



Universidad Michoacana de  
San Nicolás de Hidalgo

Facultad de Ingeniería  
Eléctrica



# **IMPLEMENTACIÓN DE UN SISTEMA IoT Wi-Fi MONTADO EN UN SERVIDOR AWS CON INTERFAZ WEB**

Tesis para obtener el título profesional de Ingeniero en Electrónica

**Juan Manuel García Ruiz**

**Asesor**

M. C. Félix Jiménez Pérez

Morelia Michoacán – México  
Junio del 2019



# **IMPLEMENTACIÓN DE UN SISTEMA IOT Wi-Fi MONTADO EN UN SERVIDOR AWS CON INTERFAZ WEB**

## DEDICATORIA

*A mis padres por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo.*

*Todo este trabajo ha sido posible gracias a ellos.*

## AGRADECIMIENTOS

*Al personal docente de la Facultad de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo, claves en mi formación académica.*

*Al personal administrativo y manual de la Facultad de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo.*

*A mi familia y amigos por su apoyo incondicional.*

## TABLA DE CONTENIDO

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| CAPÍTULO 1: INTRODUCCIÓN.....                                              | 1  |
| 1.1 Antecedentes.....                                                      | 1  |
| 1.1.1 Electrónica.....                                                     | 1  |
| 1.1.2 Computación.....                                                     | 2  |
| 1.1.3 Automatización.....                                                  | 3  |
| 1.1.4 Informática.....                                                     | 3  |
| 1.1.5 Sistemas de Control.....                                             | 4  |
| 1.2 Domótica.....                                                          | 5  |
| 1.3 Internet de las cosas.....                                             | 6  |
| 1.4 Panorama general.....                                                  | 7  |
| 1.5 Retos y expectativas.....                                              | 8  |
| 1.6 Objetivo general.....                                                  | 9  |
| 1.7 Objetivos particulares.....                                            | 9  |
| 1.8 Justificación.....                                                     | 10 |
| 1.9 Descripción del contenido de los capítulos.....                        | 10 |
| CAPÍTULO 2: MARCO TEÓRICO.....                                             | 12 |
| 2.1 Microcontroladores.....                                                | 12 |
| 2.2 Electrónica de potencia.....                                           | 13 |
| 2.3 Sensores.....                                                          | 15 |
| 2.4 Actuadores.....                                                        | 15 |
| 2.5 Redes y comunicación modelo Cliente-Servidor.....                      | 18 |
| 2.6 Lenguajes de programación para desarrollo web.....                     | 21 |
| 2.7 Bases de datos.....                                                    | 22 |
| 2.8 Access Point.....                                                      | 23 |
| 2.10 Protocolo de Seguridad SSL.....                                       | 24 |
| 2.11 Protocolo de comunicación MQTT.....                                   | 27 |
| 2.12 Sistemas actuales de Domótica.....                                    | 28 |
| CAPÍTULO 3: HARDWARE UTILIZADO.....                                        | 32 |
| 3.1 ESP-12E Development Board.....                                         | 32 |
| 3.2 NodeMCU Cloud Build Service.....                                       | 37 |
| 3.3 Esptool.py.....                                                        | 37 |
| 3.4 Conexión para la programación del módulo ESP-01.....                   | 39 |
| 3.5 Circuito para control ON/OFF utilizando el módulo ESP-01.....          | 40 |
| 3.6 Lectura de temperatura y humedad utilizando el sensor DHT11.....       | 41 |
| CAPÍTULO 4: SOFTWARE UTILIZADO.....                                        | 45 |
| 4.1 ESPlorer.....                                                          | 45 |
| 4.2 Lenguaje Lua.....                                                      | 46 |
| 4.3 Node.js y Node Package Manager.....                                    | 46 |
| 4.4 Implementación del Broker MQTT con protocolo SSL usando Mosquitto..... | 48 |
| 4.5 Base de Datos MySQL.....                                               | 53 |
| 4.6 Amazon Web Services.....                                               | 55 |

|                                                                   |     |
|-------------------------------------------------------------------|-----|
| 4.7 Desarrollo de la página Web.....                              | 57  |
| 4.8 Obtención de un dominio web.....                              | 65  |
| 4.9 Módulo “Enduser Setup”.....                                   | 67  |
| CAPÍTULO 5: DESCRIPCIÓN DEL SISTEMA.....                          | 68  |
| 5.1 Diagrama de conexiones.....                                   | 68  |
| 5.2 Diagramas de flujo de información.....                        | 69  |
| 5.3 Esquema de la base de datos MySQL.....                        | 72  |
| 5.4 Interfaz de usuario.....                                      | 74  |
| 5.5 Diagramas de funcionamiento del sistema.....                  | 77  |
| CAPÍTULO 6: PRUEBAS Y RESULTADOS.....                             | 81  |
| 6.1 Conexión de los módulos a la red Wi-Fi.....                   | 81  |
| 6.2 Registro como nuevo usuario del sistema en la página Web..... | 82  |
| 6.3 Inicio de sesión.....                                         | 85  |
| 6.4 Agregar Redes Wi-Fi y visualizar dispositivos.....            | 86  |
| 6.5 Envío de comandos y lectura de dispositivos.....              | 88  |
| 6.6 Borrar una Red Wi-Fi.....                                     | 90  |
| 6.7 Cerrar Sesión.....                                            | 92  |
| 6.8 Eliminar cuenta.....                                          | 92  |
| CAPÍTULO 7: CONCLUSIONES.....                                     | 94  |
| 7.1 Conclusiones generales.....                                   | 94  |
| 7.2 Conclusiones particulares.....                                | 96  |
| CAPÍTULO 8: TRABAJOS FUTUROS.....                                 | 98  |
| REFERENCIAS.....                                                  | 100 |
| APÉNDICES.....                                                    | 103 |
| Apéndice A: Base de Datos MySQL, Comandos de interés.....         | 103 |
| Apéndice B: Renta y configuración de un servidor EC2 de AWS.....  | 105 |
| Apéndice D: Glosario.....                                         | 119 |
| Apéndice E: Abreviaturas y acrónimos.....                         | 121 |

## LISTA DE FIGURAS

|                                                                                                                       |    |
|-----------------------------------------------------------------------------------------------------------------------|----|
| Figura 1: Microcontrolador, fuente: [13].....                                                                         | 12 |
| Figura 2: Comunicación Cliente-Servidor, fuente: [20].....                                                            | 19 |
| Figura 3: Arquitectura de redes por capas, modelo OSI, fuente: [28].....                                              | 25 |
| Figura 4: Establecimiento de conexión segura con SSL, fuente: [27].....                                               | 26 |
| Figura 5: Protocolo MQTT, fuente: [30].....                                                                           | 28 |
| Figura 6: Google Home, fuente: [31].....                                                                              | 29 |
| Figura 7: Amazon Alexa, fuente: [32].....                                                                             | 30 |
| Figura 8: MyCroft Mark II, fuente: [34].....                                                                          | 31 |
| Figura 9: Apple HomeKit, fuente: [35].....                                                                            | 31 |
| Figura 10: ESP-12E Development Board, fuente: [36].....                                                               | 32 |
| Figura 11: ESP8266, Diagrama de Bloques, fuente: [36].....                                                            | 33 |
| Figura 12: ESP-01, fuente: [37].....                                                                                  | 36 |
| Figura 13: ESP-05, fuente: [37].....                                                                                  | 36 |
| Figura 14: ESP-12, fuente: [37].....                                                                                  | 36 |
| Figura 15: ESP-201, fuente: [37].....                                                                                 | 37 |
| Figura 16: Borrado de la Memoria Flash del NodeMCU.....                                                               | 38 |
| Figura 17: Programación del Firmware mediante esptool.py.....                                                         | 39 |
| Figura 18: Conexión necesaria para la programación del módulo ESP-01.....                                             | 40 |
| Figura 19: Circuito para el control On-Off utilizando un módulo ESP-01.....                                           | 41 |
| Figura 20: Circuito para la medición de temperatura y humedad.....                                                    | 41 |
| Figura 21: Señal de inicio para la lectura del sensor DHT11 [38].....                                                 | 43 |
| Figura 22: Respuesta del sensor tras la señal de inicio, DHT11[38].....                                               | 43 |
| Figura 23: Bits 0 y 1 de la respuesta del sensor DHT11[38].....                                                       | 43 |
| Figura 24: Lectura típica del sensor DHT11[38].....                                                                   | 44 |
| Figura 25: MQTT Broker, clientes MQTT, Mosquitto, fuente: [42].....                                                   | 48 |
| Figura 26: Bases de Datos, MySQL.....                                                                                 | 55 |
| Figura 27: Tabla, Base de Datos, MySQL, fuente: [44].....                                                             | 55 |
| Figura 28: Página web con CSS (izquierda) y sin CSS (derecha), fuente: [48].....                                      | 60 |
| Figura 29: Fragmento de Código, CSS.....                                                                              | 61 |
| Figura 30: Búsqueda de un dominio, <a href="http://www.freenom.com/">http://www.freenom.com/</a> .....                | 65 |
| Figura 31: Selección del periodo para el dominio, <a href="http://www.freenom.com/">http://www.freenom.com/</a> ..... | 66 |
| Figura 32: Dominios del usuario, <a href="http://www.freenom.com/">http://www.freenom.com/</a> .....                  | 66 |
| Figura 33: Asociar una IP a un dominio, <a href="http://www.freenom.com/">http://www.freenom.com/</a> .....           | 67 |
| Figura 34: Diagrama de interacción entre los elementos del sistema.....                                               | 69 |
| Figura 35: Diagrama de Flujo de información del sistema.....                                                          | 70 |
| Figura 36: Flujo de Información del sistema.....                                                                      | 72 |
| Figura 37: Diagrama de Flujo de la Interfaz de Usuario, Página de Bienvenida.....                                     | 75 |
| Figura 38: Diagrama de Flujo de la Interfaz de Usuario, Página del Usuario.....                                       | 76 |
| Figura 39: Diagrama de Flujo de la Interfaz de Usuario, Envío de Comandos.....                                        | 76 |
| Figura 40: Diagrama de funcionamiento, NodeJS.....                                                                    | 77 |
| Figura 41: Diagrama de funcionamiento, Servidor HTTP.....                                                             | 78 |



|                                                                                                                    |     |
|--------------------------------------------------------------------------------------------------------------------|-----|
| Figura 42: Diagrama de funcionamiento, Broker MQTT.....                                                            | 79  |
| Figura 43: Diagrama de funcionamiento, Módulos ESP.....                                                            | 80  |
| Figura 44: Enduser Setup, IP del Módulo.....                                                                       | 81  |
| Figura 45: Enduser Setup, Selección de Red Wi-Fi.....                                                              | 81  |
| Figura 46: Enduser Setup, Conexión realizada.....                                                                  | 82  |
| Figura 47: página web desarrollada para el sistema: <a href="http://www.juangarcia.tk">www.juangarcia.tk</a> ..... | 83  |
| Figura 48: Página de registro.....                                                                                 | 83  |
| Figura 49: Error de Registro: campo vacío.....                                                                     | 84  |
| Figura 50: Error de Registro: contraseña no coincide.....                                                          | 84  |
| Figura 51: Página del usuario.....                                                                                 | 85  |
| Figura 52: Página para Iniciar Sesión.....                                                                         | 85  |
| Figura 53: Error al Iniciar Sesión: campo faltante.....                                                            | 86  |
| Figura 54: Error al Iniciar Sesión: usuario no existente.....                                                      | 86  |
| Figura 55: Página para añadir una Red Wi-Fi.....                                                                   | 87  |
| Figura 56: Error al agregar una red Wi-Fi, Campos vacíos.....                                                      | 87  |
| Figura 57: Página del usuario, Lista de dispositivos.....                                                          | 88  |
| Figura 58: Envío del Comando “Rename device”.....                                                                  | 89  |
| Figura 59: Cambio del nombre del dispositivo a “Habitación principal”.....                                         | 90  |
| Figura 60: Página para borrar Red Wi-Fi.....                                                                       | 91  |
| Figura 61: Notificación al borrar red(es) Wi-Fi.....                                                               | 91  |
| Figura 62: Página del usuario tras borrar la red Wi-Fi.....                                                        | 92  |
| Figura 63: Página para Eliminar la Cuenta.....                                                                     | 92  |
| Figura 64: Error al Borrar la cuenta: Campos vacíos.....                                                           | 93  |
| Figura 65: Error al Borrar la Cuenta: Contraseña incorrecta.....                                                   | 93  |
| Figura 66: Services>Compute>EC2, AWS.....                                                                          | 105 |
| Figura 67: EC2 Dashboard>Create Instance.....                                                                      | 106 |
| Figura 68: Ubuntu Server AML.....                                                                                  | 107 |
| Figura 69: Tipos de Instancias EC2.....                                                                            | 108 |
| Figura 70: Configuración de la Instancia EC2.....                                                                  | 109 |
| Figura 71: Configuración de Almacenamiento de la Instancia EC2.....                                                | 110 |
| Figura 72: Grupo de Seguridad de la instancia EC2.....                                                             | 111 |
| Figura 73: Resumen de la Instancia EC2 a lanzar.....                                                               | 112 |
| Figura 74: EC2 Dashboard tras lanzar la instancia EC2.....                                                         | 113 |
| Figura 75: Configuración de una IP elástica, AWS.....                                                              | 114 |
| Figura 76: IP's elásticas, AWS.....                                                                                | 115 |
| Figura 77: Opciones para una IP elástica, AWS.....                                                                 | 116 |
| Figura 78: Menú para asociar una IP elástica a una Instancia EC2.....                                              | 117 |
| Figura 79: Información de red de la instancia EC2, AWS.....                                                        | 118 |
| Figura 80: Opciones de Estado de una instancia EC2, AWS.....                                                       | 119 |

## LISTA DE TABLAS

|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| Tabla 1: Módulo ESP-01 Modos de Operación.....                                      | 39  |
| Tabla 2: DHT11, ejemplo de lectura del sensor.....                                  | 44  |
| Tabla 3: Comandos principales de MQTT.....                                          | 52  |
| Tabla 4: Etiquetas HTML para modificar características del texto, fuente: [47]..... | 59  |
| Tabla 5: Ejemplos de eventos JavaScript, fuente: [49].....                          | 62  |
| Tabla 6: Características de peticiones HTTP GET y HTTP POST, fuente: [51].....      | 64  |
| Tabla 7: Comandos MySQL.....                                                        | 103 |

## ABSTRACT

In this thesis are described the tools, hardware and software, needed to create a home automation system, which is oriented to the control and monitoring of devices through a web page. The system consists of sensor and actuator modules, implemented with NodeMCU v3 development boards and ESP-01 and their respective sensor (exemplified by the use of a DHT11 digital sensor) or actuator (relays and a CD motor were used), which is connected to Internet and send information that is stored in a database and is displayed on the website. To send and receive information between the modules and the server was done through the MQTT protocol and additionally with a standard security and encryption protocol SSL. This information is received and stored in a server, to present it to the user who requests it. The website has user registration and identification; each user has an account, password and associated IoT devices. The server uses the MySQL database to store all the information of devices and users. The development of the server was developed in Node.js, which manages the MySQL database, the MQTT Broker, and the HTTP requests of the web page. The website was developed with HTML, CSS and JavaScript. The Cloud Build Service tool was used to design the firmware of the ESP-01 and NodeMCU modules, esptool.py to flash the firmware in the modules, Lua code and the ESPlorer user interface for programming them.

## RESUMEN

En esta tesis se exponen las herramientas necesarias, de hardware y software, para crear un sistema de domótica, el cual está orientado al control y monitoreo de dispositivos a través de una página web. El sistema consta de módulos sensores y actuadores, implementados con tarjetas de desarrollo NodeMCU v3 y ESP-01 y su respectivo sensor (se ejemplificó con el uso de un sensor digital DHT11) y como actuador (se emplearon relevadores y un motor de CD), los cuales se conectan a Internet y envían información que se almacena en una base de datos y se muestra en una página web. El envío y recepción de información entre los módulos y el servidor se realiza mediante el protocolo MQTT conjuntamente al protocolo de seguridad y encriptación estándar SSL. Dicha información es recibida y almacenada en un servidor, para posteriormente presentarla al usuario que la solicite. La página web cuenta con registro e identificación de usuarios; cada usuario tiene su cuenta, contraseña y dispositivos IoT asociados. El servidor utiliza la base de datos MySQL para almacenar la información de los dispositivos y usuarios. El desarrollo del servidor se implementó en Node.js, el cual gestiona la base de datos MySQL, el Broker MQTT, y las peticiones HTTP de la página web. La página web se desarrolló con HTML, CSS y JavaScript. Se utiliza la herramienta Cloud Build Service para el diseño del firmware de los módulos ESP-01 y NodeMCU, esptool.py para programar el firmware en los módulos, código Lua y la interfaz ESPlorer para la programación de los mismos.

**Palabras clave:** base de datos, sistemas embebidos, firmware, cómputo elástico en la nube, sistema en chip.

## CAPÍTULO 1: INTRODUCCIÓN

En este capítulo se ofrece un panorama general de la domótica, el internet de las cosas, el origen y desarrollo de algunas de las tecnologías gracias a las cuales se desarrollaron dichos sistemas, también se abordan los retos y expectativas actuales así como los objetivos que se tienen al desarrollar el sistema propuesto.

### 1.1 Antecedentes

Los intereses y las necesidades actuales han hecho replantear la solución de diversos problemas en entornos laborales, viviendas y zonas de recreación a la hora de automatizar funciones y tareas simples a primera vista. Para ello se ha recurrido a la electrónica, informática y, según sea el caso, a expertos en diferentes áreas relacionadas con la automatización. [1]

La evolución tecnológica de diferentes disciplinas, como la informática, la ingeniería de control, la electrónica, las telecomunicaciones y la automatización, entre otras, ha posibilitado el concepto de edificio inteligente, el cual tiene como propósito el automatizar actividades y tareas.

Durante los últimos años el avance tecnológico ha sido capaz de proporcionar al mercado equipos y sistemas que realizan tareas que pueden estar integradas dentro de un único sistema flexible en cuanto a modificaciones, ampliaciones y mantenimiento cubriendo las necesidades del usuario final utilizando la automatización, la domótica, el internet de las cosas entre otras tecnologías. [1]

#### 1.1.1 Electrónica

La Electrónica probablemente no se inicia hasta que Lorentz postuló en 1895 la existencia de cargas discretas denominadas electrones. En 1904, Fleming inventó el diodo que denominó válvula el cual consistía en un filamento caliente, emisor de electrones, situado en el vacío a corta distancia de una placa. En función de la tensión positiva o negativa de la placa, se producía paso de corriente en una dirección. Esta válvula se empleó como detector de señales inalámbricas y vino a sustituir a los detectores de galena utilizados hasta ese momento. Posteriormente en 1906, uno de los acontecimientos más importantes en la historia de la Electrónica fue cuando De Forest interpuso un tercer electrodo (rejilla) en una válvula de Fleming creando el tubo tríodo denominado audión, siendo el primer amplificador de la historia. [2]

La mayor revolución tecnológica de la Electrónica surge con la invención de los dispositivos basados en semiconductores, y más en concreto, con la invención del transistor. Los primeros trabajos sobre semiconductores fueron comenzados por Hall en 1879 sobre el efecto que lleva su nombre. Los primeros rectificadores de unión metal-semiconductor se estudian entre 1920 y 1938, y es en 1938 cuando Shottky y Mott realizan separadamente un estudio sistemático sobre las propiedades de estos dispositivos, proponiendo la primera teoría del espacio de carga. En esta época, se realizan muchos estudios sobre semiconductores y se perfeccionan las técnicas de crecimiento de cristales. En 1943, se obtiene la primera unión P-N sobre cristal único de silicio. [2]

El desarrollo de la Electrónica del siglo XX está ligado al desarrollo de la radio. Basados en tubos de vacío se construyeron diferentes tipos de circuitos con aplicación en las comunicaciones por radio. Con diodos y tríodos fueron diseñados los amplificadores en cascada, amplificadores regenerativos, osciladores, el receptor heterodino, entre otros. La evolución del tríodo dio lugar a técnicas de calentamiento indirecto del cátodo y a la introducción de los tetrodos y pentodos. [2]

### 1.1.2 Computación

La revolución microelectrónica introdujo una nueva industria: la computación. Esta industria surgió por la gran expansión que se produce en el campo de la electrónica digital. En 1960, Noyce y Norman introdujeron la primera familia lógica semiconductor, lógica resistencia-transistor (RTL), que sirvió de base para la construcción de los primeros circuitos integrados digitales. Seguidamente, en 1961, apareció la familia de acoplo directo (DCTL), y un año más tarde la lógica diodo transistor (DTL). En 1964, Texas Instrument presenta la lógica transistor-transistor (TTL), pero se logra la máxima expansión en electrónica digital con las familias lógicas basadas en el transistor MOS, debido a que su proceso de fabricación es más sencillo, permite mayor escala de integración y los consumos de potencia son más reducidos. El proceso de miniaturización en tecnología MOS se encuentra por debajo de 1 micra aproximándose rápidamente a su límite físico. Esto ha permitido que se puedan realizar circuitos integrados que incorporan millones de dispositivos. [2]

En 1971 se introduce el microprocesador, esto debido a la necesidad de producir un circuito estándar de propósito general y gran flexibilidad que sirviera para las calculadoras y fuera apto a otras muchas aplicaciones. En 1971, Intel introdujo en el mercado el microprocesador de cuatro bits conocido como el modelo 4004. Era una CPU (*Central Processing Unit*) completa monolítica con 45 instrucciones en tecnología PMOS con 2300 transistores. El éxito del procesador fue inmediato y su amplia difusión supusieron el inicio de una auténtica revolución industrial. Dos años posteriores a la presentación del primer procesador, Intel desarrolla el microprocesador de 8 bits 8008 con una velocidad de 50000 instrucciones por segundo. Este continuo desarrollo de los microprocesadores ha permitido en la actualidad construir procesadores de 64 bits con altas velocidades de procesamiento. [2]

La evolución de los microprocesadores es actualmente muy rápida, con creciente implantación en los procesos de automatización industrial, robótica, instrumentación inteligente, y en los elementos de sociedad de consumo, automóviles, electrodomésticos, automatización de oficinas, comunicaciones, redes informáticas, etc.

### 1.1.3 Automatización

La división del trabajo, o sea, la reducción de un proceso de fabricación a sus fases independientes más pequeñas, se desarrolló en la segunda mitad del siglo XVII, lo cual permitió incrementar la producción y reducir el nivel de especialización de los obreros. La mecanización fue la siguiente etapa necesaria para la evolución hacia la automatización. La simplificación del trabajo permitida por la división del trabajo posibilitó el diseño y construcción de máquinas que reproducían los movimientos o funciones del trabajador. A medida que evolucionó la tecnología de transferencia de energía, estas máquinas especializadas se motorizaron, aumentando así su eficacia productiva. El desarrollo de la tecnología energética también dio lugar al surgimiento del sistema industrial de producción, ya que todos los trabajadores y máquinas debían estar situados junto a la fuente de energía. [3]

Es por ello que la automatización comenzó a pasos agigantados durante el siglo XIX con el desarrollo industrial, puesto que permitió controlar y establecer secuencialmente los procesos productivos.

Los robots industriales, diseñados en un principio para realizar tareas sencillas en entornos peligrosos para los trabajadores, son hoy extremadamente hábiles y se utilizan para trasladar, manipular y situar piezas ligeras y pesadas, etc. [3]

En los edificios, las primeras funciones que se controlaban eran el clima, para lograr un grado de confort y un óptimo consumo energético. Posteriormente se comenzó a controlar otras funciones como el grado de humedad, la presión, el caudal del aire, etc. Además el desarrollo de la electrónica permitió un control descentralizado (división o reparto de los controladores) de estos procesos, y finalmente la informática permitió una gestión del edificio en su control. [1]

Hoy en día la gran mayoría de industrias están automatizadas, o bien utilizan tecnología de automatización en alguna etapa de sus actividades, esto con la finalidad de efectuar diversas tareas con mucha mayor rapidez y calidad de lo que podría hacerlo un ser humano. [3]

### 1.1.4 Informática

Un desarrollo importante realizado en la revolución industrial fueron las tarjetas perforadas, usadas para controlar los telares textiles, posteriormente la Informática se basó en esta poderosa herramienta para el procesamiento de datos (con la lógica de perforación o ausencia de perforación).

Hoy en día las han remplazado por medios magnéticos y ópticos de ingreso de información cuyo tamaño, capacidad y velocidades de acceso no tiene comparación con las Tarjetas Perforadas. [4] Con este tipo de tecnologías se puede almacenar y procesar enormes cantidades de información, lo cual puede ser aprovechado por un controlador para lograr obtener el resultado deseado.

### 1.1.5 Sistemas de Control

La automatización de los procesos industriales comenzó con la necesidad de almacenar la secuencia de operaciones y de los tiempos de aplicación de las mismas. Dado que en los procesos industriales se requirió uniformidad y calidad en la fabricación, se tomaron medidas de las variables críticas y modificaciones del proceso cuando existan diferencias con los valores de referencia. A este principio se le denomina retroalimentación y es uno de los principales elementos del control automático. Está claro que no en todos los casos se requiere, pero es imprescindible cuando se busca precisión y calidad. [3]

Desde el punto de vista teórico, la Ingeniería de Control se consolidó cuando se produjo el traslado y aplicación de los conocimientos y datos adquiridos en los problemas de amplificación de señales a los problemas de control industrial. [5]

Los primeros estudios sobre control automático empezaron en el siglo XVIII, y fue Watt en 1769, el que demostró su utilidad al diseñar y aplicar un regulador centrífugo de velocidad, en el control de máquinas de vapor. A partir de este momento se utilizaron masivamente controladores, pero pronto se detectaron problemas al querer aumentar la precisión. Esto conllevó al primer estudio de la teoría de control, realizada por J.C. Maxwell en 1868. [3]

La aparición de grandes sistemas de generación eléctrica y en la industria química durante la segunda guerra mundial, acrecentó la demanda de sistemas de regulación y control. En este periodo los criterios de estabilidad de Routh y Hurwitz, de finales del siglo XIX, adquirieron gran importancia y además se empezaron a utilizar los reguladores PID (proporcional, integral y derivativo). En 1932 Nyquist, marcó un gran avance en la teoría de control, permitiendo con su método determinar el comportamiento de un sistema retroalimentado mediante su respuesta en frecuencia en lazo abierto (sin retroalimentación). A este estudio se añadieron los de Bode y Black, con lo que se logró el asentamiento completo de los métodos de análisis frecuencial. [3]

Los sistemas de control juegan un papel importante en nuestra sociedad debido a su gran número de aplicaciones y que éstas invariablemente son cada vez más complejas. Los sistemas de control son de suma importancia, siendo su utilización vital en el avance de la ingeniería y la ciencia, permitiendo liberar al hombre de tareas repetitivas y su exposición a ambientes inseguros que pueden ser ejecutadas fácilmente por algún sistema de control automático con alta precisión. [6]



## 1.2 Domótica

La Domótica engloba diferentes sistemas con muy diversas cualidades, capaces de realizar diferentes acciones y comunicarse por distintos medios de transmisión. Estos sistemas posibilitan los niveles de automatización demandados con factores determinantes como la facilidad de uso, la integración de las funciones y la interactividad tanto entre ellos mismos como con el usuario. [1]

La Domótica, acorde a [1], abarca diferentes y muy diversos objetivos, por ejemplo:

- Accesibilidad: de tal forma que se permita, incluso a las personas con capacidad de movimiento o comunicación reducidas, el acceso y la circulación por el edificio.
- Acceso a los servicios de telecomunicación, audiovisuales y de información.
- Seguridad en caso de algún accidente o siniestro, de tal forma que los ocupantes puedan desalojar el edificio en condiciones seguras o tomar las medidas preventivas que sean necesarias.
- Higiene, salud y protección del medio ambiente, de tal forma que se alcancen condiciones aceptables de salubridad en el ambiente interior del edificio y que éste no deteriore el medio ambiente en su entorno inmediato.
- Protección o reducción de ruido, de tal forma que el ruido percibido permita a las personas realizar satisfactoriamente sus actividades.
- Ahorro de energía, de tal manera que se consiga un uso racional de la energía necesaria para la adecuada utilización del edificio.

Un edificio automatizado es aquel que, ante una solicitud o condiciones previstas, da una respuesta específica. Incluye tres áreas: confort, ahorro energético y seguridad.

Para que todos estos dispositivos puedan trabajar de forma conjunta es necesario que estén conectados a través de una red interna, red que generalmente se suele conocer por HAN (*Home Area Network*). Esta red, cableada o inalámbrica, suele dividirse en tres tipos de redes, según el tipo de dispositivos a interconectar y aplicaciones a ofrecer: la red de control, la red de datos y la red multimedia. Estas tres redes suelen estar integradas teniendo conexión con el exterior, lo cual se realiza a través de las redes públicas de telecomunicación: por ejemplo la red de telefonía celular, RDSI (Red Digital de Servicios Integrados), Internet, entre otras. La incorporación e integración de estas redes y dispositivos en la vivienda posibilitan una cantidad ilimitada de nuevas aplicaciones y servicios en el hogar, incrementando así la calidad de vida de sus habitantes. [7]

La AIDA (Asociación de Domótica e Inmótica Avanzada) define la Domótica como “*la integración en los servicios e instalaciones residenciales de toda tecnología que permita una gestión*

*energéticamente eficiente, remota, confortable y segura, posibilitando una comunicación entre todos ellos”.*

### 1.3 Internet de las cosas

El Internet de las Cosas es un conjunto de dispositivos que envían datos por Internet con diferentes métodos y finalidades, esto gracias a la convergencia de varias tecnologías, incluyendo las comunicaciones inalámbricas, los sistemas microelectrónicos y por supuesto, Internet. Un dispositivo IoT (*Internet of Things*) es cualquier objeto con electrónica integrada capaz de transferir datos sobre una red, sin ninguna interacción humana. Algunos ejemplos son los sensores medioambientales, la maquinaria de las fábricas, los dispositivos en hogares y edificios o los componentes de un vehículo. Las zonas IoT son aquellos lugares o configuraciones donde los datos de IoT procedentes de dispositivos conectados, combinados con sistemas de computación cognitiva, facilitarán la transformación innovadora de la sociedad. [8]

A medida que aumenta la conectividad en todos los ámbitos, desde coches hasta campos de cultivo o cintas transportadoras, IoT está cambiando los paradigmas de la sociedad actual, tanto tecnológicamente como también en el sector productivo. El Internet de las cosas hace posibles innovadores modelos de economía: por ejemplo, en un edificio conectado, los datos se pueden utilizar para ahorrar energía y hacer mantenimiento predictivo. Los sensores en línea de producción incrementan la eficiencia. Los dispositivos en tiendas ayudan a los comercios a mejorar la experiencia del cliente. Los datos médicos procedentes de diferentes dispositivos, como relojes o teléfonos inteligentes, se pueden utilizar para salvar vidas. [8]

El uso realmente transformador del Internet de las Cosas consiste en combinar los datos estructurados y no estructurados con la computación cognitiva. Los datos estructurados son datos que pueden ser ordenados y procesados fácilmente, mientras que los datos no estructurados son un conglomerado masivo y desorganizado de varios objetos que no tienen valor hasta que se identifican y almacenan de manera organizada. El objetivo de la computación cognitiva es crear sistemas informáticos automatizados que procesen los datos recopilados para ser capaces de resolver problemas sin necesidad de asistencia humana; las tecnologías cognitivas son capaces de darle sentido a grandes cantidades de datos IoT para comprender en profundidad lo que está sucediendo en el sistema de donde son recopilados. Dotando de inteligencia a los sistemas y procesos, se puede incrementar la eficiencia de una acción, mejorar la satisfacción del cliente, descubrir nuevas oportunidades de negocio y reducir riesgos y amenazas de forma proactiva. [8]

## 1.4 Panorama general

Uno de los principales problemas de la implementación concreta de la Domótica es la creencia extendida de que es algo muy caro y por tanto son casas sólo para personas con alto poder adquisitivo. Pero esta idea no es totalmente cierta, pues existen sistemas económicos. [7]

Aunque la introducción de la Domótica en los edificios nuevos depende principalmente de los promotores inmobiliarios, el interés en la misma es una realidad palpable en la sociedad y además su introducción en los edificios nuevos resulta bastante más barata que en los edificios preexistentes; esto se debe a diversos motivos: la dificultad de integrar los dispositivos con el resto de instalaciones del edificio, las redes de interconexión de los distintos dispositivos pueden modificar la infraestructura del edificio, el usuario no suele contar con experiencia en diseño e instalación de sistemas domóticos, etc. Las alternativas en cuanto a las redes de comunicación entre los distintos dispositivos del edificio preexistente son menores que en los edificios nuevos, pues en los edificios ya construidos es preferible evitar las posibles modificaciones a la infraestructura puesto que hace algunos años, las tecnologías inalámbricas no estaban suficientemente desarrolladas; la red de control y de datos utilizaban habitualmente medios cableados. Sin embargo, un edificio que cuente con la automatización y control propio de los sistemas de domótica tiene, por lo general, mayor atractivo de cara al público. [7]

Debido al actual uso de Internet, las nuevas tecnologías y comunicación, se habla cada vez más de conceptos como la vivienda en red, el hogar digital y la casa conectada, que permiten nuevas funciones: control a distancia o teleoperación, entretenimiento, trabajo en casa, etc.

En la actualidad los ejemplos típicos de edificios automatizados son los grandes centros comerciales y los edificios de oficinas y bancos, a los cuales desde hace años se han ido añadiendo servicios: las escaleras mecánicas, la calefacción, control de la iluminación, sistemas antirobo y sistemas contra incendios, etc.

Las principales características o rasgos generales que debe tener un sistema de gestión técnica de un edificio inteligente, acorde a [1], se pueden resumir en los siguientes puntos:

- Simple y fácil de utilizar. El sistema de control debe ser simple y fácil de utilizar para que sea aceptado por los usuarios finales. La interfaz de usuario deberá ser sencilla e intuitiva de utilizar, para permitir un aumento de confort.
- Flexible. Debe tener prevista la posibilidad de adaptaciones futuras, de forma que ampliaciones y modificaciones se puedan realizar sin un costo elevado ni un esfuerzo grande.
- Modular. El sistema de control del edificio debe ser modular, para evitar fallos que puedan llegar a afectar a todo el edificio, y además debe permitir la fácil ampliación de nuevos servicios.

- Integral. El sistema debe permitir el intercambio de información y comunicación entre diferentes áreas de gestión del edificio, de forma que los diferentes subsistemas estén perfectamente integrados.

En la actualidad se están desarrollando trabajos de normalización relacionados con la Domótica tanto en organismos europeos, por ejemplo: CENELEC (Comité Europeo de Normalización Electrónica) el CEN (Comité Europeo de Normalización), como en organismos internacionales, por ejemplo la ISO (Organización Internacional de Normalización) o la IEC (Comisión Electrónica Internacional), por nombrar algunos. La ISO, creada en 1947, es una federación mundial no gubernamental de organismos nacionales de normalización, que produce normas internacionales industriales y comerciales. Dichas normas se conocen como normas ISO (estrechamente ligada a CEN) y su finalidad es la coordinación de las normas nacionales con el propósito de facilitar el comercio, facilitar el intercambio de información y contribuir a la transferencia de tecnologías. [9]

La ITU (Unión Internacional de Telecomunicaciones) se creó en 1932, cuenta con 189 países miembros y más de 600 participantes del sector. Su misión es adoptar recomendaciones con el objetivo de normalizar las telecomunicaciones a nivel mundial. El contenido de estas recomendaciones, a nivel de relaciones internacionales, es considerado como obligatorio por las Administraciones y Empresas Operadoras. La ITU está muy ligada a la ETSI (Instituto de Estándares de Telecomunicación Europeos). [10]

Otra institución de regulación fundamental para el desarrollo de la Domótica es la comisión Electrónica Internacional (CEI, o IEC en siglas inglesas), es una organización de normalización en los campos eléctrico, electrónico y tecnologías relacionadas. [11]

### 1.5 Retos y expectativas

En la actualidad tanto los sistemas IoT como la domótica son un área comercial poco implementada en edificios tipo casa habitación, centrándose en mayor parte en grandes instalaciones industriales, centros comerciales, bancos, oficinas, es por ello que se ha creado la mentalidad errónea de reservar este tipo de tecnologías y los beneficios que conlleva únicamente para personas físicas o morales con un poder adquisitivo considerablemente mayor al de la media nacional. Aunado a esto, se tiene un bajo nivel de comercialización y divulgación, y los clientes potencialmente activos pasan totalmente desapercibidos, dejando un mercado poco aprovechado y básicamente virgen en nuestro país.

A pesar que en el INEGI (Instituto Nacional de Estadística y Geografía) se puede encontrar información precisa sobre pavimentación, electrificación, drenaje, edad de los habitantes, entre otras características de las viviendas de la población, formalmente no hay datos que permitan identificar el nivel de automatización de los hogares mexicanos ni sus necesidades de integración de control de

iluminación, riego, temporizar actividades domésticas, actividades de apoyo a personas de la tercera edad y las orientadas hacia los menos protegidos, además de servicios médicos, etc. [12]

Sin embargo, la tendencia es interconectar los componentes electrónicos del hogar con otros de éste o de otro sitio, sin que las personas participen en las funciones básicas de operatividad. La Domótica está en continua evolución; las casas no solamente serán controladas de forma automática, sino que también aprenderá de sus procesos de rutina y posteriores ventajas que tentativamente proporcionarán los avances en nanotecnología, mecánica, bioingeniería, así como los sensores y actuadores utilizados, procesamiento de datos en tiempo real, etc. [12]

### 1.6 Objetivo general

Implementar un sistema IoT centralizado para monitorear y controlar dispositivos por medio de una página web utilizando mayoritariamente hardware de bajo costo y software libre, esto con la finalidad de proporcionando seguridad, confort y uso eficiente de la energía a las personas usuarias de edificios tipo casa habitación, siendo lo suficientemente flexible y adaptable para ser implementado en cualquier infraestructura y a un costo accesible en comparación con otras opciones del mismo tipo de sistema que se encuentran actualmente en el mercado.

### 1.7 Objetivos particulares

Instanciar una máquina virtual EC2 (*Elastic Compute Cloud*) de AWS (*Amazon web services*) con Ubuntu Server 18.04 y montar en ella un servidor HTTP (*Hyper Text Transfer Protocol*), un Broker MQTT (*Message Queue Telemetry Transport*) y una base de datos SQL (*Structured Query Language*). Utilizar NodeJS para recibir y responder las peticiones HTTP, recibir y responder mensajes MQTT y almacenar, consultar y borrar información de la base de datos SQL.

Utilizar el protocolo SSL (*Secure Sockets Layer*) para encriptar la información enviada entre los módulos, el usuario y el servidor.

Elaborar un módulo actuador tipo On-Off utilizando un relevador, un módulo utilizando el sensor digital DHT11 y un módulo tipo actuador que controle la velocidad y el sentido de giro de un motor de corriente directa con resolución de 10 bits. Utilizar el protocolo MQTTS (*Message Queue Telemetry Transport Secure*) para hacer la conexión y comunicación entre los módulos y el Broker MQTT montado en el servidor AWS.

Utilizar “*Enduser Setup*” para realizar la conexión de los módulos a una red Wi-Fi (*Wireless fidelity*) de manera sencilla (esto evita la reprogramación de los módulos cuando se cambie de red Wi-Fi).

Diseñar una interfaz web intuitiva y fácil de utilizar, además de solicitar una contraseña (proporcionada por el usuario al momento del registro) para autenticar de manera segura al usuario. La interfaz web también ofrece al usuario la posibilidad de cambiar su nombre y su contraseña, renombrar los módulos para hacer más intuitiva la visualización de información, agregar o eliminar dispositivos, visualizar las redes Wi-Fi que el usuario tenga registradas. Utilizar peticiones HTTP tipo GET y POST para interactuar con el servidor AWS.

Almacenar la información necesaria para autenticar usuarios, verificar la información de las posibles redes Wi-Fi utilizadas por los usuarios, y la información relacionada a los módulos, por ejemplo la dirección MAC (*Media Access Control*), estado actual, nombre, entre otros. Cuando haya un cambio en los módulos, redes Wi-Fi o en algún usuario (registro, cambio de nombre, cambio de contraseña o eliminación de la cuenta) actualizar la base de datos. El acceso a la base de datos se realiza restringiendo las direcciones IP (*Internet Protocol*) que no sean reconocidas por el servidor para evitar que la información contenida en la base de datos sea modificada de manera remota.

### 1.8 Justificación

Actualmente el desarrollo y comercialización de los sistemas de domótica y sistemas embebidos IoT ha ido incrementando en los últimos años aprovechando el desarrollo de diferentes tecnologías y protocolos de comunicación, sin embargo, la adquisición de dichos sistemas suele implicar la compra de productos adicionales a los ya existentes en un hogar: focos con la capacidad de controlarse por medio de una aplicación o programa para sustituir a los focos convencionales, sistemas de calefacción o ventilación controlables de manera remota, productos mediante los cuales se pueda interactuar haciendo uso de comandos de voz para controlar diferentes dispositivos, etc. además de limitarse a una aplicación móvil que puede tener limitaciones en cuanto a los sistemas operativos con los que tiene compatibilidad. Por estas razones se desarrolló un sistema competitivo en costo y funcionalidad para el control y monitoreo de dispositivos en el cual se pueden incorporar elementos ya existentes en los hogares, por ejemplo focos, reduciendo así los costos de adquisición de dispositivos adicionales compatibles con el sistema, además de tener la versatilidad de tener una página web para la interfaz de usuario, siendo compatible con cualquier dispositivo (celular, tableta, computadora, etc.).

### 1.9 Descripción del contenido de los capítulos

- **CAPÍTULO 1: INTRODUCCIÓN:** se explica de manera general el desarrollo tecnológico clave para el surgimiento de la Domótica, posteriormente se explica qué es la Domótica, su desarrollo e implementación. Se plantea el problema a resolver así como también los objetivos, general y particulares, y la justificación.

## CAPÍTULO 1: INTRODUCCIÓN

- **CAPÍTULO 2: MARCO TEÓRICO:** se expone el contexto de hardware, software y otras herramientas tecnológicas claves para el desarrollo de la tesis. El capítulo tiene el objetivo de poner al lector en contexto en las temáticas expuestas, esto con el fin de mejorar la comprensión del trabajo.
- **CAPÍTULO 3: HARDWARE UTILIZADO:** se describen los componentes de hardware empleados para la realización de la tesis, su funcionamiento y características. En este capítulo se encuentran los circuitos implementados para los módulos que el usuario puede visualizar en la página web.
- **CAPÍTULO 4: SOFTWARE UTILIZADO:** se profundiza en el software utilizado para el desarrollo de la tesis, describiendo su uso y características.
- **CAPÍTULO 5: DESCRIPCIÓN DEL SISTEMA:** se utilizan diagramas y esquemas para explicar la manera en que se integran los componentes del sistema, el funcionamiento del sistema en términos de la interacción del hardware y software y la forma en que éstos gestionan la información, y finalmente se da una descripción de cómo funciona la interfaz de usuario.
- **CAPÍTULO 6: PRUEBAS Y RESULTADOS:** se muestra el sistema en funcionamiento en apartados claves (registro de un nuevo usuario, gestión de dispositivos, gestión de redes Wi-Fi, entre otros), exponiendo los resultados de dichas acciones.
- **CAPÍTULO 7: CONCLUSIONES:** en este capítulo se exponen las conclusiones a las que se llegaron tras la realización de la tesis.

## CAPÍTULO 2: MARCO TEÓRICO

Para un mejor entendimiento del desarrollo de este trabajo es recomendable tener claros diferentes conceptos tecnológicos, herramientas de hardware y recursos de software, mismos que son expuestos de manera general en este capítulo. Posteriormente, se trata de manera más específica lo referente al hardware y software utilizado en el CAPÍTULO 3: HARDWARE UTILIZADO y el CAPÍTULO 4: SOFTWARE UTILIZADO respectivamente.

### 2.1 Microcontroladores

Un microcontrolador es un circuito integrado que en su interior contiene una unidad CPU, memorias RAM (*Random Access Memory*) y ROM (*Read Only Memory*), puertos de entrada salida y periféricos como ADC (*Analog to Digital Converter*) y DAC (*Digital to Analog Converter*), por nombrar algunos. Estas partes están interconectadas dentro del circuito integrado, y en su conjunto es una microcomputadora completa, véase la Figura 1. Toda microcomputadora requiere de un programa para realizar una función específica. Este se almacena normalmente en la memoria ROM. El propósito general de los microcontroladores es el de leer y ejecutar los programas que el usuario escribe, por esto la programación es una actividad básica e indispensable cuando se diseñan circuitos y sistemas que los incluyen. El carácter programable de los microcontroladores simplifica el diseño de circuitos electrónicos; permiten modularidad y flexibilidad. [13]

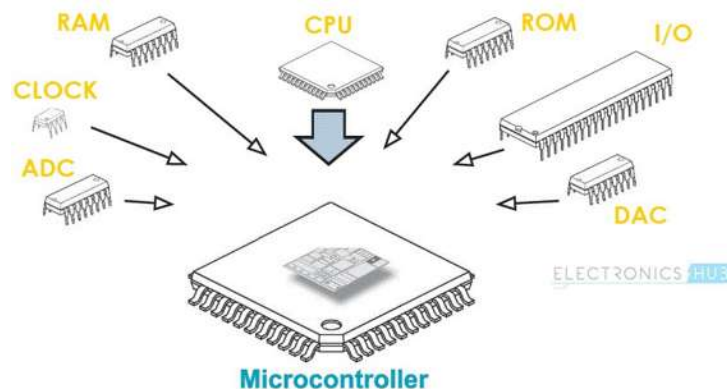


Figura 1: Microcontrolador, fuente: [13]

Los microcontroladores están diseñados para interpretar y procesar datos e instrucciones en forma binaria: 1's y 0's. Estos dígitos son conocidos como "bits", siendo la unidad mínima de información.



Sin embargo, por comodidad para el usuario, la programación de los microcontroladores comúnmente se lleva a cabo en un lenguaje de alto nivel, es decir, que utilice frases o palabras semejantes o propias del lenguaje humano, lo cual facilita enormemente la programación. Cualquiera que sea el lenguaje que se utilice en la programación de los microcontroladores, lo mas recomendable profundizar en su arquitectura interna, con este conocimiento se pueden aprovechar mas y mejor las capacidades de un microcontrolador. [13]

Todo programa escrito en un lenguaje de alto nivel debe ser transformado en código de máquina. Un software de computadora, llamado compilador, “traduce” el programa que realiza el usuario a código de máquina para poder ser interpretado por el microcontrolador.

Existen varios fabricantes de microcontroladores, tales como Texas Instruments, Motorola, Atmel, Intel, Microchip, Toshiba, por mencionar algunos. Evidentemente algunos modelos tienen más capacidad que otros, en cuanto a memoria, velocidad, periféricos, etc.

## 2.2 Electrónica de potencia

La historia de la electrónica de potencia se inicia en 1900 con la introducción del rectificador de arco de mercurio. Después se introdujeron en forma gradual el rectificador de tanque metálico, el tubo al vacío controlado por rejilla, el ignitrón, el fanotrón y el tiratrón. Estos dispositivos se aplicaban para el control de potencia hasta la década de 1950. La primera revolución electrónica comenzó en 1948, con la invención del transistor de silicio en los Bell Telephone Laboratories, por Bardeen, Brattain y Schockley. El siguiente adelanto, en 1956, también logrado en los Bell Laboratories, fue el transistor de disparo PNP (el cual alterna materiales tipo P y N), que se definió como tiristor, o SCR (rectificador controlado de silicio). La segunda revolución electrónica comenzó en 1958, con el desarrollo del tiristor comercial por la General Electric Company. Desde que se desarrolló el primer tiristor de SCR a finales de 1957 ha habido progresos impresionantes en los dispositivos semiconductores de potencia. A partir de 1970 se desarrollaron varios tipos de dispositivos semiconductores de potencia que entraron al comercio, sin embargo, los dispositivos de carburo de silicio todavía están en desarrollo y la mayor parte de los dispositivos se fabrican con silicio. Estos dispositivos se pueden dividir, de acuerdo con [14], en forma general en tres clases :

- Diodos de potencia
- Transistores
- Tiristores

También se pueden dividir en general en seis tipos:

1. Diodos de potencia

2. Tiristores
3. BJT (Transistores de unión bipolar)
4. MOSFET (Transistores de efecto de campo de óxido de metal semiconductor )
5. IGBT (Transistores bipolares de compuerta aislada)
6. SIT (Transistores de inducción estática)

Un diodo tiene dos terminales: ánodo y cátodo, pueden ser de tres tipos: de propósito general, de alta velocidad y Schottky. Un tiristor tiene tres terminales: un ánodo, un cátodo y una compuerta. Cuando se hace pasar una corriente pequeña por la terminal de la compuerta , hacia el cátodo, el tiristor conduce siempre que la terminal del ánodo tenga mayor potencial que el cátodo. Los transistores de potencia son de cuatro clases: BJT, MOSFET, IGBT y SIT. Un transistor tiene tres terminales: base, emisor y colector. En el caso normal funciona como un interruptor en la configuración de emisor común. Mientras la base de un transistor NPN (dos materiales tipo N y en medio tipo P) esté a mayor potencial que el emisor, y la corriente de base tenga el valor suficiente para activar el transistor en la región de saturación, el transistor permanece cerrado, siempre que la unión de colector a emisor tenga la polarización correcta. [14]

En electrónica de potencia se combinan la potencia, la electrónica y el control. El control tiene que ver con las características de estado estable y dinámicas de sistemas de lazo cerrado. La potencia tiene que ver con el equipo estático y rotatorio para la generación, transmisión y distribución de la energía eléctrica. La electrónica tiene que ver con los dispositivos y circuitos de estado sólido para el procesamiento de señales que cumplan con los objetivos deseados en el control. La electrónica de potencia se puede definir como las aplicaciones de la electrónica de estado sólido para el control y la conversión de la energía eléctrica. [14]

La electrónica de potencia se basa principalmente en la conmutación de dispositivos semiconductores de potencia. Con el desarrollo de la tecnología de semiconductores de potencia, las capacidades de manejo de potencia y la velocidad de conmutación de los dispositivos de potencia ha mejorado en forma considerable. El desarrollo de los microprocesadores y la tecnología de las microcomputadoras tiene un gran impacto sobre el control y la sintetización de la estrategia de control para los dispositivos semiconductores de potencia. [14]

Se utiliza en una gran diversidad de dispositivos, como controles de temperatura, de iluminación, de motores, fuentes de poder, sistemas de impulsión de vehículos y sistemas de corriente directa en alto voltaje (HVDC). [14]

## 2.3 Sensores

Los sensores son los dispositivos encargados de recopilar información de los diferentes parámetros que controla el sistema (gas, temperatura, agua, humedad, luz, movimiento, etc.). En general, los sensores, de acuerdo con [7], se clasifican en cinco tipos principales: de contacto, infrarrojos, ultrasónicos, vibración y microondas:

1. Los sensores de contacto (electromecánicos) se basan en la apertura o cierre de un circuito al actuar sobre el sensor, que puede ser un interruptor eléctrico que se activa al abrir una puerta o un sensor magnético que no necesita contacto entre las partes en movimiento. Estos detectores son muy económicos y apenas requieren mantenimiento, suelen ser utilizados para protección periférica. Permiten activar la carga directamente, como puede ser una lámpara.
2. Los sensores infrarrojos son sistemas activos que emiten radiación no visible. El sistema puede ser de barrera, con un emisor y un receptor separados unos cuantos metros, o “reflex”, donde el emisor y el receptor se encuentran juntos y el haz de luz se refleja en un espejo frente al mismo, siendo en este caso el alcance menor.
3. Los sensores ultrasónicos miden la distancia mediante el uso de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde el objeto contando el tiempo entre la emisión y la recepción. El tiempo tiene una relación directa con la distancia al objeto.
4. Los sensores de vibración utilizan sensores piezoeléctricos que son capaces de convertir variaciones de movimiento del cuerpo al que están adheridos en variaciones de tensión eléctrica. Estos sensores se utilizan en vigilancia perimétrica.
5. Los sensores de microondas son detectores volumétricos que se basan en la emisión de ondas electromagnéticas. Constan de un emisor y un receptor en la banda de 10 GHz y su principal ventaja es su gran sensibilidad, y son capaces de detectar movimientos muy pequeños, en distancias de hasta 50 metros.

## 2.4 Actuadores

Un actuador es un dispositivo capaz de transformar presión neumática, presión hidráulica y fuerza motriz eléctrica para modificar el estado de ciertos equipos o instalaciones con la finalidad de generar un efecto sobre un proceso. Actualmente existen dos tipos de actuadores:

1. Lineales
2. Rotatorios

## CAPÍTULO 2: MARCO TEÓRICO

Los actuadores lineales generan una fuerza lineal recta, tal como haría un pistón. Los actuadores rotatorios generan una fuerza rotatoria, como lo haría un motor eléctrico. Los actuadores hidráulicos son los de mayor antigüedad, pueden ser clasificados de acuerdo con la forma de operación, funcionan en base a fluidos a presión. Los actuadores neumáticos son los mecanismos que convierten la energía del aire comprimido en trabajo mecánico y aunque en esencia son idénticos a los actuadores hidráulicos, el rango de compresión es mayor, además de que hay una pequeña diferencia en cuanto al uso y en lo que se refiere a la estructura, debido a que estos tienen poca viscosidad.

La estructura de un actuador eléctrico es simple, ya que sólo requieren de energía eléctrica como fuente de poder. Es altamente versátil y prácticamente no hay restricciones respecto a la distancia entre la fuente de poder y el actuador.

Los actuadores eléctricos se dividen principalmente en cinco tipos:

- **Relés:** un relé es una solución de interrupción de potencia que puede usarse para asignar energía sin tener que abrir y cerrar un conmutador de forma manual. Para conectar y desconectar la energía, un relé solo requiere una pequeña señal eléctrica. Esta señal actúa como una habilitación para una señal eléctrica mucho más grande. Existen EMR (relés electromecánicos) y SSR (relés de estado sólido). Un relé electromecánico usa una pieza móvil física para conectar los contactos dentro del componente de salida del relé; el movimiento de este contacto se genera mediante fuerzas electromagnéticas provenientes de la señal de entrada de baja potencia, lo que permite completar el circuito que contiene la señal de alta potencia. Un relé de estado sólido usa una señal eléctrica de baja potencia para generar una señal de semiconductor óptico, generalmente un optoacoplador, que transmite y energiza la señal de salida; cuando se activa, la señal óptica de entrada actúa como el conmutador que permite que una señal de mayor voltaje o potencia pase a través del componente de salida del SSR. [15]
- **Solenoides:** un solenoide se define como una bobina de alambre, normalmente con la forma de un cilindro largo, que al transportar una corriente se asemeja a un imán de modo que un núcleo móvil es atraído a la bobina. Un solenoide convierte energía eléctrica en energía mecánica. Pueden variar en tamaño de menos de un cuarto de pulgada a más de 15 pulgadas de diámetro. Generalmente se utilizan como pulsador de algún otro botón a distancia, para abrir o cerrar una puerta eléctrica, etc. Los solenoides lineales tienen aplicaciones en electrodomésticos, máquinas vendedoras, seguros de puerta, cambiadores de monedas, disyuntores de circuito, bombas, aparatos médicos, transmisiones automotrices, entre otras. Los solenoides giratorios tienen aplicaciones en máquinas, herramientas, rayos láser, procesamiento fotográfico, almacenamiento de medios, aparatos médicos, por nombrar algunas. [16]
- **Motores CC (Corriente Continua):** los motores son máquinas que convierten energía eléctrica en energía mecánica usando el principio de inducción magnética, o sea que la alimentación del

motor es eléctrica y se aplica un voltaje en las terminales que produce una corriente, la potencia de salida es mecánica; la parte giratoria es el rotor y la potencia mecánica se transmite a través del eje de éste en forma de un par. Tienen un devanado de campo y una armadura con escobillas y conmutador. El conmutador mantiene a la armadura girando a través del campo magnético del devanado de campo. También cambia el flujo de corriente con relación al devanado de campo y la armadura, siendo esta acción creada por los polos norte y sur de los devanados de campo y armadura. En la medida que un motor acciona la carga, la potencia mecánica se toma de éste y a su vez obtiene la potencia eléctrica de la red. [17]

- Motores CA (Corriente Alterna): al igual que los motores de CC tiene una parte móvil (rotor) y una parte fija (estator) y su finalidad es transformar energía eléctrica en energía mecánica. Una característica común de todos los motores de CA, monofásicos o trifásicos, es un campo magnético rotatorio producido por el devanado del estator. Existen diferentes tipos de motores de CA: motores de jaula de ardilla (trifásico); el rotor de un motor con jaula de ardilla está hecho de barras conductoras que están en paralelo con el eje y en corto circuito por medio de unos anillos en los extremos en los que se soportan físicamente, el tamaño de la barra, forma y resistencia influyen en forma significativa en las características par-velocidad. El motor (trifásico) de inducción de rotor devanado opera bajo los mismos principios que los motores de jaula de ardilla, pero difieren en la construcción del rotor: en lugar de las barras en corto circuito, el rotor está construido por bobinas cuyas terminales llegan a unos anillos rozantes montados sobre el eje, además, la conexión de resistencias externas al circuito del rotor a través de los anillos rozantes permite la variación de las características par-velocidad. El motor monofásico cuando está en operación desarrolla un campo magnético rotatorio, pero antes que el rotor inicie la rotación, el estator produce sólo un campo estacionario pulsante. Para producir un campo rotatorio y, por lo tanto, un par de arranque, se debe tener un devanado auxiliar defasado  $90^\circ$  con respecto al devanado principal. Una vez que el motor ha arrancado el devanado auxiliar se desconecta del circuito. [17]
- Motores a pasos: es un dispositivo electromecánico que convierte una serie de pulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de girar una cantidad de grados dependiendo de sus entradas de control. Existen tres tipos de motores a pasos: de reluctancia variable, de imanes permanentes e híbridos. En el motor a pasos de reluctancia variable el rotor está hecho de acero de bajo carbono y tiene cuatro polos, mientras que el estator tiene seis. Cuando una de las fases es excitada debido a una corriente de CD (Corriente Directa) que pasa a través de las bobinas alrededor de los polos, las posiciones del rotor completan la trayectoria del flujo. Cuando los dientes del rotor y el estator están alineados de esta manera, la reluctancia es minimizada y el rotor está en reposo. El motor a pasos de imanes permanentes tiene un comportamiento similar al de reluctancia variable; tiene dos bobinas, cada

una de las cuales produce cuatro polos, pero que están desplazadas una de la otra por medio paso polar. El rotor es de construcción de imanes permanentes y tiene cuatro polos, cada polo está enrollado con devanado de campo y las bobinas que hay en los pares opuestos de polos están en serie. La corriente es suministrada desde una fuente de CD hasta el devanado a través de interruptores. Los motores a pasos híbridos combinan las características de los motores de reluctancia variable y de imanes permanentes, teniendo un imán permanente interconectado en cubiertas de hierro que están cortadas para tener dientes. El rotor se coloca en la posición de reluctancia mínima en respuesta a que se estén energizando un par de bobinas en el estator. Cualquiera que sea el tipo de motor a pasos, la velocidad con la que se aplican los impulsos determina la velocidad del motor, el número total de impulsos determina el desplazamiento angular y el orden en el que se energizan las bobinas en la primera instancia determina la dirección de rotación. [18]

## 2.5 Redes y comunicación modelo Cliente-Servidor

Una red es un medio de comunicación que permite a personas o grupos compartir información y servicios. La tecnología de las redes informáticas está compuesta por el conjunto de herramientas que permiten a los dispositivos compartir información y recursos. Una red está constituida por equipos llamados nodos. Las redes se categorizan en función de su amplitud y de su ámbito de aplicación. Para comunicarse entre sí, los nodos utilizan protocolos, o lenguajes, comprensibles para todos ellos. [19]

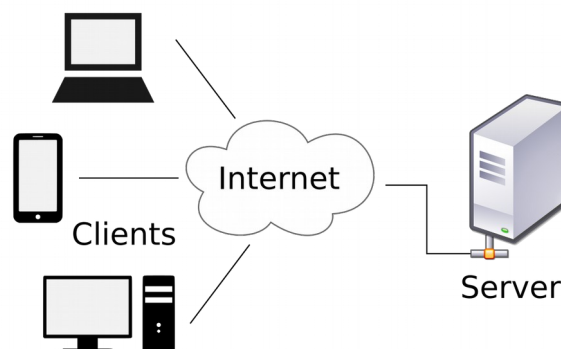
De acuerdo con [19], existen cuatro tipos principales de redes informáticas:

1. Red personal: centrada en el usuario, designa una interconexión de equipos informáticos en un espacio de una decena de metros en torno al usuario. También es conocida como red doméstica.
2. Red local: se extiende hasta algunos centenares de metros, también conocida como LAN (*Local Area Network*). Generalmente se utiliza para compartir recursos comunes, como periféricos, datos o aplicaciones.
3. Red metropolitana: garantiza la comunicación a distancias más extensas y a menudo interconecta varias redes LAN. Puede servir para interconectar, por conexión privada o pública, diferentes departamentos.
4. Red extendida: las redes con mayor alcance se clasifican como WAN (*Wide Area Network*). Están compuestas por redes de tipo LAN o incluso MAN (*Metropolitan Area Network*). Son capaces de transmitir la información a miles de kilómetros por todo el mundo. La WAN más famosa es la red pública Internet, cuyo nombre procede de Inter Networking, o interconexión de redes.

El modelo cliente-servidor representa la forma en la que se producen las comunicaciones entre dos nodos de red. En este modelo, uno de los nodos que forma parte de la comunicación tiene el rol de cliente y el otro tiene el rol de servidor. Un cliente es el elemento de la comunicación que pide o solicita un servicio de red, por ejemplo, el acceso a una página web, la descarga de un archivo o el envío de un correo electrónico. El servidor es el elemento de la comunicación que responde a las peticiones de los clientes, proporcionando el servicio requerido, es decir, enviando la página web, el archivo solicitado o el correo electrónico.

Este modelo cliente-servidor empezó a ser aceptado a finales de los 80's. Su funcionamiento es sencillo: se tiene un dispositivo cliente, que requiere un servicio de un dispositivo servidor, no tratándose necesariamente de dispositivos diferentes; es decir, un mismo dispositivo por sí solo puede ser ambos cliente y servidor dependiendo del software de configuración. [20]

El modelo cliente-servidor, véase la Figura 2, puede definirse como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. El cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este le envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada dispositivo puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el o los procesos cliente sólo se ocupan de la interacción con el usuario (aunque esto puede variar): es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento y desempeño al permitir distribuir físicamente los procesos y los datos en forma más eficiente. [20]



*Figura 2: Comunicación Cliente-Servidor, fuente: [20]*

El cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, desarrollados sobre plataformas que permiten construir GUI's (*Graphical User Interfaces*), además de acceder a los servicios distribuidos en cualquier parte de una red. [20]

De acuerdo con [20], las funciones que lleva a cabo el cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de datos.
- Recibir resultados del servidor
- Formatear resultados

Con base en [20], el servidor normalmente maneja todas las funciones relacionadas con la mayoría de los recursos de datos, siendo las funciones que lleva a cabo el servidor :

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc. Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como la velocidad del procesador, memoria, capacidad de almacenamiento, dispositivos de entrada-salida entre otros. [20]

La relación establecida puede ser de muchos a uno, en la que un servidor puede dar un servicio a muchos clientes, regulando su acceso a recursos compartidos. Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores, teniendo estos últimos un carácter pasivo ya que esperan las peticiones de los clientes. La plataforma de hardware y el sistema operativo del cliente no siempre son la misma, por lo que es posible conectar clientes y servidores independientemente de sus plataformas. [20]

El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema cliente-servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor agregar múltiples servidores. Es importante que los clientes y los servidores utilicen el mismo mecanismo, por ejemplo: sockets o por RPC (*Remote Procedure Call*), entre otros, lo cual implica que se deben tener mecanismos generales que existan en diferentes plataformas. Además, hay que tener estrategias para el manejo de errores y para mantener la consistencia de los datos. [20]



## 2.6 Lenguajes de programación para desarrollo web

El World Wide Web (www) ha recorrido un gran camino desde los días en que era un modesto sistema de hipertexto usado por unos cuantos científicos para compartir información en Internet. En la actualidad el World Wide Web es un medio para intercambiar información de millones de personas, las cuales comparten texto, video, sonido y datos, y cada vez son más las personas que intentan hacer sus páginas Web interactivas. Las compañías intentan vender sus productos, los artistas producen nuevas formas de arte interactivo y los programadores producen programas de ayuda para el diseño, todo por medio de Internet, cuyo acceso se realiza por medio de aplicaciones de visualización web. [21]

En la actualidad existe una gran diversidad de lenguajes de programación web, algunos de ellos son los siguientes:

- JavaScript: en un lenguaje de programación independiente de la plataforma, diseñado para aplicaciones distribuidas en Internet. En la actualidad, JavaScript incrementa la funcionalidad de la visualización web al permitir a los autores y diseñadores Web producir contenidos que cambian en respuesta a acciones del usuario, es decir, datos dinámicos que hacen que la información sea más accesible y más fácil de organizar y dirigir. [21]
- HTML (Hyper Text Markup Language): es un lenguaje de marcado descriptivo que se escribe en forma de etiquetas para definir la estructura de una página web y su contenido como texto, imágenes, entre otros. Las etiquetas son fragmentos de texto rodeados por corchetes angulares que se utilizan para escribir código HTML, en HTML existen etiquetas de apertura y etiquetas de cierre. Cabe destacar que HTML no es un lenguaje de programación ya que no cuenta con funciones aritméticas, variables o estructuras de control propias de los lenguajes de programación, por lo que HTML genera únicamente páginas web estáticas, sin embargo HTML se puede usar en conjunto con diversos lenguajes de programación para la creación de páginas web dinámicas. Básicamente sirve para describir la estructura básica de una página y organizar la forma en que se mostrará su contenido, además de que HTML permite incluir enlaces hacia otras páginas o documentos. [22]
- CSS (*Cascading Style Sheets*): es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML, siendo CSS una de las mejores opciones para separar los contenidos de las páginas web y su presentación. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas: mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. Con CSS se puede definir el aspecto de cada elemento: su color, tamaño, tipo de letra de texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. [23]

## 2.7 Bases de datos

Las bases de datos permiten la utilización de datos electrónicos en forma estructurada, permitiendo facilitar la introducción de datos en un esquema predefinido (por ejemplo, en los campos de un formulario), además de permitir su tratamiento de manera óptima, clasificados desde su introducción. Los directorios informáticos residen en una base de datos, optimizada para la lectura. El lenguaje estándar para efectuar consultas de lectura y escritura es LDAP (*Lighthouse Directory Access Protocol*). [19]

El trabajo con datos en forma de tablas, que pueden ser dependientes unas de otras, es propio de los SGBDR (Sistemas de Gestión de Bases de Datos Relacionales) o RDMS (*Relational DataBase Management System*). La programación de consultas estándar se realiza en el lenguaje SQL. [19]

Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada o estructurada. Desde el punto de vista de la informática, la base de datos es un sistema formado por un conjunto de datos almacenados que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos. Una base de datos contiene entidades de información que están relacionadas vía organización y asociación. La arquitectura lógica de una base de datos se define mediante un esquema que representa las definiciones de las relaciones entre las entidades de información. La arquitectura física de una base de datos depende de la configuración del hardware residente. Sin embargo, tanto el esquema (descripción lógica) como la organización (descripción física) deben adecuarse para satisfacer los requerimientos funcionales y de comportamiento para el acceso al análisis y creación de informes. [24]

La utilización de base de datos como plataforma para el desarrollo de sistemas de Aplicación en las organizaciones se ha incrementado notablemente en los últimos años. Algunas de las ventajas que ofrece su utilización:

- Globalización de la información.
- Permite mantener la integridad en la información al únicamente almacenar la información correcta.
- Manejo independiente de información y los programas que la gestionan.

Existen distintos modos de organizar la información y representar las relaciones entre los datos de una base de datos. Los sistemas administradores de bases de datos convencionales usan uno de los tres modelos lógicos de base de datos para hacer seguimiento de las entidades, atributos y relaciones. Los tres modelos lógicos principalmente de bases de datos son el jerárquico, de redes y el relacional. De acuerdo con [24], cada modelo lógico tiene ciertas ventajas de procesamiento:

- **Modelo jerárquico:** Tiene una estructura arborescente; un registro se subdividen segmentos que se interconectan en relaciones “padre e hijo” que puede representar dos tipos de relaciones entre datos: relaciones de uno a uno y relaciones de uno a muchos.
- **Modelo de datos en red:** Es una variación del modelo de datos jerárquico. Describen relaciones de muchos a muchos. Las bases de datos pueden traducirse de jerárquicas a bases de datos en redes y viceversa con el objetivo de optimizar la velocidad y la conveniencia del procesamiento.
- **Modelo relacional:** Representa todos los datos como sencillas tablas de dos dimensiones llamadas relaciones. Las tablas son semejantes a los archivos planos, pero la información en más de un archivo puede ser fácilmente extraída y combinada.

Las bases de datos orientadas a objetos e hipermedia son capaces de almacenar tanto procesos como datos. Por este motivo las bases de datos orientadas al objeto deben poder almacenar información no convencional (como imágenes estáticas o en movimiento, sonidos, entre otros). Este tipo de bases de datos deriva directamente de la llamada programación orientada a objetos. Entre las ventajas de las bases de datos orientadas al objeto destaca la posibilidad de tratar los casos inusuales. [24]

### 2.8 Access Point

Los puntos de acceso, también llamados AP's o *access point*, son equipos de hardware configurados en redes Wi-Fi y que hacen de intermediario entre el equipo en cuestión (por ejemplo una computadora) y la red externa (local o Internet). El punto de acceso hace de transmisor central y receptor de las señales en una red Wireless componiéndose de un adaptador de red, una antena y un transmisor. [25]

Existen redes Wireless pequeñas que pueden funcionar sin puntos de acceso, llamadas redes *ad-hoc* o modo *peer-to-peer*, las cuales sólo utilizan tarjetas de red para comunicarse. Las redes más usuales son en modo estructurado, es decir, los puntos de acceso serán intermediarios o puente entre los equipos Wi-Fi y una red Ethernet cableada. También harán la función de escalar más usuarios según se necesite y pueden dotar de algunos elementos de seguridad. [25]

Los puntos de acceso normalmente van conectados físicamente por medio de un cable de pares a otro elemento de red, en caso de una oficina, o directamente a la línea telefónica o servicio de televisión por cable si es una conexión doméstica. En este último caso, el AP estará haciendo también el papel de Router. Son los llamados Wireless Routers, los cuales soportan los estándar 802.11a, 802.11b, y 802.11g. [25]

Los puntos de acceso inalámbricos pueden funcionar de tres maneras diferentes:

1. **Maestro (Root):** en este modo, usuarios con diferentes dispositivos electrónicos se pueden conectar a Internet a través de un solo Acces Point compartiendo la conexión.

2. Repetidores (Repeater): se utiliza cuando se quiere extender la señal más allá de los límites actuales de alcance. Necesita disponer del punto de acceso en modo repetidor dentro del área de un punto de acceso en modo Root. Con esto la señal de AP maestro se extenderá por medio de este AP repetidor mejorando el alcance.
3. Puente (Bridge): este tipo de conexión es útil cuando se conectan dos edificios o localizaciones separados donde instalar cableado no resulta fácil o económicamente viable. Se necesitan dos puntos de acceso y dos antenas direccionales. Dependiendo de las distancias podría ser necesario contar con algún software para comprobar la conectividad entre equipos.

De acuerdo con [26], algunas ventajas del punto de acceso son:

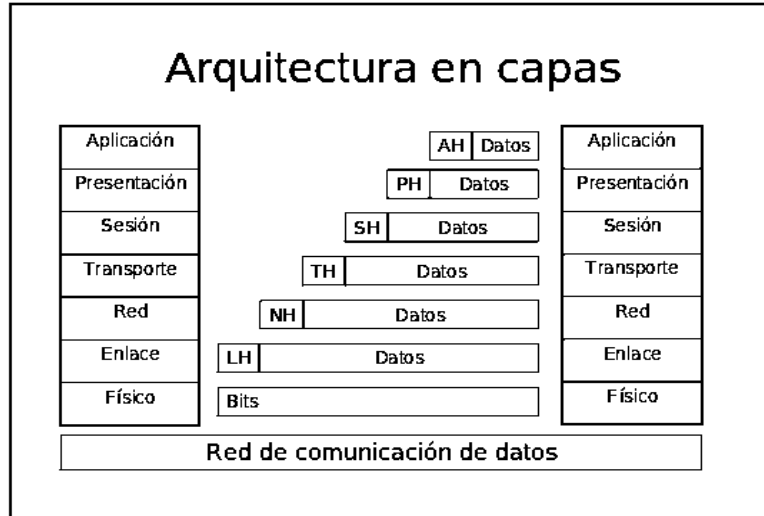
- El punto de acceso permite compartir el acceso a Internet entre todas sus terminales, esto permite compartir el acceso de banda ancha entre todas las terminales que conforman la red.
- El punto de acceso permite crear redes con mayor número de terminales.
- Al igual que cualquier red local, permite compartir recursos de las terminales que formen la red (archivos, impresoras, etc.).

### 2.10 Protocolo de Seguridad SSL

SSL es un protocolo de seguridad estándar de encriptación entre dos puntos de red, generalmente un cliente y un servidor, y asegura que toda la información transmitida permanezca encriptada. [27]

Considerando un modelo de arquitectura de redes por capas, véase Figura 3, el protocolo SSL se utiliza entre la capa de aplicación y la capa de transporte. Uno de sus usos más extendidos se realiza junto al protocolo HTTP, dando lugar al protocolo HTTPS (*Hypertext Transfer Protocol Secure*). De esta forma se consigue que la información transmitida entre un sitio web y un usuario sea segura, especialmente importante cuando se trata de información sensible: datos confidenciales, contraseñas, información bancaria, etc. [28]

Se utiliza tanto la criptografía asimétrica (utiliza dos claves: la pública, que puede ser difundida, y la privada, que no debe ser revelada) como la criptografía simétrica (solo se utiliza una clave para cifrar y descifrar el mensaje la cual tienen que conocer el emisor y el receptor). La primera se utiliza para realizar el intercambio de claves, que a su vez serán usadas para cifrar la comunicación mediante un algoritmo simétrico. [28]



*Figura 3: Arquitectura de redes por capas, modelo OSI, fuente: [28]*

En base en [29], la criptografía consta de tres elementos esenciales:

1. Protocolo: criptografía que proporciona privacidad e integridad en la comunicación entre dos puntos.
2. Certificado: opera como una credencial electrónica.
3. Autoridad de certificación: emite los certificados SSL.

El protocolo garantiza que la información transmitida por la red no pueda ser interceptada ni modificada por elementos no autorizados, garantizando que sólo los emisores y los receptores legítimos tengan acceso a la comunicación de manera íntegra. [28]

El certificado es necesario para crear la conexión SSL, se requiere proporcionar información acerca de la identidad del sitio web o servidor, compañía, etc., con la información son creadas dos llaves; una llave pública y otra privada. Cuando se inicia una conexión utilizando el protocolo SSL primeramente se evalúa si dicho certificado es válido o no. [27]

De acuerdo con [27] para realizar una conexión segura mediante el protocolo SSL el flujo de información, mostrado en la Figura 4, se describe mediante el siguiente procedimiento:

- Un cliente realiza la petición de establecer una conexión segura.
- El navegador obtiene la IP del cliente por medio de un servidor DNS (*Domain Name System*) y realiza una petición de conexión segura al servidor.

- Para iniciar la conexión segura, el navegador solicita al servidor que se identifique enviando una copia de su certificado SSL al navegador.
- El navegador revisa que el certificado esté firmado por una autoridad de certificación confiable que sea válida, que confirme las características de la llave requerida y que el dominio que especifica el certificado concuerde con el dominio que solicitó el cliente.
- Cuando el navegador confirma que el servidor es confiable, crea una llave de sesión asimétrica con la llave pública que está en el certificado del servidor. Posteriormente la llave es enviada al servidor.
- El servidor utiliza la llave privada para descifrar la llave de sesión simétrica.
- El servidor envía de regreso la notificación (“*acknowledgment*”) que está encriptada con la llave de sesión.
- Se establece la conexión segura, siendo encriptada toda la información intercambiada entre el cliente y el servidor.

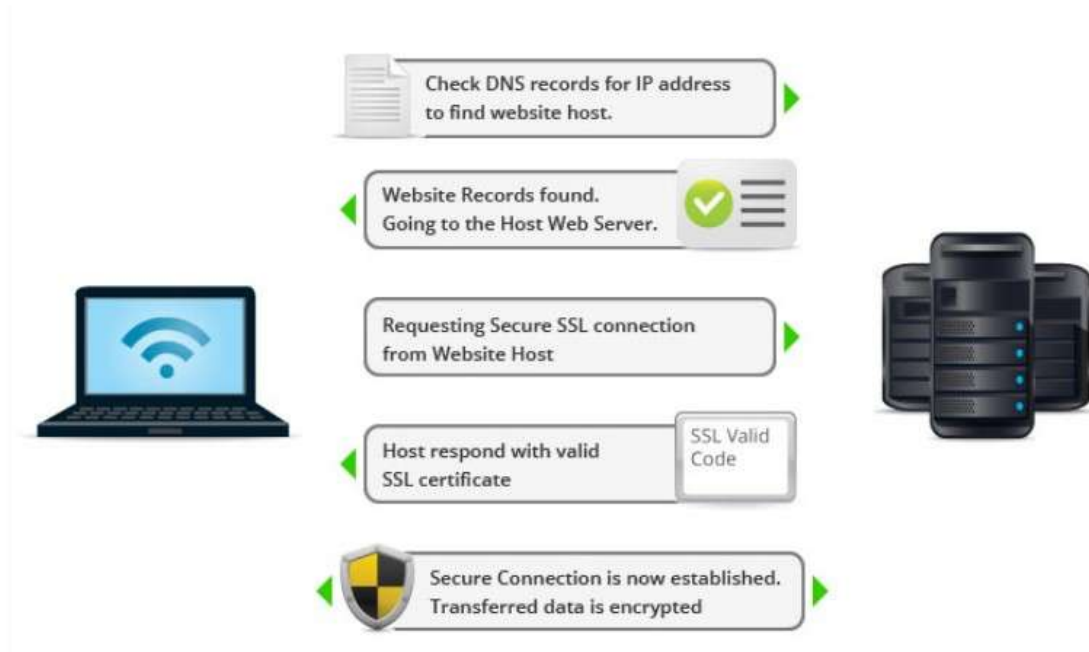


Figura 4: Establecimiento de conexión segura con SSL, fuente: [27]

## 2.11 Protocolo de comunicación MQTT

Para dispositivos IoT es necesaria la conexión a Internet. La conexión a Internet permite que los dispositivos funcionen entre ellos y realicen diferentes acciones, por ejemplo la comunicación con el servidor. El protocolo de red subyacente es TCP (*Transmission Control Protocol*) /IP. MQTT, está construido sobre la pila de TCP/IP, se ha convertido en el estándar para las comunicaciones de IoT. Originalmente, MQTT fue inventado y desarrollado por IBM (*International Business Machines*) a finales de los 90. Su aplicación original era conectar sensores de los oleoductos con satélites. Tal como sugiere su nombre, es un protocolo de mensajería que soporta la comunicación asíncrona entre las partes. Un protocolo de mensajería asíncrona disocia el emisor y al receptor de los mensajes tanto en espacio como en tiempo y, por lo tanto, es escalable en entornos de red no confiables. Utiliza un modelo de publicación y suscripción. [30]

MQTT es soportado en lenguajes de programación populares mediante la utilización de varias implementaciones de código abierto. De acuerdo con [30], MQTT es un protocolo de red liviano y flexible que logra el equilibrio adecuado para los desarrolladores de IoT:

- El protocolo liviano le permite implementar en hardware de dispositivos altamente limitados y en redes con ancho de banda de alta latencia/limitado.
- Su flexibilidad hace que pueda soportar varios escenarios de aplicaciones para dispositivos y servicios de IoT.

El protocolo MQTT define los tipos de entidades en la red: un intermediario de mensajes y un número de clientes. El intermediario es un servidor que recibe todos los mensajes de los clientes y luego los redirige a clientes de destinos relevantes. Un cliente es cualquier cosa que pueda interactuar con el intermediario para enviar y recibir mensajes. Un cliente puede ser un sensor de IoT en el campo o una aplicación del centro de datos que procesa datos de IoT. [30]

El funcionamiento del protocolo MQTT es:

1. El cliente se conecta con el intermediario. Se puede suscribir a cualquier “tema” de mensajes de intermediario. Esta conexión puede ser una conexión TCP/IP simple o una conexión TLS (*Transport Layer Security*) cifrada para mensajes confidenciales.
2. El cliente publica el mensaje, sobre un tema, enviando el mensaje y el tema al intermediario.
3. Después, el intermediario redirige el mensaje a todos los clientes que están suscritos a ese tema.

Ya que los mensajes de MQTT están organizados por temas, el desarrollador de aplicaciones tiene la flexibilidad de especificar que ciertos clientes sólo puedan interactuar con determinados mensajes. Por ejemplo, los sensores publicarán sus lecturas sobre el tema “sensor\_data” y se suscribirán al tema “config\_change”, véase la Figura 5, posteriormente las aplicaciones de procesamiento de información

que guardan las lecturas del sensor en una base de datos se suscribirán al tema “*sensor\_data*”. Una aplicación de consola de administración puede recibir los comandos del administrador del sistema para ajustar las configuraciones de los sensores, como la sensibilidad y la frecuencia de muestreo, y publicar estos cambios en el tema “*config\_change*”. [30]

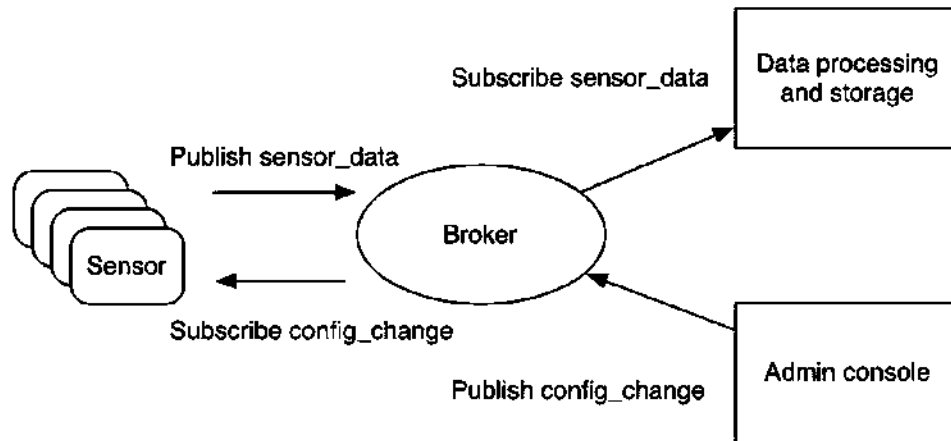


Figura 5: Protocolo MQTT, fuente: [30]

Al mismo tiempo, MQTT es liviano. Tiene una cabecera simple para especificar el tipo de mensaje, un tema basado en texto, y, a continuación, una carga útil binaria y arbitraria. La aplicación puede utilizar cualquier formato de datos para la carga útil, como JSON (*JavaScript Object Notation*), XML (*Extensible Markup Language*), cifrado binario o Base64, siempre que los clientes destino puedan analizar la carga útil. [30]

## 2.12 Sistemas actuales de Domótica

Algunos de los principales desarrolladores de sistemas de Domótica a nivel mundial son:

- Google Home, mostrado en la Figura 6: es un dispositivo controlado por voz con bocinas integradas que puede realizar diferentes acciones como: controlar la reproducción de contenido multimedia, realizar búsquedas de información en Internet, acceder a canales de radio, reproducir audio libros o reproducir música usando Bluetooth. Puede almacenar rutinas del día a día que incluyan el clima, recordatorios, noticias, reuniones, tiempos de desplazamiento y tráfico terrestre o aéreo. Tiene la capacidad de expresar resultados numéricos o financieros, traducciones, búsquedas de deportes, nutrición, diccionario. En cuanto a manipulación de otros dispositivos electrónicos: puede reproducir shows de TV o YouTube, visualizar fotos en la televisión, encender o apagar la televisión. A pesar de no ser código abierto, tiene una



compatibilidad domótica con más de 1,000 dispositivos inteligentes provenientes de más de 150 fabricantes, por lo cual es fácil el control de una casa habitación utilizando solamente la voz. Tiene la capacidad para almacenar acciones IFTTT (“Si esto, entonces eso”). Actualmente solo está disponible en Canadá, Estados Unidos, Australia, Francia, Alemania, Japón y México. Costo: US\$399. [31]



*Figura 6: Google Home, fuente: [31]*

- Amazon Alexa, véase la Figura 7: es un dispositivo electrónico controlado por voz de código abierto que incorpora las tecnologías de inteligencia artificial y *machine learning*. Tiene compatibilidad con dispositivos inteligentes de terceros. Es de código abierto, se pueden desarrollar experiencias de manejo por voz que sean más naturales para el usuario, propiciando una forma más intuitiva de interactuar con la tecnología que utilizan en su vida diaria. Se pueden desarrollar nuevas capacidades para Alexa, conectando a un mayor número de dispositivos o integrando directamente a un producto en especial. Cuenta con API's (Application Programming Interface) a disposición de desarrolladores e incentiva a la comunidad a crear más contenido para Alexa. Con Alexa se puede reproducir música, estaciones de radio o audio libros, buscar noticias, hacer peticiones a sitios de Internet o realizar búsqueda de información, realizar llamadas telefónicas, controlar dispositivos electrónicos inteligentes además de incorporar acciones IFTTT, esto para crear rutinas automatizadas que enlacen servicios, aplicaciones o gadgets. Actualmente está en expansión a más de 28 países, incluyendo América del Sur y México. Costo: US\$84.99. [32]



*Figura 7: Amazon Alexa, fuente: [32]*

- Jarvis IA, de Mark Zuckerberg: es un dispositivo con inteligencia artificial capaz de controlar tareas básicas como iluminación, temperatura, reproducción de contenido multimedia, apertura y cierre de puertas entre otras. Utiliza lenguaje natural, tiene reconocimiento de pronunciación y entiende la voz del usuario y el contexto del comando. Incorpora detección de rostros. Puede almacenar actividades a realizar y recordatorios de las mismas, enseñar idiomas, informar lo que otras personas están haciendo en la casa, realizar video conferencias. Incorpora la tecnología machine learning para adquirir información cada vez más precisa acerca de los hábitos del usuario. Actualmente se encuentra en desarrollo, no está a la venta. [33]
- MyCroft Mark II, mostrado en la Figura 8: es un sistema de código abierto, actualmente en desarrollo, cuya principal diferencia es la protección de la privacidad. Es un dispositivo controlador por voz, realizado con Arduino y Raspberry Pi. Su funcionamiento depende de los servidores de Google, Amazon o Microsoft, pero a diferencia de Amazon Alexa y Google Home, no recolecta los datos del usuario. Tiene una pantalla de cuatro pulgadas para mostrar información relevante. El precio esperado es de US\$129. Se encuentran disponibles características especiales de personalización y modificación orientadas a desarrolladores. MyCroft Mark II puede almacenar listas de compra, reproducir música, integra compatibilidad con YouTube, Gmail, WolframAlpha, Twitter, Wikipedia, entre otros. Dado a que está en desarrollo, cuenta con más de 140 funcionalidades, aumentando día a día gracias al aporte de desarrolladores y comunidad. [34]



*Figura 8: MyCroft Mark II, fuente: [34]*

- Apple HomeKit, véase la Figura 9: consta de una aplicación para dispositivos móviles iOS y el asistente virtual Siri. Mediante comandos de voz se pueden controlar de forma remota diversos dispositivos IoT (que sean compatibles con Apple HomeKit) como lámparas, interruptores, enchufes, termostatos, ventanas ventiladores, aires acondicionados, humidificadores, purificadores de aire, diversos sensores, cerraduras, cámaras, otros dispositivos de seguridad, timbres, etc. La aplicación agrupa los accesorios por habitación para que sea más fácil controlarlos desde cualquier parte de la casa. Es posible crear ambientes que activen varios accesorios a través de un solo comando. También permite que se realicen ciertas tareas de forma automática, como controlar un accesorio o un ambiente según la hora del día, la ubicación del usuario o lo que detecten los sensores. Es un sistema cerrado. [35]



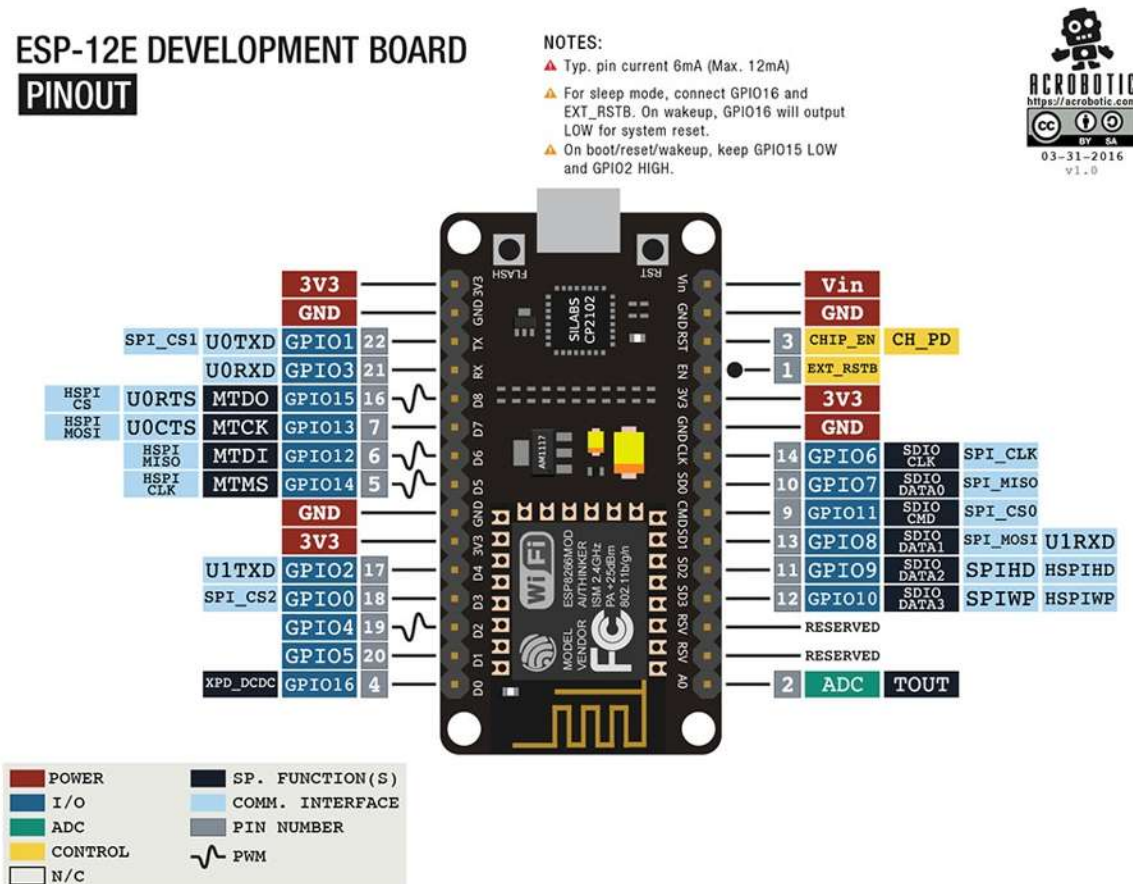
*Figura 9: Apple HomeKit, fuente: [35]*

## CAPÍTULO 3: HARDWARE UTILIZADO

En este capítulo se describen los componentes de hardware utilizados para el desarrollo del sistema, mencionando información relevante como: principales características eléctricas, compatibilidad con diferentes entornos de desarrollo, manejo básico del hardware mencionado, por nombrar algunos ejemplos, para así tener una mejor comprensión del CAPÍTULO 5: DESCRIPCIÓN DEL SISTEMA.

### 3.1 ESP-12E Development Board

El ESP-12E, mostrado en la Figura 10, es una tarjeta de desarrollo de Espressif Systems. Cuenta con integración inalámbrica de alto nivel SOC (*System On Chip*) capaz de interactuar con otros sistemas Wi-Fi o funcionar de manera independiente, con el mínimo costo y requerimiento de espacio.



Este dispositivo puede ser utilizado como host para una aplicación o funcionar en una red sin Wi-Fi. Incorpora memoria externa de 4MB con almacenamiento caché para mejorar el desempeño del sistema en ciertas aplicaciones. Adicionalmente, puede servir como un adaptador Wi-Fi que puede ser incorporado a cualquier microcontrolador mediante conectividad SPI (*Serial Peripheral Interface*), SDIO (*Secure Digital Input Output*), I2C (*Inter-integrated Circuit*) o UART (*Universal Asynchronous Receiver-Transmitter*). Integra una antena Wi-Fi, amplificador de recepción de bajo ruido, reguladores y amplificadores de voltaje y requiere mínima conexión de circuitería. Además integra una versión del procesador Tensilica serie L106 de 32 bits con 64KB de memoria SRAM. En la Figura 11 se ejemplifica con un diagrama de bloques las prestaciones de hardware y de comunicación inalámbrica del SOC ESP12-E. Incluye funciones de bajo consumo eléctrico. [36]

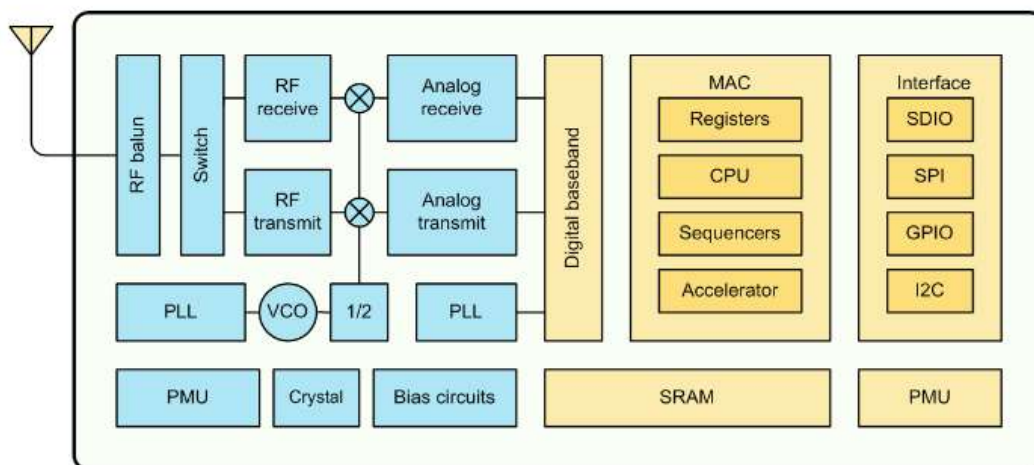


Figura 11: ESP8266, Diagrama de Bloques, fuente: [36]

De acuerdo con [36], algunas de las características del módulo ESP-12E son:

- MCU integrado de 32 bits de bajo consumo.
- Protocolo 802.11 b/g/n.
- ADC con resolución de 10 bits integrado.
- Wi-Fi 2.4GHz, con soporte de WPA/WPA2.
- Periféricos SDIO 2.0, SPI, UART, I2C, IR, PWM (*Pulse Width Modulation*), GPIO (*General Purpose Input Output*).
- Modo de operación de bajo consumo menor a 10uA.
- Consumo energético menor a 1.0mW en modo de espera.
- Rango de temperatura de operación entre 40 y 125°C

- Certificados FCC, CE, TELEC, Wi-Fi y SRRC.
- Chip SPI de memoria Flash programable de 8MB o 4MB, dependiendo del modelo.
- Memoria RAM de 36KB.
- Soporta osciladores de 40MHz, 26MHz y 24MHz.
- Frecuencia de procesador: 80MHz (160MHz máximo).
- Como se describe en [36], el módulo tiene cinco modos de operación:
  1. OFF: todos los registros se borran, RTC (*Real Time Clock*) deshabilitado, pin CHIP\_PD (pin de habilitación del módulo) en bajo.
  2. DEEP\_SLEEP: únicamente RTC habilitado, el resto está apagado. La recuperación de memoria del RTC puede mantener funciones básicas de conectividad Wi-Fi.
  3. WAKEUP: en este estado, el sistema cambia de SLEEP a PWR. El oscilador y el módulo PLL están habilitados.
  4. SLEEP: solamente está operando el módulo RTC. El oscilador está inactivo. Cualquier evento (MAC, host, timer RTC, interrupciones externas) pondrán al sistema en estado WAKEUP.
  5. ON: el reloj de alta velocidad está operando, el CPU está habilitado.

El módulo ESP8266 fue lanzado al mercado a mediados del año 2014 en su versión ESP-01. Poco a poco la comunidad de desarrolladores tradujeron la documentación originalmente en chino y fueron creando firmwares (software de bajo nivel) para el ESP8266.

Las características de conexión inalámbrica del módulo lo hacen ideal para aplicaciones de automatización y control vía remota o integración de diversos componentes en sistemas IoT y cualquier aplicación donde se requiera conexión a una red o a Internet.

El módulo NodeMCU utiliza un firmware basado en lenguaje Lua. El firmware está basado en el Espressif NON-OS SDK (*Software Development Kit*) y utiliza un sistema de archivos basado en SPIFFS (SPI Flash File Sistem). El repositorio consta en un 98% de lenguaje C. El firmware NodeMCU es un sistema de código abierto para las tarjetas de desarrollo ESP8266-12E. La programación del módulo NodeMCU es similar a Node.js pero en Lua. Es manejado por eventos asíncronos.

Para crear un firmware que se adapte a las necesidades del sistema existen fundamentalmente tres maneras de hacerlo:

### CAPÍTULO 3: HARDWARE UTILIZADO

- Cloud Build Service: es un servicio en línea para desarrolladores, se encarga de generar firmware dependiendo de los módulos (ADC, bit, GPIO, HTTP, etc.) que requiera el usuario. Consta de una interfaz de usuario y es compatible con cualquier sistema operativo, puesto que funciona por medio del navegador web.
- Docker Image: herramienta orientada a desarrolladores que no necesitan un control total de las opciones de configuración del firmware.
- Linux Build Enviroment: herramienta que dispone de control total de configuración de firmware, incluye la opción de aportar sistemas y avances por medio de GitHub.

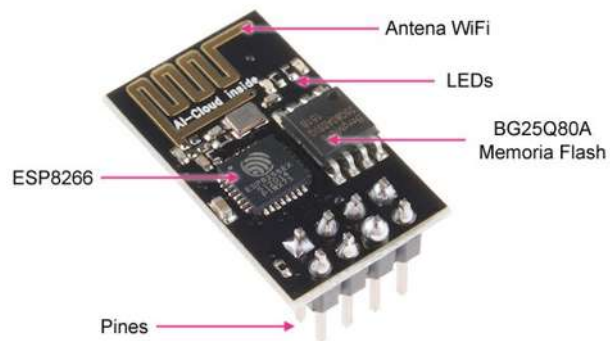
Para poder cargar el firmware al módulo NodeMCU es necesario disponer de una de las siguientes herramientas de software:

- esptool.py: plataforma independiente basada en Python de código abierto que es utilizado para comunicarse con la memoria ROM donde se ubica el bootloader en el ESP8266. Es compatible con OS X, Linux, Windows.
- NodeMCU PyFlasher: software basado en esptool.py y wxPython que incorpora una interfaz de usuario.
- NodeMCU Flasher: software solamente compatible con Windows que utiliza el framework QT.

Una vez que el módulo NodeMCU tiene el firmware correspondiente, es necesario subir código de la computadora al módulo. Esto se puede realizar mediante diferentes opciones e interfaces de programación: Arduino IDE (*Integrated Development Environment*), comandos AT (abreviatura de “*attention*”), incorporarlo a otro microcontrolador, etc., siendo ESPlorer el entorno de programación usado para el desarrollo de este trabajo, el cuenta con una interfaz gráfica de programación y desarrollo que soporta los comandos AT, muestra los archivos presentes en la memoria flash del NodeMCU, ejecuta bloques de código o archivos independientes, dispone de una línea de comandos para la interacción con la tarjeta de desarrollo NodeMCU, además de ser compatible con cualquier sistema operativo que pueda incorporar Java (edición estándar, SE 7 o superior).

Además del módulo ESP-12E, el fabricante AI-Thinker proporciona diferentes placas que incorporan algún módulo ESP [37]:

- ESP-01, mostrado en la Figura 12: tiene disponibles dos pines GPIO digitales, pines seriales Rx(GPIO3) y Tx(GPIO1), pin de habilitación CH\_PD, pin de reset y pines de alimentación Vcc y GND. El ESP-01 soporta la comunicación I2C. Es importante señalar que todos los pines funcionan a un voltaje de 3.3V. Para alimentar la placa se recomienda una fuente de alimentación de 3.3V, máximo 3.6V y una corriente mínima de 200mA.



*Figura 12: ESP-01, fuente: [37]*

- ESP-05, mostrado en la Figura 13: es el módulo más simple. No dispone de ningún puerto GPIO accesible.



*Figura 13: ESP-05, fuente: [37]*

- ESP-12, mostrado en la Figura 14: tiene acceso a 11 pines GPIO, de los cuales uno puede ser utilizado como convertidor analógico a digital con una resolución de 10 bits. Sin embargo la conexión con una tarjeta prototipo no es sencillo; es necesario soldar los pines o utilizar un adaptador.



*Figura 14: ESP-12, fuente: [37]*



- ESP-201, mostrado en la Figura 15: se puede acceder a 11 pines GPIO, además de ser usado fácilmente en una tarjeta de prototipos. Permite la conexión de una antena externa para mayor alcance.



Figura 15: ESP-201, fuente: [37]

### 3.2 NodeMCU Cloud Build Service

Para la programación del módulo NodeMCU es necesario el firmware específico para la aplicación a realizar. El firmware es el “sistema operativo” del módulo (por esto que el tema es incorporado en el apartado de hardware), el firmware se programa en el módulo para posteriormente utilizar lenguajes de alto nivel para los diferentes módulos incorporados al firmware.

Dentro de las opciones para la generación de firmware, descritas en 3.1 ESP-12E Development Board , se utilizó la opción *Cloud Build Service*, la interfaz de usuario en una página web que ofrece al usuario un conjunto de opciones para configurar los módulos a utilizar, por ejemplo: convertidor AD, pines GPIO, pines con operación PWM, pines de comunicación UART, etc. Una vez seleccionados los módulos a utilizar, el firmware es enviado por correo electrónico al usuario en formato *.bit* para ser programado con la herramienta *esptool.py*.

Además de las opciones de configuración de software, la página <http://nodemcu-build.com/> ofrece información mediante gráficas acerca de temas como: construcciones de firmware por semana, módulos más utilizados, regiones en el mundo donde es más utilizado *Build Cloud Service*, entre otros.

### 3.3 Esptool.py

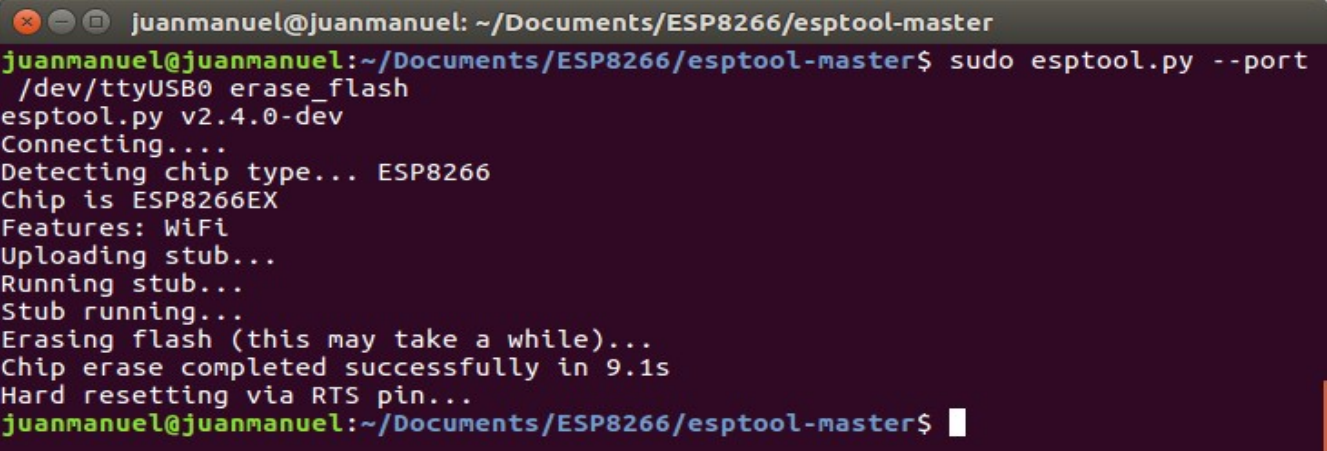
*Esptool.py* es una herramienta realizada en Python que facilita leer el contenido y características de la memoria flash del módulo, borrar o grabar firmware en el módulo NodeMCU desde la línea de comandos. Dentro de los argumentos de esta herramienta se encuentran: puerto donde está conectado el NodeMCU, velocidad en baudios (bits por segundo) a utilizar, dirección de memoria a modificar, entre otros.

En este sistema se utilizó la herramienta esptool.py para borrar la memoria flash del módulo, véase la Figura 16, mediante el comando:

```
sudo esptool.py --port /dev/ttyUSB0 erase_flash
```

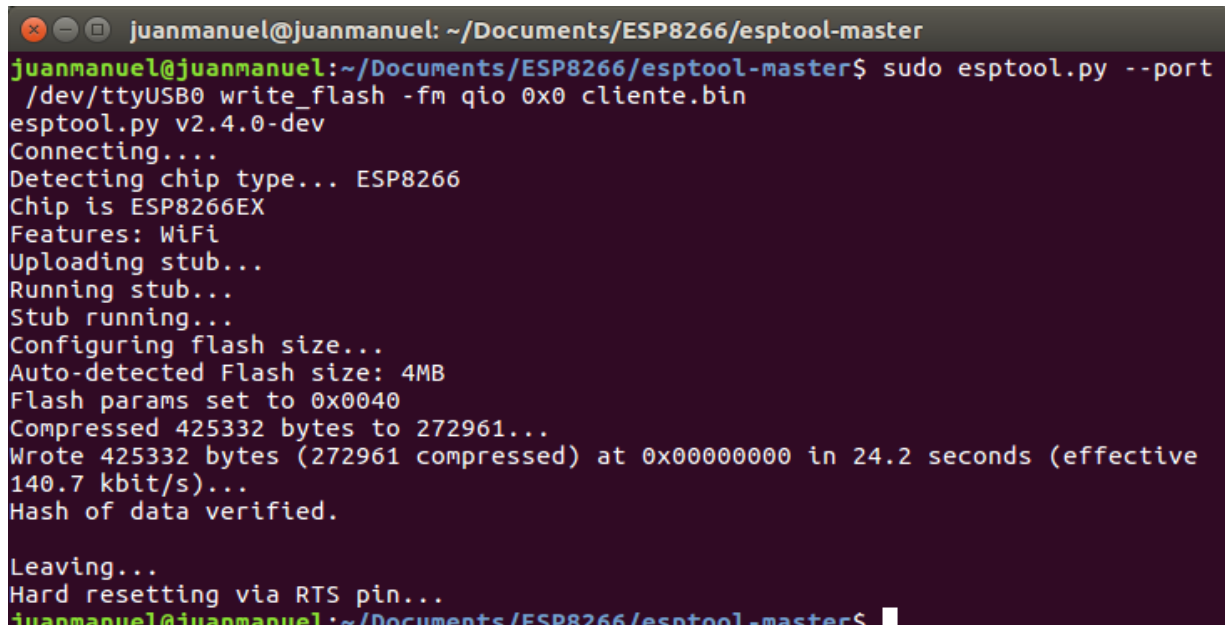
Posteriormente se programa un firmware, véase la Figura 17, diseñado mediante Cloud Build Service para desempeñar funciones de cliente, dentro del modelo de comunicaciones cliente-servidor, con el comando:

```
sudo esptool.py --port /dev/ttyUSB0 write_flash -fm qio 0x0 cliente.bin
```

A screenshot of a terminal window with a dark background. The window title is 'juanmanuel@juanmanuel: ~/Documents/ESP8266/esptool-master'. The prompt is 'juanmanuel@juanmanuel:~/Documents/ESP8266/esptool-master\$'. The command entered is 'sudo esptool.py --port /dev/ttyUSB0 erase\_flash'. The output shows the tool connecting to the device, detecting it as an ESP8266EX, and successfully erasing the flash memory in 9.1s. The final prompt is 'juanmanuel@juanmanuel:~/Documents/ESP8266/esptool-master\$' with a cursor.

```
juanmanuel@juanmanuel: ~/Documents/ESP8266/esptool-master
juanmanuel@juanmanuel:~/Documents/ESP8266/esptool-master$ sudo esptool.py --port
/dev/ttyUSB0 erase_flash
esptool.py v2.4.0-dev
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 9.1s
Hard resetting via RTS pin...
juanmanuel@juanmanuel:~/Documents/ESP8266/esptool-master$
```

*Figura 16: Borrado de la Memoria Flash del NodeMCU*



```

juanmanuel@juanmanuel: ~/Documents/ESP8266/esptool-master
juanmanuel@juanmanuel:~/Documents/ESP8266/esptool-master$ sudo esptool.py --port
/dev/ttyUSB0 write_flash -fm qio 0x0 cliente.bin
esptool.py v2.4.0-dev
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0040
Compressed 425332 bytes to 272961...
Wrote 425332 bytes (272961 compressed) at 0x00000000 in 24.2 seconds (effective
140.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
juanmanuel@juanmanuel:~/Documents/ESP8266/esptool-master$

```

Figura 17: Programación del Firmware mediante esptool.py

### 3.4 Conexión para la programación del módulo ESP-01

La programación del módulo ESP-01 consta de dos modos de operación:

Tabla 1: Módulo ESP-01 Modos de Operación

| Modo de Operación | GPIO0 | CH_PD | RESET |
|-------------------|-------|-------|-------|
| Cargar programa   | Low   | High  | High  |
| Ejecutar programa | High  | High  | High  |

Para la programación de los módulos ESP-01 se realizaron las conexiones mostradas en la Figura 18. Para reiniciar el módulo se presiona el pulsador indicado como RESET. Para cargar un programa al módulo ESP-01 es necesario presionar el pulsador LOAD/EXECUTE, en caso contrario el módulo ejecutará el último programa que le haya sido programado en la memoria FLASH.

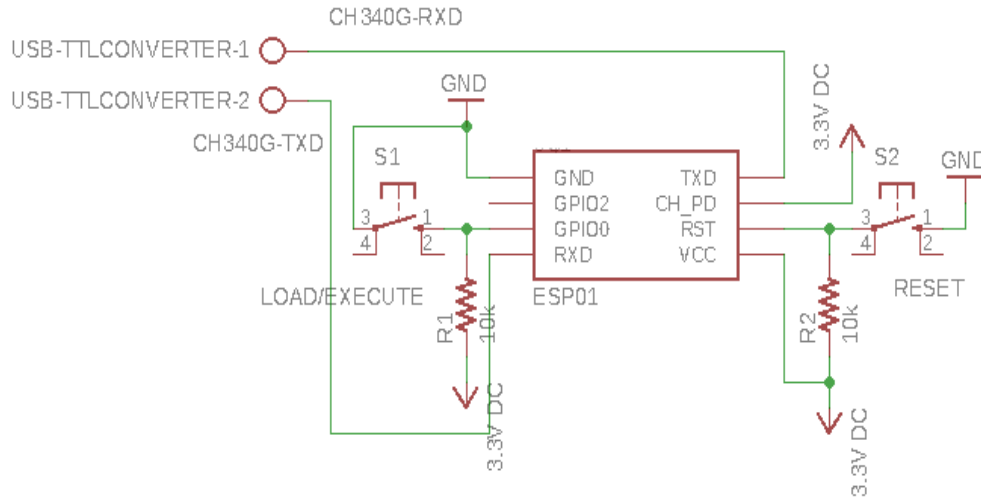


Figura 18: Conexión necesaria para la programación del módulo ESP-01

En caso que los pines de recepción y transmisión del convertidor de USB (*Universal Serial Bus*) a serial operen a 5 volts es necesario agregar divisores de voltaje para evitar dañar el módulo ESP-01, ya que opera a un voltaje de 3.3 voltios.

El módulo ESP-01 consume una corriente de pico mayor a 200mA, superior a la que el convertidor de USB a serial puede proporcionar. Es necesario contar con una fuente de alimentación externa de 3.3 voltios de al menos 200mA y un capacitor para soportar los transitorios.

### 3.5 Circuito para control ON/OFF utilizando el módulo ESP-01

El circuito implementado para realizar el control on/off consta de un módulo ESP-01, mostrado en la Figura 19, utiliza un pin digital para accionar un relevador para encender o apagar un dispositivo. La bobina del relevador opera a 5 voltios y consume una corriente de 120mA, dado que el módulo ESP-01 no proporciona los niveles de voltaje y corriente requeridos se utilizó un transistor TIP41C. La resistencia para la base del transistor es calculada con la ecuación (1).

$$\frac{V_{BE}-0.7}{\frac{I_{CE}}{h_{FE}}} = \frac{3.3-0.7}{\frac{0.072}{15}} = 541.6666 \quad (1)$$

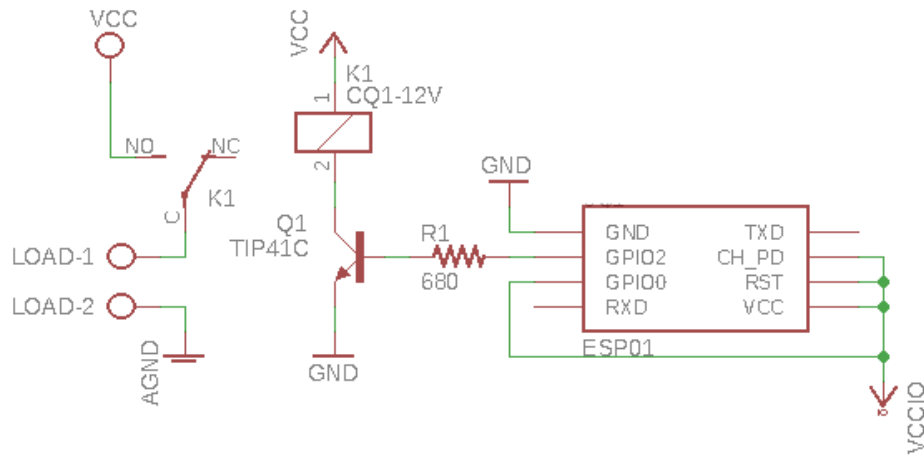


Figura 19: Circuito para el control On-Off utilizando un módulo ESP-01

### 3.6 Lectura de temperatura y humedad utilizando el sensor DHT11

El circuito utilizado para la lectura de temperatura y humedad, véase la Figura 20, se realiza empleando el sensor DHT11 y el módulo ESP-01.

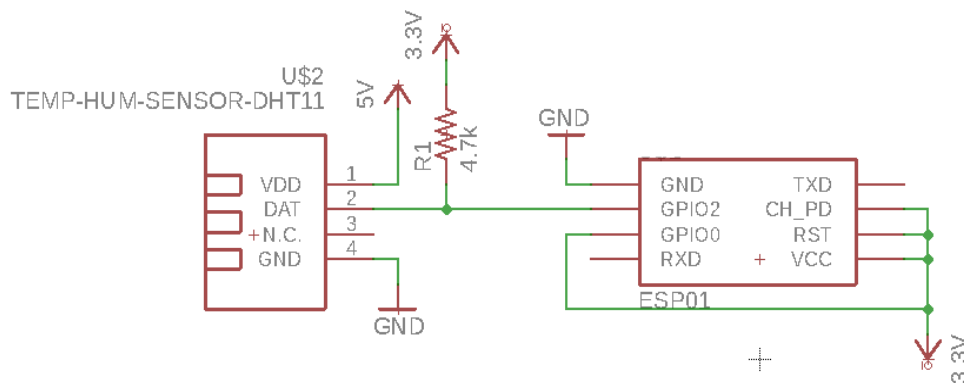


Figura 20: Circuito para la medición de temperatura y humedad

De acuerdo con [38], las principales características del sensor DHT11 son:

- Bajo costo.
- Gran estabilidad.

- Medición de Temperatura y Humedad relativa.
- Excelente calidad.
- Respuesta rápida.
- Protección anti-interferencia.
- Transmisión de señal a larga distancia.
- Señal de salida digital.
- Calibración precisa.
- Voltaje de alimentación de 3.5V a 5.5V.

La comunicación con el sensor utiliza el protocolo *single-wire* (utiliza una sola línea de datos) bidireccional para el intercambio de información. La línea de datos generalmente requiere una resistencia *pull-up* de 5.1k $\Omega$  porque la salida es de tercer estado (alta impedancia); cuando la línea de datos está inactivo entonces el estado es un 1 lógico, o  $V_{CC}$ . [38]

Con base en [38], la secuencia para la lectura del sensor:

1. Después de energizar esperar un segundo antes de enviar instrucciones al sensor, esto permite estabilizar y realizar la medición de Humedad Relativa y Temperatura del ambiente. En este estado la línea de datos está en alto mediante la resistencia de *pull-up* y una vez finalizada la medición el pin es puesto como entrada, esperando la recepción de una señal externa.
2. El microprocesador envía un 0 lógico, o *GND*, con una duración mínima de 18ms mediante la línea de datos y posteriormente el pin del microprocesador es configurado como entrada. Una vez que el sensor detecte el 0 enviado por el microcontrolador, la línea de datos es puesta en 1 mediante la resistencia de *pull-up* hasta que el sensor envíe las mediciones de temperatura, humedad y 8 bits de paridad (utilizado para corroborar las mediciones del sensor), véase la Figura 21.

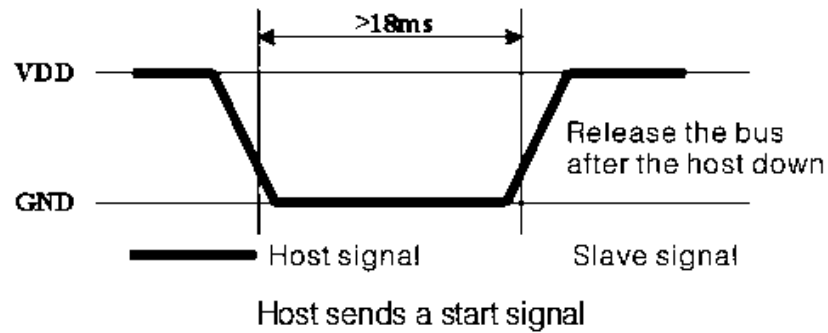


Figura 21: Señal de inicio para la lectura del sensor DHT11 [38]

- Después de detectar el 0 lógico, la línea de datos se mantiene en bajo y posteriormente pasa a nivel alto aproximadamente por 80 $\mu$ s. Posteriormente el sensor comienza a enviar la información de las mediciones, véase la Figura 22.

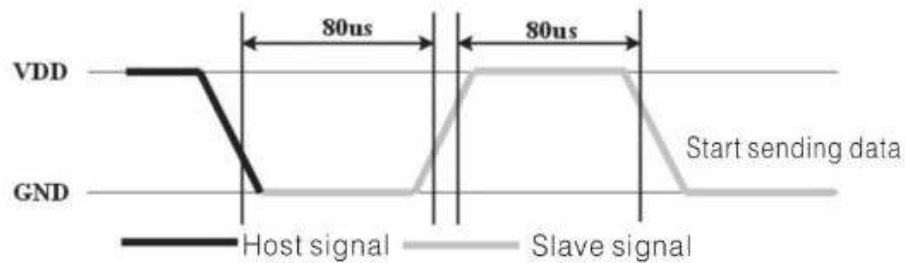


Figura 22: Respuesta del sensor tras la señal de inicio, DHT11[38]

- Mediante la línea de datos, el microcontrolador recibe 40 bits de datos, siendo un 0 lógico un pulso de 50 $\mu$ s en bajo seguido de 26 $\mu$ s-28 $\mu$ s de un pulso en alto, y un 1 lógico un pulso en bajo de 50 $\mu$ s seguido de 70 $\mu$ s en alto, véase la Figura 23.

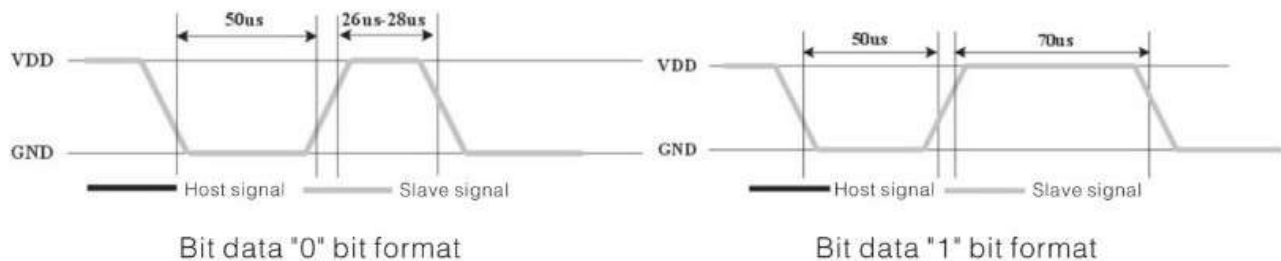


Figura 23: Bits 0 y 1 de la respuesta del sensor DHT11[38]

5. Una vez enviados los 40 bits de datos provenientes del sensor, envía un pulso en bajo de 50us y posteriormente la línea de datos será puesta en 1 mediante la resistencia *pull-up*, posteriormente vuelve a medir la Temperatura y Humedad relativa y finalmente se pone a la espera de una señal externa. Un ejemplo de lectura típica del sensor DHT11 se muestra en la Figura 24:

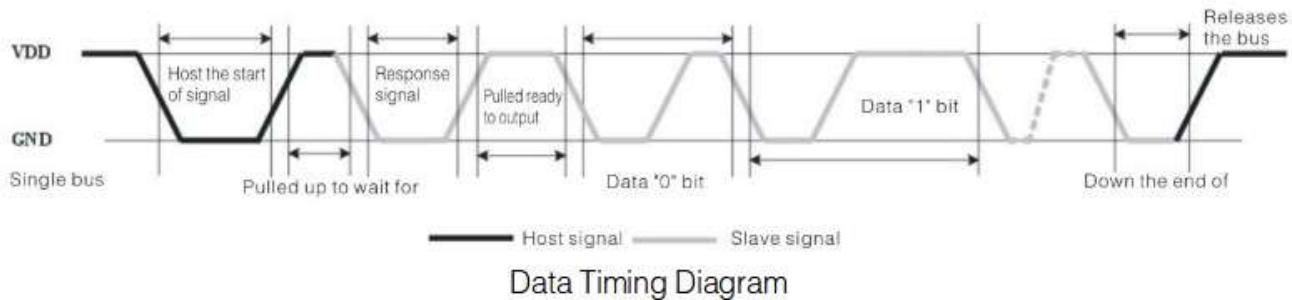


Figura 24: Lectura típica del sensor DHT11[38]

En mediciones de Humedad Relativa se tiene una exactitud de  $\pm 5\%$  RH (*Relative Humidity*) a  $25^{\circ}\text{C}$  con una variación del  $\pm 1\%$  entre mediciones y una histéresis menor a  $\pm 0.3\%$  RH. Para las mediciones de temperatura se tiene una variación de  $\pm 0.2^{\circ}\text{C}$  entre mediciones. [38] En ambos casos, humedad relativa y temperatura, la medición consta de 16 bits: 8 bits para la parte entera de la medición y 8 bits para la parte fraccional. En la Tabla 2 se ejemplifica la interpretación de los datos tras la lectura del sensor ( $24^{\circ}\text{C}$  y  $53\%\text{RH}$ ).

*Tabla 2: DHT11, ejemplo de lectura del sensor*

|               |           |           |               |               |           |
|---------------|-----------|-----------|---------------|---------------|-----------|
| Dato recibido | 0011 0101 | 0000 0000 | 0001 1000     | 0000 0000     | 0100 1101 |
| Descripción   | Humedad H | Humedad L | Temperatura H | Temperatura L | Paridad   |

$$\text{Paridad} = \text{Humedad H} + \text{Humedad L} + \text{Temperatura H} + \text{Temperatura L}$$

$$\text{Paridad} = 0011\ 0101 + 0000\ 0000 + 0001\ 1000 + 0000\ 0000 = 0100\ 1101$$

Humedad Relativa: 0011 0101 = 35H = 53%RH y la Temperatura: 0001 1000 = 18H = 24°C.

En caso que la suma de los 32 bits correspondientes a las mediciones de temperatura y humedad relativa no concuerden con los ocho bits de paridad entonces los datos son erróneos y es necesario repetir la lectura del sensor.



## CAPÍTULO 4: SOFTWARE UTILIZADO

En este capítulo se describe de manera general el software utilizado para el desarrollo y funcionamiento del sistema, mencionando sus principales características, entornos de desarrollo, configuraciones realizadas haciendo uso de la línea de comandos.

Puesto que algunos de los apartados del presente capítulo son, por lo general, ajenos a la formación académica de un ingeniero en electrónica, se abordarán de manera que se tenga la mejor comprensión posible de los mismos.

La interacción entre los diversos componentes del sistema y la manera en que fluye información entre ellos será descrita en el CAPÍTULO 5: DESCRIPCIÓN DEL SISTEMA.

### 4.1 ESPlorer

ESPlorer es una software multiplataforma para los desarrolladores que utilizan el módulo ESP8266. Es compatible con MicroPython y LUA para el NodeMCU, además de comandos AT. Requiere JAVA SE en versión 7 o posterior.

Tiene compatibilidad con los sistemas operativos: Windows, Linux, Solaris y Mac OS.

Una vez que el módulo NodeMCU tiene programado el firmware, se ejecuta ESPlorer mediante el comando:

```
java -jar ESPlorer.jar
```

La interfaz de usuario, tiene diferentes opciones de configuración y desarrollo, siendo una secuencia típica la siguiente:

1. Clic en el botón de actualizar.
2. Seleccionar el puerto donde se encuentre conectado el módulo ESP8266.
3. Seleccionar la tasa de baudios correspondiente.
4. Clic en el botón “Open”.
5. Seleccionar la opción NodeMCU+MicroPython, AT-based o RN2483 a conveniencia.
6. Seleccionar la opción “Scripts”.
7. Crear un documento *.lua* y los archivos necesarios para la aplicación a desarrollar.

8. Guardar los archivos a ESP.
9. Clic en el botón “Upload”.
10. Clic en el botón “Run”.

## 4.2 Lenguaje Lua

Lua es un lenguaje creado en 1993 que soporta programación secuencial, orientado a objetos, orientado a funciones, programación por eventos y lenguaje de descripción, esto hace de Lua un lenguaje de programación muy poderoso, eficiente y ligero. Combina sintaxis secuencial simple con descripción de estructuras de datos basado en arreglos asociativos y semántica extensible. Funciona mediante la interpretación de código bit con una máquina virtual basada en registros, incluyendo un manejo automático de memoria, por lo que es ideal para configuración y elaboración de prototipos en poco tiempo. [39]

Lua es distribuido con compatibilidad en cualquier sistema que tenga un compilador estándar de lenguaje C y es ejecutable en todas las versiones de Unix, Windows, dispositivos móviles (con sistema operativo Android, iOS, BREW, Symbian, Windows Phone), en microprocesadores embebidos (como los microprocesadores ARM), etc. [39]

El lenguaje Lua ha sido implementado en diversas aplicaciones industriales (por ejemplo, en Adobe Photoshop Lightroom), sistemas embebidos y juegos. [39]

Otra gran ventaja del código Lua, además de ser uno de los lenguajes interpretados más rápidos, es su integración con otros lenguajes de programación como C, C++, Java, C#, Smalltalk, Fortran, Ada, Erlang, entre otros, lo cual se debe a que a bajo nivel el lenguaje Lua esta desarrollado en lenguaje C. [39]

Es importante mencionar que se trata de un lenguaje de programación de código abierto, multipropósito y gratuito.

## 4.3 Node.js y Node Package Manager

Node.js es un entorno de ejecución de código abierto, eficiente, multiplataforma, asíncrono y manejado por eventos que ejecuta código JavaScript fuera del explorador de Internet. Diseñado para realizar aplicaciones escalables en redes informáticas. La principal ventaja al ser manejado por eventos es que no hay procesos bloqueantes. Dada su característica no-bloqueante es posible realizar aplicaciones escalables. [40]

A diferencia de otros frameworks, Node no es estrictamente para desarrollo web, sin embargo es extensamente utilizado para realizar servidores IRC (*Internet Relay Chat*), servidores chat, servidores

http, etc. Una de las diferencias entre JavaScript y *Node* es que, utilizando JavaScript en el navegador, cualquier función o variable creada fuera de una función u objeto es ligada de manera global, por lo que estará disponible independientemente de dónde sea utilizada; esto no ocurre en *Node*. [40]

*Node* dispone de una amplia gama de bibliotecas para el desarrollo de sistemas. Para utilizar una biblioteca simplemente se incluye mediante la función “*require*”, la cual regresa un objeto que puede ser guardado en una variable. La gestión de bibliotecas se realiza mediante el software *npm* (*Node Package Manager*), dicho software es manejado mediante la interfaz de línea de comandos y, de acuerdo con [41], es utilizado para:

- Compartir bibliotecas o paquetes de desarrollo.
- Descargar o actualizar bibliotecas.
- Administrar múltiples versiones de código y bibliotecas.

La instalación de una biblioteca se puede realizar de dos maneras: local o global. En la instalación global se puede trabajar con la biblioteca en cuestión independientemente del directorio en el cual se esté desarrollando la aplicación. [41] La instalación local es utilizada para emplear de manera específica las bibliotecas que se necesiten, haciendo uso de la función “*require*”. La instalación de una biblioteca local se realiza mediante el comando: `> npm install <nombre>`, mientras que una instalación global es: `> npm install -g <nombre>`. Para que la biblioteca instalada aparezca en la lista de dependencias del sistema se pueden emplear las siguientes banderas:

- `-S, --save`: el paquete se guarda en las dependencias del sistema.
- `-D, --save-dev`: el paquete se guarda en las dependencias de desarrollo del sistema.
- `-O, --save-optional`: el paquete se guarda en las dependencias opcionales del sistema.

Una manera de administrar los paquetes instalados es crear un archivo *package.json* mediante *npm*, dicho archivo enlistará los paquetes y bibliotecas que incluye el sistema, permitirá al usuario especificar las versiones de los paquetes y bibliotecas utilizadas además de facilitar compartir el sistema con otros desarrolladores de manera sencilla. [41] El archivo *package.json* debe contener el nombre del sistema en minúsculas y sin espacios, además la versión del sistema, esto con el formato *x.x.x*, por ejemplo:

```
{
  "nombre": "mi-sistema"
  "version": "1.00"
}
```

Con base en [41], para crear el archivo *package.json* se puede proceder de dos maneras:

- > *npm init*: este comando iniciará, mediante la línea de comandos, un cuestionario que concluirá con la creación del archivo *package.json* en el directorio en el que se ejecutó el comando.
- > *npm init --yes*: se generará un archivo *package.json* estándar utilizando la información extraída del directorio actual.

Una vez instaladas las dependencias y bibliotecas a utilizar y creado el archivo *package.json*, entonces se procede a crear el archivo que contiene el código Node, por ejemplo *mi-sistema.js*, y se ejecutará mediante el comando “*node mi-sistema.js*”.

## 4.4 Implementación del Broker MQTT con protocolo SSL usando Mosquitto

Mosquitto es una aplicación de Eclipse utilizado para generar tanto servidores (Broker MQTT) como clientes. Es utilizado para conectar dispositivos, por ejemplo: computadoras, microcontroladores, servidores, etc., un ejemplo de ello se muestra en la Figura 25. Utiliza una dinámica de suscripción y publicación de mensajes, cuyo protocolo es similar al protocolo TCP/IP cliente-servidor donde el gateway realiza la función del cliente y el Broker MQTT asemeja un servidor, el cual se comunica a la IP del cliente utilizando HTTP. Una variación del protocolo MQTT es MQTT-SN, cuya principal diferencia es que su implementación se enfoca a dispositivos embebidos, como ZigBee, que se encuentren en redes informáticas no compatibles con protocolos TCP/IP. [42]



Figura 25: MQTT Broker, clientes MQTT, Mosquitto, fuente: [42]

## CAPÍTULO 4: SOFTWARE UTILIZADO

Mosquitto puede ser instalado desde el repositorio de software en distribuciones de Linux, por ejemplo en Ubuntu se instala mediante el comando:

```
$ sudo apt-get install mosquitto mosquitto-clients
```

Por defecto, Ubuntu iniciará Mosquitto después de la instalación.

Para agregar certificados de encriptación y hacer lo más seguro posible el uso de MQTT, se instaló el software Cerbot [43], el cual ofrece gratuitamente certificados SSL a través de su API. Se puede añadir el repositorio más actual de Cerbot con el comando:

```
$ sudo add-apt-repository ppa:cerbot/cerbot
```

Posteriormente se debe actualizar la lista de paquetes para que el sistema reconozca la información de los paquetes nuevos:

```
$ sudo apt-get update
```

Finalmente , se instala el cliente Cerbot desde el repositorio:

```
$ sudo apt-get install cerbot
```

Cerbot utiliza el puerto 80 (HTTP) y/o el puerto 443 (HTTPS) para comprobar la encriptación de nuestro dominio, se deben abrir estos puertos con los comandos:

```
$ sudo ufw allow http
```

```
$ sudo ufw allow https
```

De acuerdo con [43], Cerbot puede ser ejecutado para obtener el certificado SSL mediante el comando:

```
$ sudo certbot certonly --standalone --standalone-supported-challenges http-01 -d <IP/dominio>
```

Los argumentos que recibe el comando son:

- *certonly*: especifica que no habrá pasos adicionales en la configuración para la obtención de certificados.
- *--standalone*: habilita a Cerbot para manejar las peticiones que se realicen mediante HTTP o HTTPS.
- *--standalone-supported-challenges http-01*: limita la comunicación a los puertos 80 (como es el caso del ejemplo) y/o 443.
- *-d <IP/dominio>*: se especifica el dominio o la dirección IP para la que se desea obtener el certificado SSL.

## CAPÍTULO 4: SOFTWARE UTILIZADO

Los certificados obtenidos serán válidos únicamente por 90 días, y deberán renovarse continuamente. Para renovar los certificados de manera automática se debe editar un archivo llamado *contrab*:

```
$ sudo contrab -e
```

Con base en [43], una vez seleccionado el editor de textos para abrir el archivo, se debe agregar al final:

```
15 3 * * * certbot renew --noninteractive --post-hook "systemctl restart mosquitto"
```

Siendo los argumentos:

- *15 3 \* \* \**: hace que el comando se ejecute todos los días a las 3:15 a.m., la hora puede ser modificada a conveniencia.
- *renew*: Cerbot revisará si los certificados instalados en el equipo están actualizados y si expirarán en menos de 30 días.
- *--noninteractive*: Cerbot no esperará una acción del usuario para proceder.
- *--post-hook "systemctl restart mosquitto"*: se reiniciará Mosquitto para incluir el nuevo certificado, pero solamente si el certificado ha sido renovado.

Al utilizar el software Mosquitto es preferible que se configure para utilizar contraseñas. Mosquitto incluye una utilidad para generar un archivo dedicado para contraseñas llamado *mosquitto\_passwd*. Para añadir una contraseña asociada a un usuario en el archivo *mosquitto\_passwd* se ejecuta el comando [42]:

```
$ sudo mosquitto_passwd -c /etc/mosquitto/passwd <username>
```

Una vez ejecutado este comando se tendrá que introducir la contraseña para el usuario *<username>* y posteriormente corroborarla al ingresarla nuevamente. Los cambios serán guardados en el archivo */etc/mosquitto/passwd*.

Para configurar el uso de contraseñas para autenticar usuarios en Mosquitto es necesario modificar el archivo */etc/mosquitto/conf.d/default.conf*, para editar el archivo se ejecuta el comando [42]:

```
$ sudo nano /etc/mosquitto/conf.d/default.conf
```

Al abrirse el archivo se deberá añadir lo siguiente:

```
allow_anonymous false
```

```
password_file /etc/mosquitto/passwd
```

Las acciones que realiza Mosquitto mediante la configuración anterior son:

- *allow\_anonymous false*: deshabilita todas las conexiones que no especifiquen un nombre de usuario y contraseña correctos.
- *password\_file /etc/mosquitto/passwd*: se le indica a Mosquitto dónde buscar la información referente a usuarios y contraseñas.

Para que Mosquitto se ejecute con las modificaciones anteriores es necesario reiniciar mosquitto:

```
$ sudo systemctl restart mosquitto
```

De acuerdo con [42], para habilitar la encriptación SSL en Mosquitto, es necesario especificar dónde están almacenados los certificados SSL, es necesario editar nuevamente el archivo de configuración de Mosquitto */etc/mosquitto/conf.d/default.conf* y añadir:

```
listener 1883 localhost
```

```
listener 8883
```

```
certfile /etc/letsencrypt/live/<IP/dominio>/cert.pem
```

```
cafile /etc/letsencrypt/live/<IP/dominio>/chain.pem
```

```
keyfile /etc/letsencrypt/live/<IP/dominio>/privkey.pem
```

Mediante la configuración “*listener 1883 localhost*” se actualiza el puerto que utiliza MQTT; el puerto 1883 es utilizado para MQTT no encriptado. Al especificar “*localhost*” solo se enlazarán a Mosquitto peticiones MQTT locales; todas las peticiones externas serán bloqueadas por el firewall del sistema. Adicionalmente se configura el puerto 8883, esto mediante “*listener 8883*”, que es el puerto estándar para el protocolo MQTT con certificado SSL, también llamado MQTTS. Las últimas tres líneas, “*certfile ...*”, “*cafile ...*” y “*keyfile ...*”, le indican a Mosquitto la ruta de los archivos de configuración para las conexiones encriptadas. Nótese que es necesario especificar la IP o el dominio para el cual se tiene certificado SSL [42]. Una vez guardados los cambios en el archivo de configuración, es necesario reiniciar:

```
$ sudo systemctl restart mosquitto
```

Finalmente se deshabilita el firewall para los puertos MQTT y MQTTS:

```
$ sudo ufw allow 1883
```

```
$ sudo ufw allow 8883
```

Ahora se puede abrir un Broker MQTTS mediante el comando [42]:

```
$ mosquitto_sub [-v] [-h <IP/dominio>] -t <topic> [-p <port>] --capath /etc/ssl/certs/ -u
"<username>" -P "<password>"
```

Para hacer una publicación es necesario conectarse como cliente MQTTS, lo cual se hace con el siguiente comando:

```
$ mosquitto_pub [-h <IP/dominio>] -t <topic> [-p <port>] --capath /etc/ssl/certs/ -u
"<username>" -P "<password>" -m "<message>" [-i <clientId>]
```

Los argumentos utilizados son [42]:

*Tabla 3: Comandos principales de MQTT*

| Argumento                | Descripción                                                                                                                                                                                                                                                                             |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -v                       | El servidor MQTT desplegará los mensajes y el topic en el que se envió. En caso de no habilitarse únicamente se mostrarán los mensajes recibidos.                                                                                                                                       |
| [-h <IP/dominio>]        | Especifica el host del servidor. En caso de no especificar el host por defecto será localhost.                                                                                                                                                                                          |
| -t <topic>               | Define el topic al cual se hará la suscripción o publicación de mensajes. Para suscribirse o enviar un mensaje a múltiples topics esta opción deberá repetirse.                                                                                                                         |
| [-p <port>]              | Expresa el puerto en el cual se hará la conexión. Por defecto, en caso de no especificar, se conectará al puerto 1883.                                                                                                                                                                  |
| --capath /etc/ssl/certs/ | Establece la ruta o directorio en donde se encuentran los certificados de confianza. Especialmente utilizado para habilitar la comunicación SSL. La ruta "/etc/ssl/certs/" puede cambiar dependiendo el sistema operativo y elecciones de ruta para guardado que haya hecho el usuario. |
| -u "<username>"          | Proporciona el nombre de usuario para la autenticación con el Broker MQTT.                                                                                                                                                                                                              |
| -P "<password>"          | Contraseña utilizada para autenticar al usuario especificado ante el Broker MQTT. Utilizar este argumento será inválido sin un usuario al cual relacionar.                                                                                                                              |
| -m "<message>"           | Mensaje a publicar.                                                                                                                                                                                                                                                                     |
| [-i <clientId>]          | Identificador a usar para el cliente desde donde se publicará un mensaje y así evitar colisión de mensajes. De no ser especificado, se generará un identificador aleatorio.                                                                                                             |

Una descripción del resto de los posibles argumentos a utilizar puede ser encontrada en [https://mosquitto.org/man/mosquitto\\_sub-1.html#](https://mosquitto.org/man/mosquitto_sub-1.html#).



## 4.5 Base de Datos MySQL

MySQL es un sistema de gestión de bases de datos de código abierto (*Open-Source Relational Database Management System*, RDBMS) que opera bajo un lenguaje de consulta estructurado (*Structured Query Language*), escrito en lenguaje C y C++. MySQL se puede ejecutar en diversas plataformas incluyendo: AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, macOS, Microsoft Windows, NetBSD, entre otros. Ofrece soporte mediante foros, además de disponer de un manual oficial. MySQL también es integrado en software de terceros, por ejemplo: MariaDB y Percona. [44]

De acuerdo con [44], entre las principales ventajas que tiene el uso de MySQL destacan:

- Funciona considerablemente bien en la mayoría de las aplicaciones.
- Fácil acceso a la documentación y foros de soporte.
- Dispone de un servidor multi-usuario y multi-tarea rápido y estable.
- Soporta el protocolo de seguridad SSL.
- Es compatible con múltiples rutinas de almacenamiento.
- Puede ser ejecutado en plataformas de informática basadas en la nube, tales como: Microsoft Azure, Amazon EC2, Oracle Cloud Infrastructure.
- Además del manejo mediante línea de comandos, ofrece un entorno gráfico para administrar, modelar y diseñar bases de datos, llamado MySQL Workbench, cuyo entorno gráfico será utilizado para ejemplificar posteriormente el estado y características de la base de datos utilizada .

MySQL puede ser instalado mediante diferentes métodos; por ejemplo, utilizando los repositorios APT de Linux:

```
$ sudo apt-get install mysql-server
```

Tras haber instalado MySQL se crearán diversos directorios: todos los archivos de configuración (*my.cnf*, por ejemplo) estarán en la ruta */etc/mysql*, el directorio de información será */var/lib/mysql*, y los archivos binarios, bibliotecas, headers etc. tendrán los directorios */usr/bin* y */usr/sbin*. Es importante conocer la ubicación de los archivos y directorios que permiten modificar la configuración inicial y la configuración de arranque.

Otros comandos de interés:

- `$ sudo service mysql status`: revisa el estado del servicio MySQL.
- `$ sudo service mysql stop`: detiene el servicio MySQL.
- `$ sudo service mysql start`: reinicia el servicio MySQL.
- `$ sudo apt-get install mysql-client`: instala los paquetes necesarios para administrar clientes de MySQL.

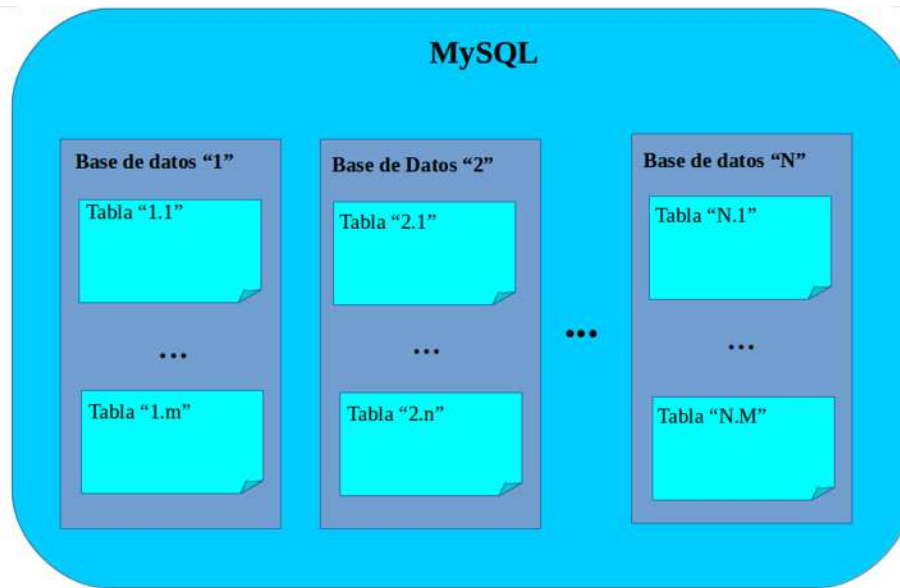
La conexión con el servidor se inicia con el siguiente comando:

```
$ mysql -h <host> -u <usuario> -p
```

Posteriormente se introduce la contraseña para el usuario registrado de MySQL, o un usuario registrado y habilitado para ingresar o modificar la base de datos. El host es donde se esté ejecutando el servidor MySQL. Si el servicio se está ejecutando en el mismo equipo desde el cual se quiere ingresar a la base de datos, entonces este parámetro se puede omitir.

MySQL puede administrar múltiples bases de datos, cada una de las cuales puede contener ‘*n*’ tablas, véase la Figura 26. La información almacenada en las tablas es ordenada en columnas, definidas por el usuario, y renglones que representan la información contenida en la tabla, véase la Figura 27. Una tabla puede estar relacionada con una o varias tablas, las cuales pueden estar contenidas en la misma base de datos o en una diferente.

Es importante mencionar que un renglón perteneciente a una tabla de la base de datos podrá ser almacenado sin contener información específica para todas las columnas de la tabla; en dicho caso la información será del tipo “*NULL*”, véase la Figura 27.



*Figura 26: Bases de Datos, MySQL*

| name     | owner  | species | sex | birth      | death      |
|----------|--------|---------|-----|------------|------------|
| Fluffy   | Harold | cat     | f   | 1993-02-04 |            |
| Claws    | Gwen   | cat     | m   | 1994-03-17 |            |
| Buffy    | Harold | dog     | f   | 1989-05-13 |            |
| Fang     | Benny  | dog     | m   | 1990-08-27 |            |
| Bowser   | Diane  | dog     | m   | 1979-08-31 | 1995-07-29 |
| Chirpy   | Gwen   | bird    | f   | 1998-09-11 |            |
| Whistler | Gwen   | bird    |     | 1997-12-09 |            |
| Slim     | Benny  | snake   | m   | 1996-04-29 |            |

*Figura 27: Tabla, Base de Datos, MySQL, fuente: [44]*

En este trabajo, se empleó una base de datos MySQL, siendo administrada mediante la biblioteca correspondiente en Node.js.

## 4.6 Amazon Web Services

Con el propósito de reducir costos se utilizó un servidor Amazon, para evitar la necesidad de un servidor local (como lo pudiera ser computadora, un celular, una tableta, una raspberry, etc) que

siempre debe permanecer encendido para realizar el monitoreo y control del sistema en todo momento, además de abrir los puertos utilizados por el protocolo MQTT y HTTP para acceder desde cualquier parte del mundo. Amazon ofrece un servicio de Cloud Hosting que simplifica las problemáticas descritas. [45]

Dentro de los servicios que ofrece Amazon Web Service, se utilizó el servicio EC2. EC2 es un servicio web de Amazon que provee seguridad y capacidad computacional escalable en la nube. AWS EC2 está diseñado para facilitar aplicaciones web escalables para los desarrolladores. [45]

La interfaz de Amazon EC2 permite a los desarrolladores obtener y configurar la capacidad de procesamiento que requieran, además provee control de todas las instancias solicitadas. Una instancia es un servidor que se solicita y gestiona por medio de AWS EC2, una instancia es una máquina que se administra remotamente, se puede especificar la potencia de procesamiento que se requiera (esto hace variar el costo), el sistema operativo del servidor, los puertos de acceso, entre otros. Es importante aclarar que el acceso al servidor web para el desarrollador es a través de SSH; Amazon proporcionará “llaves” y “credenciales” de seguridad para tener acceso por medio de SSH. [45]

Algunos de los beneficios de Amazon EC2 son [45]:

- Escalabilidad: Amazon EC2 permite aumentar o disminuir la capacidad de procesamiento en cuestión de minutos y con una duración especificada por el usuario. Esto permite tener solamente una instancia o cientos para cubrir la potencia de procesamiento requerida por el servidor. Amazon cuenta con la opción de auto-escalar la instancia EC2 en función de la capacidad demandada por la aplicación que esté desarrollando el usuario, permitiendo que el desempeño siempre sea óptimo.
- Control total: El usuario tiene en todo momento control de las instancias, incluyendo el acceso como root. El usuario puede detener una instancia mientras retiene la información en la partición de arranque y posteriormente reiniciar la instancia. Las instancias pueden ser reiniciadas de manera remota utilizando el servicio web de Amazon, además de tener acceso a su terminal de control.
- Servicio de Cloud Hosting flexible: Se pueden tener múltiples instancias, diferentes sistemas operativos, e inclusive distintos paquetes de software. Amazon EC2 le permite al usuario seleccionar la configuración de memoria, CPU y tamaño de particiones que sean óptimas para el sistema operativo y la aplicación elegidos. Entre algunos de los sistemas operativos se encuentran numerosas distribuciones Linux y el servidor Web de Microsoft.
- Integración de Servicios: Amazon EC2 se integra con otros servicios de Amazon como: Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service

(Amazon RDS) y Amazon Virtual Private Cloud (Amazon VPC) para proporcionar una solución completa y segura para cualquier tipo de aplicación.

- **Confiable:** Amazon EC2 ofrece alta confiabilidad, teniendo la capacidad de reemplazar instancias de manera rápida en caso de ser necesario. El servicio se ejecuta en todo momento ya que Amazon provee la red, la infraestructura y los recursos de almacenamiento, asegurando un 99.99% de disponibilidad de servicio para cada región de Amazon EC2.
- **Seguridad:** El almacenamiento y acceso remoto de información de manera segura es la mayor prioridad. Como usuario de Amazon Web Server se tiene el beneficio de tener a disposición almacenamiento de información y arquitectura de red que cumplen los más altos estándares de seguridad.
- **Bajo costo:** los beneficios financieros radican en que el usuario paga únicamente por los recursos de procesamiento que consuma, teniendo a su disposición una amplia gama de configuraciones de hardware y servicios para sus aplicaciones.
- **Fácil de utilizar:** existen muchas maneras para comenzar a utilizar el servicio EC2 de Amazon; se puede hacer uso de *AWS Management Console*, las herramientas de línea de comandos de AWS, o AWS SDKs.

Dadas las ventajas de utilizar AWS EC2 el servidor MySQL y el Broker MQTT se ejecutan en la instancia de Amazon EC2, así como lo referente a la página web que será la interfaz de usuario. Como la instancia EC2 se configura con una distribución de Linux, todos los comandos y configuraciones descritas en capítulos anteriores referentes al software utilizado serán empleados en el servidor web de Amazon. [45]

En el Apéndice B: Renta y configuración de un servidor EC2 de AWS se muestra la información detallada referente al servidor y la configuración utilizada para el desarrollo de este trabajo.

### 4.7 Desarrollo de la página Web

Para que el usuario interactúe con el sistema, se diseñó una página web, la cual tiene como pilares fundamentales para el desarrollo: HTML, CSS y JavaScript, todos ellos siendo ejecutados desde el entorno de desarrollo *Node.js*. Al ejecutarse desde el entorno *Node.js* es posible hacer uso de paquetes que faciliten la programación de la página web; para efectos de este trabajo se utilizaron los paquetes *express*, *ejs*, *body-Parser* y *nodemon*, cuyas funciones son [46]:

- **Express:** provee herramientas informáticas para desarrollar servidores HTTP, facilitando diversas tareas como direccionamiento, manejo de peticiones, etc.

- *Ejs (embedded JavaScript)*: es un paquete que permite ejecutar comandos JavaScript con la opción de importar estructuras externas a la plantilla HTML y utilizarlas para facilitar la presentación de páginas web dinámicas. Otra característica importante que hace clave la implementación del paquete *ejs* es facilitar el envío de documentos HTML, ayudando a organizar y simplificar el script principal de Node.js.
- *Body-Parser*: facilita el manejo de middleware antes de ser administrado por otras funciones que gestionen peticiones de tipo GET o POST.
- *Nodemon*: este paquete es útil para desarrollar aplicaciones empleando Node.js, reiniciando la aplicación automáticamente al guardarse un cambio en el directorio de la aplicación, agilizando el proceso de desarrollo de la página web (únicamente siendo necesario actualizar el navegador), manejo de la base de datos y gestión de dispositivos MQTT.

La estructura de la página web se diseñó en HTML, el cual es un lenguaje de marcado de hipertexto creado por Tim Berners-Lee en 1990. El hipertexto hace referencia a marcas o links contenidos en el texto que permiten identificar la estructura del mismo. Emplear un lenguaje de marcas, como es el caso de HTML, en una manera en que las computadoras se comunican entre sí para controlar cómo se procesa y se presenta la información, para ello HTML emplea etiquetas y atributos: las etiquetas son utilizadas para marcar dónde comienza un elemento de HTML y usualmente se encierran en paréntesis angulares, por ejemplo `<h1>` (las etiquetas tienen que ser abiertas, `<h1>`, y posteriormente cerradas, `</h1>`, para poder ser ejecutadas). Los atributos en HTML contienen información adicional que se asigna a la etiqueta para la cual se atribuyan, por ejemplo: `<img scr="mydog.gif" alt="A photo of my dog">`, en este caso la imagen fuente (*scr*) y el texto (*alt*) son atributos de la etiqueta `<img>`. Es importante mencionar que la mayoría de las etiquetas tienen que ser abiertas e incluir su respectivo cierre, incluyendo sus atributos entre su apertura y su cierre. Cuando se utilizan múltiples etiquetas, estas deberán ser cerradas en el mismo orden en el que fueron abiertas. [47]

Cuando se desarrolle un documento HTML, la primer etiqueta deberá ser `<!DOCTYPE html>`, esto para especificar el lenguaje en el que será redactado el script, posteriormente se abre y cierra la etiqueta que contendrá el documento HTML: `<html> </html>`, dichas etiquetas tendrán que ser contenidas en cualquier documento HTML. Otras etiquetas importantes son [47]:

- `<head> </head>`: contiene el encabezado del documento.
- `<body> </body>`: especifica cuál será el cuerpo del documento.
- `<title> </title>`: contiene el nombre que se muestra en la pestaña del navegador web.

- `<p> </p>`: se emplean para añadir texto a manera de párrafos, teniendo la opción de modificar las características del texto al utilizar etiquetas adicionales dentro de las etiquetas `<p> </p>`.

Otros ejemplos de etiquetas son mostrados en la Tabla 4.

*Tabla 4: Etiquetas HTML para modificar características del texto, fuente: [47]*

| Etiqueta                    | Función            |
|-----------------------------|--------------------|
| <code>&lt;b&gt;</code>      | Texto plano        |
| <code>&lt;strong&gt;</code> | Texto en negritas  |
| <code>&lt;i&gt;</code>      | Texto en itálicas  |
| <code>&lt;em&gt;</code>     | Texto enfatizado   |
| <code>&lt;mark&gt;</code>   | Marcar texto       |
| <code>&lt;small&gt;</code>  | Texto pequeño      |
| <code>&lt;del&gt;</code>    | Texto borrado      |
| <code>&lt;ins&gt;</code>    | Texto insertado    |
| <code>&lt;sub&gt;</code>    | Texto de subíndice |
| <code>&lt;sup&gt;</code>    | Texto superíndice  |

El lenguaje HTML ofrece otras herramientas para estructurar páginas web, por ejemplo: añadir enlaces a otras páginas web, incluir imágenes, presentar información en forma de listas, añadir tablas al documento, etc.

Para poder modificar la presentación visual de la página web se emplea CSS, el cual es un lenguaje utilizado para añadir color, imágenes (o colores) de fondo, texturas, etc. En la Figura 28 se muestra un ejemplo en donde se presenta una página web con CSS y sin el uso de estas.

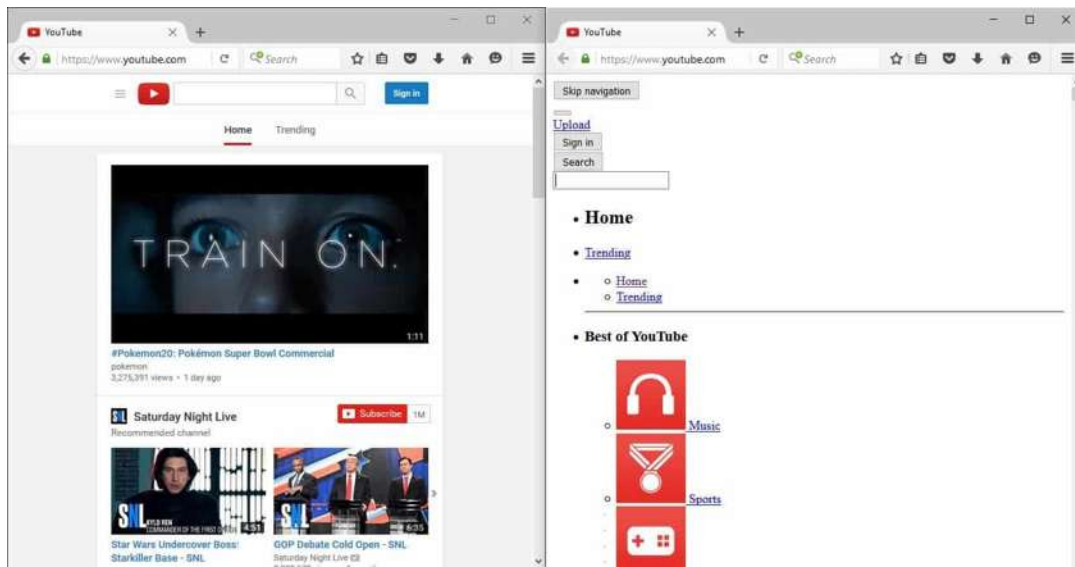


Figura 28: Página web con CSS (izquierda) y sin CSS (derecha), fuente: [48]

Como se puede ver en la Figura 28, CSS es un lenguaje utilizado para estilizar visualmente páginas web y así hacerlas más atractivas y fáciles de utilizar para el usuario. La sintaxis de CSS incluye selectores, propiedades, valores, declaraciones, bloques de declaraciones, y reglas [48]:

- **Selector:** fragmento de código utilizado para identificar el o los elementos de la página web que serán afectados por los estilos.
- **Propiedad:** es el aspecto del elemento que será afectado, siendo algunas de propiedades más comunes: color, relleno, margen, color de fondo, entre otras.
- **Valor:** es utilizado para definir una propiedad, por ejemplo, la propiedad de color deberá ser especificada con su respectivo valor, el cual puede ser rojo: “*color: red;*”.
- **Declaración:** es la combinación de propiedades con sus respectivos valores.
- **Selector(es):** en la mayoría de los casos, múltiples declaraciones son aplicadas a un selector simple. Un bloque de declaraciones se refiere a todas las declaraciones aplicadas a un selector simple.
- **Reglas:** es utilizado para crear bloques de reglas CSS que son aplicadas a la visualización web.

Un ejemplo de estilos CSS se puede observar en la Figura 29, en la cual se define el estilo de un encabezado *h1* para ser de color rojo, con un tamaño de letra igual al del elemento padre (generalmente 16 pixeles) y estar subrayado.



En el ejemplo de la Figura 29, *h1* es el selector, el cual especifica que el conjunto de declaraciones CSS que le precede serán aplicadas al contenido HTML que se encuentre definido por las etiquetas `<h1></h1>`. Posteriormente se tiene el bloque de declaraciones que contiene, valga la redundancia, tres declaraciones (color, tamaño de fuente y estilo). Cada declaración es separada de la siguiente por punto y coma. Las tabulaciones y los corchetes son opcionales, pero son utilizados por la mayoría de desarrolladores para hacer que el código sea más comprensible. Se puede especificar el mismo conjunto de declaraciones para uno o varios selectores, o un conjunto de declaraciones para cada selector. [48]

```
h1 {
    color: red;
    font-size: 3em;
    text-decoration: underline;
}
```

*Figura 29: Fragmento de Código, CSS*

Es importante mencionar que cuando se tenga una plantilla de estilos CSS, al hacer cualquier petición a la página web, se realizará una petición con la ruta `/styles.css`, y esperará que el servidor le responda con la plantilla de estilos a manera de texto plano que se deberá tratar como una petición del tipo *middleware*.

Para que la página web sea interactiva se utiliza JavaScript, el cual es un lenguaje de programación ligero que, gracias al sistema de Node.js, es posible ejecutarlo desde el servidor, y aprovechando el desarrollo de tecnologías que utilicen JSON como lenguaje de serialización de datos (data serialization language) es posible el uso de JavaScript en la aplicación web. JavaScript fue desarrollado originalmente en Netscape para ser utilizado en su buscador Netscape Navigator, a mediados del año 1990, siendo su aplicación enfocada a manipular objetos dentro del documento. [49]

Todo en JavaScript es un objeto, y por consiguiente tiene diferentes métodos de acceder a su contenido. Los objetos no son instanciados por clases, sino clonados de otros objetos. En lugar de escribir una clase, simplemente se creará un objeto prototipo. Lo anterior conlleva tanto ventajas como desventajas. Una ventaja es que si se necesita solo una instancia de un objeto no es necesario sobrescribir toda una clase y posteriormente escribir el código que lo instancia. La principal desventaja es que, en una aplicación extensa o compleja, seguramente habrá objetos prototipo no utilizados, pero clonados en nuevos objetos (recreando clases). [49]

JavaScript proporciona una manera de interactuar, y manipular, el contenido de una página web: empleando las etiquetas `<script>` `</script>` para definir un bloque de código en lenguaje JavaScript o hacer referencia a un archivo .js. [49]

JavaScript es generalmente utilizado para ser ejecutado en base a eventos. Por ejemplo, cada vez que se hace click en un elemento de la página (onclick event), hace que el código JavaScript asociado a dicho evento sea ejecutado. Otros ejemplos de posibles eventos [49]:

- Cuando se hace click en una imagen → cargar la imagen siguiente.
- Cuando se tenga el puntero del ratón en un área específica de la página → cambiar de color o hacer un acercamiento.
- Cuando el puntero del ratón se aproxime a la barra de búsqueda → abrir una ventana con una ventana de publicidad.

Los eventos de JavaScript en elementos de HTML son [49]:

*Tabla 5: Ejemplos de eventos JavaScript, fuente: [49]*

| <b>Evento</b> | <b>Descripción</b>                                   |
|---------------|------------------------------------------------------|
| onclick       | Click en un elemento.                                |
| oncontextmenu | Despliegue de un menú.                               |
| ondblclick    | Doble click a un elemento                            |
| onmousedown   | Se presiona el botón de bajar dentro de un elemento. |
| onmouseenter  | Selección de un elemento.                            |
| onmouseleave  | Se deja de seleccionar un elemento.                  |
| onmouseover   | Movimiento del mouse dentro de un elemento.          |
| onmouseout    | Salida de un elemento.                               |
| onmouseup     | Se presiona el botón de subir dentro de un elemento. |
| onerror       | Error mientras se carga un archivo externo.          |
| onload        | Se completa la carga de un archivo externo.          |
| onscroll      | Cuando el contenido es desplazado.                   |

Otras aplicaciones de JavaScript son [49]:

- **AJAX (JavaScript and XML):** es un conjunto de técnicas de desarrollo web que utiliza diferentes tecnologías en el lado del cliente (“*client side*” o “*front-end*”) para crear

aplicaciones web asíncronas. Con AJAX las aplicaciones web pueden enviar y recibir información de un servidor de forma asíncrona sin interferir con el despliegue de información de la página web, lo cual hace posible cambiar la información en la página web dinámicamente sin la necesidad de actualizar la página web.

- jQuery: es una de las bibliotecas estándar más comunes de JavaScript, la cual provee un amplio rango básico de funcionalidad, haciendo sencillas tareas como manipular objetos HTML y sus propiedades, comunicación con un servidor, interacción con el navegador web, etc.

Para poder comunicarse con el servidor se implementó el protocolo HTTP haciendo uso de peticiones tipo *GET* y *POST*.

En el modelo cliente-servidor son de suma importancia las peticiones de datos y recursos (como pueden serlo los documentos HTML). La petición es iniciada por el elemento que recibe los datos (el cliente), normalmente un navegador web, y al otro lado del canal de comunicación está el servidor, que envía los datos que ha pedido el cliente. Un servidor no tiene que ser necesariamente un único equipo físico, así como también varios servidores pueden estar funcionando en la misma computadora. [50]

Para poder mostrar una página web (la cual es un documento HTML), el navegador envía una petición de documento HTML al servidor, entonces procesa el documento y envía más peticiones para solicitar scripts, hojas de estilo (CSS) y otros datos que necesite (normalmente videos o imágenes). El navegador une todos estos documentos y datos para mostrar la página web. Entre cada petición y respuesta hay varios intermediarios, normalmente denominados proxies, los cuales realizan distintas funciones, como gateways o caches. Los proxies pueden ser transparentes o no (modificando las peticiones que pasan por ellos), y realizan varias funciones [50]:

- Caching: la caché web almacena copias de los recursos de una página web en el navegador o en servidores proxy con el objeto de evitar las limitaciones de la latencia y el ancho de banda. Ambos conceptos inciden sobre la velocidad de carga de una página. Al obtener una página o partes de ella desde las copias almacenadas en el navegador (o proxy) evitamos gastar conexiones con el servidor de origen y el ancho de banda queda disponible para otras acciones. Las páginas se cargan más rápido. Un beneficio adicional es la reducción en la carga de trabajo del servidor al disminuir el número de solicitudes. La caché puede ser pública (como la de los servidores *proxy*, los cuales actúan como intermediarios cercanos al usuario) o privada (de los dispositivos de usuario, como un navegador).
- Filtrado: como un antivirus, control parental, etc.
- Balanceo de carga de peticiones: para permitir a varios servidores responder a la carga total de peticiones.

- Autenticación: para el control de acceso de recursos y datos.
- Registro de eventos: para tener un recurso histórico de los eventos que se generan.

Los métodos principales para realizar una petición son: OPTIONS, PUT, DELETE, CONNECT, PATCH, HEAD, GET y POST ([51]), siendo de especial interés para fines de este trabajo los métodos GET y POST. El método GET es utilizado principalmente para solicitar información a una fuente específica, mientras que el método POST se utiliza para enviar información al servidor para crear o actualizar un recurso o archivo. Las principales diferencias y características de los métodos GET y POST son expuestos en la Tabla 6.

*Tabla 6: Características de peticiones HTTP GET y HTTP POST, fuente: [51]*

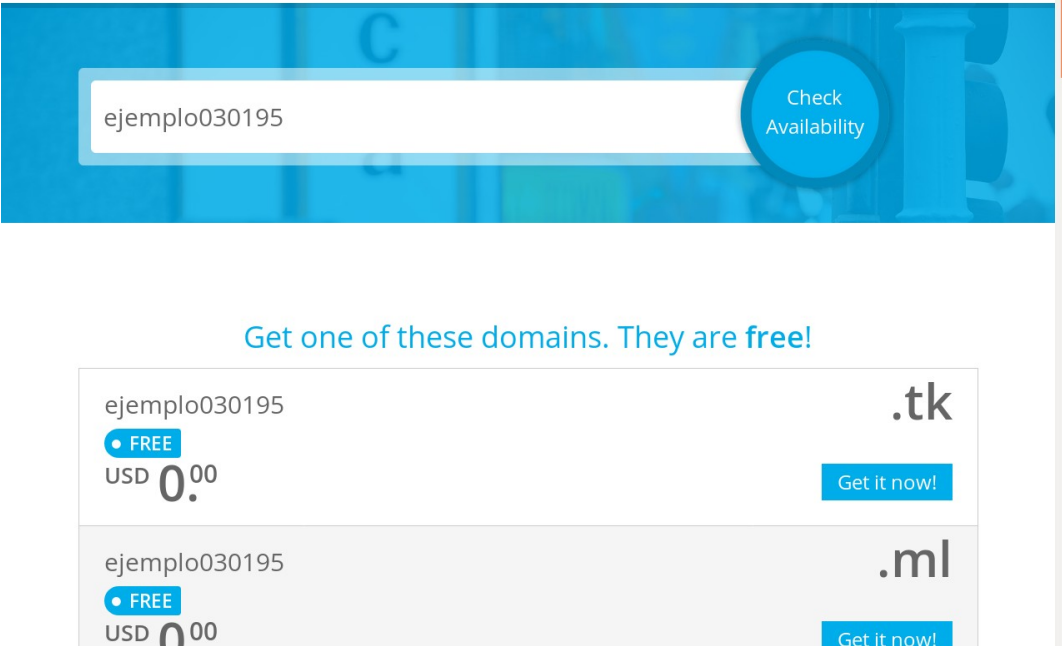
|                                               | <b>GET</b>                                      | <b>POST</b>                                                                                  |
|-----------------------------------------------|-------------------------------------------------|----------------------------------------------------------------------------------------------|
| Acción de retroceder o recargar el navegador: | No aplica.                                      | La información se enviará de nuevo. El usuario será notificado.                              |
| Marcación ( <i>Bookmarked</i> ):              | Aplica.                                         | No aplica.                                                                                   |
| Caché:                                        | Puede ser almacenado en caché.                  | No puede ser almacenado en caché.                                                            |
| Tipo de codificación:                         | /x-www-form-urlencoded                          | application/x-www-form-urlencoded o multipart/form-data                                      |
| Historial:                                    | Los parámetros son guardados en el historial.   | Los parámetros no son guardados en el historial.                                             |
| Restricción de la longitud de la información: | Limitado a 2048 caracteres.                     | Sin restricciones.                                                                           |
| Restricción de tipo de datos:                 | Solo se permiten caracteres ASCII.              | Sin restricciones.                                                                           |
| Seguridad:                                    | La información es enviada como parte de la URL. | Los parámetros no son almacenados en el historial del navegador o en registros del servidor. |
| Visibilidad:                                  | La información es visible en la URL.            | La información no es visible en la URL.                                                      |

En el caso de la aplicación web desarrollada, tanto las peticiones GET como las peticiones POST son manejadas en Node.js, haciendo uso de los paquetes descritos al principio de la sección 4.7 Desarrollo de la página Web.

## 4.8 Obtención de un dominio web

Es más fácil recordar el nombre alfanumérico de un dominio en comparación a una dirección IP, es por ello que se registraron los dominios <http://juangarcia.tk/> y <http://www.juangarcia.tk/> haciendo uso del servicio de registro de dominios que ofrece la página web <http://www.freenom.com/>.

En la opción *Servicios>Registrar un nuevo dominio* se ingresa el nombre de dominio de interés y posteriormente se corrobora que se encuentre disponible, véase la Figura 30. También ofrece información acerca de costos, en caso de que el dominio introducido no sea gratuito, y da la opción de seleccionar el periodo de tiempo por el cual se desea solicitar el dominio, lo cual se ilustra con la Figura 31.



The screenshot displays the Freenom domain registration process. At the top, a search bar contains the text 'ejemplo030195' and a 'Check Availability' button. Below this, a message states 'Get one of these domains. They are free!'. Two domain options are listed:

| Domain Name      | Price            | Action      |
|------------------|------------------|-------------|
| ejemplo030195.tk | FREE<br>USD 0.00 | Get it now! |
| ejemplo030195.ml | FREE<br>USD 0.00 | Get it now! |

Figura 30: Búsqueda de un dominio, <http://www.freenom.com/>

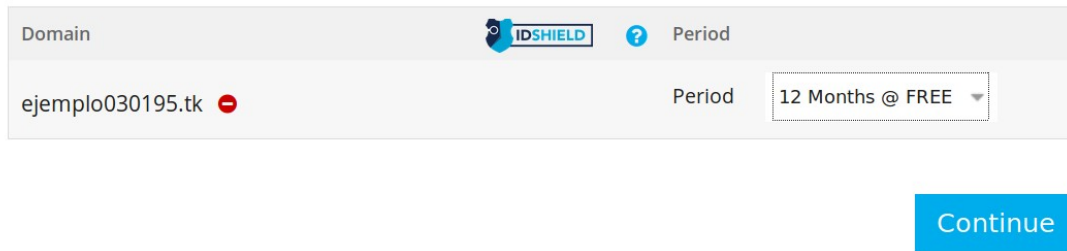


Figura 31: Selección del periodo para el dominio, <http://www.freenom.com/>

Una vez solicitado el dominio es mostrado en una lista al seleccionar la opción *Services My Domains*, véase la Figura 32.

| Domain ▾           | Registration Date ▾ | Expiry date ▾ | Status ▾ | Type ▾ |                 |               |
|--------------------|---------------------|---------------|----------|--------|-----------------|---------------|
| juangarcia.tk<br>✉ | 11/07/2018          | 11/07/2019    | ACTIVE   | Free   | Manage Domain ⚙ | Manage GoSite |

Results Per Page: 10 ▾

Figura 32: Dominios del usuario, <http://www.freenom.com/>

Finalmente, para poder asociar una dirección IP a un dominio se selecciona la opción *Manage Domain Manage Freenom DNS*, véase la Figura 33, y se especifica la IP en la opción *Target*. El servidor DNS de [www.freenom.com/](http://www.freenom.com/) dirigirá al servidor AWS cuando se haga una solicitud que incluya el nombre del dominio en un navegador de Internet.

## Modify Records

| Name | Type | TTL   | Target          |        |
|------|------|-------|-----------------|--------|
| WWW  | A    | 14440 | 255.255.255.255 | Delete |

Save Changes

Figura 33: Asociar una IP a un dominio, <http://www.freenom.com/>

## 4.9 Módulo “Enduser Setup”

Para evitar la programación de los módulos por el usuario final, se utilizó el módulo *enduser setup* proporcionado por la plataforma NodeMCU [52], el cual provee una manera simple de configurar la conexión a una red Wi-Fi sin necesidad de una interfaz serial o cambiar la programación de los módulos. Esto se logra configurando los módulos ESP-01 y NodeMCU primeramente como puntos de acceso y modo estación; el punto de acceso permite que mediante la conexión directa al módulo (como si el módulo fuera un módem o router) se pueda acceder a su dirección IP, donde se está ejecutando un servidor web desde el cual hace la configuración del módulo como estación seleccionando la red inalámbrica a la cual conectarse al proporcionar la contraseña necesaria. [52]

Su funcionamiento es descrito de manera práctica en la sección 6.1 Conexión de los módulos a la red Wi-Fi.

## CAPÍTULO 5: DESCRIPCIÓN DEL SISTEMA

En este capítulo se profundiza en los apartados clave para el desarrollo del sistema a partir del funcionamiento e interacción de los elementos de hardware y software que lo conforman, por ejemplo: las conexiones de los componentes físicos, el flujo de información entre los componentes del sistema, los protocolos de comunicación que utilizan, la gestión de la información por parte del servidor y la interfaz de usuario.

No se profundiza en componentes particulares para facilitar su comprensión y pueda ser entendido y aplicado haciendo uso de herramientas de software o hardware distintas a las expuestas.

### 5.1 Diagrama de conexiones

En el diagrama de conexiones se presenta la manera en que se conectan e interactúan físicamente los componentes utilizados. Dependiendo del módulo, se tienen etapas de control donde se requieren voltajes de 5V y 3.3V. La etapa de potencia es conectada directamente a la línea de 127VCA. Se utilizó un convertidor de corriente alterna a corriente directa y reguladores de voltaje para energizar el módulo ESP-01 y la demás electrónica que requieren bajos niveles de voltaje. La etapa de control se conectó a la etapa de potencia, la línea de 127VCA, utilizando un relevador.

Se utiliza un módem Wi-Fi conectado a Internet, al cual se conectan a los módulos. Una vez enlazados, los módulos ESP-01 se conectan al Servidor Web de Amazon para poder interactuar con el servidor configurado para el sistema utilizando el protocolo MQTT.



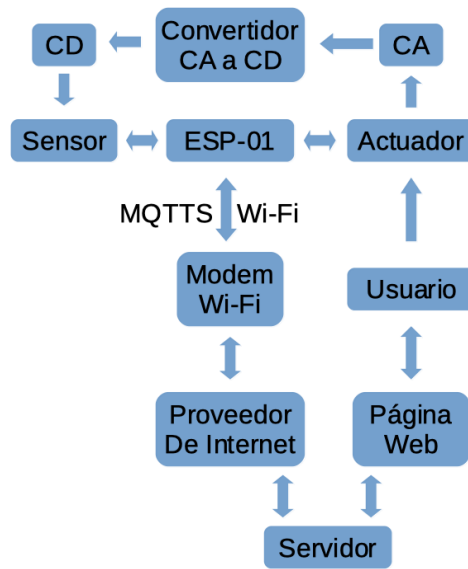


Figura 34: Diagrama de interacción entre los elementos del sistema

Finalmente, para hacer uso del sistema, el usuario deberá conectarse a través de Internet a la página web del sistema, cuyo servidor es proporcionado por *Amazon Web Services*. El diagrama de interacción entre los elementos del sistema se ilustra en la Figura 34.

## 5.2 Diagramas de flujo de información

Una vez conectados los dispositivos, estos tienen interacción entre ellos, intercambiando información dependiendo de las acciones del usuario: al acceder a la página web, el usuario debe proporcionar su nombre y contraseña para registrarse o iniciar sesión, y posteriormente ingresar las redes Wi-Fi de interés que contengan módulos ESP-01. Los módulos a los que tenga acceso el usuario aparecerán en la página web para facilitar su monitoreo o control. Toda la información proporcionada por el usuario es enviada al servidor de Amazon por medio de la página web dentro del modelo cliente-servidor y utilizando el protocolo HTTP. Las peticiones HTTP, del tipo GET y POST son atendidas por el servidor, desarrollado en Node.js, y en función de las peticiones y de la información que proporcione el usuario, se almacena la información en una base de datos MySQL, también administrada desde Node.js, para llevar el control de usuarios, dispositivos, redes Wi-Fi etc. Cuando el usuario solicite cambiar el estado de un módulo ESP-01 se hace uso del protocolo MQTT para enviar el comando solicitado al módulo, dicho mensaje es enviado por el Broker MQTT que también se está ejecutando en AWS. En caso que el usuario cambie manualmente el estado de un dispositivo entonces el módulo ESP-01 envía la actualización a la página web por medio de MQTT para agregar la información en la

base de datos y finalmente el cambio es presentado al usuario a través de la página web. Lo anterior se muestra de manera gráfica en la Figura 35.

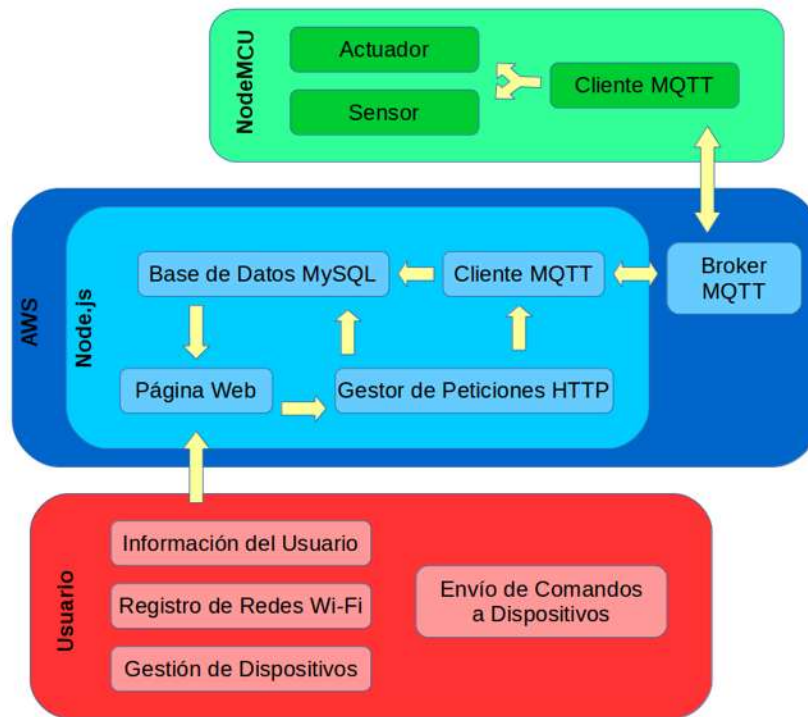


Figura 35: Diagrama de Flujo de información del sistema

Como se puede observar en la Figura 36, el software del sistema consta de cinco partes fundamentales, las cuales interaccionan entre sí teniendo como eje central al servidor NodeJS. El flujo de información es:

- NodeMCU: una vez realizada la conexión a una red Wi-Fi, el módulo ESP-01 se conecta al Broker MQTT y realiza la suscripción a los “topics” de interés. Utilizando el protocolo MQTT, envía la información del módulo al Broker MQTT en la primera vez que se conecte a una red Wi-Fi, cada vez que se modifique el estado del módulo (a manera de retroalimentación y corroborar que el módulo tiene conexión al servidor) y cada vez que el módulo realice alguna acción o lectura de algún sensor. Cuando el módulo no está enviando información a través de MQTT está en modo de bajo consumo a la espera de que el Broker MQTT le envíe un comando.
- Broker MQTT: toda la interacción entre los módulos y el servidor es realizada de manera segura mediante el Broker MQTT, el cual enruta los mensajes entre los módulos y el servidor dependiendo de los “topics” a los cuales estén suscritos los módulos.

## CAPÍTULO 5: DESCRIPCIÓN DEL SISTEMA

- Servidor HTTP: presenta al usuario una interfaz mediante la cual es posible controlar y monitorear los diferentes módulos, además de dar opción a modificar la información del usuario, gestionar las redes Wi-Fi a las que se tiene acceso, eliminar la cuenta del usuario, etc. La interacción entre el usuario y el servidor se realiza a través de la página web utilizando peticiones tipo GET y POST: las peticiones tipo GET son enviadas al servidor HTTP y se procesan mediante NodeJS para obtener la página web que solicite el usuario y enviarla como respuesta, y las peticiones tipo POST envían información que, mediante NodeJS, pueden generar diferentes acciones como modificar información del usuario o los módulos, autenticar el usuario, etc.
- Base de datos MySQL: almacena la información de todos los módulos, las redes Wi-Fi y los usuarios registrados con la finalidad de autenticar usuarios, conocer en todo momento el estado de los módulos, autenticar redes Wi-Fi, etc. La información se almacena y se consulta mediante NodeJS.
- NodeJS: se encarga de configurar el servidor HTTP haciendo uso de la biblioteca *express* (define el puerto a utilizar, recibir y responder peticiones), configurar la base de datos MySQL, autenticar usuarios con los datos almacenados, leer la información de la base de datos para ofrecer datos actualizados en caso de que el usuario lo solicite, actualizar la base de datos cada vez que se realice una modificación o actualización, se agreguen módulos o usuarios. También se suscribe al Broker MQTT para enviar comandos a los módulos y recibe la información de los mismos a través de MQTT.

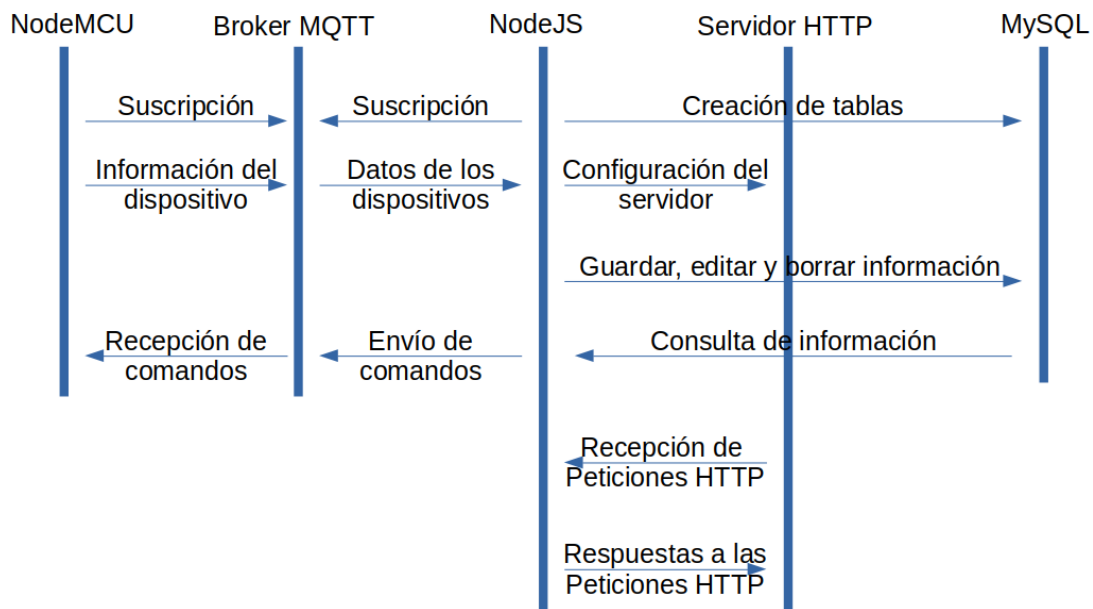


Figura 36: Flujo de Información del sistema

### 5.3 Esquema de la base de datos MySQL

Para el funcionamiento del sistema se utilizó una base de datos llamada “IoT”, en la cual se encuentran tres tablas base: *wifi*, *users* y *devices*.

La tabla “wifi” contiene información de las redes Wi-Fi que tengan conectado y activo uno o más dispositivos. Esta tabla se utiliza para tener el control de las redes Wi-Fi que agregue el usuario; primeramente se corrobora que la red que el usuario requiera agregar exista en esta tabla (lo cual implica que tiene uno o más módulos conectados) y finalmente comprueba que sea correcta la contraseña de la red Wi-Fi ingresada por el usuario al comparar la contraseña almacenada en esta tabla (proporcionada al ejecutar el módulo “*enduser setup*” y conectarse exitosamente a la red Wi-Fi). El contenido administrado en las columnas es:

- *ssid*: contiene el nombre de la red inalámbrica a la cual se conectó un módulo ESP01 o NodeMCU, esto se hace cuando el módulo en cuestión obtiene una dirección IP tras ejecutar correctamente la conexión mediante “*enduser setup*”, descrito en la sección 4.9 Módulo “Enduser Setup” .
- *password*: guarda la contraseña de la red Wi-Fi, la cual es proporcionada tras la conexión exitosa a una red Wi-Fi tras ejecutar “*enduser setup*”

## CAPÍTULO 5: DESCRIPCIÓN DEL SISTEMA

La tabla “*users*” almacena nombres de usuarios y contraseñas, los cuales se utilizan para validar información cuando un usuario solicite registrarse o iniciar sesión; cuando se registra un usuario nuevo primeramente se verifica que el nombre de usuario no exista en la tabla, en dicho caso se procede a registrar el nombre de usuario y la contraseña introducidos. Para iniciar sesión se verifica que la contraseña introducida coincida con la almacenada en la tabla para el nombre de usuario para iniciar sesión. La interfaz web tiene la opción de registrar nuevos usuarios, cambiar el nombre de usuario o la contraseña de un usuario ya registrado. El contenido de esta tabla es dinámico; conforme se agregan, modifican o eliminan usuarios se modifica la información contenida en esta tabla. El contenido administrado en las columnas es:

- *username*: contiene el nombre de usuario, el cual puede contener letras minúsculas, letras mayúsculas, números y caracteres especiales.
- *password*: almacena la contraseña proporcionada por el usuario, la cual también puede contener letras minúsculas, letras mayúsculas, números y caracteres especiales.

La tabla “*devices*” contiene la información de todos los dispositivos (módulos ESP01 y NodeMCU) conectados, actualizando la información del dispositivo cuando este cambie de estado (por ejemplo tras recibir un comando), se cambie el nombre asociado al dispositivo, sea eliminado, etc. El contenido administrado en las columnas es:

- *network*: nombre de la red Wi-Fi a la cual está conectado el dispositivo.
- *device*: especifica si el dispositivo es del tipo sensor, actuador o un dispositivo regulable mediante pwm.
- *mac\_address*: almacena la dirección mac del dispositivo, dicha dirección es única ya que depende del adaptador de red que tenga cada módulo.
- *client\_id*: contiene el nombre asociado al módulo, el cual puede ser modificado por el usuario para ser más familiar e intuitivo. En caso contrario se muestra un nombre por defecto relacionado con el tipo de dispositivo.
- *state*: en esta columna se almacena el estado actual de los dispositivos, sea actuador, sensor o dispositivo regulable.

Conforme se realice el registro, modificación o la eliminación de usuarios, la cantidad de tablas en la base de datos incrementará de forma dinámica agregando una tabla para cada nuevo usuario, la cual se identifica el nombre de usuario y contiene información de las redes Wi-Fi a las que el usuario tiene acceso.

El contenido almacenado en las tablas es del tipo *BOOLEAN* para los actuadores, *DECIMAL* para las lecturas de los sensores y *VARCHAR* para las cadenas de caracteres, además de utilizar campos tipo

*BLOB* para almacenar contraseñas encriptadas haciendo uso del algoritmo AES (*Advanced Encryption Standard*).

Por motivos de seguridad, solo el servidor local tiene acceso a la base de datos, siendo cualquier otra dirección IP inválida para acceder a la información.

### 5.4 Interfaz de usuario

El usuario puede interactuar con el sistema de dos maneras: cambiando físicamente el estado de un dispositivo o haciendo uso de la página web. La página web dispone de varias opciones de uso y configuración, administradas en diferentes páginas dependiendo de la acción a realizar por el usuario:

- **Página de Bienvenida:** esta página da la opción al usuario de registrarse (como nuevo usuario) o de iniciar sesión en caso de que ya esté registrado. En cualquiera de las dos opciones se despliega un formulario para que el usuario pueda ingresar sus datos. En caso de iniciar sesión, se busca en la base de datos si existe un usuario registrado con el nombre proporcionado, de ser así se verifica que la contraseña ingresada sea la correcta para dar acceso al usuario a la página principal, en cualquier otro caso el usuario es notificado del error al iniciar sesión y volver a intentar o podrá regresar a la página de bienvenida. Para registrarse, se muestra un formulario que solicite al usuario escribir su nombre y su contraseña; si el nombre no está registrado y la contraseña cumple los requisitos, entonces se procede a almacenar el usuario en la base de datos y se dirige al usuario a la página principal, véase la Figura 37
- **Página del Usuario:** le da opción al usuario de cerrar su sesión (y posteriormente presentar la página de Bienvenida), eliminar su cuenta, agregar o borrar una red Wi-Fi, o gestionar sus dispositivos, mostrado en la Figura 38. Para eliminar su cuenta es necesario confirmar la contraseña del usuario para proceder, si dicha contraseña es ingresada correctamente se eliminará la cuenta del usuario de la base de datos y se mostrará la página de Bienvenida. Para agregar una red Wi-Fi es necesario que en dicha red existan módulos ESP conectados, en este caso se procede a solicitar el nombre y la contraseña de dicha red; nuevamente, de ser requerido, se notifica al usuario de errores o se da la opción de cancelar la operación. Para borrar una red solamente son necesarias dos condiciones: que la red a borrar haya sido registrada anteriormente, y que el usuario introduzca el nombre de la red. Una vez que se han agregado redes Wi-Fi, se mostrará al usuario todos los módulos ESP disponibles, así como también otra información de interés, por ejemplo el nombre del dispositivo. El usuario puede enviar instrucciones a un módulo siempre y cuando el módulo se encuentre registrado y sea compatible con el comando que el usuario envía, para esto, el usuario tiene a su disposición información acerca de los comandos compatibles con cada tipo de módulo, véase la Figura 39.

- Página para agregar redes Wi-Fi: mediante esta página el usuario puede agregar redes Wi-Fi, siempre y cuando la red de interés ya esté registrada en la base de datos. Además se muestran las redes que tenga registradas el usuario, ilustrado posteriormente en la Figura 55.
- Página para borrar redes Wi-Fi: se muestran mediante una lista todas las redes a las que el usuario tiene acceso y se da la opción al usuario de seleccionar una, varias o todas las redes Wi-Fi, las cuales serán eliminadas de la base de datos y el usuario ya no tendrá acceso a los dispositivos conectados a ellas.
- Página para editar los datos del usuario: en caso que el usuario desee cambiar su nombre o contraseña, dispone de esta página para hacerlo. Es necesario que el usuario se autentifique introduciendo su contraseña. Al guardar cualquier cambio, la información en la base de datos es actualizada.

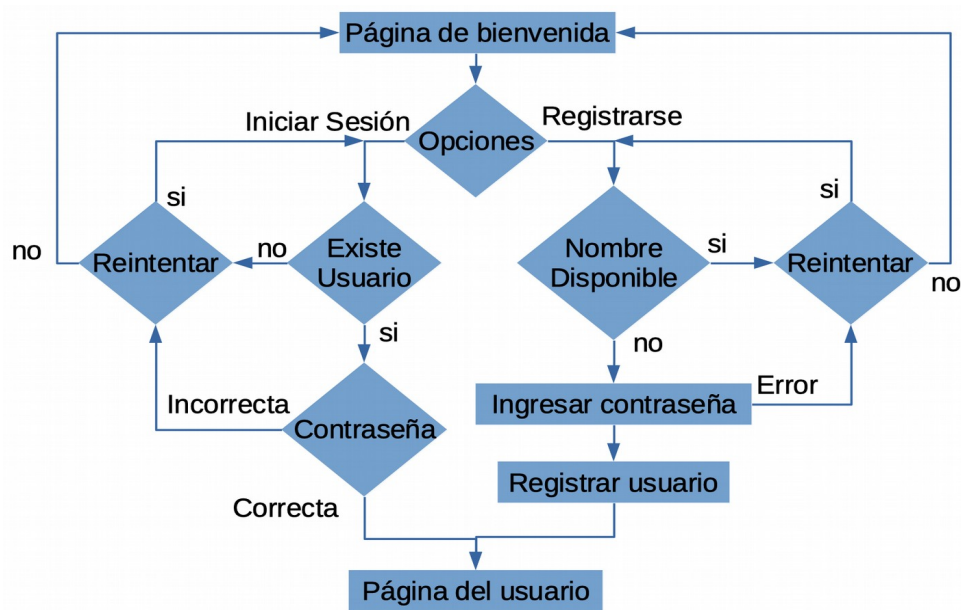


Figura 37: Diagrama de Flujo de la Interfaz de Usuario, Página de Bienvenida

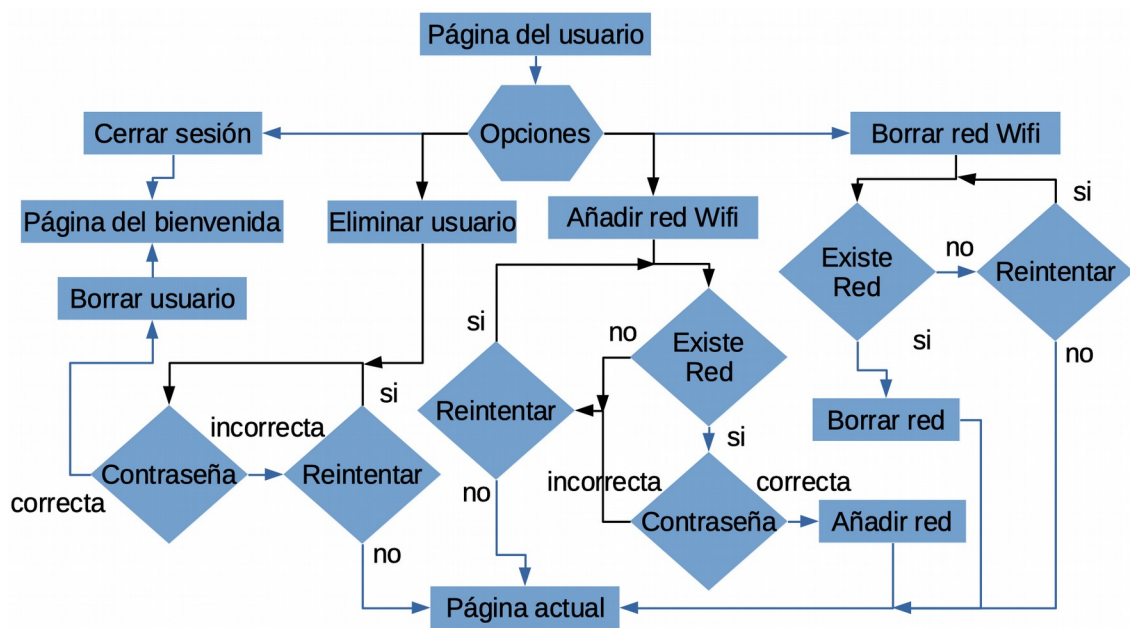


Figura 38: Diagrama de Flujo de la Interfaz de Usuario, Página del Usuario

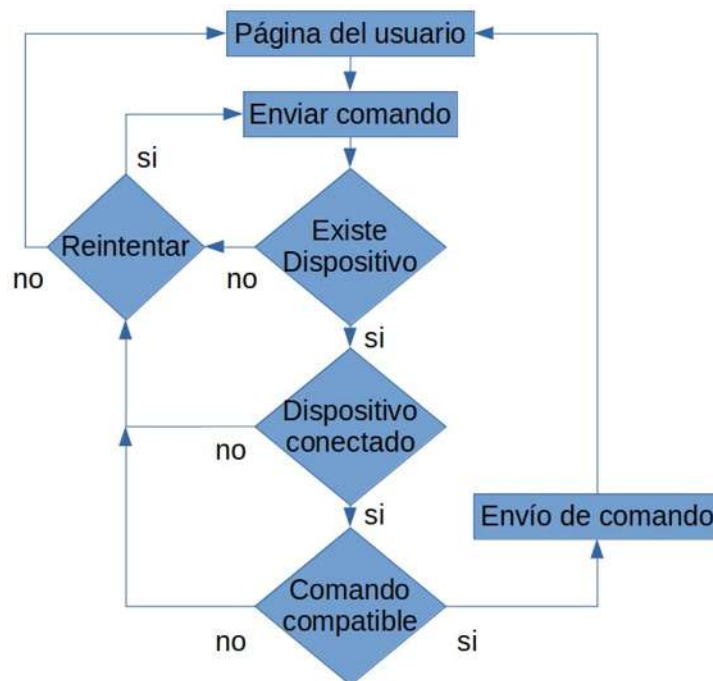


Figura 39: Diagrama de Flujo de la Interfaz de Usuario, Envío de Comandos



## 5.5 Diagramas de funcionamiento del sistema

Como se mencionó, es posible analizar el sistema en cinco partes fundamentales:

1. NodeJS
2. Servidor HTTP
3. Broker MQTT
4. Módulos NodeMCU
5. Base de datos MySQL

Mediante NodeJS se crean las tablas a utilizar en la Base de datos MySQL (usuarios, dispositivos y redes Wi-Fi, inicialmente vacías). Una vez creadas las tablas, la información se actualiza con la conexión o modificación de un dispositivo, cuando un usuario se registra o modifica su información. NodeJS también es responsable de configurar el servidor HTTP y atender a las peticiones que se realicen: las peticiones tipo POST envían información para ser analizada o actualizada en la base de datos, y las peticiones tipo GET tienen como respuesta una página Web. Así mismo, se realiza la conexión con el Broker MQTT para el envío y recepción de información y comandos a los módulos, véase la Figura 40.

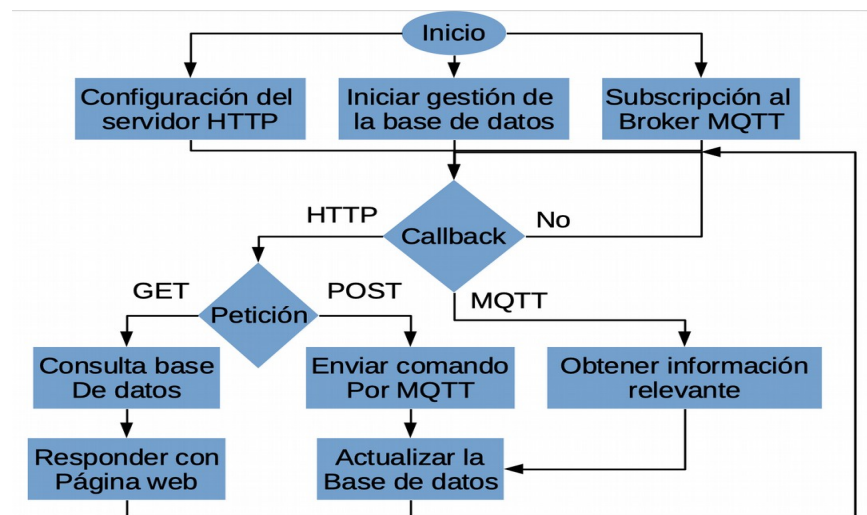


Figura 40: Diagrama de funcionamiento, NodeJS

Una vez configurado el servidor HTTP, se dispone de un conjunto de páginas donde el usuario puede acceder y modificar su información, consultar o modificar el estado de los módulos a los que tiene acceso, agregar o eliminar redes Wi-Fi, etc. Dicha interacción se realiza mediante peticiones tipo GET y POST las cuales son realizadas desde la interfaz de usuario, y se recibe como respuesta una página HTML del servidor; lo anterior se muestra con el diagrama de flujo mostrado en la Figura 41.



Figura 41: Diagrama de funcionamiento, Servidor HTTP

Cuando el usuario modifica su información, nombre de usuario o contraseña, realice la consulta o modificación de algún módulo, se envía una petición tipo POST al servidor, mediante el protocolo MQTT para la publicación y lectura de mensajes de los módulos para leer o actualizar su información. Cuando un dispositivo se agregue a una red Wi-Fi, se haga el cambio de estado de un módulo de manera física o un dispositivo tipo sensor realice una lectura, se envía la información por MQTT, la cual se recibe mediante NodeJS y en caso de ser necesario, se actualiza la información en la base de datos. El diagrama de funcionamiento del Broker MQTT se ilustra con la Figura 42.

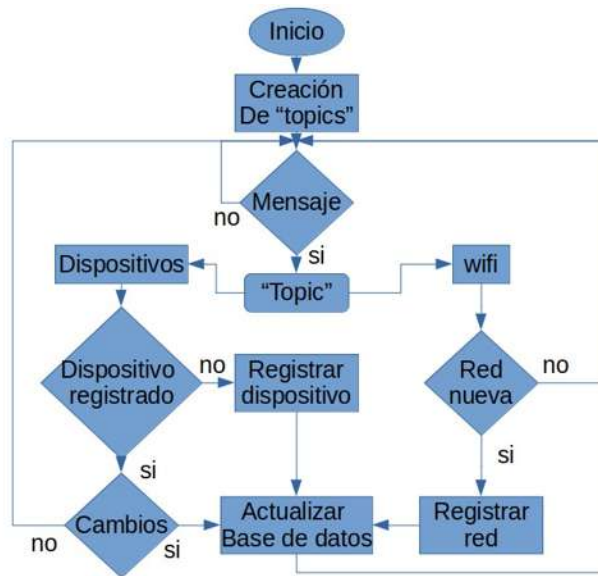


Figura 42: Diagrama de funcionamiento, Broker MQTT

Para realizar la conexión de un módulo a una red Wi-Fi se accede a la página que genera el módulo operando como punto de acceso, una vez seleccionada la red Wi-Fi e introducida correctamente la contraseña entonces la red Wi-Fi le asigna una dirección IP al módulo. Una vez realizada la conexión Wi-Fi, el módulo se suscribe al Broker MQTT y envía información referente al tipo de módulo, su estado actual o alguna otra información, para finalmente entrar en modo de bajo consumo. Cuando el servidor envía un comando por MQTT a alguno de los módulos ESP-01 entonces el módulo verifica si el comando enviado se dirige a él, en dicho caso ejecutará el comando y publicará un mensaje mediante MQTT para indicar al servidor que el módulo está activo y recibió el comando exitosamente y regresa al modo de bajo consumo, en otro caso el módulo permanece en modo de bajo consumo. Los módulos tipo sensor periódicamente publican la información de la medición que realizan mediante MQTT para después continuar en modo de bajo consumo, véase la Figura 43.

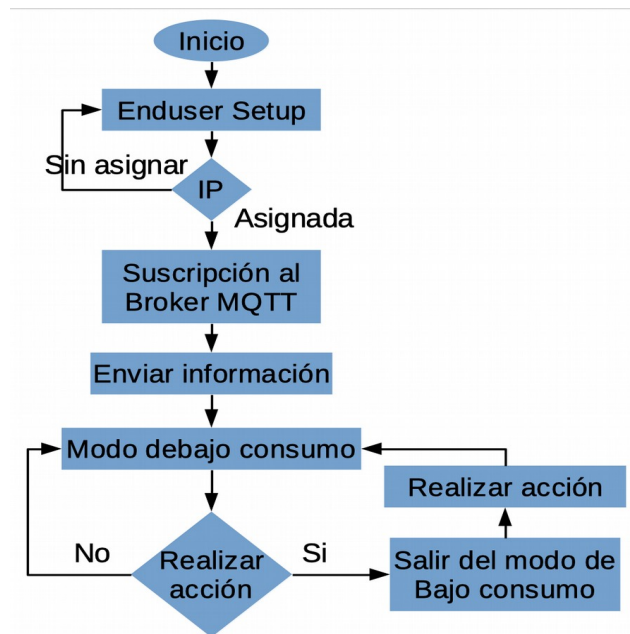


Figura 43: Diagrama de funcionamiento, Módulos ESP

La información referente a la base de datos se aborda a detalle en la sección Error: Reference source not found.

## CAPÍTULO 6: PRUEBAS Y RESULTADOS

### 6.1 Conexión de los módulos a la red Wi-Fi

Para propósitos de este trabajo, después de programar el *enduser setup* [52], introducido en la sección 4.9 Módulo “Enduser Setup” , una vez que se energiza el módulo, este genera una red Wi-Fi llamada "MyPersonalSSID" (el nombre es definido mediante programación) a la cual se debe conectar. Una vez que el usuario está conectado a la red, se abre en el navegador la dirección IP del módulo, véase la Figura 44.

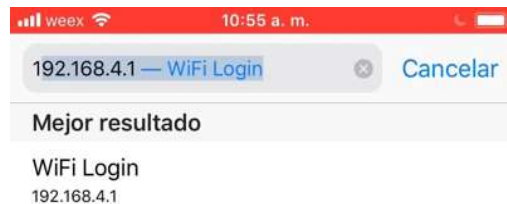


Figura 44: Enduser Setup, IP del Módulo

Una vez conectado a la IP del módulo, se muestra en el navegador un formulario para introducir el nombre de la red y contraseña a la que posteriormente se conecta el módulo ESP-01, véase la Figura 45.

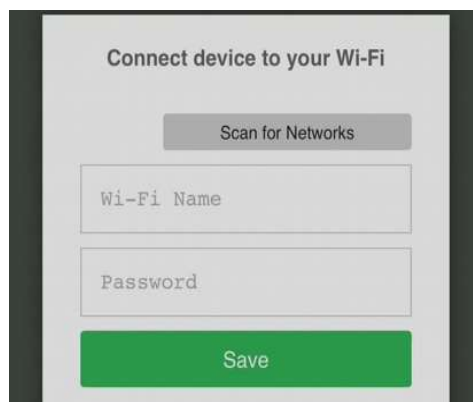


Figura 45: Enduser Setup, Selección de Red Wi-Fi

Al realizar una conexión exitosa a la red Wi-Fi seleccionada, ilustrado con la Figura 46, el módulo guarda en su memoria flash los datos de la red, esto para que la próxima vez que se energice el módulo se conecte automáticamente a la misma red Wi-Fi, esto es muy útil orientado a un producto final.



*Figura 46: Enduser Setup, Conexión realizada*

Otra alternativa al “*enduser setup*” es hacer la conexión de los módulos a la red Wi-Fi mediante WPS (*Wi-Fi Protect Setup*). El método WPS no fue implementado debido a que el módem a disposición no fue compatible con WPS.

Al conectar el dispositivo, el servidor lo agrega a la tabla “*devices*” y registra la red Wi-Fi de la cual se conectó en la tabla “*wifi*”, esto siempre y cuando el dispositivo o la red no hayan sido registrado anteriormente.

## 6.2 Registro como nuevo usuario del sistema en la página Web

Al abrir con el explorador de Internet la página de [www.juangarcia.tk](http://www.juangarcia.tk) se muestra la página de bienvenida, la cual ofrece dos opciones: iniciar sesión en caso de tener una cuenta ya registrada, o registrarse como nuevo usuario, véase la Figura 47.



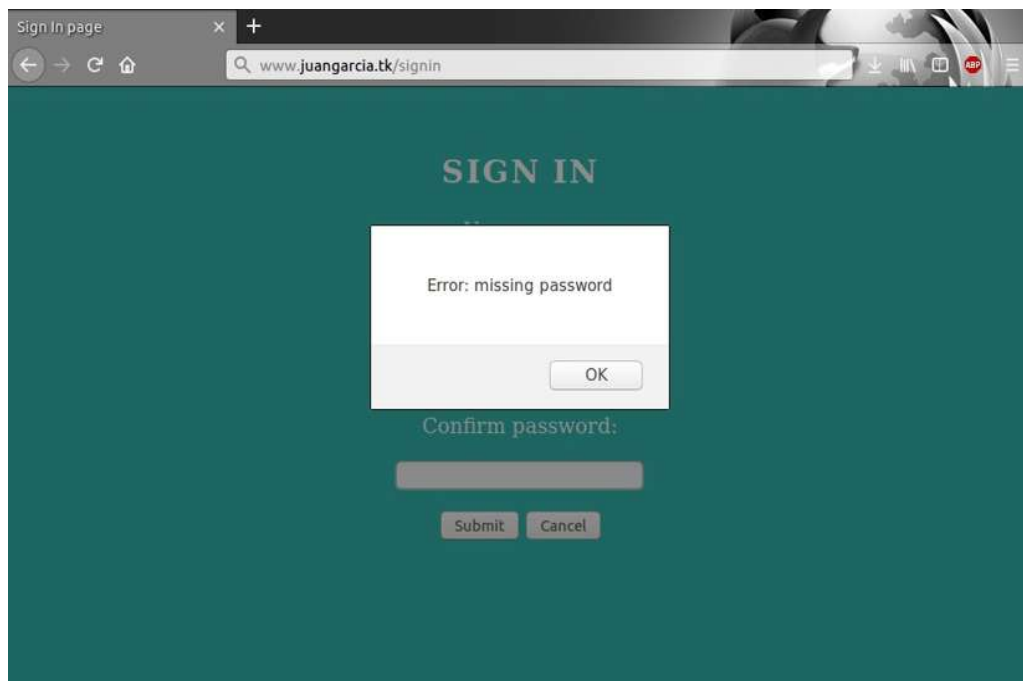
*Figura 47: página web desarrollada para el sistema: www.juangarcia.tk*

Dado que inicialmente no hay usuarios entonces se procede a hacer el registro como nuevo usuario, para lo cual se utiliza el formulario de la página web mostrada en la Figura 48.

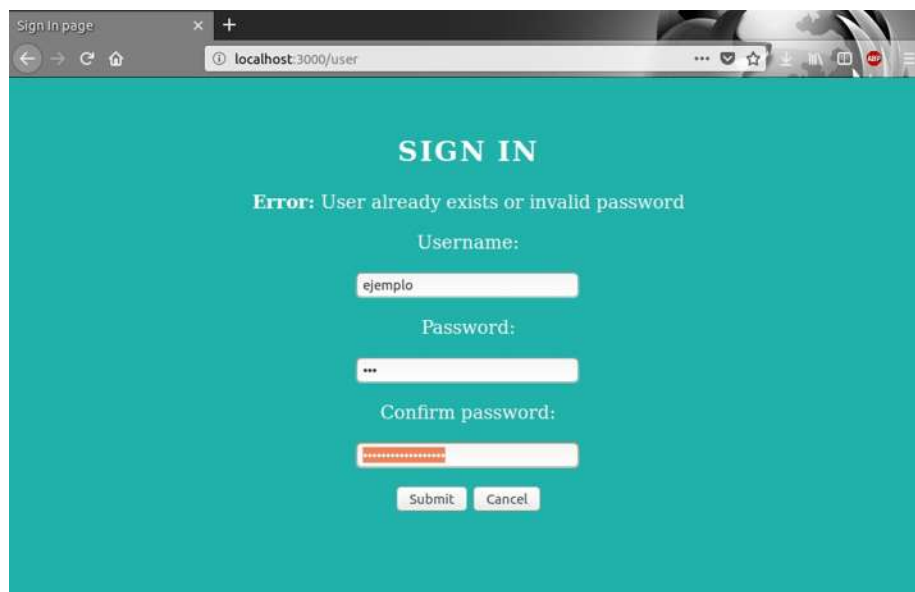
A screenshot of a web browser showing the 'Sign In' page. The browser's address bar displays 'www.juangarcia.tk/signin'. The page has a teal background with the text 'SIGN IN' in white. Below the text are three input fields labeled 'Username:', 'Password:', and 'Confirm password:'. At the bottom are two buttons: 'Submit' and 'Cancel'.

*Figura 48: Página de registro*

Si algún campo de texto está vacío, como el nombre de usuario, contraseña o confirmar contraseña, un ejemplo de ello se muestra en la Figura 49, el usuario será notificado con una alerta en la página. De igual manera cuando existe un error con la contraseña, mostrado en la Figura 50.



*Figura 49: Error de Registro: campo vacío*



*Figura 50: Error de Registro: contraseña no coincide*

Una vez que el registro es exitoso, se mostrará la página del usuario, como se aprecia en la Figura 51, y se agrega el usuario y su contraseña a la base de datos correspondiente, además se crea una tabla



única para el usuario, donde se registrarán las redes Wi-Fi a las que pueda acceder a los dispositivos, inicialmente vacía.



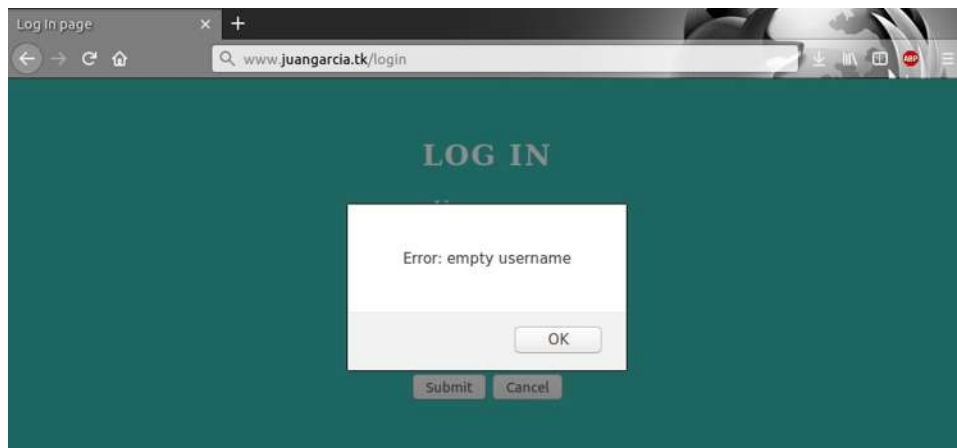
*Figura 51: Página del usuario*

## 6.3 Inicio de sesión

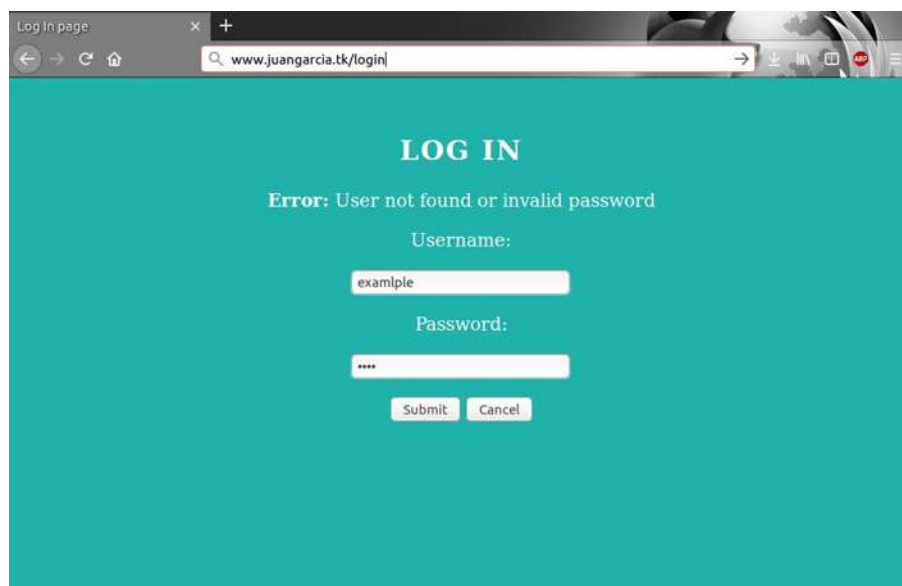
Cuando se selecciona iniciar sesión en la página de bienvenida, se muestra su respectiva página, un ejemplo de ello se muestra en la Figura 52. En caso que exista algún error, por ejemplo un campo vacío, véase la Figura 53, o que el nombre de usuario no esté registrado, error mostrado en la Figura 54, se notifica con un mensaje de error o alerta.



*Figura 52: Página para Iniciar Sesión*



*Figura 53: Error al Iniciar Sesión: campo faltante*



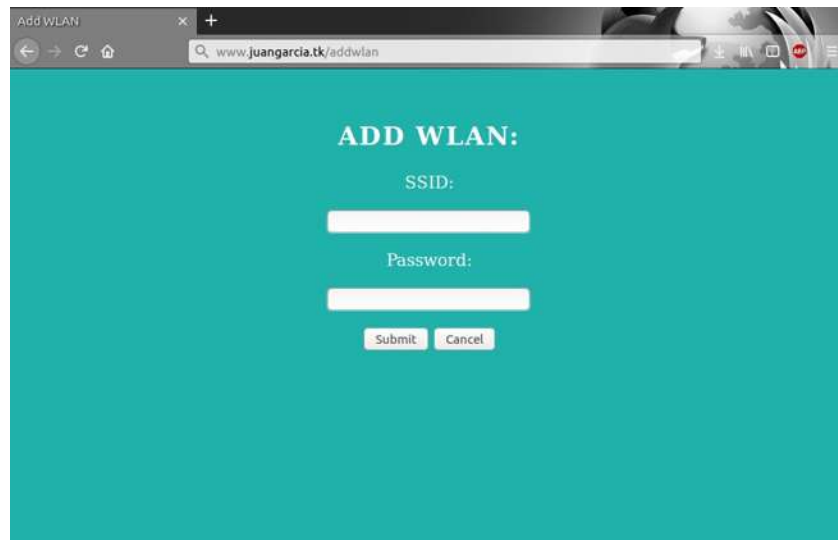
*Figura 54: Error al Iniciar Sesión: usuario no existente*

Una vez que se introduce el nombre de usuario y contraseña correctos, se muestra la página del usuario.

## 6.4 Agregar Redes Wi-Fi y visualizar dispositivos

La tabla del usuario que registra las redes Wi-Fi a las que tiene acceso está inicialmente vacía, se dispone de una pestaña dedicada a registrar redes Wi-Fi, mostrada en la Figura 55. El registro se puede

realizar siempre y cuando el usuario no tenga registrada la red Wi-Fi, y que esta red haya sido anteriormente registrada por el servidor. Igualmente se muestra el error en caso de que existan campos vacíos al realizar la solicitud, véase la Figura 56.

A screenshot of a web browser showing a form titled "ADD WLAN:". The form has a teal background. It contains two input fields: "SSID:" and "Password:". Below the "Password:" field are two buttons: "Submit" and "Cancel". The browser's address bar shows "www.juangarcia.tk/addwlan".

*Figura 55: Página para añadir una Red Wi-Fi*

A screenshot of the same "ADD WLAN:" web form, but with a white error dialog box in the center. The dialog box contains the text "Error: empty password" and an "OK" button. The "Submit" and "Cancel" buttons at the bottom of the form are now disabled and have a greyed-out appearance.

*Figura 56: Error al agregar una red Wi-Fi, Campos vacíos*

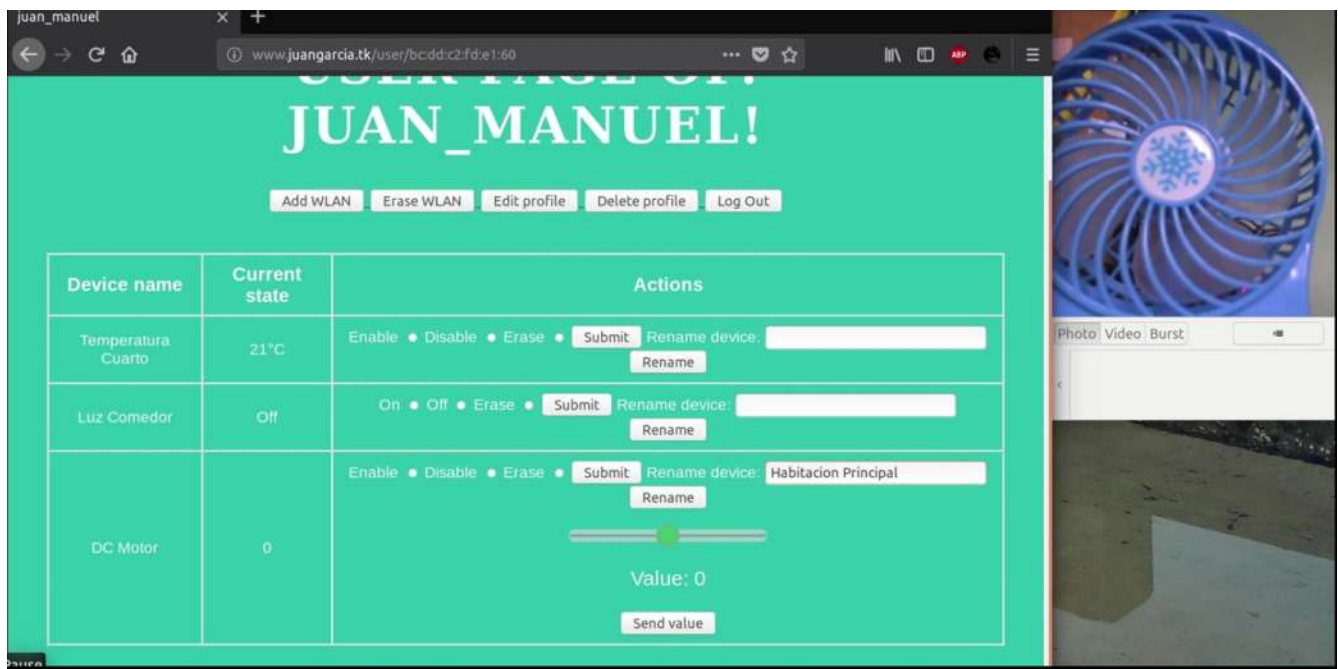
En caso que la solicitud sea procesada con éxito, se agregará la red Wi-Fi a la tabla de redes del usuario. Cuando la tabla de redes de un usuario no está vacía, se muestra la información correspondiente a todos los dispositivos existentes en las redes del usuario, así como dos campos de texto para ingresar la dirección mac de un dispositivo y el comando deseado. En la misma página se informa de los comandos con los que es compatible cada tipo de dispositivo, véase la Figura 57.



*Figura 57: Página del usuario, Lista de dispositivos*

## 6.5 Envío de comandos y lectura de dispositivos

Una vez que el usuario dispone de una lista de dispositivos, puede enviar comandos utilizando las opciones disponibles para cada dispositivo, dichas opciones dependerán del tipo de dispositivo: sensor, actuador o dispositivo regulable, véase la Figura 58.



*Figura 58: Envío del Comando “Rename device”*

Posterior al envío del comando, el dispositivo realiza una acción, dependiendo de la elección del comando del usuario, y actualiza el estado del dispositivo en la base de datos. En la página se refleja el cambio en la tabla que se muestra al usuario, se muestra un ejemplo de cambio del nombre asociado al tercer dispositivo (motor de CD) en la Figura 59.

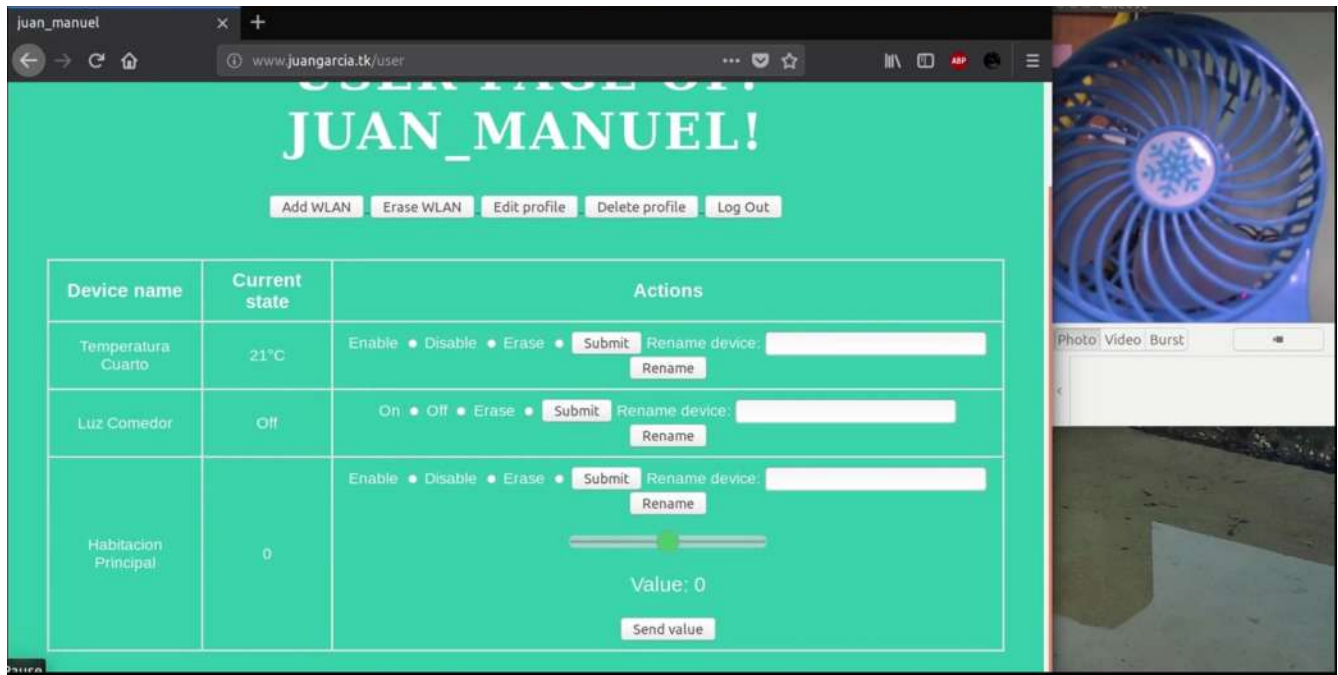


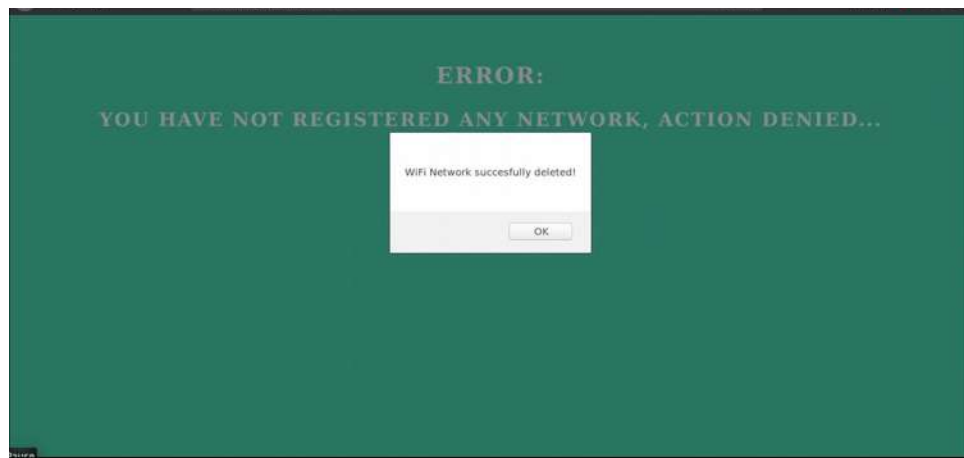
Figura 59: Cambio del nombre del dispositivo a “Habitación principal”

## 6.6 Borrar una Red Wi-Fi

En caso que el usuario desee eliminar los dispositivos de toda una red Wi-Fi, puede seleccionar la opción “Erase WLAN”, la cual contiene un campo para seleccionar la red Wi-Fi que se desea eliminar, véase la Figura 60. La red Wi-Fi es eliminada siempre y cuando esté registrada en la tabla de redes asociadas al usuario. Una vez que el usuario procede a borrar una o varias redes Wi-Fi se notifica que la operación se realizó correctamente, se muestra un ejemplo de ello en la Figura 61, dando la opción de borrar otra(s) red(es) o regresar a la página principal.



*Figura 60: Página para borrar Red Wi-Fi*



*Figura 61: Notificación al borrar red(es) Wi-Fi*

Si el usuario llega a borrar todas las redes Wi-Fi asociadas a él entonces al regresar a la página principal del usuario no tendrá ningún dispositivo para visualizar, mostrado en la Figura 62, por lo tanto, cualquier dispositivo que tuviera anteriormente disponible ya no será incluido en la página. Algunos módulos pueden operar de manera independiente de la página web, por ejemplo los módulos On-off.



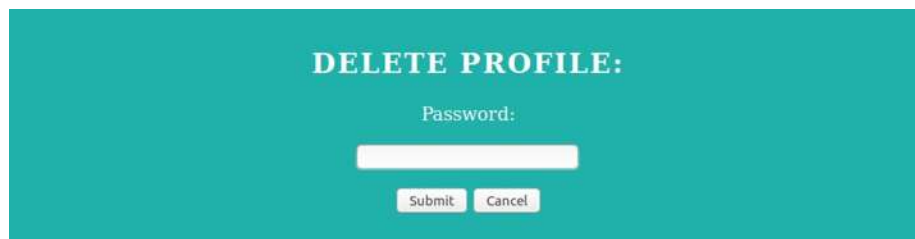
*Figura 62: Página del usuario tras borrar la red Wi-Fi*

## 6.7 Cerrar Sesión

Para cerrar sesión únicamente se tiene que seleccionar el botón *Log Out*, tras hacerlo, se muestra nuevamente la página de bienvenida, véase la Figura 47.

## 6.8 Eliminar cuenta

Para eliminar la cuenta se dispone de una ventana cuyo propósito es verificar que el usuario conoce la contraseña y tiene que introducirla, véase la Figura 63. Nuevamente, de presentarse algún error, como un campo vacío, mostrado en la Figura 64, o que la contraseña no sea la correcta, ejemplo ilustrado en la Figura 65, el usuario es informado.



*Figura 63: Página para Eliminar la Cuenta*





*Figura 64: Error al Borrar la cuenta: Campos vacíos*



*Figura 65: Error al Borrar la Cuenta: Contraseña incorrecta*

## CAPÍTULO 7: CONCLUSIONES

En este capítulo se exponen las conclusiones a las que se llegaron en relación a los objetivos generales y particulares.

### 7.1 Conclusiones generales

Se logró implementar un sistema centralizado de domótica para realizar el control y monitoreo de dispositivos tipo actuador y tipo sensor, para demostrar el funcionamiento se utilizó: un motor de corriente directa, un sensor DHT11 y un relé (el cual se empleó como etapa de potencia para el encendido y apagado de un foco). Todos los módulos tienen como base las tarjetas de desarrollo NodeMCU y ESP-01, código Lua y el software ESPlorer para su programación. Esto implicó el estudio y aprendizaje del lenguaje Lua, el uso del software ESPlorer, *NodeMCU build cloud service* y *esptool.py*.

Para implementar el protocolo MQTTS se utilizó el protocolo MQTT con seguridad SSL. Para esto fue necesario la investigación referente a la implementación de protocolos de seguridad SSL y la dinámica involucrada en la comunicación MQTT: características del protocolo, comandos y opciones de configuración.

Utilizando NodeJS se realizó la programación de un servidor HTTP, esto con la finalidad de hacer la gestión de las peticiones GET y POST realizadas a través del navegador de Internet por parte de los usuarios. Para realizar la implementación del servidor HTTP utilizado en el sistema fue de vital importancia la investigación y aprendizaje de temáticas relacionadas al modelo de comunicación cliente-servidor, web sockets y programación en NodeJS.

Se realizó la programación en NodeJS de un algoritmo para el registro de usuarios (utilizando como campos de registro un nombre y una contraseña), además dar la opción al usuario de modificar la información de registro a través de la interfaz web. Mediante la interfaz el usuario tiene la posibilidad de agregar o eliminar redes Wi-Fi, el usuario puede modificar la cantidad de módulos en la base de datos que contiene la información de los módulos a los que el usuario tiene acceso. De igual manera, el usuario puede visualizar las redes Wi-Fi que tiene agregadas. Por medio una página web se puede visualizar la información de los módulos de manera organizada, sencilla e intuitiva, siendo fácil para el usuario cambiar información de los módulos a los que tiene acceso: cambiar el nombre asociado a un módulo para que sea más fácil al usuario relacionar al módulo con una ubicación, finalidad, etc., modificar el estado actual del módulo, habilitar o deshabilitar módulos. El usuario también puede

eliminar módulos individualmente, lo cual no afecta a ningún otro módulo conectado a la misma red Wi-Fi.

La implementación de la interfaz web implicó el aprendizaje de desarrollo web con bases en HTML, hojas de estilo CSS y programación en JavaScript, esto para hacer páginas web funcionales, con presentación visual aceptable y dinámica con respecto a las diferentes acciones que puede realizar el usuario además, de manera dinámica, se envía información desde el servidor para cambiar la información presentada por medio de la interfaz web.

Para almacenar información se utilizó una base de datos MySQL cuyo acceso es únicamente para el servidor. La gestión de la base de datos se llevó a cabo mediante NodeJS. Para dicho fin fue necesario aprender a realizar la gestión de bases de datos MySQL, sus principales comandos y configuración, además de gestionar la base de datos con NodeJS.

Es importante mencionar que la experiencia de usuario se puede mejorar; simplificar el uso de la interfaz, estilizar más el aspecto visual y agregar funciones tipo IFTTT (“*if this then that*”) para agregar rutinas que se puedan programar y hacer más sencilla la manipulación de grupos de dispositivos o realizar tareas repetitivas.

Algunas de las ventajas de la implementación de este sistema son:

- Reducción de costos: tal como se presentó en el estado del arte, la variedad de sistemas comercializados para monitoreo y control de dispositivos tienen alcance limitado en función de la variedad de sensores y actuadores, sin embargo actualmente son muy costosos, una de las ventajas del sistema presentado es un menor costo.
- Es inclusivo: se puede agregar al sistema una gran variedad de sensores y actuadores, aumentando el campo de aplicación y el número de clientes potenciales. Se pueden desarrollar soluciones a la medida del cliente y aplicación específica.
- Se puede adaptar a cualquier infraestructura: el sistema es lo suficientemente flexible para ser adaptado a edificios tipo casa habitación y aplicaciones industriales con la posibilidad de ser un sistema no invasivo, lo cual implica el aprovechamiento del equipo eléctrico y electrónico e instalaciones físicas y eléctricas.
- Es posible incluir en el sistema soluciones a problemáticas específicas de una persona o empresa; se puede ofrecer una solución a un problema específico, por ejemplo el monitoreo de nivel de agua de un tinaco, medición de la cual dependa el encendido y apagado de una bomba y mostrar el encendido y apagado de la bomba y nivel de agua del tinaco con gráficos en una interfaz web, rastreo de vehículos utilizando GPS aprovechando la cobertura de la red de antenas para telefonía celular a nivel nacional e internacional, monitorear en nivel de gas de tanques estacionarios y mostrar con indicadores gráficos la información en la página web, inclusión de cámaras IP con las cuales proporcionar vigilancia constante mediante *streaming* de

video en tiempo real, etc. Gracias a la versatilidad del sistema es posible diseñar una gran cantidad de aplicaciones.

Algunos de los usos domésticos son: control y monitoreo del encendido y apagado de luminaria, apertura y cierre de puertas, medición de temperatura, control de aire acondicionado y calefacción, control de acceso, control y monitoreo del encendido y apagado de bombas de agua, etc. Algunos de los posibles usos industriales son: aumentar la eficiencia energética promoviendo un uso adaptativo de la luminaria, ventilación, calefacción, etc., promover un uso eficiente de pipas y recursos humanos dedicados a reabastecer tanques contenedores de recursos, por ejemplo gas natural, al hacer el constante monitoreo del nivel, evitando así desplazar personal a tanques o contenedores que no lo requieran, monitorear el correcto funcionamiento de las instalaciones, por ejemplo tuberías de gas, agua, temperatura de operación, etc. al hacer la lectura constante de variables de interés cuya frecuencia de muestreo ayude a realizar mantenimiento preventivo o correctivo de las instalaciones, etc.

Dados los avances de las tecnologías actuales y el amplio desarrollo de sistemas embebidos IoT, el reto principal que se tiene para la implementación y comercialización del sistema propuesto es competir en costo, versatilidad, cobertura, seguridad, escalabilidad, accesibilidad, etc. con otros sistemas similares que actualmente están en el mercado.

Una de las dificultades para la implementación del sistema propuesto es la necesidad de conexión a Internet por medio de Wi-Fi, lo cual descarta la posibilidad de que el sistema se implemente en lugares donde no esté disponible una red Wi-Fi o en contextos en los que el sistema esté en constante movimiento (por ejemplo en un vehículo), pero se podrían agregar otros métodos para realizar el monitoreo y control de los dispositivos como: uso de módulos GSM (*Global System for Mobile Communications*), SigFox, Xbee, etc.

### 7.2 Conclusiones particulares

- El módulo ESP-01 y la tarjeta de desarrollo NodeMCU son ideales para sistemas de domótica de bajo costo, teniendo grandes ventajas como su versatilidad de programación, poco requerimiento de espacio, bajo consumo energético, numerosos ejemplos de uso, bibliotecas disponibles y gran comunidad de desarrolladores activa.
- La programación en Node.js y código Lua simplifica el funcionamiento del sistema, tanto en los clientes (módulos) como del lado del servidor, pues al ser manejado por eventos y *callbacks* el programa se desempeña de manera fluida y no bloqueante, lo cual es de suma importancia dada la naturaleza asíncrona y multi-hilo del sistema.
- El desarrollo de la página web se realizó de manera minimalista en cuanto a la presentación y uso, lo cual se puede mejorar en diversos aspectos utilizando más herramientas propias de HTML, CSS y JavaScript.

## CAPÍTULO 7: CONCLUSIONES

- El estado actual de los dispositivos de manera ideal podría ser mostrado de manera gráfica, por ejemplo: implementando indicadores tipo barra para mostrar valores de lectura de los sensores (como la temperatura), indicadores con imágenes que representen el encendido, apagado y habilitación de dispositivos, etc.
- Dada la versatilidad de los módulos al tener puertos seriales, comunicación I2C, SPI, entre otros, se pueden agregar al sistema una gran variedad de sensores y actuadores o inclusive implementar una red de dispositivos aprovechando su característica de punto de acceso para extender la cobertura provista por el módem que se esté utilizando.
- Es posible ampliar las posibilidades de desarrollo al desarrollar las bibliotecas HTTP propias de los módulos, teniendo acceso a peticiones de tipo GET y POST directamente de los módulos, lo cual podría resultar interesante al momento de hacer más interactiva la página web.
- La implementación del protocolo de seguridad SSL fue de suma importancia ya que el intercambio de información entre los módulos y el servidor por medio de MQTT puede ser interceptado por un tercero, accediendo a información personal y sensible del usuario puesto que los módulos podrían ser implementados para la apertura y cierre de puertas, cocheras, la regulación de calefacción, aire acondicionado, monitoreo de movimiento, temperatura, humedad, vigilancia, etc.

## CAPÍTULO 8: TRABAJOS FUTUROS

Las metas posteriores a la realización de este trabajo tienen relación a la mejora del mismo, teniendo primordial importancia las temáticas referentes a la experiencia de usuario, la optimización de recursos informáticos y hardware:

- Profundizar conocimientos en hojas de estilo CSS, lo cual proporciona la capacidad de mejorar el aspecto visual de la página web que utiliza el usuario para interactúa con el sistema.
- Hacer mayor uso de JavaScript, JSON, Ajax, jQuery, etc., para que la interfaz web sea más interactiva, dinámica y atractiva al usuario, además de mejorar el formato de intercambio de información entre el cliente y el servidor.
- Optimizar el código utilizado en la programación del servidor en NodeJS para mejorar el tiempo de respuesta del servidor, uso de memoria, estabilidad y seguridad del sitio web, etc.
- Mejorar el atractivo visual del sitio web al implementar recursos gráficos con relación al tipo de módulos disponibles y sus posibles estados de operación, por ejemplo, si se utiliza un módulo para controlar el encendido y apagado de un foco, en lugar de mostrar “on” u “off” como estados del módulo en la página web, utilizar una imagen de un foco encendido o apagado dependiendo del estado actual del módulo.
- Agregar una mayor variedad de sensores y actuadores, mediante los cuales poder agregar funciones al sistema: detección de movimiento, detección de humo, transmisión de video, implementación de servomecanismos, motores a pasos, control de motores de corriente alterna, etc.
- Incrementar la funcionalidad del sistema al hacer que los módulos sensores y actuadores puedan interactuar, dando lugar a opciones de configuración del tipo “si hay movimiento en el sensor X, entonces enciende el foco Y”, o “si la temperatura X es mayor que Y grados entonces encender el ventilador Z”, etc.
- Dar posibilidad al usuario de establecer rutinas que especifiquen el estado o valor de uno o más dispositivos. Dichas rutinas podrán ser establecidas para operar cuando el usuario lo indique y con una duración igualmente designada por el usuario. También se podrán establecer rutinas repetitivas; la activación y desactivación de los dispositivos podrá depender de la fecha y/o la hora, con opción a repetirse periódicamente.

## CAPÍTULO 8: TRABAJOS FUTUROS

- Establecer tipos de usuarios con distintos tipos de permisos, esto a manera de control parental, ayudando a personalizar la experiencia de usuario en situaciones donde jerarquizar usuarios sea primordial, por ejemplo: al tener invitados en la casa, limitar los permisos de los hijos, tener un mayor control de empleados en una fábrica o industria, etc.
- Implementar técnicas de mercado para ofrecer de la mejor manera posible el sistema desarrollado en este trabajo al público y con ello lograr la implementación y comercialización del mismo.
- Agregar el campo de correo electrónico a los requisitos para registrarse como nuevo usuario para aumentar la seguridad de la cuenta. Una vez implementado, para completar el registro se enviaría una clave numérica aleatoria al correo del usuario, misma que se tendría que introducir en la interfaz web para validar la cuenta y concluir con el registro. Además, el correo proporcionado por el usuario se utilizaría para dar opción de recuperar la contraseña del usuario o modificarla.
- Agregar al sistema la posibilidad de enviar alertas a usuarios, esto al aprovechar las bibliotecas disponibles utilizadas para el desarrollo del servidor para el envío de correos electrónicos, hacer uso de módulos GSM para el envío de alertas a manera de mensaje de texto, etc.

## REFERENCIAS

### Referencias

- [1] Romero, Vázquez, Castro. (2011). Domótica e Inmótica. España: Alfaomega.
- [2] Gustavo A., Ruiz Robredo. (2001). Electrónica Básica para Ingenieros. Universidad de Cantabria: Dpto. Electrónica y Computadores.
- [3] Alberto Sanfeliu Cortés. (2001). Evolución histórica de la Automatización de los Procesos Industriales. 2018, de IUA, Subsede Funes Sitio web: [http://members.tripod.com/iua\\_informatica.ar/informatica\\_industrial/download/historia.pdf](http://members.tripod.com/iua_informatica.ar/informatica_industrial/download/historia.pdf)
- [4] William Crespo. (2011). Historia de la Automatización Industrial. 2018, de WordPress Sitio web: <https://automatizacionindustrial.wordpress.com/2011/02/17/historia-de-la-automatizacion-industrial/>
- [5] Ramón Piedra Moreno. (1999). Evolución histórica de la Ingeniería de Control. 2018, de Automata Sitio web: <http://automata.cps.unizar.es/regulacionautomatica/historia.PDF>
- [6] Isidro I. Lázaro Castillo. (2015). Ingeniería de sistemas de Control Continuo. México: PROFOCIE.
- [7] Manuel Hidobro, Beatriz Novel, Cristhian Calafat, Eduardo Suller, Adrian Nogales, Carlos Toledano. (2007). La Domótica como Solución de Futuro. Madrid: Asociación Profesional de Empresarios de Instalaciones Eléctricas y Telecomunicaciones de Madrid.
- [8] Watson. (2007). Internet of Things. 2018, de IBM Sitio web: <https://www.ibm.com/internet-of-things/es-es/resources/library/what-is-iot/>
- [9] ISO. (2018). Organización Internacional para la Estandarización. 2018, de ISO Sitio web: [www.iso.org](http://www.iso.org)
- [10] ITU. (2018). Unión Internacional de Telecomunicaciones. 2018, de ITU Sitio web: [www.itu.int](http://www.itu.int)
- [11] CEI. (2018). Comisión Electrónica Internacional. 2018, de CEI Sitio web: [www.iec.ch](http://www.iec.ch)
- [12] José Torres Ventura. (2014). Domótica en México. 2018, de Ciencia y Desarrollo Sitio web: <http://cienciaydesarrollo.mx/?p=articulo&id=306>
- [13] Electrónica Estudio. (2012). ¿Qué es un Microcontrolador?. 2018, de Electrónica Estudio Sitio web: <http://www.electronicaestudio.com/microcontrolador.htm>
- [14] Muhammad H. Rashid. (2004). Electrónica de Potencia. México: Pearson Prentice Hall.
- [15] Zach Wendt. (2017). Relés de estado sólido y relés electromecánicos. 2018, de Arrow Sitio web: <https://www.arrow.com/es-mx/research-and-events/articles/crydom-solid-state-relays-vs-electromechanical-relays>
- [16] Zona Vertigo 2040. (2018). Selenoide. 2018, de Vertigo 2040 Sitio web: <https://vertigo2040.wordpress.com/2011/08/04/selenoide-actuador-electromagnetico/>
- [17] Gilberto Enríquez Harper. (2008). Control de Motores Eléctricos. México: Limusa.
- [18] Subir Kumar Saha. (2008). Introducción a la Robótica. México: Mc. Graw Hill.
- [19] José Dordoigne. (2008). Redes Informáticas. Nociones fundamentales. México: ENI ediciones.
- [20] Marquez. (2018). Cliente-Servidor. 2018, de Catarina Sitio web: [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/marquez\\_a\\_bm/capitulo5.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf)
- [21] Arman Danesh. (1996). Aprendiendo JavaScript. México: pHH.



- [22] Acerca de HTML. (2018). ¿Qué es HTML y para qué sirve?. 2018, de Acerca de HTML Sitio web: <http://www.acercadehtml.com/manual-html/que-es-html.html>
- [23] Javier Eguiluz. (2008). Introducción a CSS. México: Creative.
- [24] WordPress. (2008). Bases de Datos. 2018, de Fundamentos Informáticos Sitio web: <https://fundamentosinformaticosjl.wordpress.com/category/base-de-datos/>
- [25] Ieshaira. (2006). Puntos de Acceso. 2008, de SMR Sitio web: <https://smr.iesharia.org/wiki/doku.php/src:ut7:proyectos:ap>
- [26] Ivan Ahuja, Rodrigo Arriaga & Leslie Medellín & Angelica Mendoza. (2010). Redes WLAN con Access Point. México: Instituto Politécnico Nacional.
- [27] Instant SSL. (2018). What is SSL?. 2018, de Instant SSL Sitio web: [www.instantssl.com](http://www.instantssl.com)
- [28] Redalia. (2018). Qué es el protocolo SSL/TLS. 2018, de Redalia Sitio web: [www.redalia.es](http://www.redalia.es)
- [29] Certsuperior Digicert. (2018). SSL. 2018, de Certsuperior Sitio web: <https://certsuperior.com>
- [30] Michael Yuan. (2018). Conociendo MQTT. 2018, de IBM Sitio web: <https://www.ibm.com/developerworks/ssa/library/iot-mqtt-why-good-for-iot/index.html>
- [31] Google. (2018). Google Home Help. 2018, de Gogle Sitio web: [https://support.google.com/googlehome/answer/7130274?hl=en&ref\\_topic=7195641](https://support.google.com/googlehome/answer/7130274?hl=en&ref_topic=7195641)
- [32] Amazon. (2018). Amazon Alexa. 2018, de Amazon Sitio web: <https://developer.amazon.com/alexa>
- [33] Mark Zuckerberg. (2018). Jarvis IA. 2018, de Facebook Sitio web: <https://www.facebook.com/notes/mark-zuckerberg/building-jarvis/10154361492931634/>
- [34] Kick Starter. (2018). MyCroft Mark II: The Open Voice Assistant. 2018, de Kick Starter Sitio web: <https://www.kickstarter.com/projects/aiforeveryone/mycroft-mark-ii-the-open-voice-assistant?ref=390850&token=acb30e8f>
- [35] Apple Inc.. (2018). Apple HomeKit. 2018, de Apple Inc. Sitio web: <https://www.apple.com/mx/ios/home/>
- [36] Espressif Systems IOT Team. (2015). ESP8266EX Datasheet. 2018, de Espressif Systems Sitio web: [https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266\\_\\_Datasheet\\_\\_EN\\_v4.3.pdf](https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf)
- [37] Luis del Valle Hernández. (2018). ESP8266. 2018, de Programar Facil Sitio web: <https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/>
- [38] AOSONG. (2018). Temperature and humidity module DHT11 Product Manual. 2018, de AOSONG Sitio web: <http://akizukidenshi.com/download/ds/aosong/DHT11.pdf>
- [39] PUC RIO. (2018). Lenguaje LUA. 2019, de PUC RIO Sitio web: <https://www.lua.org/>
- [40] NodeJS. (2018). Node.js. 2018, de NodeJS Sitio web: <https://nodejs.org/>
- [41] NPM Inc. (2018). npm. 2018, de NPM Inc (US) Sitio web: <https://docs.npmjs.com>
- [42] Eclipse. (2018). Eclipse Mosquitto. 2018, de Eclipse Sitio web: <https://mosquitto.org>
- [43] Cerbot. (2018). Cerbot. 2018, de Cerbot Sitio web: <https://cerbot.eff.org>
- [44] ORACLE Corporation. (2018). MySQL Documentation. 2018, de ORACLE Corporation Sitio web: <https://dev.mysql.com/doc/>
- [45] Amazon. (2018). Amazon Web Server. 2018, de Amazon Sitio web: <https://aws.amazon.com/>
- [46] NPMJS. (2018). npm. 2018, de NPMJS Sitio web: <https://www.npmjs.com>
- [47] HTML.com. (2018). HTML. 2018, de HTML.com Sitio web: <https://html.com/>
- [48] HTML.com. (2018). Learn CSS Today The Easy Way. 2018, de HTML.com Sitio web: <https://html.com/css/>
- [49] HTML.com. (2018). JavaScript. 2018, de HTML.com Sitio web: <https://html.com/javascript/>

## REFERENCIAS

- [50] Daninz. (2018). Generalidades del protocolo HTTP. 2018, de Developer Mozilla Sitio web: <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>
- [51] w3schools.com. (2018). HTTP Request Methods. 2018, de w3schools.com Sitio web: [https://www.w3schools.com/tags/ref\\_httpmethods.asp](https://www.w3schools.com/tags/ref_httpmethods.asp)
- [52] NodeMCU Readthedocs. (2018). Enduser Setup Module. 2018, de NodeMCU Readthedocs Sitio web: [https://nodemcu.readthedocs.io/en/master/en/modules/enduser-setup/#enduser\\_setupstop](https://nodemcu.readthedocs.io/en/master/en/modules/enduser-setup/#enduser_setupstop)

## APÉNDICES

### Apéndice A: Base de Datos MySQL, Comandos de interés.

MySQL ofrece una gran variedad de comandos y opciones dedicados a administrar bases de datos, algunos de los cuales son [44]:

Tabla 7: Comandos MySQL

| Comando                                                                                                                       | Descripción                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mysql [-h <host>] -u <user> -p[password] [database]                                                                           | <p>Inicia la conexión con el servidor MySQL. Recibe distintos parámetros:</p> <ul style="list-style-type: none"> <li>[-h &lt;host&gt;]: dirección IP o dominio en donde se encuentra el servidor MySQL. En caso de no especificar, se hará la conexión a 'localhost'.</li> <li>-u &lt;user&gt;: usuario que tiene permisos de lectura y/o escritura a la base de datos.</li> <li>-p[password]: contraseña para el usuario especificado. En caso de no introducir la contraseña en el comando, posteriormente será solicitada.</li> <li>[database]: si la base de datos de interés se encuentra en existencia en el servidor, entonces se puede especificar para ser utilizada.</li> </ul> |
| create database <name>;                                                                                                       | Crea una base de datos llamada '<name>'                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| use <database>;                                                                                                               | Especifica la base de datos a administrar.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| drop database <database>;                                                                                                     | Elimina la base de datos llamada '<database>'                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| select database();                                                                                                            | Informa cuál es la base de datos que se esté utilizando.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| show tables;                                                                                                                  | Muestra la(s) tabla(s) asociada(s) con la base de datos en uso.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| describe <table>;                                                                                                             | Muestra las características, por ejemplo: las columnas y tipo de datos de las columnas, de una tabla específica de la base de datos actual.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| create [temporary] table [if not exist] <name><br>[(create_definition,...)]<br>[table_options]<br>[partition_options] [ignore | <p>Crea una tabla, la cual será asociada a la base de datos previamente seleccionada. Algunos de los parámetros que puede recibir son:</p> <ul style="list-style-type: none"> <li>[temporary]: la tabla será borrada al finalizar la sesión shell de MySQL.</li> <li>[if not exist]: evita crear una tabla con un nombre que ya</li> </ul>                                                                                                                                                                                                                                                                                                                                                |

| Comando                                                  | Descripción                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| replace] [as] query_expression;                          | <p>haya sido utilizado.</p> <ul style="list-style-type: none"> <li>• [(create_definition,...)]: especifica las características y tipo de datos de las columnas que tendrá la tabla.</li> <li>• [table_options]: especifica diversas características de la tabla: compresión, auto incremento, encriptación, máximo o mínimo de renglones (datos ingresados), contraseña de acceso a la tabla, etc.</li> <li>• [partition_options]: dimensionamiento de la tabla.</li> <li>• [ignore   replace]: especifica qué hacer con los renglones duplicados producidos al copiar una tabla.</li> <li>• [as] query_expression: crea una tabla a partir de otra.</li> </ul> <p>Más información en: <a href="https://dev.mysql.com/doc/refman/8.0/en/create-table.html">https://dev.mysql.com/doc/refman/8.0/en/create-table.html</a></p> |
| load data local infile '<file.txt>' into table <table>;  | Introduce valores a una tabla a partir del contenido de un archivo de texto.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| insert into <table> values (value1,...);                 | Introduce valores a una tabla.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| select * from <table> where <conditions>;                | Selecciona todos los renglones de una tabla que cumplan las condiciones especificadas en '<conditions>'.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| select <what_to_select> from <table> where <conditions>; | Selecciona los renglones de las columnas especificadas por '<what_to_select>' y que cumplan las condiciones especificadas por '<conditions>'.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## Apéndice B: Renta y configuración de un servidor EC2 de AWS

Una instancia EC2 es un servidor virtual que puede ser solicitado por los usuarios de *Amazon Web Services* para desarrollar diversas aplicaciones. Las características del servidor se pueden configurar de diferentes maneras, dando la flexibilidad al usuario para definir la potencia de la unidad de procesamiento de la instancia EC2, la cantidad de memoria, el sistema operativo, etc., además de proporcionar al usuario el control del encendido, apagado y activación de la instancia.

En esta sección se explica el lanzamiento de una instancia EC2 de AWS y su configuración básica, dando por hecho que se tiene una cuneta registrada en AWS.

Después de iniciar sesión, se hace clic en la opción “Services” en la esquina superior izquierda de la página. Esto desplegará diversas opciones, siendo la opción de interés *Compute>EC2*, véase la Figura 66.

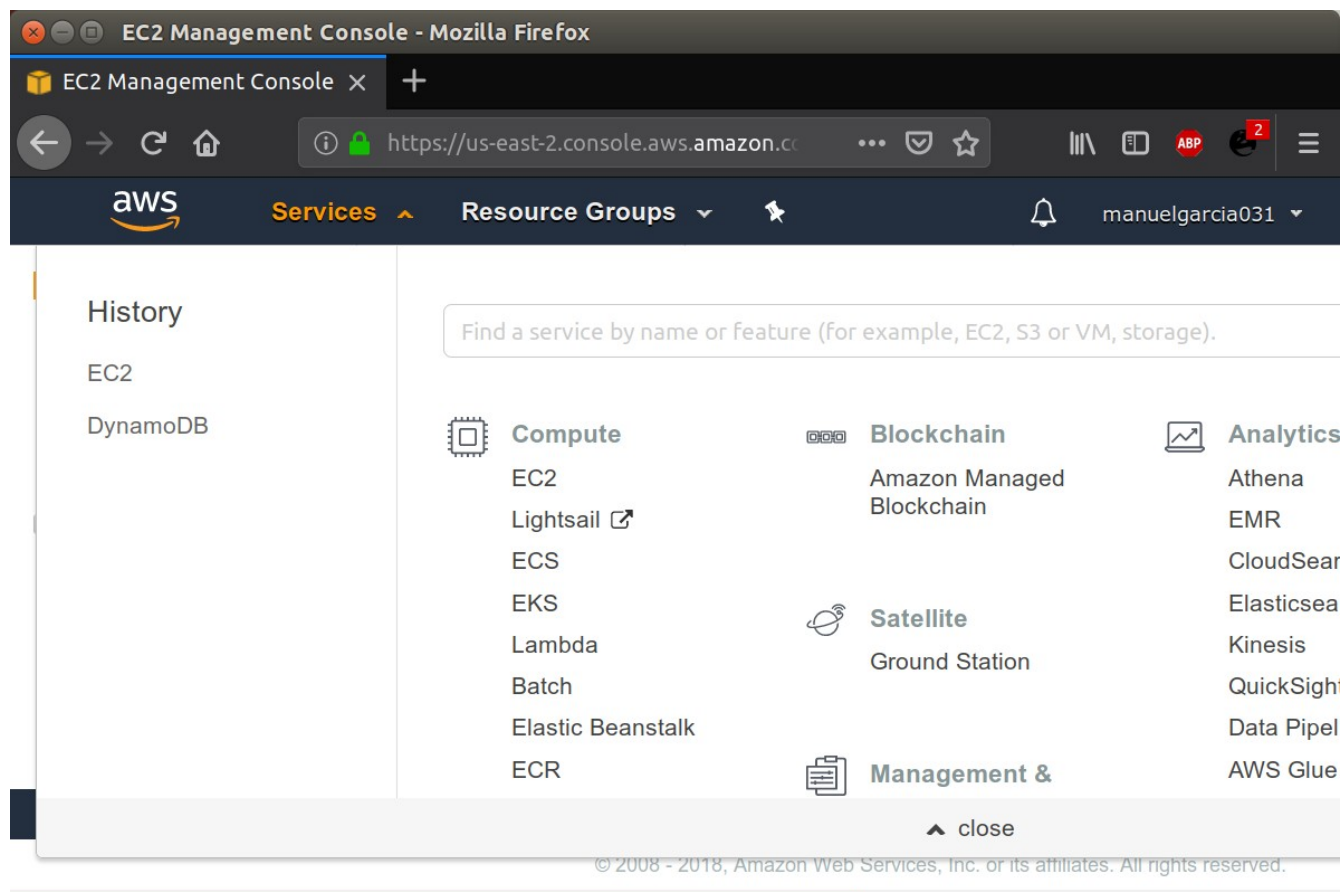
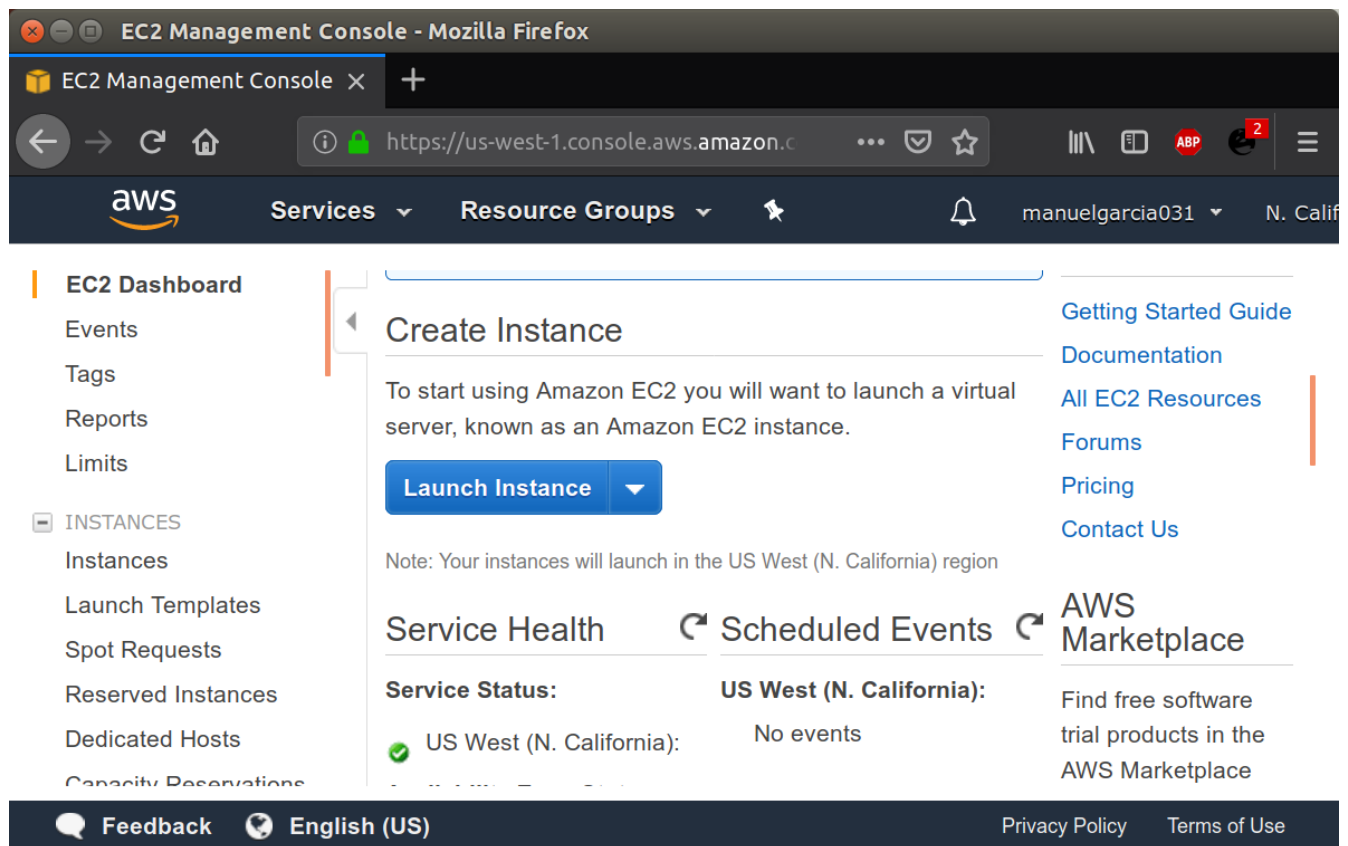


Figura 66: Services>Compute>EC2, AWS

Esto desplegará el menú de instancias EC2 de Amazon clasificadas por región. Se puede seleccionar la región en la cual la instancia a crear será lanzada. Se procede a seleccionar la opción “*Launch instance*”, véase la Figura 67.



© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

*Figura 67: EC2 Dashboard>Create Instance*

Se desplegará una lista de AMI's (Amazon Machine Image) disponibles, de las cuales se seleccionó Ubuntu Server, véase la Figura 68.

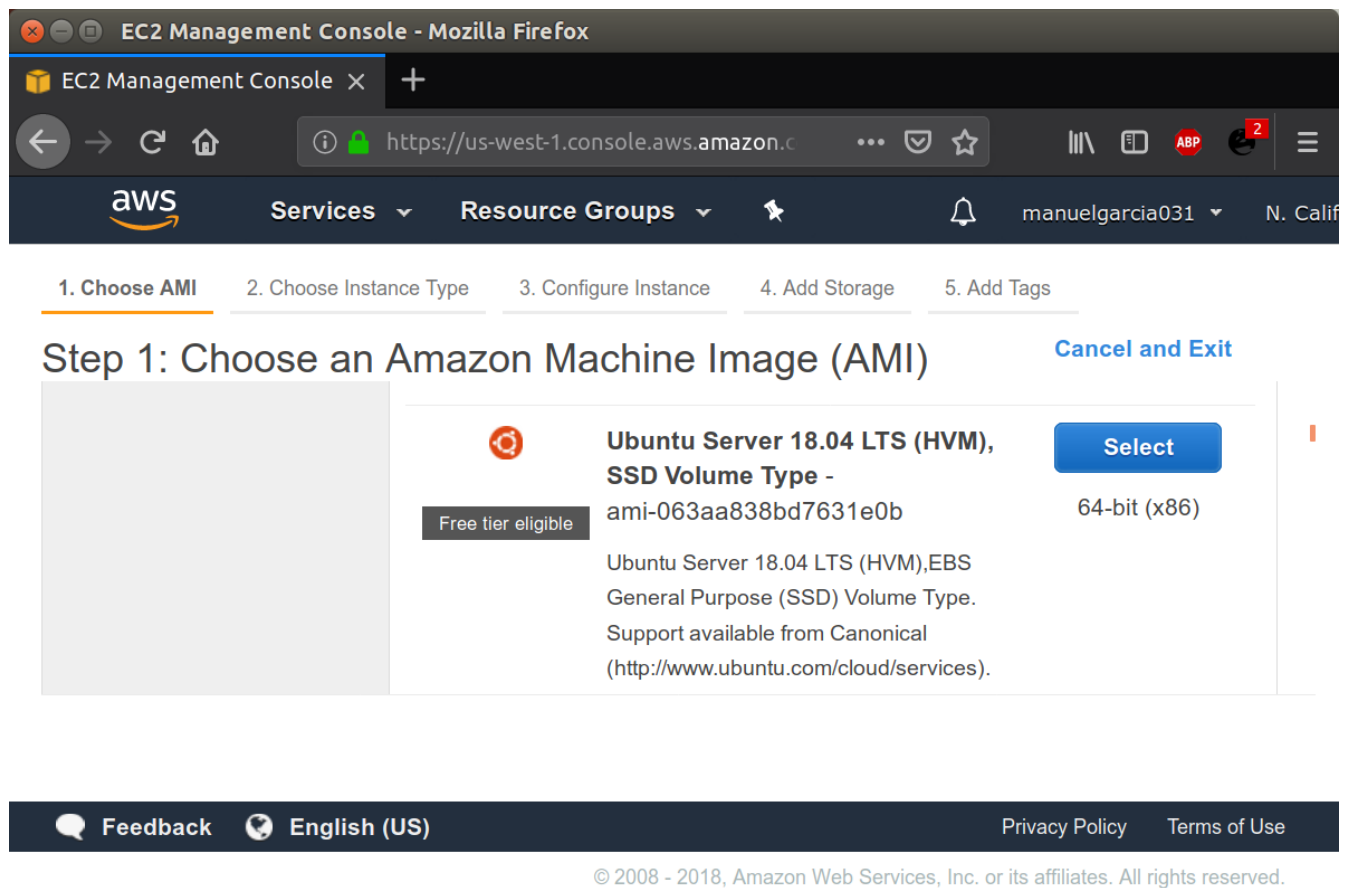
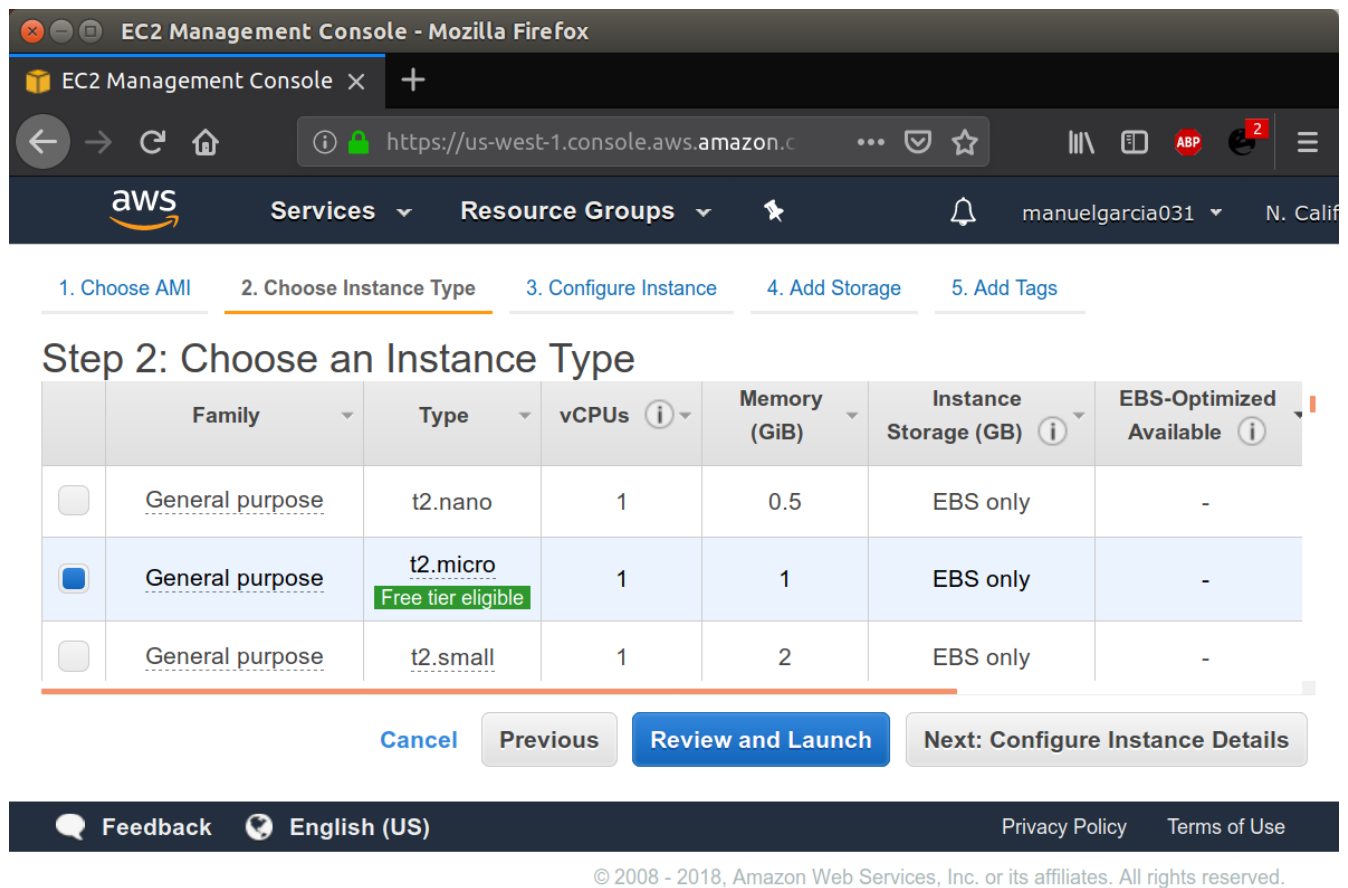


Figura 68: Ubuntu Server AMI

Una vez seleccionada la AMI de Ubuntu Server de propósito general, se despliegan en una lista las posibles configuraciones de CPU, GPU (*Graphics Processing Unit*), memoria y otras características con las que se puede lanzar la instancia. En este caso se seleccionó la opción marcada como “Free tier eligible”, véase la Figura 69.



*Figura 69: Tipos de Instancias EC2*

Si se desea, en el apartado de “*Configure Instance Details*” se pueden configurar opciones red, número de instancias, por nombrar algunas, véase la Figura 70.



EC2 Management Console - Mozilla Firefox

EC2 Management Console x +

https://us-west-1.console.aws.amazon.c

aws Services Resource Groups

manuelgarcia031 N. Calif

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

**Number of instances** ⓘ

1 [Launch into Auto Scaling Group](#) ⓘ

**Purchasing option** ⓘ

☐ Request Spot instances

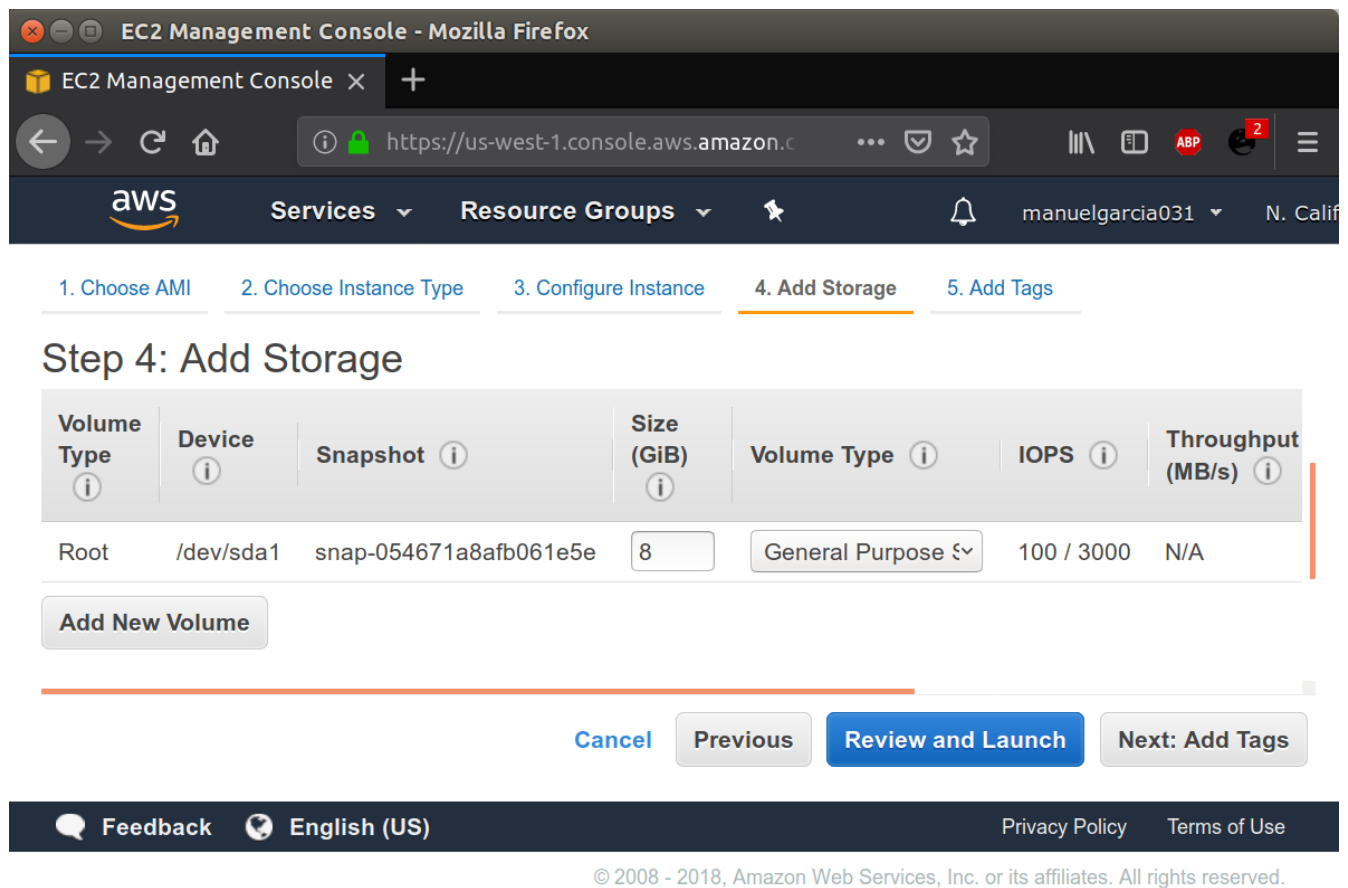
[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

[Feedback](#) [English \(US\)](#) [Privacy Policy](#) [Terms of Use](#)

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

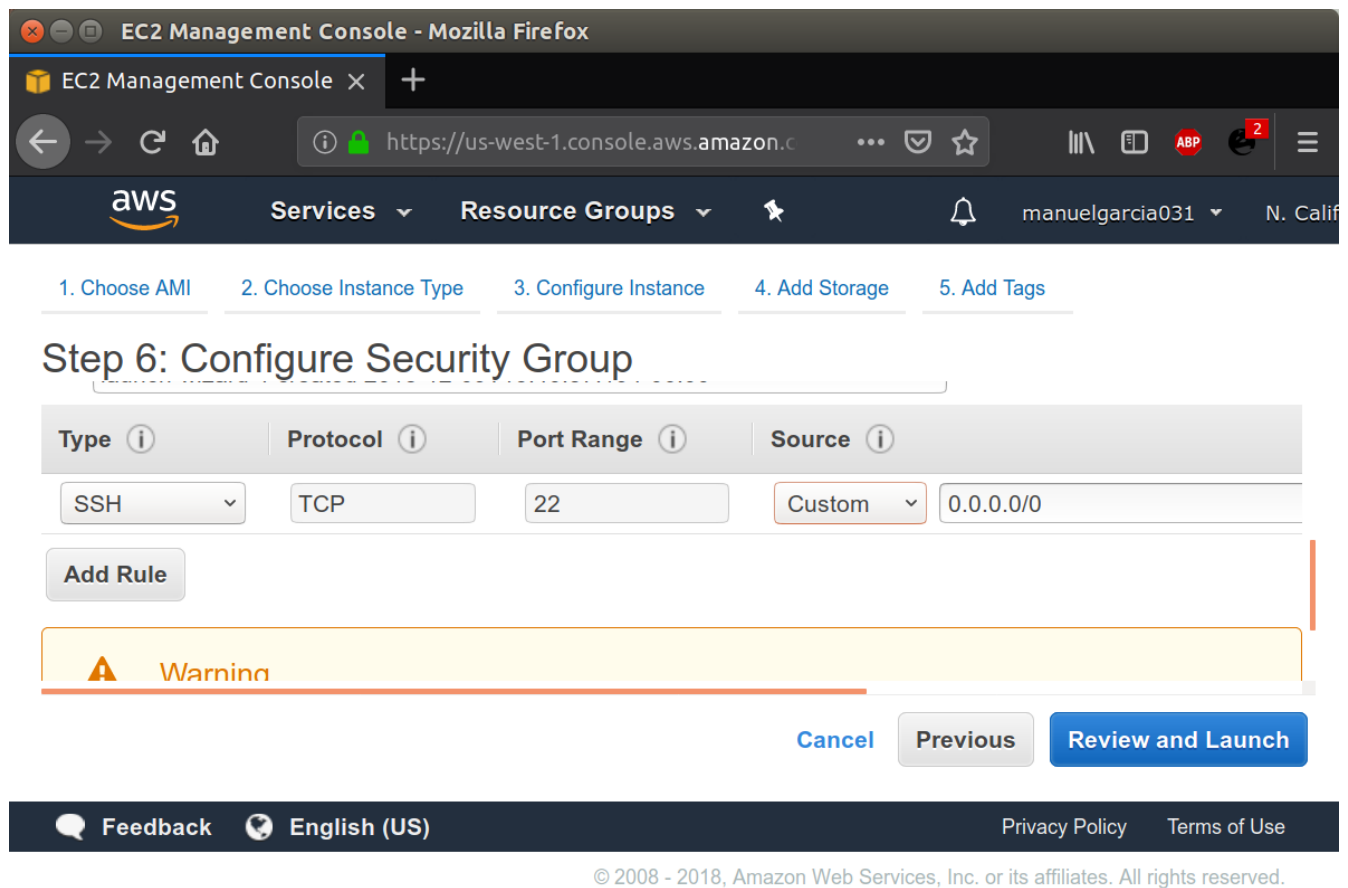
*Figura 70: Configuración de la Instancia EC2*

Es posible seleccionar la cantidad y tipo de almacenamiento (de propósito general o almacenamiento magnético) de la instancia, siendo un máximo de 30GB de almacenamiento para el tipo de instancia utilizada, véase la Figura 71.



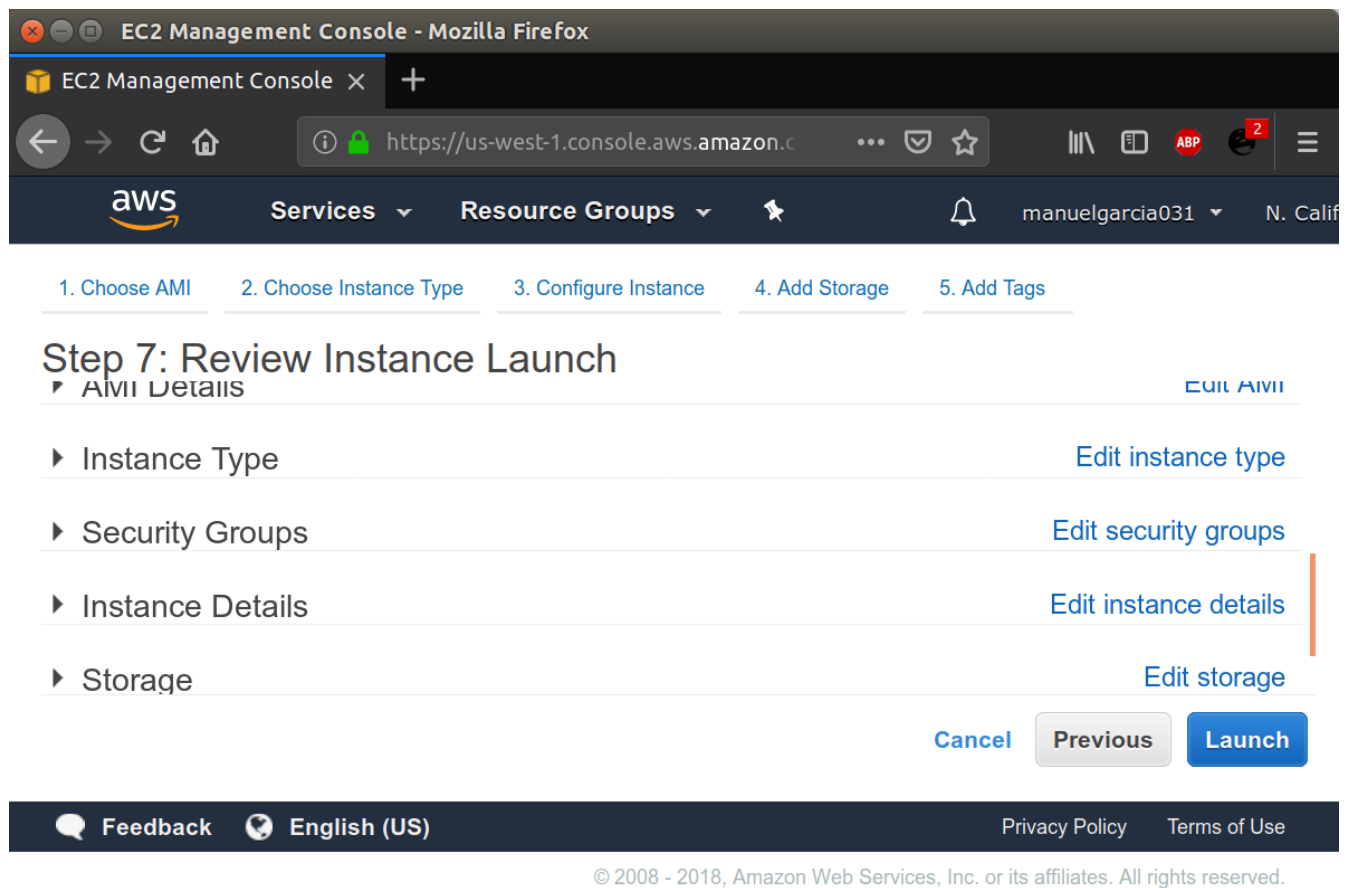
*Figura 71: Configuración de Almacenamiento de la Instancia EC2*

Es imprescindible especificar en el grupo de seguridad asociado a la instancia EC2 los puertos y protocolos a utilizar, así como también delimitar el acceso, según sea el caso, a direcciones IP específicas; es recomendable que el acceso a los puertos relevantes como el 22, SSH, sea restringido para uso de direcciones IP conocidas y aprobadas por el usuario, véase la Figura 72.



*Figura 72: Grupo de Seguridad de la instancia EC2*

Antes de lanzar la instancia EC2, se presenta al usuario un resumen con las características y configuraciones escogidas por el usuario, véase la Figura 73.



*Figura 73: Resumen de la Instancia EC2 a lanzar*

Para poder ser lanzada la instancia, por motivos de seguridad, se asocia una llave a un archivo que solamente se puede descargar una vez. Esta llave es utilizada para poder conectarse por medio de SSH a la instancia.

Una vez se haya lanzado la instancia, esta será mostrada en el apartado de la región en la cual se lanzó. Esta sección muestra información acerca de las características principales y el estado actual de la instancia, véase la Figura 74.

EC2 Management Console

Services Resource Groups

manuelgarcia031 Ohio Support

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Launch Templates

Spot Requests

Reserved Instances

Dedicated Hosts

Capacity Reservations

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

Create Volume Actions

Filter by tags and attributes or search by keyword

| Name | Volume ID             | Size  | Volume Type | IOPS | Snapshot               | Created                           | Availability Zone | State  | Alarm Status |
|------|-----------------------|-------|-------------|------|------------------------|-----------------------------------|-------------------|--------|--------------|
|      | vol-01ac2695627afbcf4 | 8 GiB | gp2         | 100  | snap-015e0c9bfb72bf22e | July 11, 2018 at 8:39:13 PM UTC-5 | us-east-2b        | in-use | None         |

Volumes: vol-01ac2695627afbcf4

Description Status Checks Monitoring Tags

Volume ID: vol-01ac2695627afbcf4

Size: 8 GiB

Created: July 11, 2018 at 8:39:13 PM UTC-5

State: in-use

Attachment information: i-01f64650e7c44691d (Domotics) /dev/sda1 (attached)

Volume type: gp2

Product codes: marketplace:

IOPS: 100

Alarm status: None

Snapshot: snap-015e0c9bfb72bf22e

Availability Zone: us-east-2b

Encrypted: Not Encrypted

KMS Key ID

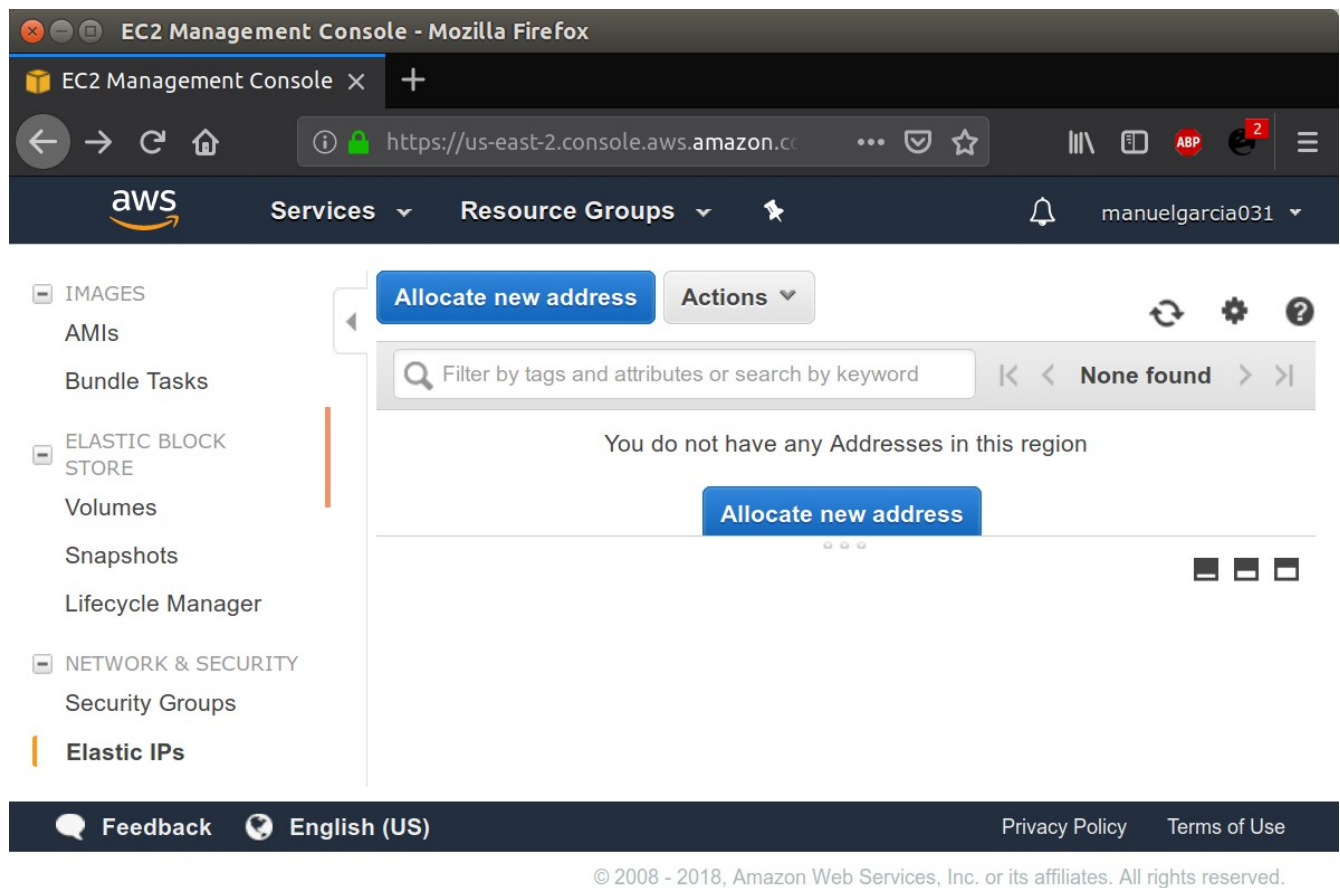
KMS Key Aliases

KMS Key ARN

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

*Figura 74: EC2 Dashboard tras lanzar la instancia EC2*

Para evitar el cambio de IP de la instancia EC2 al reiniciar el servidor virtual, se procede a configurar una IP elástica para que sea asociada a éste. En el apartado “Network & Security” se selecciona la opción “Elastic Ips”, véase la Figura 75.



*Figura 75: Configuración de una IP elástica, AWS*

De igual manera, todas las IP's elásticas se muestran en el apartado de Redes y seguridad,

The screenshot shows the AWS Management Console interface for Elastic IPs. The left sidebar lists navigation options: IMAGES (AMIs, Bundle Tasks), ELASTIC BLOCK STORE (Volumes, Snapshots, Lifecycle Manager), and NETWORK & SECURITY (Security Groups, Elastic IPs). The main content area has a header with 'Allocate new address' and 'Actions'. Below is a search bar and a table with one Elastic IP.

| Name | Elastic IP    | Allocation ID         | Instance |
|------|---------------|-----------------------|----------|
|      | 52.15.132.254 | eipalloc-0b1350be4... | -        |

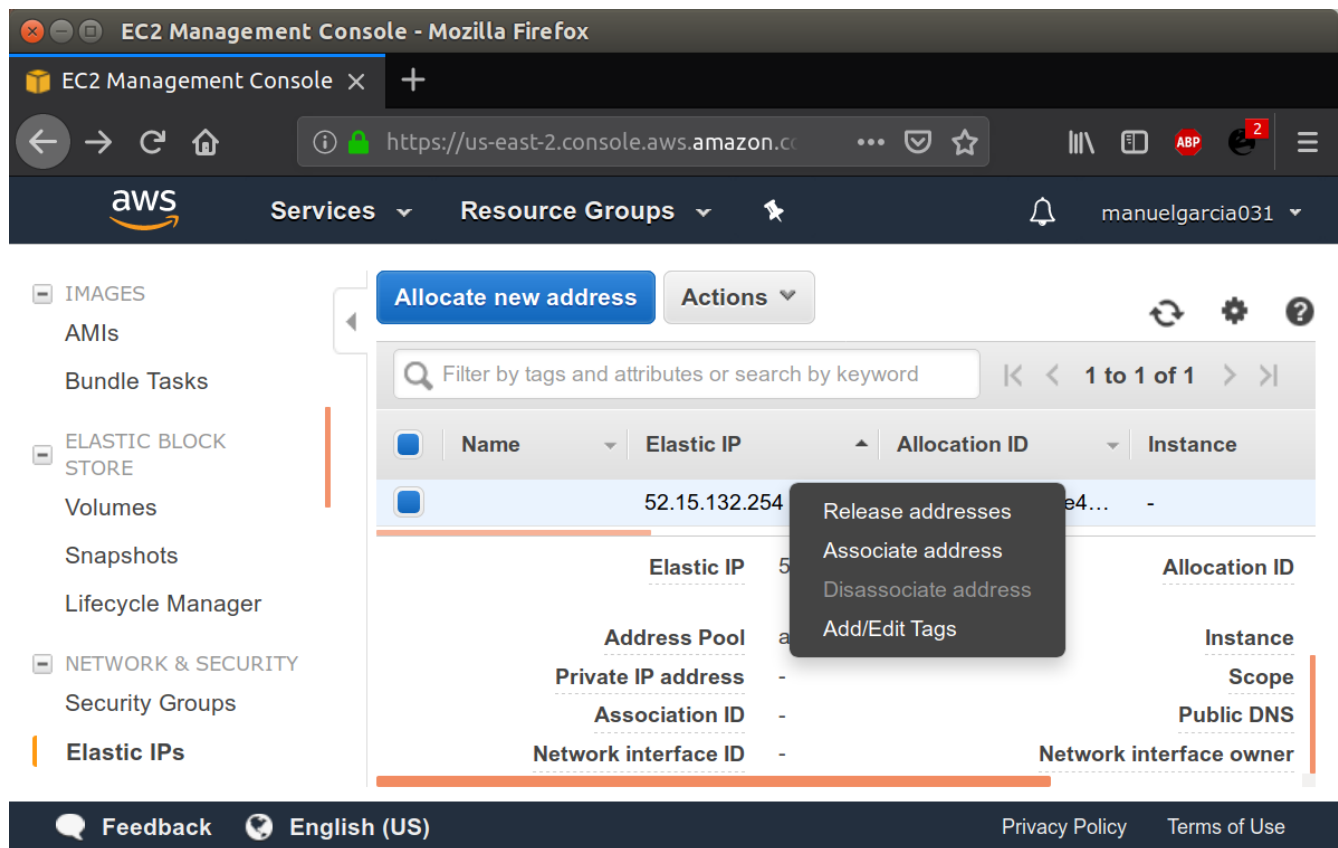
Below the table, detailed information for the selected Elastic IP is displayed:

- Elastic IP:** 52.15.132.254
- Allocation ID:**
- Address Pool:** amazon
- Instance:**
- Private IP address:** -
- Scope:**
- Association ID:** -
- Public DNS:**
- Network interface ID:** -
- Network interface owner:**

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

*Figura 76: IP's elásticas, AWS*

Para asociar la IP elástica a la instancia EC2, se hace clic derecho sobre la IP elástica y se selecciona la opción “Associate Address”, véase la Figura 77.

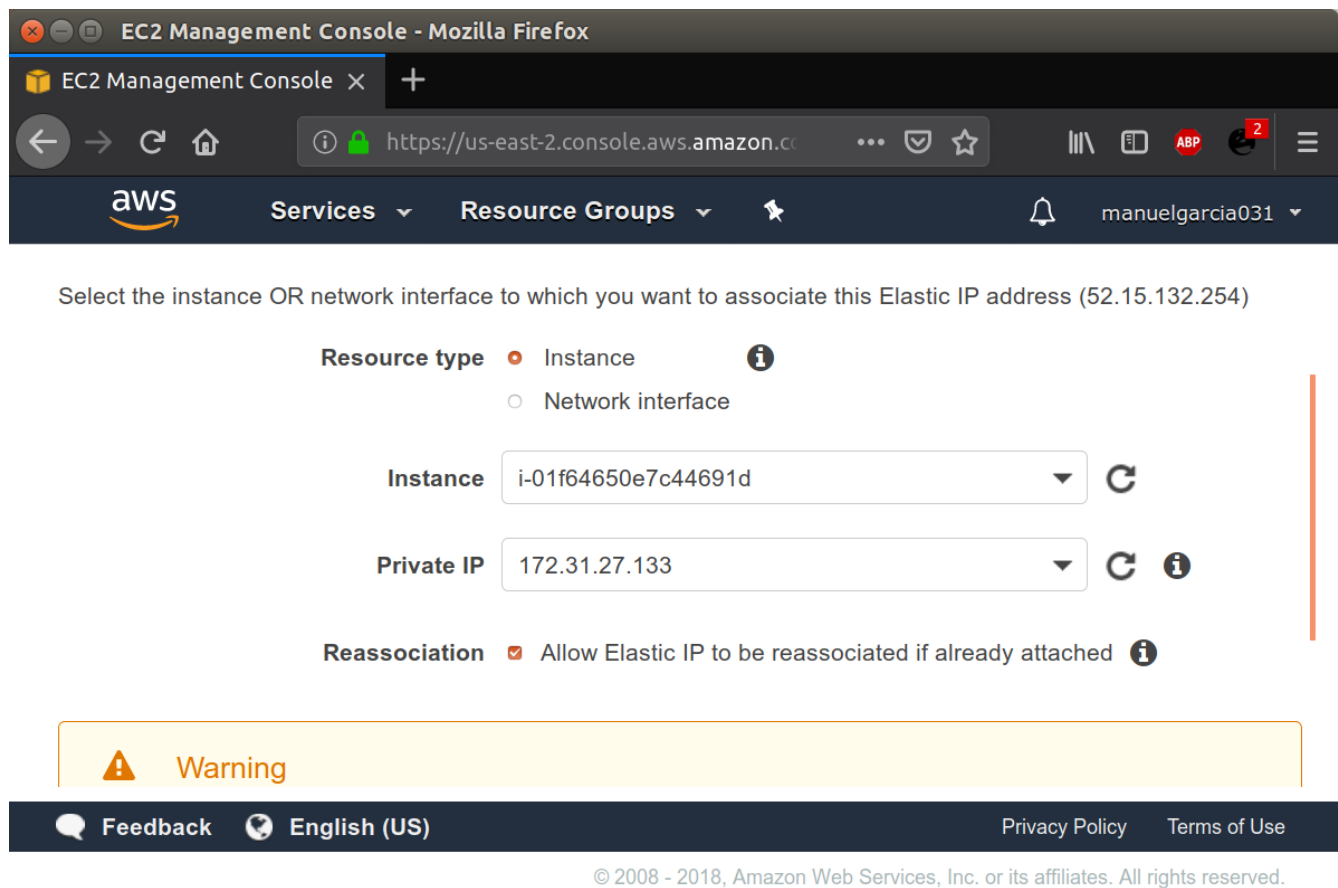


© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

*Figura 77: Opciones para una IP elástica, AWS*

Al seleccionar la opción “Associate Address” se desplegará una página mediante la cual se selecciona la instancia EC2 y la IP privada a las cual asociar la IP elástica, véase la Figura 78.





*Figura 78: Menú para asociar una IP elástica a una Instancia EC2*

Una vez hecho esto, la conexión mediante SSH a la instancia se realiza con el comando:

```
ssh -i /<path_de_la_llave>/<nombre_de_la_llave>.pem <username>@<dirección>
```

La dirección puede ser:

- Public DNS (IPv4)
- Private DNS
- IPv4 Public IP

Dicha información es proporcionada en la sección INSTANCES>Instances, véase la Figura 79.

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with categories like EC2 Dashboard, INSTANCES, IMAGES, and ELASTIC BLOCK STORE. The main area displays the details of an EC2 instance named 'Domotics' with ID 'i-01f64650e7c44691d'. The instance is a 't2.micro' type in the 'us-east-2b' availability zone, currently in a 'stopped' state. A table below the instance details lists various attributes:

| Attribute             | Value                                                                             |
|-----------------------|-----------------------------------------------------------------------------------|
| Instance ID           | i-01f64650e7c44691d                                                               |
| Instance state        | stopped                                                                           |
| Instance type         | t2.micro                                                                          |
| Elastic IPs           | 52.15.132.254*                                                                    |
| Availability zone     | us-east-2b                                                                        |
| Security groups       | launch-wizard-1 . view inbound rules .<br>view outbound rules                     |
| Scheduled events      | -                                                                                 |
| AMI ID                | ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20180627<br>(ami-5e8bb23b) |
| Platform              | -                                                                                 |
| IAM role              | -                                                                                 |
| Key pair name         | AWS_EC2_Juan_Manuel                                                               |
| Owner                 | 489019327223                                                                      |
| Public DNS (IPv4)     | ec2-52-15-132-254.us-east-2.compute.amazonaws.com                                 |
| IPV4 Public IP        | 52.15.132.254                                                                     |
| IPV6 IPs              | -                                                                                 |
| Private DNS           | ip-172-31-27-133.us-east-2.compute.internal                                       |
| Private IPs           | 172.31.27.133                                                                     |
| Secondary private IPs | -                                                                                 |
| VPC ID                | vpc-e8f2d180                                                                      |
| Subnet ID             | subnet-198d0063                                                                   |
| Network interfaces    | eth0                                                                              |
| Source/dest. check    | True                                                                              |
| T2/T3 Unlimited       | Disabled                                                                          |
| EBS-optimized         | False                                                                             |

Figura 79: Información de red de la instancia EC2, AWS

Haciendo clic derecho sobre la instancia, se accede a un menú que presenta opciones para detener la instancia, iniciarla, hibernar, reiniciar, modificar su IP, cambiar el tipo de instancia, etc., véase la Figura 80.

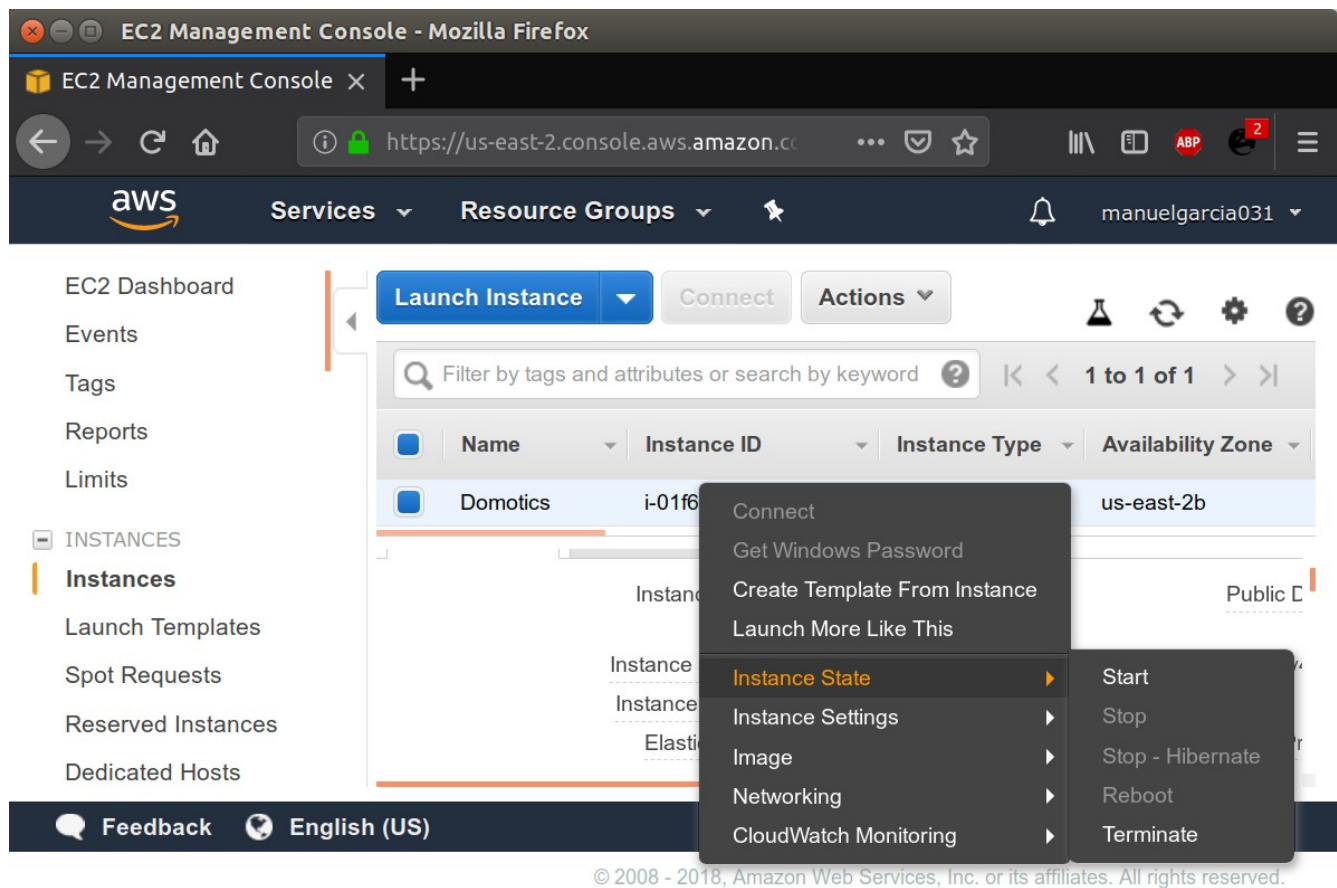


Figura 80: Opciones de Estado de una instancia EC2, AWS

## Apéndice D: Glosario.

| Término             | Descripción                                                                                                                                                                                                                                                    |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Baudios             | Número de símbolos por segundo en un medio de transmisión digital. Cada símbolo puede comprender 1 o más bits.                                                                                                                                                 |
| Circuito monofásico | Sistema de producción, distribución o consumo de energía eléctrica formado por una única corriente alterna o fase y por lo tanto todo el voltaje varía de la misma forma.                                                                                      |
| Circuito trifásico  | Sistema de producción, distribución o consumo de energía eléctrica formado por tres corrientes alternas monofásicas de igual frecuencia y amplitud, que presentan una diferencia de fase entre ellas de 120° eléctricos, y están dadas en un orden determinado |

| <b>Término</b>                | <b>Descripción</b>                                                                                                                                                                                  |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Computación cognitiva         | Sistemas de aprendizaje computacional que utilizan la minería de datos, el reconocimiento de patrones y el procesamiento del lenguaje natural para imitar el funcionamiento del cerebro humano.     |
| Desplazamiento angular        | Distancia recorrida por un cuerpo que sigue una trayectoria circular.                                                                                                                               |
| Framework                     | Esquema o estructura que se establece y que se aprovecha para desarrollar y organizar un software determinado, esto para poder escribir código o desarrollar una aplicación de manera más sencilla. |
| Frecuencia de muestreo        | Número de muestras por unidad de tiempo que se toman de una señal.                                                                                                                                  |
| Hardware                      | Partes físicas de un sistema informático.                                                                                                                                                           |
| Heterodino                    | Generador de frecuencias mediante la mezcla de dos o más señales en un dispositivo no lineal, tal como un diodo, una válvula termoiónica o un transistor                                            |
| Histéresis                    | Tendencia de un material a conservar una de sus propiedades, en ausencia del estímulo que la ha generado.                                                                                           |
| Inteligencia artificial       | Capacidad de un sistema para interpretar correctamente datos externos, para aprender de dichos datos y emplear esos conocimientos para lograr tareas y metas concretas a través de la adaptación.   |
| IP privada                    | Dirección IP reservada para uso interno detrás de un enrutador u otro dispositivo de traducción de direcciones de red.                                                                              |
| IP pública                    | Dirección IP visible mundialmente.                                                                                                                                                                  |
| <i>Machine learning</i>       | Método de análisis de datos que automatiza la construcción de modelos analíticos.                                                                                                                   |
| Nanotecnología                | Tecnología que se dedica al diseño y manipulación de la materia a escala nanométrica.                                                                                                               |
| Oscilador                     | Dispositivo capaz de convertir la energía de corriente continua en corriente alterna de una determinada frecuencia.                                                                                 |
| Par                           | Momento de fuerza que ejerce un motor sobre el eje de transmisión de potencia                                                                                                                       |
| Punto de acceso               | Dispositivo de red que interconecta equipos de comunicación inalámbricos.                                                                                                                           |
| Reluctancia variable          | Resistencia variable que posee un material al paso del flujo magnético cuando es influenciado por un campo magnético.                                                                               |
| Resistencia de <i>pull-up</i> | Resistencia que permite establecer un valor lógico (1 lógico o VCC) de reposo para cuando la señal de entrada no está presente y poder asegurar una lectura correcta.                               |
| Sensores piezoeléctricos      | Dispositivo que utiliza el efecto piezoeléctrico para medir presión, aceleración, tensión o fuerza, transformando las lecturas en señales                                                           |

| Término  | Descripción                                                                                                                                                                                               |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | eléctricas.                                                                                                                                                                                               |
| Software | Programa o conjunto de programas de cómputo que incluye datos, procedimientos y pautas que permiten realizar distintas tareas en un sistema informático.                                                  |
| Triodo   | Válvula electrónica que consta de tres electrodos dispuestos en el interior de una envoltura de vidrio en la que se ha hecho el vacío: un filamento calentado o cátodo, una rejilla, y una placa o ánodo. |

## Apéndice E: Abreviaturas y acrónimos

|         |                                                                           |
|---------|---------------------------------------------------------------------------|
| ADC     | <i>Analog to digital converter</i> (convertidor analógico a digital)      |
| AES     | <i>Advanced encryption standard</i> (estándar de encriptación avanzada)   |
| AIDA    | Asociación de domótica e inmótica avanzada                                |
| AJAX    | <i>Asynchronous JavaScript and XML</i> (JavaScript asíncrono y XML)       |
| AT      | Abreviatura de <i>attention</i> (atención)                                |
| AWS     | <i>Amazon web services</i> (servicios web de Amazon)                      |
| BJT     | <i>Bipolar Junction Transistor</i> (transistor de unión bipolar)          |
| CA      | Corriente alterna                                                         |
| CC      | Corriente continua                                                        |
| CD      | Corriente directa                                                         |
| CEN     | Comité Europeo de normalización                                           |
| CENELEC | Comité Europeo de normalización electrónica                               |
| CPU     | <i>Central processing unit</i> (unidad central de procesos)               |
| CSS     | <i>Cascading style sheets</i> (Hojas de estilo en cascada)                |
| DAC     | <i>Digital to analog converter</i> (convertidor digital a analógico)      |
| DNS     | <i>Domain name system</i> (sistema de nombres de dominio)                 |
| EC2     | <i>Elastic compute cloud</i> (cómputo elástico en la nube)                |
| EMR     | <i>Electromechanical relay</i> (relevador electromecánico)                |
| ETSI    | Instituto de estándares de telecomunicaciones europeo                     |
| GPIO    | <i>General purpose input output</i> (entrada-salida de propósito general) |

|        |                                                                                                                      |
|--------|----------------------------------------------------------------------------------------------------------------------|
| GPU    | <i>Graphics processing unit</i> (unidad de procesamiento gráfico)                                                    |
| GSM    | <i>Global System for Mobile Communications</i> (sistema global para las comunicaciones móviles)                      |
| GUI    | <i>Graphical user interface</i> (interfaz gráfica de usuario)                                                        |
| HAN    | <i>Home area network</i> (red de área doméstica)                                                                     |
| HTML   | <i>Hypertext Markup Language</i> (lenguaje de marcado de hipertexto)                                                 |
| HTTP   | <i>Hypertext Transfer Protocol</i> (protocolo de transferencia de hipertextos)                                       |
| HTTPS  | <i>Hypertext transfer protocol secure</i> (protocolo seguro de transferencia de hipertexto)                          |
| I2C    | <i>Inter-integrated circuit</i> (circuito inter-integrado)                                                           |
| IBM    | <i>International Business Machines</i> (Máquinas de negocios internacionales)                                        |
| IDE    | <i>Integrated development environment</i> (entorno de desarrollo integrado)                                          |
| IEC    | Comisión electrónica internacional                                                                                   |
| INEGI  | Instituto Nacional de Estadística y Geografía                                                                        |
| IoT    | <i>Internet of things</i> (internet de las cosas)                                                                    |
| IP     | <i>Internet protocol</i> (protocolo de internet)                                                                     |
| IRC    | <i>Internet Relay Chat</i> (protocolo de comunicación en tiempo real basado en texto)                                |
| ISO    | <i>International organization for standardization</i> (organización internacional de estandarización)                |
| ITU    | Unión internacional de telecomunicaciones                                                                            |
| JSON   | <i>JavaScript object notation</i> (notación de objetos de JavaScript)                                                |
| LAN    | <i>Local area network</i> (red de área local)                                                                        |
| LDAP   | <i>Lightweight directory access protocol</i> (protocolo ligero de acceso a directorios)                              |
| MAC    | <i>Media access control</i> (control de acceso a medios)                                                             |
| MAN    | Metropolitan area network (red de área metropolitana)                                                                |
| MOSFET | Metal oxide semiconductor field effect transistor (transistor de efecto de campo de semiconductor de óxido de metal) |
| MQTT   | Message queue telemetry transport (transporte por telemetría de cola de mensajes)                                    |
| PWM    | <i>Pulse width modulation</i> (modulación por ancho de pulso)                                                        |
| RAM    | <i>Random access memory</i> (memoria de acceso aleatorio)                                                            |
| RDMS   | <i>Relational database management system</i> (sistema de gestión de bases de datos relacionales)                     |
| RDSI   | Red digital de servicios integrados                                                                                  |

|       |                                                                                              |
|-------|----------------------------------------------------------------------------------------------|
| RH    | <i>Relative humidity</i> (humedad relativa)                                                  |
| ROM   | <i>Read only memory</i> (memoria de solo lectura)                                            |
| RPC   | <i>Remote procedure call</i> (llamada a procedimiento remoto)                                |
| RTC   | <i>Real time clock</i> (reloj de tiempo real)                                                |
| SCR   | <i>Silicon controlled rectifier</i> (rectificador controlado de silicio)                     |
| SDIO  | <i>Secure digital input output</i> (entrada salida digital segura)                           |
| SDK   | <i>Software Development Kit</i> (kit de desarrollo de software)                              |
| SIT   | <i>Static induction transistor</i> (transistor de inducción eléctrica)                       |
| SOC   | <i>Sistem on chip</i> (sistema en chip)                                                      |
| SPI   | <i>Serial peripheral interface</i> (protocolo de comunicación serial)                        |
| SQL   | Structured query language (lenguaje de consulta estructurada)                                |
| SSL   | <i>Secure sockets layer</i> (capa de sockets seguros)                                        |
| SSR   | <i>Solid state relay</i> (relevador de estado sólido)                                        |
| TCP   | <i>Transmission control protocol</i> (protocolo de control de transmisión)                   |
| TLS   | <i>Transport layer security</i> (seguridad en capas de transporte)                           |
| UART  | <i>Universal asynchronous receiver-transmitter</i> (transmisor-receptor asíncrono universal) |
| USB   | <i>Universal serial bus</i> (bus serie universal)                                            |
| WAN   | <i>Wide area network</i> (red de área amplia)                                                |
| Wi-Fi | <i>Wireless fidelity</i> (fidelidad inalámbrica)                                             |
| WPS   | <i>Wi-Fi protect setup</i> (configuración de protección de Wi-Fi)                            |
| WWW   | <i>World wide web</i> (red mundial)                                                          |
| XML   | <i>Extensible markup language</i> (lenguaje de marcas extensible)                            |