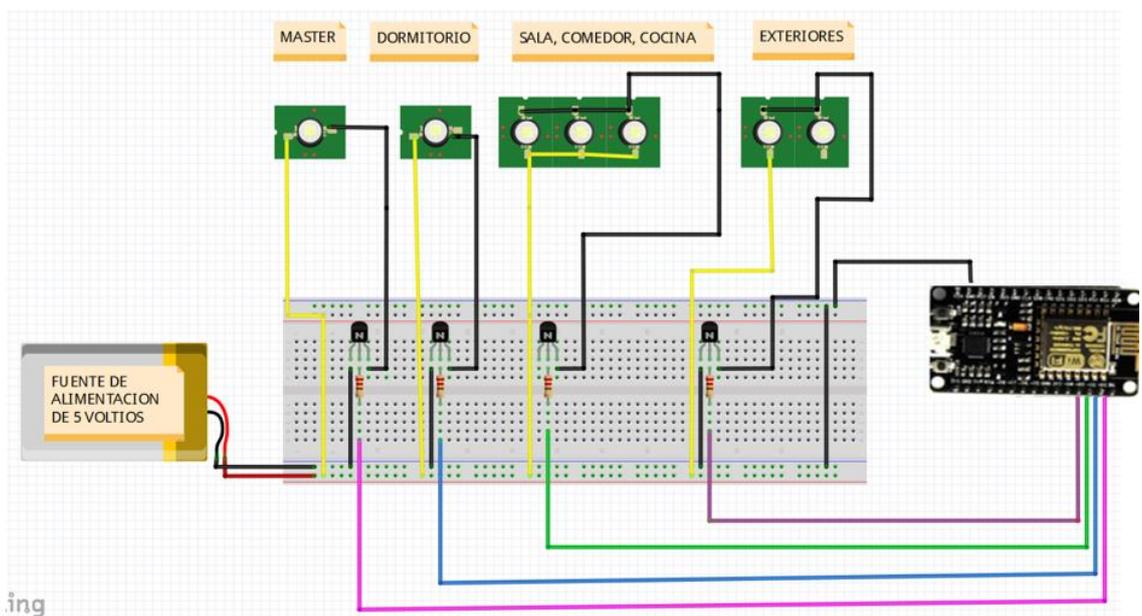


Figura 25. Circuito de iluminación

Elaborado por: Autora

3.2.3.2 Temperatura y ventilación.

Para el desarrollo de este circuito se adicióno un transistor 2N2222 para los ventiladores, y cada uno con una resistencia de $1k \Omega$. Los ventiladores se han conectado en paralelo al pin D7. Por otro lado tenemos el ventilador conectado al pin D8, el cual simula la climatización en la sala. Se utilizó una fuente de voltaje de 12V para alimentar a los tres ventiladores.

Para medir la temperatura se utilizó sensores digitales DS18B20 estos sensores se han conectado en el pin D6 del NodeMcu esp8266. Además, para el área de la sala, comedor y cocina se utilizó un sensor LM35, el mismo que se ha conectado al pin A0, debido a que este pin es el único que permite una entrada analógica.

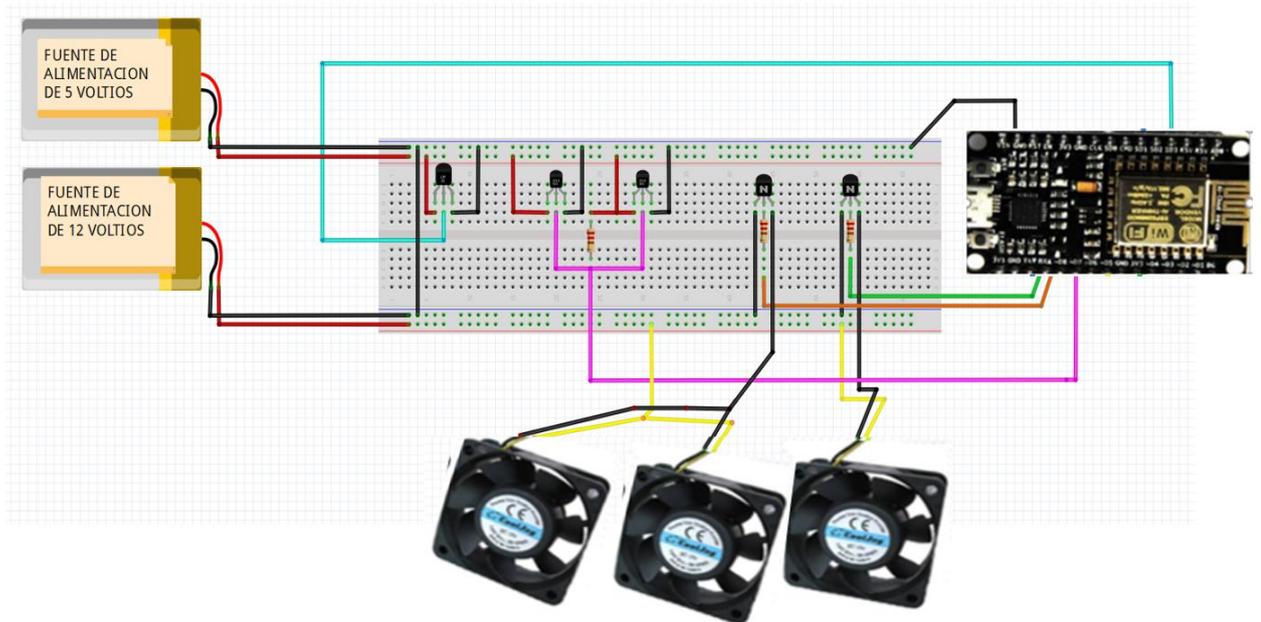


Figura 26. Circuito de climatización y temperatura
Elaborado por: Autora

3.2.3.3 Servo motor y motor dc.

Para el desarrollo de este circuito se añaden dos transistores 2N2222 para el motor DC, y uno más para el servo-motor, cada uno con su respectiva resistencia. El motor DC, además como se requiere su cambio de polaridad para la simulación de apertura y cierre de la puerta, ha sido conectado a un puente L293D, el mismo ha sido conectado como se muestra en la imagen. Además, para alimentar a ambos motores, se ha utilizado una fuente de 5V, los pines designados para el motor son D4 y D5, y el pin D10 para el servo-motor.

El servo-motor sirve para la simulación de las persianas, gracias a su movimiento de 180 grados, se puede apreciar la elevación y bajada de las persianas. El motor DC simula una puerta enrollable, conectado a la misma por un tipo de mecanismo realizado.

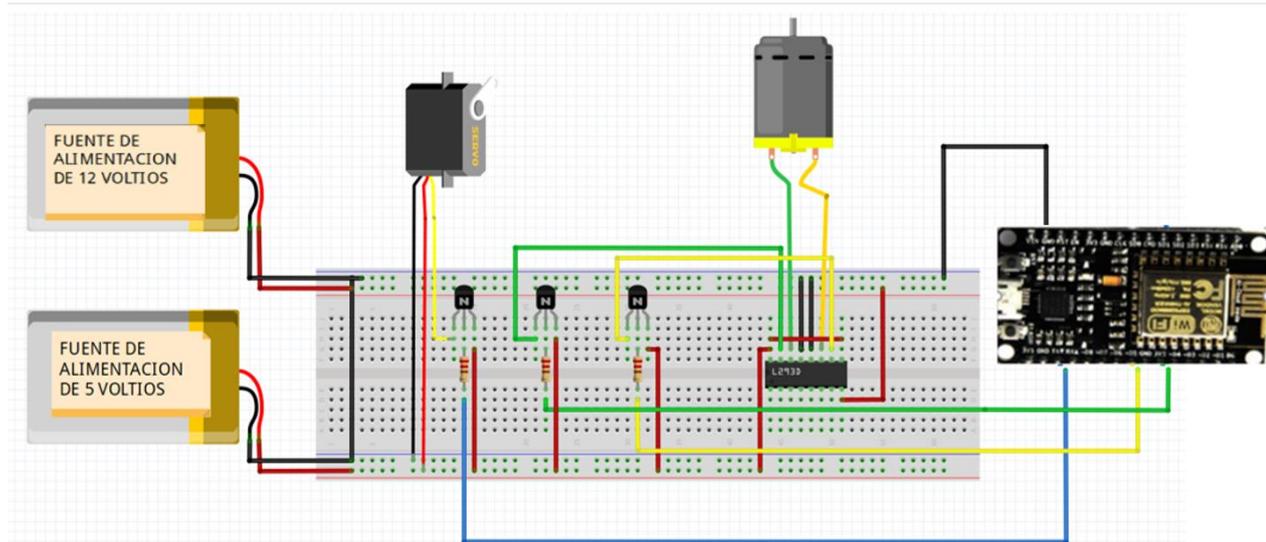


Figura 27. Circuito de servomotor para persianas y motor dc para puerta
Elaborado por: Autora

3.3 Diseño del software.

Para el diseño de software se trabajó en Android Studio y Matlab donde se realizaron dos interfaces, una que podrá usarse desde la computadora y la otra desde un teléfono Android, pero ambas comunicadas por medio de Wi-Fi. La programación del dispositivo NodeMcu ESP8266, se la realizó en el IDE de Arduino, como ya se ha mencionado este dispositivo cuenta con conexión Wi-Fi, la misma que permiten el control en tiempo real por medio de las interfaces. Matlab, Android Studio y el software del NodeMcu ESP8266 están interconectadas al servidor FireBase que permite almacenar datos e información en la nube. Así se logra que las tres puedan trabajar en conjunto, y el dispositivo NodeMcu ESP8266 pueda recibir órdenes de las interfaces. El desarrollo de la interfaz en Matlab es un panel de control, por otro lado, para la interfaz diseñada en Android Studio se ha desarrollado algo más elaborado dando como resultado una aplicación.

A continuación, se explica cada una de estas interfaces junto a su respectiva programación, así mismo el software del dispositivo NodeMcu ESP8266. En esta sección se ha decidido dividir cada parte del software completo, para lograr una explicación más a detalle de cada paso realizado para el diseño final del software de nuestro control.

3.3.1 Diseño del software en Android Studio.

El diseño del software en Android Studio se trata de una aplicación, la cual cumple y cubre con todas las variables a controlar propuestas. El lenguaje de programación que se utiliza es JAVA, y se trabaja con dos IDE de programación, uno que es el del diseño de la aplicación, y el otro donde se le agregan las funciones a los botones, textos, etc. Para poder darle inicio al diseño de nuestra aplicación se debe conectar la aplicación con firebase, esto se realiza siguiendo varios pasos que el servidor provee. Una vez realizado esto se puede iniciar con la parte del diseño, la cual es sumamente sencilla, se la puede hacer de dos formas, escribiendo en el código, o arrastrando y moviendo los botones o textos de forma gráfica. Luego de definir el diseño final de la aplicación, se procede a programar y es aquí donde se utiliza el lenguaje JAVA, la programación es básicamente hacer que los botones de la aplicación envíen información al servidor. La aplicación final contiene una ventana de bienvenida, la misma que da acceso al inicio de sesión, y en la parte superior izquierda da la opción de registro haciendo de la aplicación dinámica.

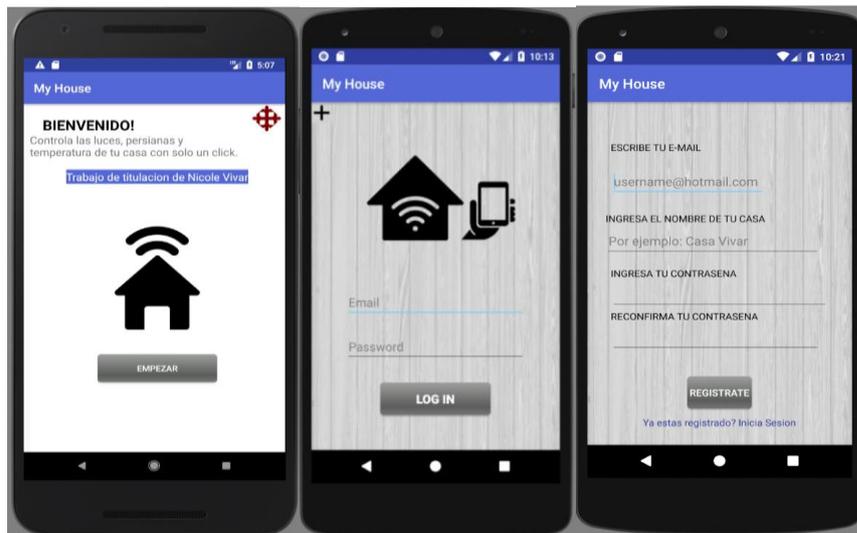


Figura 28. Ventana de bienvenida, inicio de sesión y registro de la aplicación
Elaborado por: Autora

Cuando se ingresa un correo no registrado, aparece un mensaje diciéndole al usuario que no está registrado, además si se deja un campo vacío de igual forma solicita al usuario que ingrese lo que hace falta, y por último si el usuario ha ingresado mal su contraseña le dice que la contraseña es incorrecta.

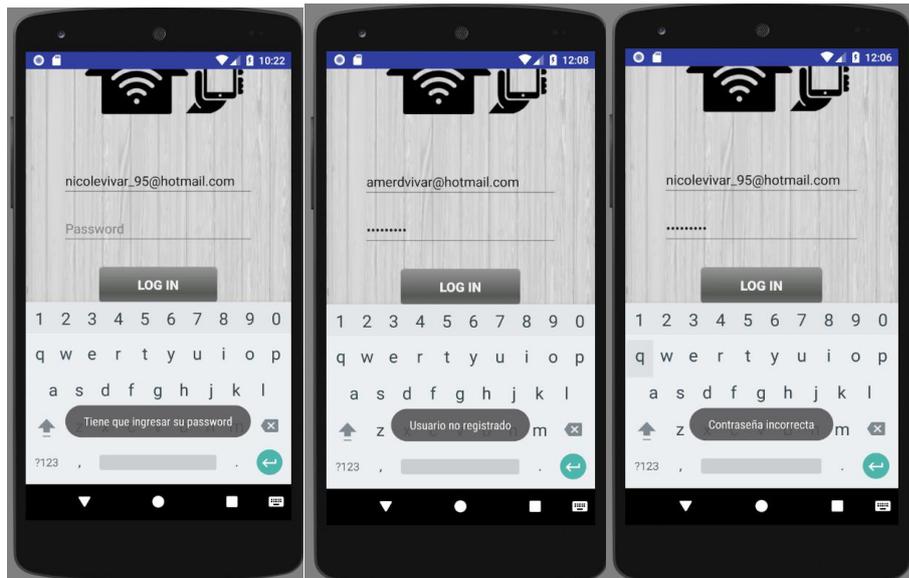


Figura 29. Mensajes de error.
Elaborado por: Autora

Para la ventana de registro sucede algo parecido en cuanto a dejar sin llenar los campos vacios, saldra un mensaje de error pidiendole al usuario que ingrese el correcto. Una vez que se registra el usuario te redirige de forma inmediata al menú de la casa. Se pudo haber hecho una aplicación mas dinamica, en cuanto a setear el menú según lo que se desee utilizar, pero por cuestion de tiempo, se ha dejado establecido el menú con las cuatro variables a controlar como son iluminación, temperatura, persianas y puerta del garaje.

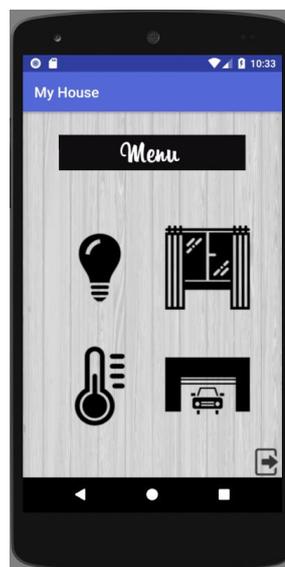


Figura 30. Menú de la aplicación.
Elaborado por: Autora

Esta parte cuenta con un botón de log out, en la parte inferior derecha, lo que hace que el usuario cierre sesión y lo regresa automáticamente a la ventana de inicio

de sesión. En este menú se seleccionará lo que se desee controlar, en la sección de iluminación, tenemos lo que es sala, comedor, cocina, dormitorios y exteriores, en las persianas un botón controla las persianas de la sala, con el garaje un botón controla la puerta, y por último en la temperatura podemos ingresar el valor deseado y encender y apagar el ventilador de forma manual o automática para alcanzar este valor.

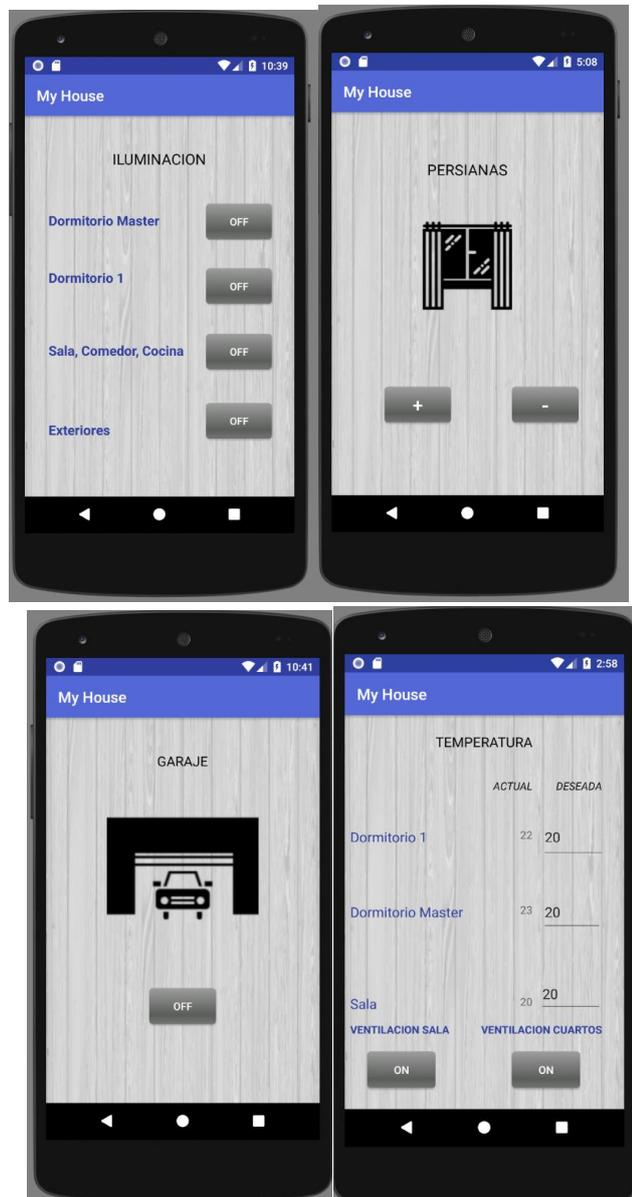


Figura 31. Ventanas de las variables
Elaborado por: Autora

En paralelo a todo esto Android Studio trabaja con Firebase, le envía los datos según se vayan escribiendo, por ejemplo, cada que presionamos un botón en la base de datos se guarda en una variable designada un valor, como en la iluminación cuando se activa o se enciende una luz, el servidor escribe el número 1 en el estado de la

habitación que se haya presionado, y escribe un 0 cuando esta se desactiva. Como se puede observar en la figura 19. En el caso de la temperatura, estos valores se escriben cuando presionamos el botón de activación de la ventilación. En el caso de las persianas, debido a que se trabaja con servo-motores se han utilizado dos botones, uno con el signo más (+), lo que significa que abre la persiana, y con el signo menos (-), que significa que la va cerrando.

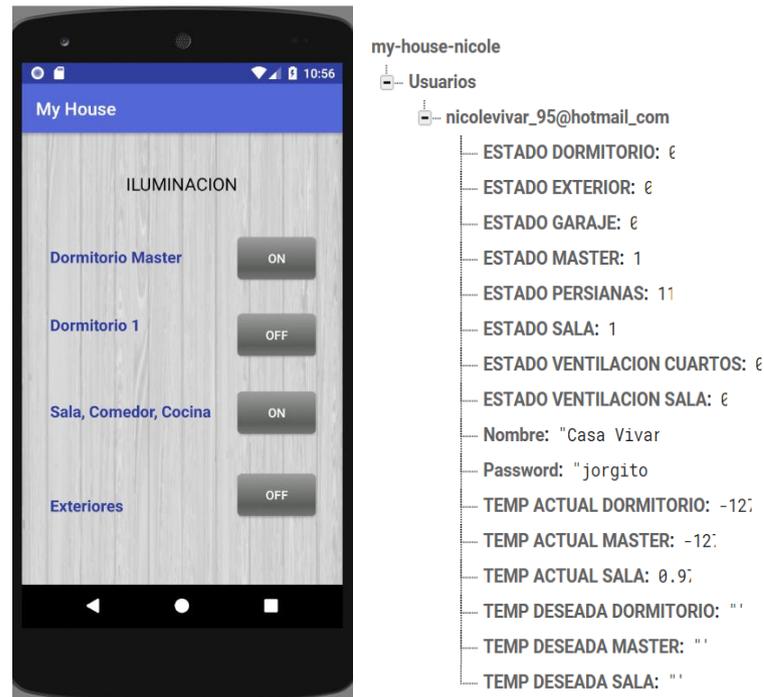


Figura 32. Interacción entre Android Studio y Firebase
Elaborado por: Autora

Para la programación de la aplicación en Android Studio, se utiliza lenguaje JAVA, el código para las distintas ventanas posee un faceplate que se crea a medida que se va elaborando cada diseño.

1. La primera ventana que se observa en la aplicación, es la de bienvenida al usuario, donde solo se programó el botón de empezar, el cual lo redirige hacia el inicio de sesión.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button btEmp = findViewById(R.id.btEmp);
    btEmp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent btEmp = new Intent( packageContext: MainActivity.this,Main3Activity.class);
            startActivity(btEmp);
        }
    });
}
}

```

2. En la ventana de inicio de sesión, se definen las variables, y se procede a programarlas. Se llama a los botones y a las funciones que se usan para conectarlo al servidor firebase, se desarrolla las siguientes condiciones, donde indica que si el usuario está registrado al presionar el botón Login, se dirija a la ventana de menú. En la parte superior izquierda, se ha programado el botón MÁS (+), el mismo que sirve para cuando el usuario no se encuentra registrado y proceda a hacerlo.

```

public class Main3Activity extends AppCompatActivity {
    private EditText textName;
    private EditText textPas;
    private Button btLogin;
    public ImageButton Mas;
    private FirebaseAuth mAuth;
    private FirebaseAuth.AuthStateListener mAuthListener;
    private ProgressDialog progressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main3);

        mAuth = FirebaseAuth.getInstance();
        textName = (EditText) findViewById(R.id.textName);
        textPas = (EditText) findViewById(R.id.textPas);
        btLogin = (Button) findViewById(R.id.btLogin);
        Mas = (ImageButton) findViewById(R.id.Mas);
        progressDialog = new ProgressDialog( context: this);

        mAuthListener = (AuthStateListener) (firebaseAuth) -> {
            if(firebaseAuth.getCurrentUser() != null){
                Intent btLogin = new Intent( packageContext: Main3Activity.this, menu.class);
                startActivity(btLogin);
            }
        };

        Mas.setOnClickListener((View) -> {
            Intent Mas = new Intent( packageContext: Main3Activity.this, Main2Activity.class);
            startActivity(Mas);
        });

        btLogin.setOnClickListener((view) -> { Login(); });
    }
}

```

3. Se declara la función de inicio, y a continuación se desarrolla la función Login, a la misma se le han puesto varias condiciones, para que muestre

mensajes de error. Si estas condiciones no se cumplen, se mostrara un mensaje de espera.

```
@Override
3 protected void onStart() {
    super.onStart();
    mAuth.addAuthStateListener(mAuthListener);
3 }

3 private void Login() {
    String email = textName.getText().toString();
    String password = textPas.getText().toString();

    if (TextUtils.isEmpty(email) && TextUtils.isEmpty(password)) {
        Toast.makeText(context, this, text: "Tiene que ingresar su email y password", Toast.LENGTH_SHORT).show();
        return;
    }

    if (TextUtils.isEmpty(email)) {
        Toast.makeText(context, this, text: "Tiene que ingresar su email", Toast.LENGTH_SHORT).show();
        return;
    }

    if (TextUtils.isEmpty(password)) {
        Toast.makeText(context, this, text: "Tiene que ingresar su password", Toast.LENGTH_SHORT).show();
        return;
    }

    progressDialog.setMessage("Por favor espere...");
    progressDialog.show();
}
```

4. Se crea la función donde se comprueba que el inicio de sesión es exitoso, y lo lleva al usuario al menú si cumple con los parámetros para hacerlo. De modo contrario muestra los mensajes de error correspondientes.

```
mAuth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener( activity: this, (task) -> {
        progressDialog.dismiss();
        if (task.isSuccessful()) {
            Intent btLogin = new Intent( packageContext: Main3Activity.this, menu.class);
            startActivity(btLogin);
        }
    }).addOnFailureListener( activity: this, (e) -> {
        if (e.getMessage().contains("There is no user record")) {
            Toast.makeText(context: Main3Activity.this, text: "Usuario no registrado", Toast.LENGTH_SHORT).show();
            return;
        } else {
            Toast.makeText(context: Main3Activity.this, text: "Contraseña incorrecta", Toast.LENGTH_SHORT).show();
            return;
        }
    });
}
```

5. Para el registro, el procedimiento es similar, se agregan las variables y botones que van a usarse, se los llama, y se procede a crear las funciones de cada botón. Se colocan varias condiciones para mostrar mensajes de error, si esto no sucede aparece un mensaje donde avisa que el usuario está siendo registrado, y si la tarea es exitosa se crea el usuario en la base de datos, y lo lleva hacia el menú directamente. Caso contrario, mostrara un mensaje de error.

```

if (TextUtils.isEmpty(email) || TextUtils.isEmpty(houseName) || TextUtils.isEmpty(password) || TextUtils.isEmpty(confirmacion)) {
    Toast.makeText(context, this, text: "Tiene que llenar todos los campos", Toast.LENGTH_SHORT).show();
    return;
}

if (password.equals(confirmacion)) {} else {
    Toast.makeText(context, this, text: "LAS CONTRASENAS NO COINCIDEN", Toast.LENGTH_SHORT).show();
    return;
}

progressDialog.setMessage("Registrando Usuario");
progressDialog.show();
 mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener((task) -> {

    if (task.isSuccessful()) {

        String user_id = mAuth.getCurrentUser().getEmail();
        user_id = user_id.replace( oldChar: '.', newChar: '_');
        DatabaseReference current_id = mDataBase.child(user_id);
        current_id.child("Password").setValue(password);
        current_id.child("Nombre").setValue(houseName);
        progressDialog.dismiss();
        Toast.makeText(getApplicationContext(), text: "SE HA REGISTRADO CON EXITO", Toast.LENGTH_SHORT).show();
    }
    else {
        progressDialog.dismiss();
        Toast.makeText(getApplicationContext(), text: "ESTA CUENTA YA EXISTE", Toast.LENGTH_SHORT).show();
    }
});
});
}

```

6. Para el menú, luego de haber definido las variables y botones. Se les da función a los mismos, se llama a la base de datos para el botón de logout, para que se pueda cerrar sesión. En el menú se observaran las variables a controlar. En este caso, luces, temperatura, garaje y persianas, de modo que lleva al usuario a la ventana que haya seleccionado.

```

 mAuth = FirebaseAuth.getInstance();

 btLogout = (ImageButton) findViewById(R.id.btLogout);
 btLogout.setOnClickListener((view) -> {
     mAuth.signOut();
     finish();
     Intent textLogout = new Intent ( packageContext: menu.this, Main3Activity.class);
     startActivity(textLogout);
 });

 btiluminacion = (ImageButton) findViewById(R.id.btiluminacion);
 btiluminacion.setOnClickListener((view) -> {
     Intent btiluminacion = new Intent( packageContext: menu.this, Main4iluminacion.class);
     startActivity(btiluminacion);
 });

 btpersianas = (ImageButton) findViewById(R.id.btpersianasonoff);
 btpersianas.setOnClickListener ((View) -> {
     Intent btpersianas = new Intent ( packageContext: menu.this, Main4Persianas.class);
     startActivity(btpersianas);
 });
 bttemperatura = (ImageButton) findViewById(R.id.bttemperatura);
 bttemperatura.setOnClickListener ((View) -> {
     Intent bttemperatura = new Intent ( packageContext: menu.this, Main4Temperatura.class);
     startActivity(bttemperatura);
 });

 btgaraje = (ImageButton) findViewById(R.id.btgaraje);
 btgaraje.setOnClickListener ((View) -> {
     Intent btgaraje = new Intent ( packageContext: menu.this, Main4GARAJE.class);
     startActivity(btgaraje);
 });
}

```

7. Para el caso de la iluminación, la programación es la misma, definir variables y botones. Solo se agrega, para que al presionar el botón se genere un cambio en la variable asignada en el servidor Firebase.

```
btmaster.setOnClickListener((view) -> {
    boolean checkedmaster = btmaster.isChecked();
    String user_id = mAuth.getCurrentUser().getEmail();
    user_id = user_id.replace( oldChar: '.', newChar: '_' );
    DatabaseReference current_id = mDataBase.child(user_id);
    if (checkedmaster) {
        current_id.child("ESTADO MASTER").setValue(1);
    } else {
        current_id.child("ESTADO MASTER").setValue(0);
    }
});

btcuarto.setOnClickListener((view) -> {
    boolean checkedcuarto = btcuarto.isChecked();
    String user_id = mAuth.getCurrentUser().getEmail();
    user_id = user_id.replace( oldChar: '.', newChar: '_' );
    DatabaseReference current_id = mDataBase.child(user_id);
    if (checkedcuarto){
        current_id.child("ESTADO DORMITORIO").setValue(1);
    } else {
        current_id.child("ESTADO DORMITORIO").setValue(0);
    }
});

btsala.setOnClickListener((view) -> {
    boolean checkedsala = btsala.isChecked();
    String user_id = mAuth.getCurrentUser().getEmail();
    user_id = user_id.replace( oldChar: '.', newChar: '_' );
    DatabaseReference current_id = mDataBase.child(user_id);
    if (checkedsala) {
        current_id.child("ESTADO SALA").setValue(1);
    }else {
        current_id.child("ESTADO SALA").setValue(0);
    }
});

btexterior.setOnClickListener((view) -> {
    boolean checkedexterior = btexterior.isChecked();
    String user_id = mAuth.getCurrentUser().getEmail();
    user_id = user_id.replace( oldChar: '.', newChar: '_' );
    DatabaseReference current_id = mDataBase.child(user_id);
    if (checkedexterior) {
        current_id.child("ESTADO EXTERIOR").setValue(1);
    }else {
        current_id.child("ESTADO EXTERIOR").setValue(0);
    }
});
}
}
```

8. Este procedimiento se repite para el caso del garaje. Sin embargo para las secciones de persianas existe una diferencia, en este caso se han agregado dos botones, MAS (+) y MENOS (-), se genera un contador en el servidor, el mismo que llega hasta 18, y no desciende menos de 0.

```

        boton2.setOnClickListener((view) -> {
            String user_id = mAuth.getCurrentUser().getEmail();
            user_id = user_id.replace( oldChar: '.', newChar: '_' );
            DatabaseReference current_id = mDataBase.child(user_id);
            if (contador<18){
                current_id.child("ESTADO PERSIANAS").setValue(contador);
                contador++;
            }
        });

        boton3.setOnClickListener((view) -> {
            String user_id = mAuth.getCurrentUser().getEmail();
            user_id = user_id.replace( oldChar: '.', newChar: '_' );
            DatabaseReference current_id = mDataBase.child(user_id);
            if (contador>0){
                current_id.child("ESTADO PERSIANAS").setValue(contador);
                contador--;
            }
        });
    }
}

```

9. En el caso de la sección de temperatura, después de realizar el mismo procedimiento de definir variables y botones, se procede a darle función a los mismos, en el botón de encendido y apagado de los ventiladores la programación ha sido la misma que en los botones anteriores, solo se adicionó la escritura de valores en los espacios donde se le permite al usuario ingresar la temperatura que desee.

```

btvent_salaonoff.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        boolean checked = btvent_salaonoff.isChecked();
        String user_id = mAuth.getCurrentUser().getEmail();
        user_id = user_id.replace( oldChar: '.', newChar: '_' );
        DatabaseReference current_id = mDataBase.child(user_id);
        String valor_sala = Deseado3.getText().toString().trim();
        current_id.child("TEMP DESEADA SALA").setValue(valor_sala);

        if (checked) {
            current_id.child("ESTADO VENTILACION SALA").setValue(1);
        } else {
            current_id.child("ESTADO VENTILACION SALA").setValue(0);
        }
    }
});

btvent_cuartosonoff.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        boolean checked = btvent_cuartosonoff.isChecked();
        String user_id = mAuth.getCurrentUser().getEmail();
        user_id = user_id.replace( oldChar: '.', newChar: '_' );
        DatabaseReference current_id = mDataBase.child(user_id);
        String valor_cuarto = Deseado1.getText().toString().trim();
        String valor_master = Deseado2.getText().toString().trim();
        current_id.child("TEMP DESEADA MASTER").setValue(valor_master);
        current_id.child("TEMP DESEADA DORMITORIO").setValue(valor_cuarto);

        if (checked) {
            current_id.child("ESTADO VENTILACION CUARTOS").setValue(1);
        } else {
            current_id.child("ESTADO VENTILACION CUARTOS").setValue(0);
        }
    }
});

```

10. En esta misma sección, se dejó un texto visible, donde se le permitirá al usuario observar la temperatura actual que el sensor está leyendo. Aquí sucede el efecto contrario, pues desde el nodeMcu ESP8266, se escribe en el servidor el valor sensado, y este lo muestra en la pantalla de la aplicación del teléfono.

```
mDataBase.child("nicolevivar_95@hotmail_com").addValueEventListener(new ValueEventListener() {  
    @Override  
    public void onDataChange(DataSnapshot dataSnapshot) {  
  
        Integer n = dataSnapshot.child("TEMP ACTUAL DORMITORIO").getValue(Integer.class);  
        Integer i = dataSnapshot.child("TEMP ACTUAL MASTER").getValue(Integer.class);  
        Integer c = dataSnapshot.child("TEMP ACTUAL SALA").getValue(Integer.class);  
  
        Actual1.setText(String.valueOf(n));  
        Actual2.setText(String.valueOf(i));  
        Actual3.setText(String.valueOf(c));  
  
    }  
  
    @Override  
    public void onCancelled(DatabaseError databaseError) {  
  
    }  
  
});
```

3.3.2 Diseño del software en Matlab.

Para el software de Matlab se diseñó una interfaz mediante el GUIDE, agregando una conexión de usuario, un bloque para el control de la iluminación, de las persianas y garaje, y en la parte inferior de la temperatura. Todos los botones que se pueden observar en la interfaz son de encendido y apagado, los cuales funcionan de forma similar a los de la aplicación en Android Studio.

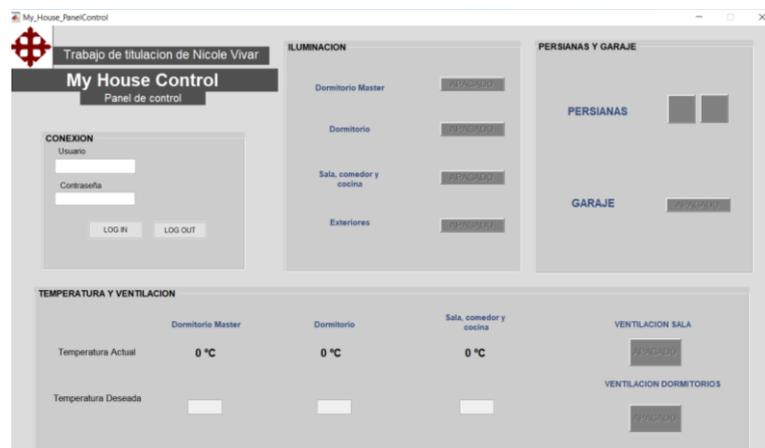


Figura 33. Interface de Matlab
Elaborado por: Autora

- Panel de conexión: Este panel permite habilitar el resto de paneles, para que los mismos puedan funcionar es necesario el inicio de sesión. En caso de que no se llenen los campos aparecerán mensajes de error, de igual forma si el usuario o contraseña son incorrectos o si no se cuenta con un usuario creado.

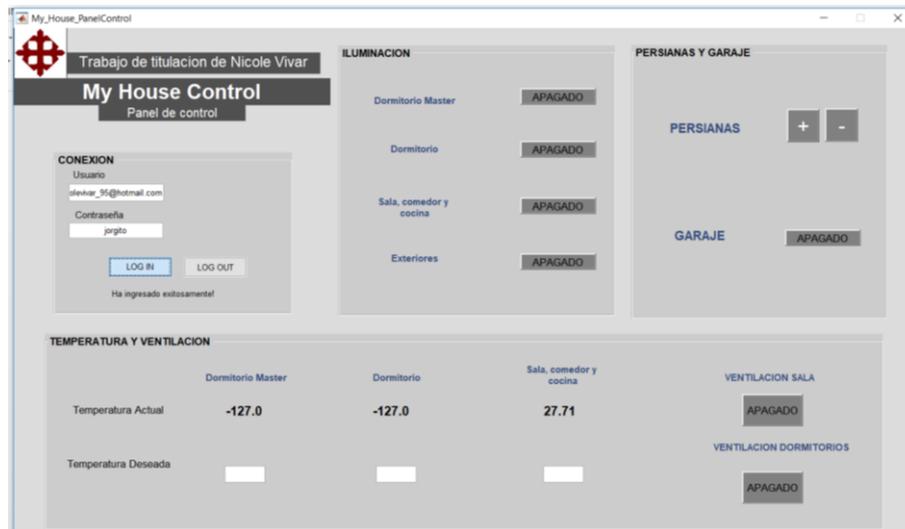


Figura 34. Interface de Matlab, después de iniciar sesión.

Elaborado por: Autora

- Panel de iluminación: Este panel se habilita una vez que se ha iniciado sesión, y permite la interacción entre el usuario, y el sistema. Permitiéndole encender o apagar luces de dormitorios, sala, comedor, cocina y exteriores.
- Panel persianas y garaje: Este panel se habilita una vez que se ha iniciado sesión, le permite al usuario abrir cerrar las persianas según se desee, y abrir cerrar la puerta del garaje.
- Panel de temperatura y ventilación: Este panel hace posible la interacción con la ventilación de la vivienda, en este caso nos muestra la temperatura actual que está leyendo el sensor, y permite el ingreso de la temperatura que se desee. Así mismo, encender y apagar los ventiladores, según se desee.

La programación para la interfaz de Matlab, como se conoce, una vez creada la interfaz en el GUIDE, el código se desarrolla por sí solo, lo que se procede a hacer es darle función a los botones.

1. Se creó la función `updateTemps`, la misma que permite leer y actualizar los valores del sensor.

```
function updateTemps(obj, evt)
    global ghandles db

    t_master = db.getCurTemp('MASTER');
    t_cuarto = db.getCurTemp('CUARTO');
    t_sala = db.getCurTemp('SALA');

    str = sprintf('%s °C', t_master);
    set(ghandles.actual1, 'String', str);
    str = sprintf('%s °C', t_cuarto);
    set(ghandles.actual2, 'String', str);
    str = sprintf('%s °C', t_sala);
    set(ghandles.actual3, 'String', str);
    fprintf('funciona\n');
```

2. En el botón login se ha programado el inicio de sesión, donde se escribe el usuario y la contraseña, en caso de encontrar algún error envía avisos, o mensajes. Se agregan las funciones respectivas, las cuales se han agregado a la carpeta en archivos comprimidos, estas funciones han sido realizadas para facilitar la tarea de la programación. Se designa las variables de usuario y contraseña como string, y se escribe la condición donde se indica que si los campos están vacíos, dé un aviso con un mensaje que indique el problema.

```
% --- Executes on button press in boton_login.
function boton_login_Callback(hObject, eventdata, handles)
% hObject    handle to boton_login (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of boton_login
global db;
global ttimer;
global ghandles

ghandles = handles;
raw_user = get(handles.usuario, 'String');
password = get(handles.contrasena, 'String');

if isempty(raw_user) || isempty(password)
    msgbox('Debe llenar los campos de usuario y contraseña.', 'Aviso');
    return
end
```

3. Se ingresa las variables para la conexión con Firebase, se agrega el host, y token. Además, se añade la condición que habilita todos los botones una vez que el inicio de sesión sea exitoso, de modo contrario, se mostraran distintos mensajes de error, según sea su condición.

```

host = 'my-house-nicole.firebaseio.com'; %host del proyecto en firebase
token = 'Z30I6kDYUQzcJsn2JiYr2ixYSOfRqC3PP5K4sLDJ'; %token del proyecto en firebase
user = strrep(raw_user, '.', '_');

db = db_nv(host, token, user);
s = db.verifyUserPassword(password);
if s == -1
    set(handles.status_login, 'String', 'El usuario no existe.');
```

```

elseif s == -2
    set(handles.status_login, 'String', 'La contraseña ingresada es incorrecta.');
```

```

elseif s == 1
    set(handles.status_login, 'String', 'Ha ingresado exitosamente!');
    set(handles.boton_master, 'Enable', 'on');
    set(handles.boton_dormitorio, 'Enable', 'on');
    set(handles.boton_sala, 'Enable', 'on');
    set(handles.boton_exterior, 'Enable', 'on');
    set(handles.abrir_persianas, 'Enable', 'on');
    set(handles.cerrar_persianas, 'Enable', 'on');
    set(handles.boton_garaje, 'Enable', 'on');
    set(handles.temp_master, 'Enable', 'on');
    set(handles.temp_dormitorio, 'Enable', 'on');
    set(handles.temp_sala, 'Enable', 'on');
    set(handles.vent_sala, 'Enable', 'on');
    set(handles.vent_cuartos, 'Enable', 'on');
```

```

    ttimer = timer('ExecutionMode', 'fixedRate', 'Period', 10, 'TimerFcn', @updateTemps);
    start(ttimer);
else
    set(handles.status_login, 'String', 'Ha ocurrido un error desconocido.');
```

```

end

```

4. Se programó el botón para la ventilación de la sala de modo que si no se agrega un texto este no se pueda encender y que si existe un valor se encienda y escriba en el servidor que se ha generado este cambio, del mismo modo pasa cuando se apaga.

```

% --- Executes on button press in vent_sala.
function vent_sala_Callback(hObject, eventdata, handles)
% hObject handle to vent_sala (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of vent_sala
global db

stemp_sala_deseada = get(handles.temp_sala, 'String');
if isempty(stemp_sala_deseada)
    msgbox('Debe llenar el campo de temperatura deseada de la sala.', 'Aviso');
    return
end
temp_sala_deseada = str2double(get(handles.temp_sala, 'String'));

if(get(hObject,'Value') == 1)
    set(hObject, 'String', 'ENCENDIDO');
    db.turnOn('VENTILACION SALA');
    db.changeDesiredTemp('SALA', temp_sala_deseada);
else
    set(hObject, 'String', 'APAGADO');
    db.turnOff('VENTILACION SALA');
```

```

end

```

5. Se ha elaborado un procedimiento similar para el botón de ventilación de los cuartos.

```

% --- Executes on button press in vent_cuartos.
function vent_cuartos_Callback(hObject, eventdata, handles)
% hObject      handle to vent_cuartos (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of vent_cuartos
global db

stemp_master_deseada = get(handles.temp_master, 'String');
stemp_dormitorio_deseada = get(handles.temp_dormitorio, 'String');

if isempty(stemp_master_deseada) || isempty(stemp_dormitorio_deseada)
    msgbox('Debe llenar el campo de temperatura deseada de los dormitorios.', 'Aviso');
    return
end
temp_master_deseada = str2double(get(handles.temp_master, 'String'));
temp_dormitorio_deseada = str2double(get(handles.temp_dormitorio, 'String'));

if(get(hObject,'Value') == 1)
    set(hObject, 'String', 'ENCENDIDO');
    db.turnOn('VENTILACION CUARTOS');
    db.changeDesiredTemp('MASTER', temp_master_deseada);
    db.changeDesiredTemp('DORMITORIO', temp_dormitorio_deseada);
else
    set(hObject, 'String', 'APAGADO');
    db.turnOff('VENTILACION CUARTOS');
end

```

6. De igual forma para el botón de luces del master, dormitorio, sala, comedor, cocina, garaje y exteriores.

```

% --- Executes on button press in boton_master.
function boton_master_Callback(hObject, eventdata, handles)
% hObject      handle to boton_master (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of boton_master
global db

if(get(hObject,'Value') == 1)
    set(hObject, 'String', 'ENCENDIDO');
    db.turnOn('MASTER');
else
    set(hObject, 'String', 'APAGADO');
    db.turnOff('MASTER');
end

% --- Executes on button press in boton_dormitorio.
function boton_dormitorio_Callback(hObject, eventdata, handles)
% hObject      handle to boton_dormitorio (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of boton_dormitorio
global db

if(get(hObject,'Value') == 1)
    set(hObject, 'String', 'ENCENDIDO');
    db.turnOn('DORMITORIO');
else
    set(hObject, 'String', 'APAGADO');
    db.turnOff('DORMITORIO');
end

```

```

% --- Executes on button press in boton_sala.
function boton_sala_Callback(hObject, eventdata, handles)
% hObject    handle to boton_sala (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of boton_sala
global db

if(get(hObject,'Value') == 1)
    set(hObject, 'String', 'ENCENDIDO');
    db.turnOn('SALA');
else
    set(hObject, 'String', 'APAGADO');
    db.turnOff('SALA');
end

% --- Executes on button press in boton_garaje.
function boton_garaje_Callback(hObject, eventdata, handles)
% hObject    handle to boton_garaje (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of boton_garaje
global db

if(get(hObject,'Value') == 1)
    set(hObject, 'String', 'ENCENDIDO');
    db.turnOn('GARAJE');
else
    set(hObject, 'String', 'APAGADO');
    db.turnOff('GARAJE');
end
% COMO FUNCIONA LO DE LA VENTILACION??? MIRA WA

% --- Executes on button press in boton_exterior.
function boton_exterior_Callback(hObject, eventdata, handles)
% hObject    handle to boton_exterior (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of boton_exterior
global db

if(get(hObject,'Value') == 1)
    set(hObject, 'String', 'ENCENDIDO');
    db.turnOn('EXTERIOR');
else
    set(hObject, 'String', 'APAGADO');
    db.turnOff('EXTERIOR');
end

```

7. Para las persianas, se programó dos botones, como se observa en la imagen, vemos un MÁS (+) y un MENOS (-), para el botón MÁS se le ha ordenado abrir las persianas y para el botón MENOS cerrarlas. Por medio de la función db, mencionada en los pasos anteriores.

```

% --- Executes on button press in abrir_persianas.
function abrir_persianas_Callback(hObject, eventdata, handles)
% hObject    handle to abrir_persianas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global db
db.openBlinds(1);

% --- Executes on button press in cerrar_persianas.
function cerrar_persianas_Callback(hObject, eventdata, handles)
% hObject    handle to cerrar_persianas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global db
db.closeBlinds(1);

```

8. Por el último tenemos el botón logout, que permite cerrar sesión. Este botón deshabilita todos los botones, y muestra un mensaje de cierre de sesión.

```

% --- Executes on button press in boton_logout.
function boton_logout_Callback(hObject, eventdata, handles)
% hObject    handle to boton_logout (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global ttimer
stop(ttimer);
set(handles.status_login, 'String', 'Ha cerrado sesion.');
```

```

set(handles.boton_master, 'Enable', 'off');
set(handles.boton_dormitorio, 'Enable', 'off');
set(handles.boton_sala, 'Enable', 'off');
set(handles.boton_exterior, 'Enable', 'off');
set(handles.abrir_persianas, 'Enable', 'off');
set(handles.cerrar_persianas, 'Enable', 'off');
set(handles.boton_garaje, 'Enable', 'off');
set(handles.temp_master, 'Enable', 'off');
set(handles.temp_dormitorio, 'Enable', 'off');
set(handles.temp_sala, 'Enable', 'off');
set(handles.vent_sala, 'Enable', 'off');
set(handles.vent_cuartos, 'Enable', 'off');

```

3.3.3 Diseño del software del NodeMcu esp8266 en IDE de Arduino

Para la programación del NodeMcu ESP8266, se trabajó en el IDE de Arduino, siguiendo la misma estructura que se utiliza para programar normalmente un Arduino.

1. En este caso se inició agregando las librerías que se utilizaron, en este caso, fueron las siguientes:

```

#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Servo.h>

```

2. Luego, se definieron las variables que se utilizaron en la programación para las luces leds.

```

int leds [4] = {D0,D1,D2,D3};
int estados [4] = {LOW,LOW,LOW,LOW};
int totalLeds = 4;

```

3. Así mismo, las variables del servo-motor, motor DC, ventiladores, y sensores.

```

Servo MiServo;
bool lastservo = true;

#define SENSOR_SALA A0

#define PUERTA_ABRE D4
#define PUERTA_CIERRA D5

bool lastmotor = true;

#define ONEWIRE_MASTER_DORMITORIO D6

#define FAN_SALA D7
#define FAN_ROOMS D8

OneWire oneWire (ONEWIRE_MASTER_DORMITORIO);
DallasTemperature sensores (&oneWire);

```

4. Se definieron los path con los que se trabaja de firebase, uno siendo el path de la base de datos, y el otro de la autenticación. Del mismo modo, se definió el usuario y el password de la red a la que se conecta el NodeMcu ESP8266.

```

#define FirebaseHost "my-house-nicole.firebaseio.com"
#define FirebaseAuth "Z30I6kDYUQzcJsn2JiYr2ixYSOfRqC3PP5K4sLDJ"
#define Wifi_User "Netlife-Chavez"
#define Wifi_Password "10101010"

```

5. Se agregó el siguiente string, para que la tarjeta pueda entrar a la base de datos llamada usuarios.

```
String chipId = "Usuarios";
```

6. Posterior a esto, se diseñó el código que permite la conexión del NodeMcu ESP8266 a Wi-Fi, además se observó la función que permite el inicio del resto de funciones de las luces, servomotor (persianas), motor Dc (puerta del garaje), sensores de temperatura, y ventiladores.

```
void setupFirebase () {  
  Firebase.begin (FirebaseHost, FirebaseAuth);  
}
```

```
void setupWiFi () {  
  //se conecta a wifi  
  WiFi.begin (Wifi_User, Wifi_Password);  
  Serial.print ("conectando...");  
  while (WiFi.status() !=WL_CONNECTED) {  
    Serial.print (".");  
    delay (500);  
  }  
  Serial.println();  
  Serial.print("conectado: ");  
  Serial.println (WiFi.localIP());  
}
```

7. Se crea la función setup, que reconoce las demás funciones como son: Conexión Wi-fi de la tarjeta, conexión con el servidor, función del servo, del motor, las luces, y temperatura.

```
void setup() {  
  Serial.begin (9600);  
  setupWiFi ();  
  setupFirebase ();  
  setupPinMode ();  
  ServoMotor ();  
  setupMotor ();  
  setupTemp ();  
  setupSensores ();  
}
```

8. Se realiza la interacción entre la tarjeta con el servidor Firebase, el string chipId, agregado en uno de los pasos anteriores se lo necesita para agregar este

nombre a la línea del path, para que la tarjeta, logre ingresar a la base de datos asignada. Además se le indica a los pines a que dirección deben obedecer, como se puede observar en el código. Este mismo paso se lo realiza al inicio de cada función desarrollada. Luego se inicia a la programación de los pines, donde se le indica que led encender o apagar. Además se añade los serial.println para poder visualizar en el monitor serial lo que ocurre en el interior de la tarjeta.

```
void getData() {

    String path = chipId + "/nicolevivar_95@hotmail_com";
    FirebaseObject object = Firebase.get(path);
    bool led1 = object.getInt ("/ESTADO MASTER");
    bool led2 = object.getInt ("/ESTADO DORMITORIO");
    bool led3 = object.getInt ("/ESTADO SALA");
    bool led4 = object.getInt ("/ESTADO EXTERIOR");

    Serial.println ("MASTER");
    Serial.println (led1);

    Serial.println ();
    Serial.println ("DORMITORIO");
    Serial.println (led2);

    Serial.println ();
    Serial.println ("SALA");
    Serial.println (led3);

    Serial.println ();
    Serial.println ("EXTERIOR");
    Serial.println (led4);

    if (led1 == 1) {digitalWrite (D0, HIGH);} else {digitalWrite (D0, LOW);}
    if (led2 == 1) {digitalWrite (D1, HIGH);} else {digitalWrite (D1, LOW);}
    if (led3 == 1) {digitalWrite (D2, HIGH);} else {digitalWrite (D2, LOW);}
    if (led4 == 1) {digitalWrite (D3, HIGH);} else {digitalWrite (D3, LOW);}
}
```

9. Para el servo se le ha indicado que según lo que lea en el servidor, se agreguen o resten 10 grados, debido a que este pin está programado en D10, se debieron borrar todas las funciones de Serial.

```

void getServo(){
  String path = chipId + "/nicolevivar_95@hotmail_com";
  FirebaseObject object = Firebase.get(path);
  int i = object.getInt ("/ESTADO PERSIANAS");

  if (i != lastservo){
    MiServo.write(i*10);
    delay(25);
    lastservo = i;
  }

  Serial.println ();
  Serial.println ("PERSIANAS");
  Serial.println (i);
}

```

10. Para el motor se ha creado una variable, llamada 'lastmotor', la misma le indica al NodeMcu Esp8266 que si 'x' no es igual a 'lastmotor', realice lo que se encuentra entre las llaves. Si no se utilizara esta variable se generaría un bucle infinito mientras este en uno y repetirá la acción, haciendo imposible la simulación de la puerta de garaje. Se ha utilizado la función analogWrite para que trabaje enviando el PWM establecido, que en este caso fue de 250, y un tiempo de 750ms.

```

void getMotor(){
  String path = chipId + "/nicolevivar_95@hotmail_com";
  FirebaseObject object = Firebase.get(path);
  bool x = object.getInt ("/ESTADO GARAJE");

  if (x!=lastmotor){
    if(x==1){
      analogWrite(PUERTA_ABRE, 250);
      analogWrite(PUERTA_CIERRA, 0);
      delay(750);
      analogWrite(PUERTA_ABRE, 0);
      delay(750);}

    if(x==0){
      analogWrite(PUERTA_CIERRA, 250);
      analogWrite (PUERTA_ABRE, 0);
      delay (550);
      analogWrite(PUERTA_CIERRA, 0);
      delay(550);}

    lastmotor = x;
  }

  Serial.println ();
  Serial.println ("GARAJE");
  Serial.println (x);
}

```

11. Para la temperatura la estructura ha sido la misma, pero se han agregado 6 variables, una que es la que lee el sensor y envía al servidor, y la otra es la que genera el cambio, agregando la condición que si ha leído el 1 el ventilador se encienda, y si es el 0 el ventilador se apague.

```
void getTemp(){
  String path = chipId + "/nicolevivar_95@hotmail_com";
  String path1 = path + "/TEMP ACTUAL SALA";
  String path2 = path + "/TEMP ACTUAL MASTER";
  String path3 = path + "/TEMP ACTUAL DORMITORIO";
  FirebaseObject object = Firebase.get(path);
  bool A = object.getInt ("/ESTADO VENTILACION SALA");
  bool B = object.getInt ("/ESTADO VENTILACION CUARTOS");

  int t1 = 0;
  int t2 = 0;
  int t3 = 0;
  float temp1 = 0;
  float temp2 = 0;
  float temp3 = 0;

  t1= analogRead(A0);          //Lee la temperatura que ingresa en el pin A0

  float milivoltios1 = (t1/1024.0)*3300;
  temp1= milivoltios1/10;

  Firebase.setFloat (path1, temp1);

  //Serial.println ();
  //Serial.println ("TEMP SALA");
  //Serial.println (temp1);

  //Serial.println ();
  //Serial.println ("TEMP MASTER");
  sensores.requestTemperatures();
  temp2 = sensores.getTempCByIndex(0);
  //Serial.println (temp2);

  //Serial.println ();
  //Serial.println ("TEMP DORMITORIO");
  temp3 = sensores.getTempCByIndex(1);
  //Serial.println (temp3);

  Firebase.setFloat (path2, temp2);
  Firebase.setFloat (path3, temp3);

  if(A==0){
    digitalWrite (FAN_SALA,LOW);
    else if (temp1 == temp_deseada_master)
  {digitalWrite (FAN_SALA,LOW);}
    else {digitalWrite (FAN_SALA,HIGH);}

  if(B==0){
    digitalWrite (FAN_ROOMS,LOW);
    else {digitalWrite (FAN_ROOMS,HIGH);}
  }
}
```

12. Se crea la función loop, que es la que permite que se ejecuten las funciones que hemos desarrollado, en cuanto a las luces, servo, motor y temperatura.

```

void loop() {
  getData();
  getServo();
  getMotor();
  getTemp();
}

```

13. Las siguientes funciones, son donde se describen y se define las variables, como entradas o salidas, utilizadas en el programa.

```

void setupPinMode() {

  for (int i=0; i<totalLeds; i++){
    //Serial.printf("Setup Output for pin %d",leds[i]);
    pinMode (leds[i],OUTPUT);}
}

void ServoMotor() {
  MiServo.attach (3);
  MiServo.write (0);
}

void setupMotor () {
  pinMode (PUERTA_ABRE, OUTPUT);
  pinMode (PUERTA_CIERRA, OUTPUT);
}

void setupTemp () {
  pinMode (FAN_ROOMS, OUTPUT);
  pinMode (FAN_SALA, OUTPUT);
  pinMode (SENSOR_SALA, INPUT);
}

void setupSensores () {
  sensores.begin ();
  pinMode (ONEWIRE_MASTER_DORMITORIO, INPUT);
}

```

3.4 Implementación

La implementación se la realizo en una maqueta a escala 1:20, simulando una vivienda. Para la conexión de los leds, se pasaron cables por el techo de la casa (de forma oculta) y se dejó pequeños orificios para la salida de los leds. Los ventiladores se han colocado en la parte superior (techo), de forma directa a los sensores para que se pueda verificar el cambio de temperatura, se ha dejado una abertura para poder observar con facilidad el movimiento de los ventiladores cuando se encuentran encendidos. Para las persianas se diseñó un modelo mecánico donde el servo motor ha sido conectado a un rodillo de madera a escala, y las persianas se las ha simulado con tela, entonces a medida que se va subiendo o bajando las persianas, estas se abren o cierran. Para la parte de la puerta se diseñó otro tipo de mecanismo, parecido a este

pero en lugar de tela, se ha utilizado palillos de helado, dando el efecto de puertas enrollables.

3.5 Pruebas y resultados

Para las pruebas se dividieron los circuitos en dos secciones, y se realizaron pruebas de las interfaces, pruebas de iluminación, pruebas de temperatura y ventilación y pruebas de servo motor y motor DC. Estos mismos se describen a continuación indicando cuales han sido las pruebas realizadas con sus respectivos resultados. La prueba principal que se realizó entre los dos fue la del tiempo de respuesta, así como el funcionamiento, la sintonía, y la agilidad.

3.5.1 Pruebas y resultados de software

Las pruebas realizadas en la interfaz de Android, se hicieron con el fin de probar cada una de las señales, y ver su tiempo de respuesta. Sin embargo, sabemos que tanto para Android y Matlab si no se ha establecido una red estable, o lo suficientemente buena, el sistema tendrá una respuesta bastante deficiente, además carecería de su función primordial que es la de trabajar en tiempo real.

En la interfaz desarrollada en Android Studio, se obtuvo como resultado una app de fácil acceso, semi – dinámica, y que funciona casi de forma instantánea. Se logra visualizar una respuesta inmediata en el servidor, y efectivamente la respuesta al leer las variables de temperatura, también son instantáneas. Si no existe una buena señal, como ya se lo ha mencionado, la velocidad de la aplicación es bajísima, y resulta deficiente.

En la interfaz desarrollada en Matlab ocurre algo muy similar, el panel de control funciona de forma instantánea, debido a la programación es bastante explicativo al momento de encontrar un error al iniciar sesión, y existe un impedimento, pues no te permite registrarte, el único modo de registro de usuario es en Android, siendo una desventaja frente a la misma. Sin embargo, lo bueno del panel de control es que se puede visualizar todas las variables controladas y sus estados al mismo tiempo.

3.5.2 Pruebas y resultados de hardware

Las pruebas que se realizaron para el hardware fueron de sintonización con el software, y el tiempo de respuesta del sistema. Aproximadamente, las luces, ventiladores, y motores responden en un tiempo de 1 segundo, máximo 2 segundos, y en otros momentos la respuesta es instantánea, esto ocurre debido a que el nodemcu Esp8266 lee datos en determinado tiempo, y si ha terminado de hacer la lectura y en ese momento se ha enviado algo, tiene que esperar a volver a leer ese dato, pero esto sucede tan rápido que el tiempo máximo que se demoraría son 2 segundos, si excede este valor significa que la conexión a internet puede estar fallando.

En el caso de los ventiladores, se encienden de inmediato o demora de 1 – 2 segundos sin embargo al momento de apagarlo, debido a que la desconexión del mismo se genera a partir de una disminución de la velocidad, demora 5 segundos en detenerse por completo. Los sensores funcionan en buen estado, tanto los sensores DS18B20 y el LM35, muestran una respuesta inmediata.

Capítulo 4: Conclusiones y Recomendaciones

4.1 CONCLUSIONES

- Se logró un correcto funcionamiento de las dos interfaces desarrolladas para el sistema de domótica. Ambas trabajan en tiempo real, y logran controlar las variables establecidas desde cualquier lugar del mundo.
- Se concluye que el Wi-Fi tiene las mayores ventajas en cuanto a sistemas de domótica se trata, gracias a que se puede trabajar por medio de un servidor, y controlar las variables desde cualquier parte del mundo. Una de las desventajas, es que si el mismo no posee una buena señal, el sistema de domótica se entorpece y trabaja lento, o simplemente no responde.
- La tarjeta con la que se ha trabajado en este caso el nodeMcu ESP8266 es de uso educativo, de forma que para un sistema real no abastecería y se generaría un sobre procesamiento.
- Este tema también se puede aplicar para plantas de procesos industriales que tienen sistema de control distribuido, es decir que se operan a través de pantallas locales, las cuales usando este mismo método pueden hacerse remotas, con una estructura similar haciendo posible manejarlas desde cualquier parte del mundo.

4.2 RECOMENDACIONES

- Se recomienda verificar bien las conexiones antes de iniciar el programa y así evitar posibles daños al circuito, y al NodeMcu Esp8266. Además es importante verificar las hojas técnicas antes de cada conexión de los dispositivos del sistema.
- Es recomendable también verificar la conexión a internet, y tratar de mantener una buena señal, para que el sistema de domótica pueda funcionar adecuadamente.
- Es necesario no inicializar el NodeMcu ESP8266, o subir el programa, mientras está conectado al circuito energizado, pues alguno de los pines permiten la comunicación, y si están conectados pueden generar un cortocircuito y dañar el pin, y si esto sucede, por ejemplo, con el pin D3, que es el que permite la comunicación serial, le impedirá cargar programas a la tarjeta, ocasionándole un daño irreversible.
- Para trabajos futuros es recomendable mantener un buen QoS o calidad de servicio, configurando el router para que priorice las señales enviadas del sistema, en lugar de otras como Netflix, Youtube, o cualquier otro programa que requiera acceso a internet.

BIBLIOGRAFIA:

Android Studio. (2017). Android Studio Release Notes | Android Studio. Recuperado 7 de diciembre de 2017, a partir de <https://developer.android.com/studio/releases/index.html>

Arkiplus. (2013). Historia de la Domótica. Recuperado 30 de octubre de 2017, a partir de <http://www.arkiplus.com/historia-de-la-domotica>

Brodsky, I. (2016). The race to create smart homes is on | Computerworld. Recuperado 5 de enero de 2018, a partir de <https://www.computerworld.com/article/3062002/home-tech/the-race-to-create-smart-homes-is-on.html>

Cevallos, C. (2016). *Aplicación de la plataforma hardware y software raspberry pi ii y el módulo de conectividad de red inalámbrica photon wi-fly, para el diseño de aplicaciones domóticas basadas en tecnología wi-fi.* (Trabajo de Titulación). Universidad Católica de Santiago de Guayaquil, Guayaquil - Ecuador. Recuperado a partir de <http://repositorio.ucsg.edu.ec/bitstream/3317/5339/1/T-UCSG-PRE-TEC-IECA-40.pdf>

Cincodays. (2015). Conociendo la Historia – Domotica: Orígenes. Recuperado 31 de octubre de 2017, a partir de <https://cincodays.com/2015/08/12/conociendo-la-historia-domotica-origenes/>

Cisco. (2013). Connections Counter: The Internet of Everything in Motion. Recuperado 7 de noviembre de 2017, a partir de

<https://newsroom.cisco.com/feature-content?type=webcontent&articleId=1208342>

Craven, J. (2017). What is a Smart House? Ask Your Computer. Recuperado 5 de enero de 2018, a partir de <https://www.thoughtco.com/what-is-a-smart-house-domotics-177572>

Edwards, R. (2017). Home Automation Protocols Explained. Recuperado 15 de noviembre de 2017, a partir de <https://www.safewise.com/blog/makes-house-smart-home-home-automation-protocols/>

EH Contributor. (2016). Home Automation Protocols: A Round-Up. Recuperado 15 de noviembre de 2017, a partir de <https://www.electronichouse.com/smart-home/home-automation-protocols-what-technology-is-right-for-you/>

EKOS. (2017). Computerworld - La domótica e inmótica en Ecuador. Recuperado 13 de noviembre de 2017, a partir de <http://computerworld.com.ec/actualidad/tendencias/128-domoticaeinmoticaecuador.html>

Esplin, C. (2016). What is Firebase? Recuperado 6 de diciembre de 2017, a partir de <https://howtofirebase.com/what-is-firebase-fcb8614ba442>

Espressif Systems. (2015). Esp8266 Datasheet. Recuperado a partir de https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf

Flores, M. (2016). La domótica cambia el estilo de vida y va más allá del confort. Recuperado 5 de enero de 2018, a partir de

<http://laconversacion.net/2016/08/la-domotica-cambia-el-estilo-de-vida-y-va-mas-alla-del-confort/>

Fritz, R. (2017). Top Reasons To Automate Your Home. Recuperado 7 de noviembre de 2017, a partir de <https://www.lifewire.com/reasons-to-automate-your-home-817695>

Garcia Pumarino. (2015). La Estandarización de la Domótica. Recuperado 6 de enero de 2018, a partir de <https://cursodidacticoknx.wordpress.com/2015/02/20/la-estandarizacion-de-la-domotica/>

Haz, M. (2016). *Estudio y diseño de un ambiente inteligente aplicando inmótica en el auditorio Dr. Leónidas Ortega Moreira de la Universidad Católica de Santiago de Guayaquil*. Universidad Católica de Santiago de Guayaquil, Guayaquil - Ecuador. Recuperado a partir de <http://repositorio.ucsg.edu.ec/bitstream/3317/6648/1/T-UCSG-PRE-TEC-IECA-49.pdf>

Heraldo. (2017). Las casas inteligentes, ¿el futuro del hogar? Recuperado 30 de octubre de 2017, a partir de <http://www.heraldo.es/noticias/economia/inmobiliaria/2017/08/02/las-casas-inteligentes-futuro-del-hogar-1189739-1601025.html>

Hierro Duarte, & Mills, C. (2017). What is a web server? Recuperado 6 de diciembre de 2017, a partir de https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server

- IBM Knowledge Center. (2018). IBM Knowledge Center - TCP/IP protocols. Recuperado 6 de marzo de 2018, a partir de https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.1.0/com.ibm.cics.ts.internet.doc/topics/dfhtl23.html
- IECOR. (2017). Estándares Internacionales de Domótica. Recuperado 6 de enero de 2018, a partir de <https://www.iecor.com/estandares-internacionales-de-domotica/>
- MathWorks. (2017). MathWorks - Makers of MATLAB and Simulink. Recuperado 7 de diciembre de 2017, a partir de <https://www.mathworks.com/>
- Morgan, J. (2014). A Simple Explanation Of «The Internet Of Things». Recuperado 13 de noviembre de 2017, a partir de <https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/>
- Netlife. (2015). La domótica en Ecuador y cómo puedes tener una casa inteligente | Netlife. Recuperado 30 de octubre de 2017, a partir de <http://www.netlife.ec/2015/07/la-domotica-en-ecuador-y-como-puedes-tener-una-casa-inteligente/>
- Peñalver, M. Á. (2012). Lonworks ¿Qué es un Neuron Chip y para qué se utiliza? Recuperado 8 de enero de 2018, a partir de <https://domotiva.wordpress.com/2012/03/04/lonworks-que-es-un-neuron-chip-y-para-que-se-utiliza/>

Ray, B. (2018). What is M2M? Recuperado 10 de marzo de 2018, a partir de <https://www.link-labs.com/blog/what-is-m2m>

Reimón, J. (2013). De los primeros sistemas domóticos al hogar digital. Recuperado a partir de https://www.academia.edu/20955653/De_los_primeros_sistemas_dom%C3%B3ticos_al_hogar_digital

Rodrigo, J. (2016). ESP8266 y NodeMCU: la nueva generación de sistemas embebidos. Recuperado 16 de noviembre de 2017, a partir de <http://panamahitek.com/esp8266-y-nodemcu-la-nueva-generacion/>

Salazar, J. (2017). Wireless Network. Recuperado a partir de https://upcommons.upc.edu/bitstream/handle/2117/110811/LM01_F_EN.pdf

Telégrafo, E. (2011). Resultados del Censo 2010 revelan una nueva cara de Ecuador. Recuperado 7 de noviembre de 2017, a partir de <http://www.eltelegrafo.com.ec/noticias/informacion-general/1/ecuador-tiene>

Telégrafo, E. (2016). Cocinas de inducción se vendieron desde que inició el programa en 2014. Recuperado 7 de noviembre de 2017, a partir de <http://www.eltelegrafo.com.ec/noticias/economia/8/403-197-cocinas-de-induccion-se-vendieron-desde-que-inicio-el-programa-en-2014>

Torres, J. (2014). ¿Qué es y cómo funciona el Internet de las cosas? Recuperado 6 de enero de 2018, a partir de <https://hipertextual.com/archivo/2014/10/internet-cosas/>

ANEXOS

Anexo A. Datasheet de los parámetros del ESP8266.



Espressif Systems

ESP8266 Datasheet

Categories	Items	Values
WiFi Parameters	Certificates	FCC/CE/TELEC/SRRC
	WiFi Protocols	802.11 b/g/n
	Frequency Range	2.4G-2.5G (2400M-2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
802.11 g: -75 dbm (54 Mbps)		
802.11 n: -72 dbm (MCS7)		
Types of Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip	
Hardware Parameters	Peripheral Bus	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	5x5mm
	External Interface	N/A
Software Parameters	WiFi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Ssoftware Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP

3. Pins and Definitions

The chipset encapsulates variable analog and data transmission I/Os, descriptions and definitions of which are explained below in detail.

3.1. GPIO

3.1.1. General Purpose Input/Output Interface (GPIO)

There are up to 17 GPIO pins. They can be assigned to various functions by the firmware. Each GPIO can be configured with internal pull-up (except XPD_DCDC, which is configured with internal pull-down), input available for sampling by a software register, input triggering an edge or level CPU interrupt, input triggering a level wakeup interrupt, open-drain or push-pull output driver, or output source from a software register, or a sigma-delta PWM DAC.

These pins are multiplexed with other functions such as I2C, I2S, UART, PWM, IR Remote Control, etc. Data I/O soldering pad is bidirectional and tri-state that include data input and output controlling buffer. Besides, I/O can be set as a specific state and remains like this. For example, if you intend to lower the power consumption of the chip, all data input and output enable signals can be set as remaining low power state. You can transport some specific state into the I/O. When the I/O is not powered by external circuits, the I/O will remain to be the state that it was used the last time. Some positive feedback is generated by the state-remaining function of the pins, therefore, if the external driving power must be stronger than the positive feedback. Even so, the driving power that is needed is within 5uA.

Table 6 Pin Definitions of GPIOs

Variables	Symbol	Min	Max	Unit
Input Low Voltage	V_{IL}	-0.3	$0.25 \times V_{IO}$	V
Input High Voltage	V_{IH}	$0.75 \times V_{IO}$	3.3	V
Input Leakage Current	I_{IL}		50	nA
Output Low Voltage	V_{OL}		$0.1 \times V_{IO}$	V
Output High Voltage	V_{OH}	$0.8 \times V_{IO}$		V
Input Pin Resistance Value	C_{pad}		2	pF
VDDIO	V_{IO}	1.8	3.3	V
Maximum Driving Power	I_{MAX}		12	mA
Temperature	T_{amb}	-40	125	°C

All digital IO pins are protected from over-voltage with a snap-back circuit connected between the pad and ground. The snap back voltage is typically about 6V, and the holding voltage is 5.8V. This

	User Configuration	AT Instruction Set, Cloud Server, Android/ iOS App
--	--------------------	---

1.4. Ultra Low Power Technology

ESP8266EX has been designed for mobile, wearable electronics and Internet of Things applications with the aim of achieving the lowest power consumption with a combination of several proprietary techniques. The power saving architecture operates mainly in 3 modes: active mode, sleep mode and deep sleep mode.

By using advance power management techniques and logic to power-down functions not required and to control switching between sleep and active modes, ESP8266EX consumes about than 60uA in deep sleep mode (with RTC clock still running) and less than 1.0mA (DTIM=3) or less than 0.5mA (DTIM=10) to stay connected to the access point.

When in sleep mode, only the calibrated real-time clock and watchdog remains active. The real-time clock can be programmed to wake up the ESP8266EX at any required interval.

The ESP8266EX can be programmed to wake up when a specified condition is detected. This minimal wake-up time feature of the ESP8266EX can be utilized by mobile device SOCs, allowing them to remain in the low-power standby mode until WiFi is needed.

In order to satisfy the power demand of mobile and wearable electronics, ESP8266EX can be programmed to reduce the output power of the PA to fit various application profiles, by trading off range for power consumption.

1.5. Major Applications

Major fields of ESP8266EX applications to Internet-of-Things include:

- Home Appliances
- Home Automation
- Smart Plug and lights
- Mesh Network
- Industrial Wireless Control
- Baby Monitors
- IP Cameras
- Sensor Networks
- Wearable Electronics

Anexo B. Datasheet del transistor 2N2222.

2N2222A

Small Signal Switching Transistor

NPN Silicon

Features

- MIL-PRF-19500/255 Qualified
- Available as JAN, JANTX, and JANTXV

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Characteristic	Symbol	Value	Unit
Collector - Emitter Voltage	V _{CE0}	50	Vdc
Collector - Base Voltage	V _{CB0}	75	Vdc
Emitter - Base Voltage	V _{EB0}	6.0	Vdc
Collector Current - Continuous	I _C	800	mAdc
Total Device Dissipation @ T _A = 25°C	P _T	500	mW
Total Device Dissipation @ T _C = 25°C	P _T	1.0	W
Operating and Storage Junction Temperature Range	T _J , T _{stg}	-65 to +200	°C

THERMAL CHARACTERISTICS

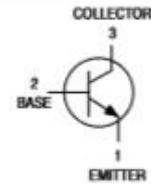
Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Ambient	R _{θJA}	325	°C/W
Thermal Resistance, Junction to Case	R _{θJC}	150	°C/W

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.



ON Semiconductor®

<http://onsemi.com>



TO-18
CASE 206AA
STYLE 1

ORDERING INFORMATION

Device	Package	Shipping
JAN2N2222A	TO-18	Bulk
JANTX2N2222A		
JANTXV2N2222A		

2N2222A

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
OFF CHARACTERISTICS				
Collector-Emitter Breakdown Voltage (I _C = 10 mAdc)	V _{(BR)CEO}	50	-	Vdc
Collector-Base Cutoff Current (V _{CB} = 75 Vdc) (V _{CB} = 60 Vdc)	I _{CBO}	-	10 10	μA nA
Emitter-Base Cutoff Current (V _{EB} = 6.0 Vdc) (V _{EB} = 4.0 Vdc)	I _{EBO}	-	10 10	μA nA
Collector-Emitter Cutoff Current (V _{CE} = 50 Vdc)	I _{CES}	-	50	nA
ON CHARACTERISTICS (Note 1)				
DC Current Gain (I _C = 0.1 mAdc, V _{CE} = 10 Vdc) (I _C = 1.0 mAdc, V _{CE} = 10 Vdc) (I _C = 10 mAdc, V _{CE} = 10 Vdc) (I _C = 150 mAdc, V _{CE} = 10 Vdc) (I _C = 500 mAdc, V _{CE} = 10 Vdc)	h _{FE}	50 75 100 100 30	- 325 - 300 -	-
Collector-Emitter Saturation Voltage (I _C = 150 mAdc, I _B = 15 mAdc) (I _C = 500 mAdc, I _B = 50 mAdc)	V _{CE(sat)}	-	0.3 1.0	Vdc
Base-Emitter Saturation Voltage (I _C = 150 mAdc, I _B = 15 mAdc) (I _C = 500 mAdc, I _B = 50 mAdc)	V _{BE(sat)}	0.6	1.2 2.0	Vdc
SMALL-SIGNAL CHARACTERISTICS				
Magnitude of Small-Signal Current Gain (I _C = 20 mAdc, V _{CE} = 20 Vdc, f = 100 MHz)	h _{fe}	2.5	-	-
Small-Signal Current Gain (I _C = 1.0 mAdc, V _{CE} = 10 Vdc, f = 1 kHz)	h _{fe}	50	-	-
Input Capacitance (V _{EB} = 0.5 Vdc, I _C = 0, 100 kHz ≤ f ≤ 1.0 MHz)	C _{ibo}	-	25	pF
Output Capacitance (V _{CB} = 10 Vdc, I _E = 0, 100 kHz ≤ f ≤ 1.0 MHz)	C _{obo}	-	8.0	pF
SWITCHING (SATURATED) CHARACTERISTICS				
Turn-On Time (Reference Figure in MIL-PRF-19500/255)	t _{on}	-	35	ns
Turn-Off Time (Reference Figure in MIL-PRF-19500/255)	t _{off}	-	300	ns

Product parametric performance is indicated in the Electrical Characteristics for the listed test conditions, unless otherwise noted. Product performance may not be indicated by the Electrical Characteristics if operated under different conditions.

1. Pulse Test: Pulse Width = 300 μs, Duty Cycle ≤ 2.0%.

2N2222A

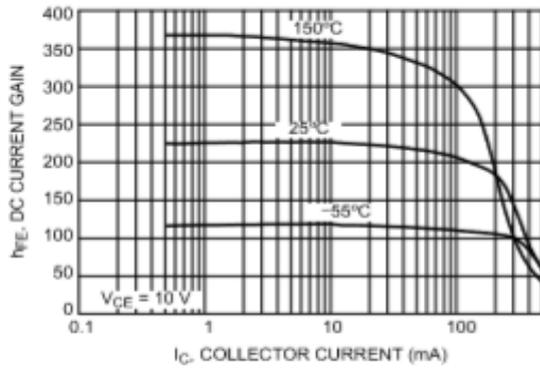


Figure 1. DC Current Gain

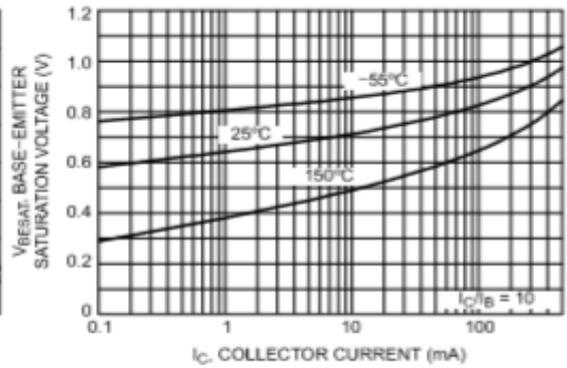


Figure 2. Base-Emitter Saturation Voltage

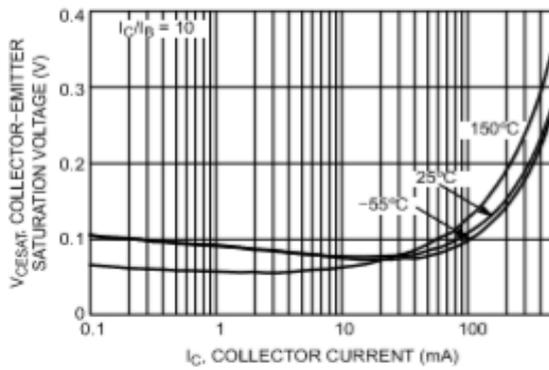


Figure 3. Collector-Emitter Saturation Voltage

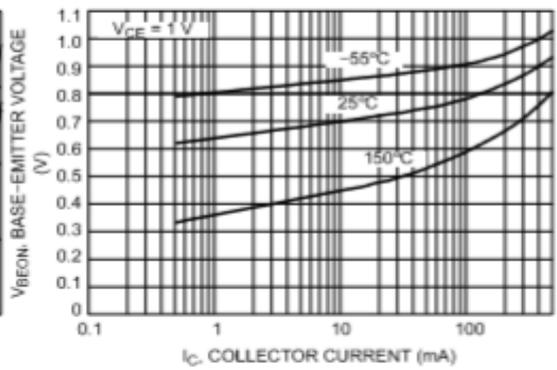


Figure 4. Base-Emitter Voltage

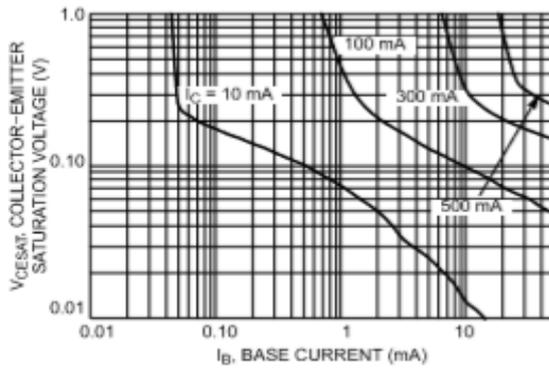


Figure 5. Collector Saturation Region

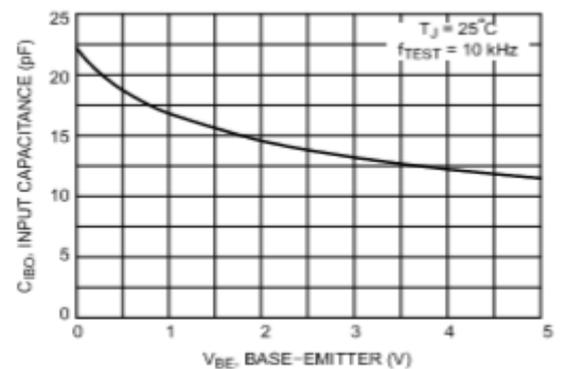


Figure 6. Input Capacitance

2N2222A

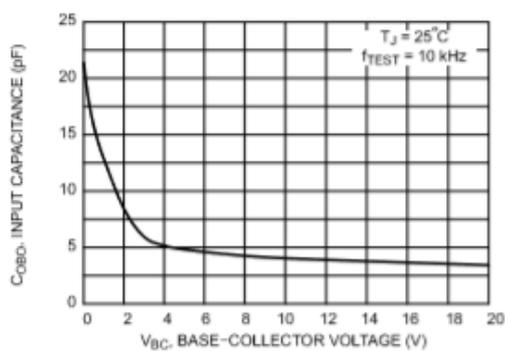


Figure 7. Output Capacitance

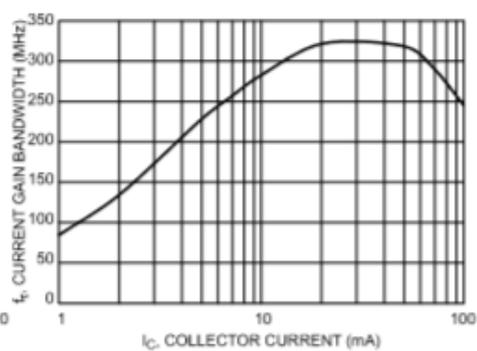
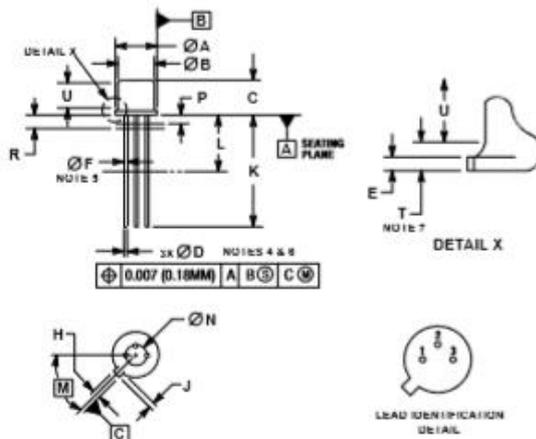


Figure 8. Current Gain Bandwidth Product

2N2222A

PACKAGE DIMENSIONS

TO-18 3
CASE 206AA
ISSUE A



- NOTES:
1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
 2. CONTROLLING DIMENSION: INCHES.
 3. DIMENSION J MEASURED FROM DIAMETER A TO EDGE.
 4. LEAD TRUE POSITION TO BE DETERMINED AT THE GAUGE PLANE DEFINED BY DIMENSION H.
 5. DIMENSION F APPLIES BETWEEN DIMENSION P AND L.
 6. DIMENSION D APPLIES BETWEEN DIMENSION L AND K.
 7. BODY CONTOUR OPTIONAL WITHIN CORE DEFINED BY DIMENSIONS A, H, AND T.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	4.31	4.96	0.170	0.195
B	4.57	4.96	0.180	0.196
C	4.32	5.30	0.170	0.210
D	0.41	0.50	0.016	0.020
E	—	0.76	—	0.030
F	0.41	0.48	0.016	0.019
H	0.91	1.17	0.036	0.046
J	0.71	1.25	0.028	0.049
K	12.70	19.05	0.500	0.750
L	6.35	—	0.250	—
M	45° BSC	—	45° BSC	—
N	7.64 BSC	—	0.300 BSC	—
P	—	1.27	—	0.050
R	1.37 BSC	—	0.054 BSC	—
T	—	0.76	—	0.030
U	2.54	—	0.100	—

- STYLE 1:
PIN 1: EMITTER
2: BASE
3: COLLECTOR

ON Semiconductor and are registered trademarks of Semiconductor Components Industries, LLC (SCILLC). SCILLC owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of SCILLC's product/patent coverage may be accessed at www.onsemi.com/pdf/Patent-Marking.pdf. SCILLC reserves the right to make changes without further notice to any products herein. SCILLC makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does SCILLC assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. "Typical" parameters which may be provided in SCILLC data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. SCILLC does not convey any license under its patent rights nor the rights of others. SCILLC products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SCILLC product could create a situation where personal injury or death may occur. Should Buyer purchase or use SCILLC products for any such unintended or unauthorized application, Buyer shall indemnify and hold SCILLC and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that SCILLC was negligent regarding the design or manufacture of the part. SCILLC is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
P.O. Box 5163, Denver, Colorado 80217 USA
Phone: 303-675-2175 or 800-344-3860 Toll Free USA/Canada
Fax: 303-675-2176 or 800-344-3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800-282-9855 Toll Free
USA/Canada
Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910
Japan Customer Focus Center
Phone: 81-3-5817-1050

ON Semiconductor Website: www.onsemi.com
Order Literature: <http://www.onsemi.com/orderlit>
For additional information, please contact your local
Sales Representative

2N2222A/D

Anexo C. Datasheet de puente L293D.

L293, L293D QUADRUPLE HALF-H DRIVERS

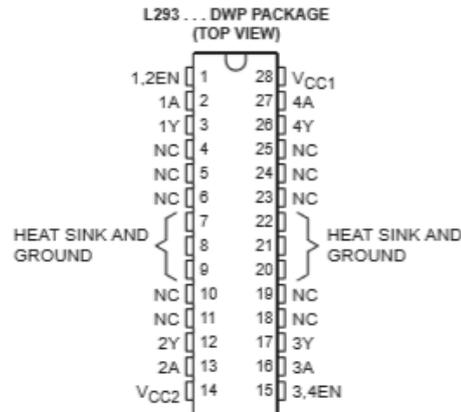
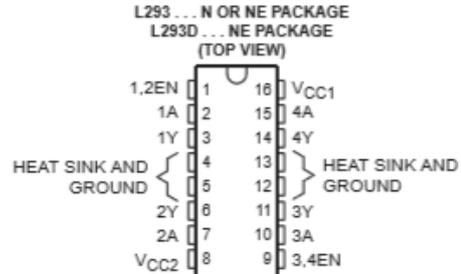
SLRS008C – SEPTEMBER 1986 – REVISED NOVEMBER 2004

- Featuring Unitrode L293 and L293D Products Now From Texas Instruments
- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Functionally Similar to SGS L293 and SGS L293D
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

description/ordering information

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.



ORDERING INFORMATION

T _A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	HSOP (DWP)	Tube of 20	L293DWP	L293DWP
	PDIP (N)	Tube of 25	L293N	L293N
	PDIP (NE)	Tube of 25	L293NE	L293NE
		Tube of 25	L293DNE	L293DNE

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA Information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS
INSTRUMENTS

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2004, Texas Instruments Incorporated

1

L293, L293D QUADRUPLE HALF-H DRIVERS

SLRS008C - SEPTEMBER 1986 - REVISED NOVEMBER 2004

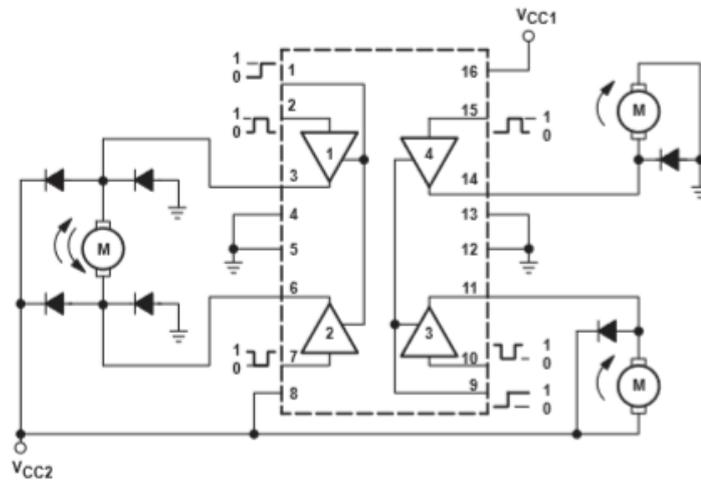
description/ordering information (continued)

On the L293, external high-speed output clamp diodes should be used for inductive transient suppression.

A V_{CC1} terminal, separate from V_{CC2} , is provided for the logic inputs to minimize device power dissipation.

The L293 and L293D are characterized for operation from 0°C to 70°C.

block diagram



NOTE: Output diodes are internal in L293D.

FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high level, L = low level, X = irrelevant, Z = high impedance (off)

† In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

 **TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

L293, L293D QUADRUPLE HALF-H DRIVERS

SLRS008C – SEPTEMBER 1986 – REVISED NOVEMBER 2004

recommended operating conditions

		MIN	MAX	UNIT
Supply voltage	V _{CC1}	4.5	7	V
	V _{CC2}	V _{CC1}	36	
V _{IH}	High-level input voltage	V _{CC1} ≤ 7 V	2.3 V _{CC1}	V
V _{IL}	Low-level output voltage	V _{CC1} ≥ 7 V	2.3	7 V
T _A	Operating free-air temperature	-0.3†	1.5	°C
		0	70	°C

† The algebraic convention, in which the least positive (most negative) designated minimum, is used in this data sheet for logic voltage levels.

electrical characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V _{OH}	High-level output voltage	L293: I _{OH} = -1 A L293D: I _{OH} = -0.6 A		V _{CC2} - 1.8	V _{CC2} - 1.4		V
V _{OL}	Low-level output voltage	L293: I _{OL} = 1 A L293D: I _{OL} = 0.6 A			1.2	1.8	V
V _{OKH}	High-level output clamp voltage	L293D: I _{OK} = -0.6 A			V _{CC2} + 1.3		V
V _{OKL}	Low-level output clamp voltage	L293D: I _{OK} = 0.6 A			1.3		V
I _{IH}	High-level input current	A	V _I = 7 V		0.2	100	μA
		EN			0.2	10	
I _{IL}	Low-level input current	A	V _I = 0		-3	-10	μA
		EN			-2	-100	
I _{CC1}	Logic supply current	I _O = 0	All outputs at high level		13	22	mA
			All outputs at low level		35	60	
			All outputs at high impedance		8	24	
I _{CC2}	Output supply current	I _O = 0	All outputs at high level		14	24	mA
			All outputs at low level		2	6	
			All outputs at high impedance		2	4	

switching characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER	TEST CONDITIONS	L293NE, L293DNE			UNIT
		MIN	TYP	MAX	
t _{PLH}	Propagation delay time, low-to-high-level output from A input		800		ns
t _{PHL}	Propagation delay time, high-to-low-level output from A input	C _L = 30 pF. See Figure 1	400		ns
t _{TLH}	Transition time, low-to-high-level output		300		ns
t _{THL}	Transition time, high-to-low-level output		300		ns

switching characteristics, V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER	TEST CONDITIONS	L293DWP, L293N L293DN			UNIT
		MIN	TYP	MAX	
t _{PLH}	Propagation delay time, low-to-high-level output from A input		750		ns
t _{PHL}	Propagation delay time, high-to-low-level output from A input	C _L = 30 pF. See Figure 1	200		ns
t _{TLH}	Transition time, low-to-high-level output		100		ns
t _{THL}	Transition time, high-to-low-level output		350		ns

Anexo D. Procesador del ESP-8266.



CP2102/9

SINGLE-CHIP USB-TO-UART BRIDGE

For newer designs, the CP2102N devices offer compatible footprints and are recommended for use instead of the CP2102/9. See the Silicon Labs website (www.silabs.com/usbxpress) for more information.

Single-Chip USB to UART Data Transfer

- Integrated USB transceiver; no external resistors required
- Integrated clock; no external crystal required
- Internal 1024-byte programmable ROM for vendor ID, product ID, serial number, power descriptor, release number, and product description strings
 - EEPROM (CP2102)
 - EPROM (One-time programmable) (CP2109)
- On-chip power-on reset circuit
- On-chip voltage regulator
 - 3.3 V output (CP2102)
 - 3.45 V output (CP2109)
- 100% pin and software compatible with CP2101

USB Function Controller

- USB Specification 2.0 compliant; full-speed (12 Mbps)
- USB suspend states supported via SUSPEND pins

Asynchronous Serial Data BUS (UART)

- All handshaking and modem interface signals
- Data formats supported:
 - Data bits: 5, 6, 7, and 8
 - Stop bits: 1, 1.5, and 2
 - Parity: odd, even, mark, space, no parity
- Baud rates: 300 bps to 1 Mbps
- 576 Byte receive buffer; 640 byte transmit buffer
- Hardware or X-On/X-Off handshaking supported
- Event character support
- Line break transmission

Virtual COM Port Device Drivers

- Works with existing COM port PC Applications
- Royalty-free distribution license
- Windows 8/7/Vista/Server 2003/XP/2000
- Mac OS-X/OS-9
- Linux

USBxpress™ Direct Driver Support

- Royalty-Free Distribution License
- Windows 7/Vista/XP/Server 2003/2000
- Windows CE

Example Applications

- Upgrade of RS-232 legacy devices to USB
- Cellular phone USB interface cable
- USB interface cable
- USB to RS-232 serial adapter

Supply Voltage

- Self-powered: 3.0 to 3.6 V
- USB bus powered: 4.0 to 5.25 V

Package

- RoHS-compliant 28-pin QFN (5x5 mm)

Ordering Part Numbers

- CP2102-GM
- CP2109-AD1-GM

Temperature Range: -40 to +85 °C

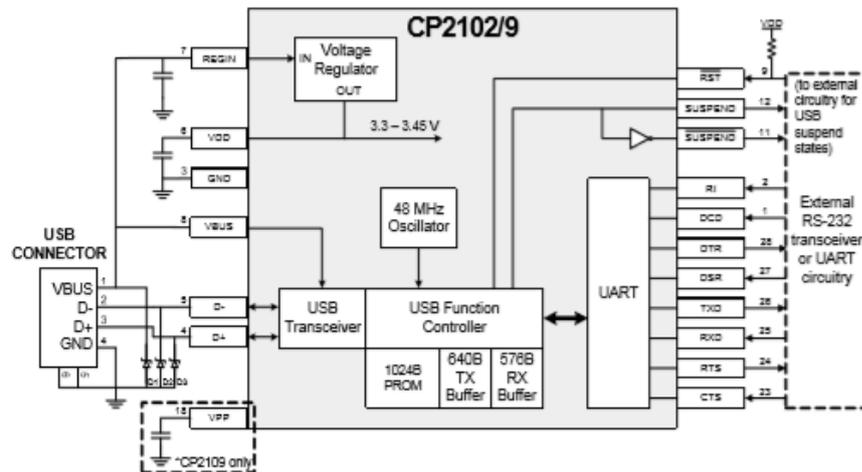
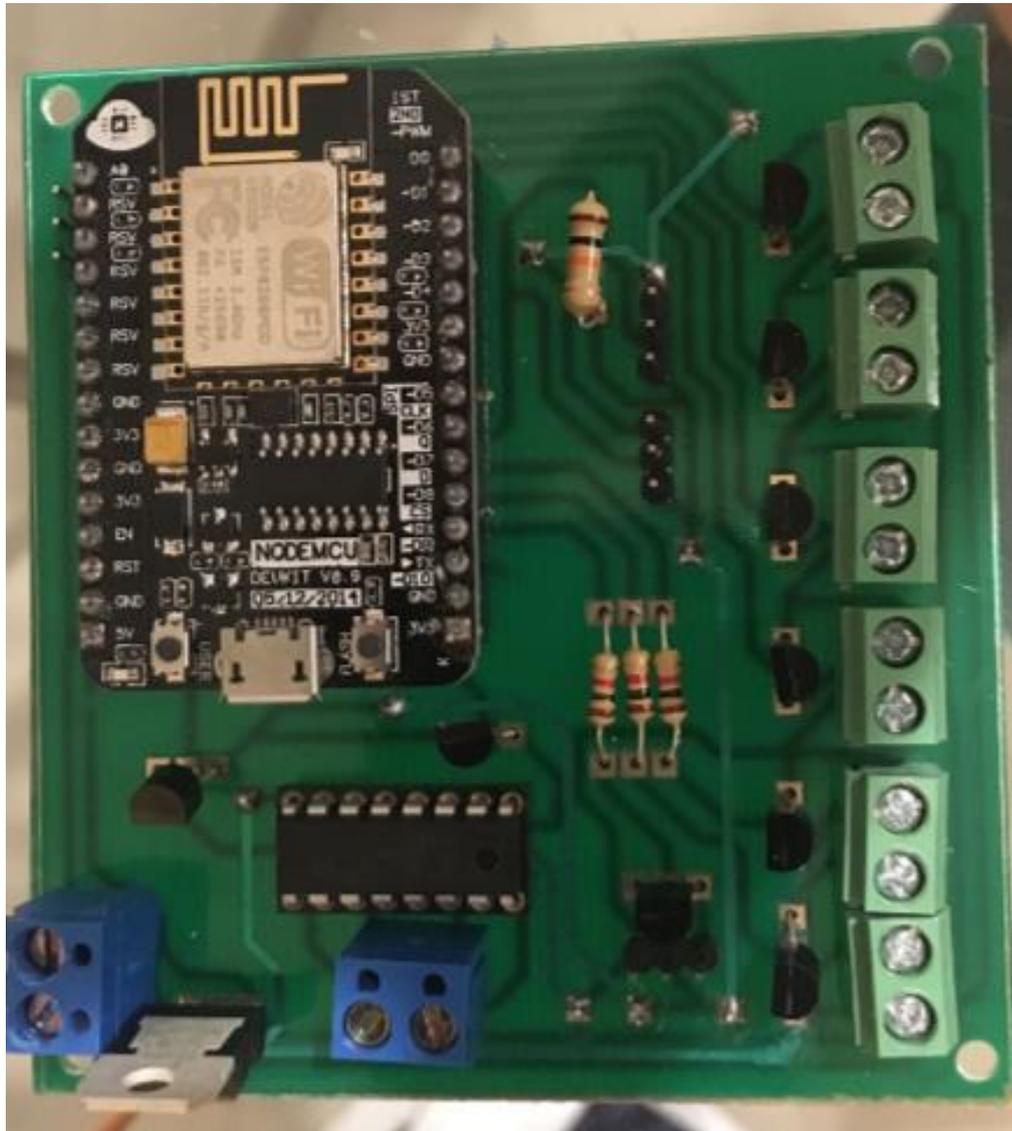


Figure 1. Example System Diagram

Anexo E. Imágenes de la PCB.



Anexo F. Imágenes de la maqueta.





Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Vivar Estrella, Ana Nicole** con C.C: # 0802882894 autora del Trabajo de Titulación: **Desarrollo de interfaces gráficas en Matlab y Android para el control de sistemas domóticos utilizando el dispositivo de comunicación inalámbrica NodeMcu Esp8266** previo a la obtención del título de **INGENIERA ELECTRÓNICA EN CONTROL Y AUTOMATISMO** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 13 de Marzo del 2018

Vivar Estrella, Ana Nicole

C.C: 0802882894

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Desarrollo de interfaces gráficas en Matlab y Android para el control de sistemas domóticos utilizando el dispositivo de comunicación inalámbrica NodeMcu Esp8266		
AUTOR(ES)	Ana Nicole Vivar Estrella		
REVISOR(ES)/TUTOR(ES)	M. Sc. Palacios Meléndez, Edwin Fernando		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería Electrónica en Control y Automatismo		
TÍTULO OBTENIDO:	Ingeniera Electrónico en Control y Automatismo		
FECHA DE PUBLICACIÓN:	13 de marzo del 2018	No. DE PÁGINAS:	105
ÁREAS TEMÁTICAS:	Domótica		
PALABRAS CLAVES/ KEYWORDS:	<i>Interfaz, Matlab, Android, NodeMcu ESP 8266, domótica, Wi-Fi, Firebase</i>		
RESUMEN/ABSTRACT (150-250 palabras):			
<p>El siguiente trabajo de titulación está ideado en el desarrollo de interfaces gráficas en Matlab y Android para el control de una vivienda inteligente donde se utiliza el dispositivo de comunicación inalámbrica NodeMcu ESP8266, donde se busca un mayor impacto en el ámbito de la domótica a nivel estudiantil en la UCSG. La idea se presenta por medio de una maqueta de una vivienda a escala, la cual permite observar la interacción entre el usuario a través de las interfaces con diferentes variables de la vivienda, como luces, puertas, persianas, y temperatura, lo novedoso es que se puede tener un control en tiempo real y además sin importar el lugar en el que se encuentre debido a que la comunicación se realiza por medio de Wi-Fi, fue necesario utilizar el servidor Firebase, el cual trabaja con una base de datos y permite enviar y recibir información desde las interfaces al sistema. En conclusión, se pudo realizar con éxito la implementación y además abre cabida a la posibilidad de no solo realizar este tipo de controles para sistemas domóticos, sino también para distintos procesos industriales.</p>			
ADJUNTO PDF:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NO
CONTACTO CON AUTOR/ES:	Teléfono: +593 994588704	E-mail: nicolevivar_95@hotmail.com	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Mendoza Merchán, Eduardo Vicente		
	Teléfono: +593 985086815		
	E-mail: eduardo.mendoza@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			