



**UNIVERSIDAD DEL AZUAY**

**FACULTAD DE CIENCIA Y TECNOLOGÍA**

**ESCUELA DE INGENIERÍA ELECTRÓNICA**

**“Diseño e Implementación de un Asistente Virtual para  
Control y Monitoreo de una Casa Inteligente”**

**Trabajo de graduación previo a la obtención del título de:**

**INGENIERO ELECTRÓNICO**

**Autores:**

**GABRIEL EDUARDO MENESES PALOMEQUE**

**CHRISTOPHER JUNIOR PESÁNTEZ ÁVILA**

**Director:**

**ING. HUGO TORRES SALAMEA Ph.D.**

**CUENCA, ECUADOR**

**2017**

## **DEDICATORIA**

Este trabajo está dedicado con gran sinceridad y afecto para toda mi familia y amigos, quienes junto a mi celebraron mis logros, mis derrotas y estuvieron junto a mí en los momentos más importantes. Especialmente quiero agradecer a mis padres, Jorge Meneses y Kerly Palomeque, también a mi prometida, Verónica Andrade; porque gracias a ellos me he formado como la persona que soy ahora. Gracias.

Gabriel Eduardo Meneses Palomeque

A mi esposa Jessica y a mi hija Daniela, por la paciencia y comprensión durante este tiempo regalado para culminar con mis estudios, sin su comprensión y apoyo constante no hubiera sido posible llegar a la finalización del mismo. A mi familia y amigos por haber sido siempre el soporte en este largo camino.

Christopher Junior Pesántez Ávila

## **AGRADECIMIENTO**

Agradezco de la manera más grata a nuestro director de tesis, el Ingeniero Hugo Torres, quién siempre supo guiarnos con paciencia y con gran veracidad, sin duda alguna, además de ser nuestro director de tesis, es un ser humano maravilloso quien merece nuestro respeto, cariño y admiración.

También quiero agradecer de una manera muy especial a mis amigos, quienes supieron colaborarme de una manera incondicional para cumplir todas las metas académicas y personales que me he propuesto, mi cariño y mis consideraciones para estas grandes personas.

Gabriel Eduardo Meneses Palomeque

A todos los profesores durante estos cinco años de estudio por impartir sus valiosos conocimientos, además de contribuir en mi formación como profesional y ser humano. A mi director de tesis Dr. Hugo Torres Salamea que con sus sabios consejos nos ha ayudado para que lleguemos a la culminación de este proyecto. A mi compañero Gabriel Meneses porque con su apoyo y dedicación, juntos hemos logrado un nuevo objetivo.

Christopher Junior Pesántez Ávila

**ÍNDICE DE CONTENIDOS**

DEDICATORIA .....	ii
AGRADECIMIENTO .....	iii
ÍNDICE DE CONTENIDOS .....	iv
ÍNDICE DE FIGURAS.....	vii
ÍNDICE DE TABLAS .....	xii
ÍNDICE DE ANEXOS .....	xiii
RESUMEN .....	xiv
ABSTRACT.....	xv
INTRODUCCIÓN .....	1
<b>1. CAPÍTULO I: INVESTIGACIÓN ACERCA DE LAS DISTINTAS TECNOLOGÍAS USADAS EN IOT .....</b>	<b>2</b>
1.1. Introducción .....	2
1.2. Tipo de comunicación. ....	4
1.3. Topología de la red.....	4
1.4. Servidor Principal.....	12
1.5. Lenguaje de Programación.....	12
1.6. Tipo de sensores y actuadores .....	13
1.6.1 Sensores.....	13
1.6.2 Actuadores.....	15
1.7 Conclusiones .....	16
<b>2 CAPÍTULO II: DESARROLLO DEL ASISTENTE VIRTUAL (AV). .....</b>	<b>17</b>
2.1 Introducción .....	17
2.2 Interfaz Gráfica .....	17
2.2.1 Iluminación.....	18
2.2.2 Seguridad.....	19
2.2.3 Multimedia .....	20
2.2.4 Automatización .....	21
2.2.5 Dispositivos Inteligentes .....	21
2.3 Base de Datos .....	22

2.4	Comunicación Wifi .....	24
2.5	Inteligencia Artificial (IA) .....	25
2.5.1	Sistemas de Inteligencia Artificial. ....	25
2.5.2	Reglas para el diseño de un Sistema Experto.....	36
2.6	Conclusiones .....	38
<b>3</b>	<b>CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DE LA RED DE SENSORES Y ACTUADORES.....</b>	<b>39</b>
3.1	Introducción .....	39
3.2	<i>Hardware</i> .....	39
3.2.1	Microcontrolador.....	39
3.2.2	Servidor .....	41
3.2.3	Alarma.....	44
3.2.4	GSM 46	
3.2.5	Dispositivos utilizados en el hogar.....	47
3.2.5.1	Sistema de Alarma .....	47
3.2.5.2	Sistema de Riego Inteligente .....	49
3.2.5.3	Multimedia.....	51
3.3	<i>Software</i> .....	52
3.3.1	Base de Datos.....	52
3.3.2	Página Web .....	53
3.3.3	Android Studio.....	56
3.3.4	Inteligencia Artificial .....	57
3.3.5	Protocolo de Comunicación .....	58
3.4	Implementación de la Red.....	62
3.4.1	Acciones derivadas de los sensores en el hogar.....	64
3.4.2	Actuadores dispuestos en el hogar .....	66
3.4.3	Alarma Inteligente.....	73
3.4.4	Módulo GSM .....	74
3.4.5	Acceso Remoto .....	75
3.4.5.1	Página Web .....	75
3.4.5.2	Aplicación Móvil en Android .....	78

3.5	Conclusiones .....	82
<b>4.</b>	<b>CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA IMPLEMENTADO.....</b>	<b>84</b>
4.1	Introducción .....	84
4.1.1	Iluminación inalámbrica .....	84
4.1.2	Portero Inteligente.....	86
4.1.3	Sistema de alarma inteligente .....	86
4.1.4	Sistema de riego inteligente .....	89
4.2	Costo de la implementación. ....	90
4.3	Pruebas del sistema. ....	92
4.3.1	Pruebas de tiempos de respuesta.....	92
4.3.1.1	Tiempo de respuesta del servidor.....	93
4.3.1.2	Tiempo de respuesta de los distintos actuadores y sensores.....	95
4.3.1.3	Tiempo de respuesta del reconocimiento de voz en Android. ....	97
4.3.2	Prueba del sistema ante distintos estímulos externos.....	99
4.3.2.1	Prueba de rendimiento del sistema. ....	99
4.3.2.2	Prueba de robustez del sistema. ....	101
4.4	Ventajas y Desventajas del Sistema .....	101
4.4.1	Ventajas: .....	101
4.4.2	Desventajas: .....	103
4.5	Conclusiones .....	104
	CONCLUSIONES .....	106
	RECOMENDACIONES.....	110
	REFERENCIAS.....	111

## ÍNDICE DE FIGURAS

Figura 1.1: Características del tipo de tecnología a usar. ....	3
Figura 1.2: Esquema topología Bus .....	5
Figura 1.3: Esquema topología Árbol .....	6
Figura 1.4: Esquema topología Estrella .....	7
Figura 1.5: Esquema topología Anillo .....	8
Figura 1.6: Esquema topología Malla .....	8
Figura 1.7: Ubicación Interruptores y Tomacorrientes dentro del Hogar .....	9
Figura 1.8: Esquema topología mixta (estrella, árbol).....	10
Figura 1.9: Esquema a instalarse en los distintos dispositivos. ....	16
Figura 2.1: Estructura del Asistente Virtual.....	17
Figura 2.2: Esquema de tareas a realizar por el AV.....	18
Figura 2.3: Matriz de iluminación generada a partir de la base de datos.....	18
Figura 2.4: Matriz de sensores generada a partir de la base de datos. ....	20
Figura 2.5: Interfaz gráfica multimedia. ....	20
Figura 2.6: Interfaz automatización del hogar. ....	21
Figura 2.7: Estructura Base de Datos.....	22
Figura 2.8: Idle Python27, algoritmo para inicializar base de datos.....	23
Figura 2.9: Idle Python27, algoritmo para actualizar datos .....	23
Figura 2.10: Idle Python27, algoritmo para eliminar datos .....	24
Figura 2.11: Idle Python27, algoritmo para insertar datos.....	24
Figura 2.12: Estructura lógica difusa basada en Mandani para sistema de Luz Ambiente.....	25
Figura 2.13: Reglas control difuso de Iluminación extraídas de Matlab. ....	25
Figura 2.14: Valor lingüístico de Iluminación ambiental extraídas de Matlab.....	26
Figura 2.15: Valor lingüístico de iluminación interior extraídas de Matlab.....	26
Figura 2.16: Monitor Serial IDE Arduino mostrando datos del sistema difuso. ....	27
Figura 2.17: Estructura para agregar acciones a la función horario de automatización. ....	28
Figura 2.18: Diagrama de Flujo para entrenamiento Facial.....	30

Figura 2.19: Diagrama de Flujo para Reconocimiento Facial. ....	31
Figura 2.20: Entrenamiento para reconocimiento facial. ....	32
Figura 2.21: Base de datos faciales. ....	32
Figura 2.22: Estructura lógica difusa basada en Mandani para sistema de Riego Inteligente. ....	34
Figura 2.23: Reglas control difuso extraídas de Matlab. ....	34
Figura 2.24: Valor lingüístico de Tiempo de Riego extraídas de Matlab. (Salida) ...	35
Figura 2.25: Valor lingüístico de Radiación Solar extraídas de Matlab. (Entrada) ...	35
Figura 2.26: Valor lingüístico de Humedad extraídas de Matlab. (Entrada). ....	35
Figura 2.28: Estructura Sistema de Riego Inteligente. ....	36
Figura 3.1: ESP8266-12E (Shenzen Anxinke Technology, 2017) ....	40
Figura 3.2: Raspberry Pi 3. (Prometec) ....	41
Figura 3.3: Waveshare HDMI TFT Resistiva (Waveshare, 2017). ....	43
Figura 3.4: Pines de conexión a la Raspberry pi (Waveshare, 2017) ....	44
Figura 3.5: Adafruit TFT Capacitiva (Adafruit, 2017) ....	45
Figura 3.6: Módulo GSM/GPRS SIM900 (HetPro, 2017). ....	46
Figura 3.7: Sensor PIR (PIR, 2017). ....	48
Figura 3.8: Electroválvula (ElectroniLab, 2017) ....	49
Figura 3.9: Panel Solar. ....	50
Figura 3.10: Diagrama de flujo del sistema de riego. ....	50
Figura 3.11: Receptor IR. ....	51
Figura 3.12: Emisor IR ....	52
Figura 3.13: Diagrama de flujo activación de dispositivos multimedia ....	52
Figura 3.14: Logo MySQL (Oracle, 21017) ....	53
Figura 3.15: Distintas tecnologías. PHP, Javascript, Ajax, JQuery. ....	54
Figura 3.16: Diferentes opciones del servicio de hosting. ....	55
Figura 3.17: Archivos que conforman la página web ....	55
Figura 3.18: Tablas de las que consta la base de datos ....	56
Figura 3.19: Android Studio (Android Studio, 2017), REST Web Service (Rodriguez, 2017) ....	57
Figura 3.20: Diagrama de flujo del proceso de conexión a la red. ....	59

Figura 3.21: Pasos para conexión a la red del Hogar.....	60
Figura 3.22: Arduino OTA .....	61
Figura 3.23: Topología de la red WSAN. ....	63
Figura 3.24: Activación sensor PIR (Casadomo, 2017) .....	64
Figura 3.25: Diagrama del acceso al sistema implementado. ....	66
Figura 3.26: Diagrama de flujo interruptor inteligente .....	67
Figura 3.27: Circuito Inteligente .....	68
Figura 3.28: PCB del circuito inteligente .....	69
Figura 3.29: Activación del integrado PC817 con respecto a la rectificación a la onda senoidal. ....	70
Figura 3.30: Activación del integrado 4n33 con respecto a la rectificación a la onda senoidal. ....	71
Figura 3.31: Corte de ambos lados de la onda senoidal.....	72
Figura 3.32: Gráfica de la onda en la iluminación así como en el convertor. ....	72
Figura 3.33: Diagrama de la alarma inteligente .....	73
Figura 3.34: Diagrama de flujo alarma inteligente. ....	74
Figura 3.35: Opción iluminación en página web .....	75
Figura 3.36: Opción sensores en página web.....	76
Figura 3.37: Opción Horario en la página web .....	77
Figura 3.38: Opciones entregadas al presionar el botón “Ingresar nuevo Horario”. .	77
Figura 3.39: Captura de pantalla de la vista principal de la aplicación. ....	78
Figura 3.40: Estructura comunicación Android-Base de Datos.....	78
Figura 3.41: Captura de pantalla del menú Iluminación.....	79
Figura 3.42: Captura de pantalla del menú Sensores .....	80
Figura 3.43: Captura de pantalla del menú Horario.....	80
Figura 3.44: Captura de pantalla del menú Actividades .....	81
Figura 3.45: Captura de pantalla del menú Reconocimiento de Voz.....	82
Figura 4.1: Aislamiento de los distintos componentes del interruptor. ....	84
Figura 4.2: Implementación del interruptor inteligente. ....	85
Figura 4.3: Interruptor inteligente instalado. ....	85
Figura 4.4: Instalación del módulo ESP8266 al portero eléctrico. ....	86

Figura 4.5: Instalación del sensor magnético en la puerta principal.....	87
Figura 4.6: Instalación del sensor de presencia en la entrada principal.....	87
Figura 4.7: Dispositivo multipropósito, sensor presencial e interruptor inteligente..	88
Figura 4.8: Sistema de alarma inteligente.....	89
Figura 4.9: Sistema de riego inteligente .....	89
Figura 4.10: Sistema de riego .....	90
Figura 4.11: Esquema de pruebas para el servidor principal.....	93
Figura 4.12: Estructura del escenario 0 para prueba de tiempo de respuesta del servidor.....	93
Figura 4.13: Shell de Python para visualización de tiempo de respuesta.....	93
Figura 4.14: Estructura del escenario 1 para prueba de tiempo de respuesta del servidor.....	94
Figura 4.15: Shell de Python para visualización de tiempo de respuesta.....	94
Figura 4.16: Estructura de escenario 2 para prueba de tiempo de respuesta del servidor.....	94
Figura 4.17: Shell de Python para visualización de tiempo de respuesta.....	95
Figura 4.18: Estructura del escenario 0 para prueba de tiempo de respuesta del interruptor.....	95
Figura 4.19: Shell de Python para visualización de tiempo de respuesta.....	96
Figura 4.20: Estructura del escenario 1 para prueba de tiempo de respuesta del interruptor.....	96
Figura 4.21: Shell de Python para visualización de tiempo de respuesta.....	96
Figura 4.22: Estructura del escenario 1 para prueba de tiempo de respuesta del interruptor.....	97
Figura 4.23: Shell de Python para visualización de tiempo de respuesta.....	97
Figura 4.24: Estructura del escenario 0 para prueba de tiempo de respuesta del interruptor.....	98
Figura 4.25: Monitor de Android para visualización de tiempo de respuesta.....	98
Figura 4.26: Shell de Python para visualización respuesta a peticiones.....	99
Figura 4.27: Shell de Python para visualización respuesta a peticiones.....	100
Figura 4.28: Shell de Python para visualización respuesta a peticiones.....	100

Figura 4.29: Shell de Python para visualización respuesta a peticiones.....	100
Figura 4.30: Resumen tiempos de respuesta del servidor.....	104
Figura 4.31: Resumen de rendimiento del servidor. ....	105
Figura 4.32: Grafico de consumo energético desde que se implementó el sistema.	105

## ÍNDICE DE TABLAS

Tabla 1.1: Hoja de datos del módulo ESP8266-12E.....	11
Tabla 4.1: Costos del sistema inteligente.....	90

## ÍNDICE DE ANEXOS

Anexo 1: Base de Datos .....	114
Anexo 2: Lógica Fuzzy Iluminación.....	117
Anexo 3: Entrenamiento EigenFaces .....	123
Anexo 4: Entrenamiento FisherFaces .....	127
Anexo 5: Reconocimiento de Rostro utilizando FisherFaces y EigenFaces.....	133
Anexo 6: Sistema lógica difusa para optimizar de la iluminación interior .....	134

## Diseño e Implementación de un Asistente Virtual para Control y Monitoreo de una Casa Inteligente

### RESUMEN

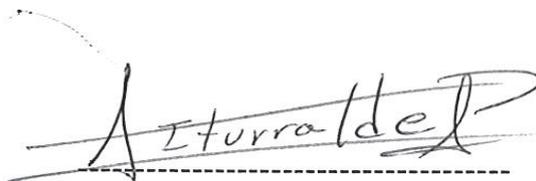
El objetivo de esta tesis es desarrollar un asistente virtual inteligente, dicho asistente provee de un control domótico a través de: aplicaciones en Android, aplicaciones web, reconocimiento de voz e interfaces visuales basadas en HMI. La comunicación entre el servidor y los diferentes nodos instalados usan un protocolo de descubrimiento basado en UDP, con un proceso continuo de lectura de datos que serán almacenados y obtenidos de la base de datos. El acceso remoto al hogar consta de página web y aplicación en Android los cuales leen la información almacenada en la base de datos, mostrándolos en tiempo real.

**Palabras clave:** asistente virtual, automatización, domótica, dispositivos inteligentes, inteligencia artificial.



Ing. Hugo Torres Ph.D.

Director de Tesis



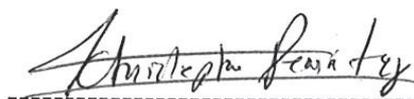
Ing. Daniel Iturralde Ph.D.

Director de Carrera



Gabriel Meneses

Autor de Tesis



Christopher Pesántez

Autor de Tesis

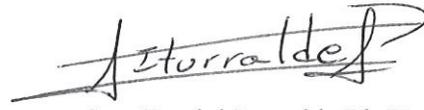
**DESIGN AND IMPLEMENTATION OF A VIRTUAL ASSISTANT FOR  
SMART HOME CONTROL AND MONITORING**

**ABSTRACT**

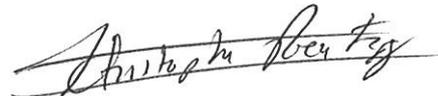
This research paper aimed to develop a smart virtual assistant that provides home automation control through applications in Android, web applications, voice recognition and visual interfaces based on HMI. The communication between the server and the different installed nodes uses a UDP-based discovery protocol with a continuous process of reading data, which will be stored and recovered from the database. Home Remote access consists of a web page and application on Android, which read the information stored in the database, showing them in real time.

**Keywords:** virtual assistant, automation, home automation, smart devices, artificial intelligence.

  
Ing. Hugo Torres Ph.D.  
**Thesis Director**

  
Ing. Daniel Iturralde Ph.D.  
**School Director**

  
Gabriel Meneses  
**Author**

  
Christopher Pesántez  
**Author**

## **INTRODUCCIÓN**

En la actualidad la automatización y el control de procesos que permiten tener mayor seguridad, confort y accesibilidad, es una necesidad que viene creciendo de forma acelerada, la domótica es un tema en constante evolución y cada vez existen nuevos dispositivos electrónicos, a menor precio y con una gran calidad, lo que facilita su presencia en hogares de clase media. Por lo tanto, esta tesis abordará la construcción, programación e implementación de un asistente virtual en un hogar de clase media, tomando en cuenta las distintas necesidades que presente cada usuario del hogar, que formará parte del sistema inteligente, para lo cual primero se realizará una investigación exhaustiva con el fin de utilizar la tecnología más idónea que se adapte a las necesidades del hogar.

Una vez elegida la tecnología ideal es necesario realizar la construcción y programación de los distintos dispositivos inteligentes, así como, la integración de electrodomésticos a la red, para controlarlos de manera automática y remota.

Esta automatización se realizará basándose en la construcción de dispositivos de bajo costo y robustos para que el sistema funcione de manera óptima, tomando en cuenta que en el hogar en el cual se implementará el sistema viven personas de distintas edades, para los cuales el sistema deberá adaptarse a cada uno.

# 1. CAPÍTULO I: INVESTIGACIÓN ACERCA DE LAS DISTINTAS TECNOLOGÍAS USADAS EN IOT

## 1.1. Introducción

Para comenzar, es necesario entender el término IOT<sup>1</sup>, el cual hace referencia a la conexión de distintos sensores o actuadores con la nube, es decir este tipo de tecnología nos permite controlar y monitorear distintos dispositivos a través del Internet.

Un sensor según la RAE<sup>2</sup> se define como: “un dispositivo que detecta una determinada acción externa, temperatura, presión, etc., y la transmite adecuadamente”, (RAE, 2014) en resumen, un sensor transforma cualquier magnitud física o química en una magnitud eléctrica, la cual es interpretada para realizar una acción dependiendo del mismo; por otra parte, un actuador según RAE se define como: “un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado.”

Para realizar una implementación de IOT en un hogar es necesario establecer la tecnología que se va a usar en cada una de sus etapas, por ende, para un mejor entendimiento de la misma, se dividirán como se puede apreciar en la Figura 1.1.

---

<sup>1</sup> IoT: Internet Of Things – Internet de las cosas

<sup>2</sup> RAE: Real Academia Española

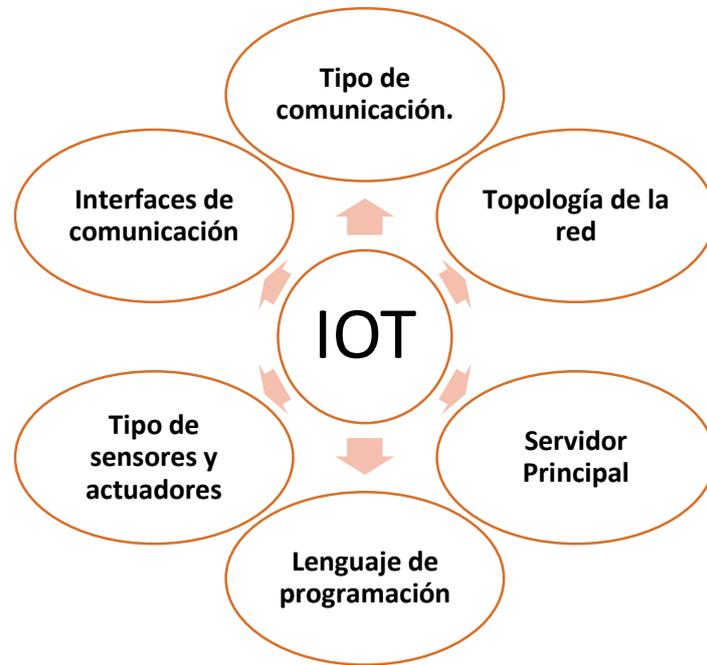


Figura 1.1: Características del tipo de tecnología a usar.

1. **Tipo de comunicación.**- Tecnología que se va a utilizar para conectar los distintos dispositivos con la nube.
2. **Topología de la red.**- Hace referencia al tipo de distribución que tendrán los distintos dispositivos en la red.
3. **Servidor Principal (SP).**- Dispositivo principal que realiza el direccionamiento de los distintos dispositivos de la red según lo que disponga el usuario.
4. **Lenguaje de programación.**- Lenguaje en el que se va a desarrollar el programa que se ejecutará sobre el servidor principal.
5. **Tipo de sensores y actuadores.**- Hace referencia a los distintos sensores y actuadores que se pueden implementar en un hogar inteligente.
6. **Interfaces de comunicación.**- Interfaces con las cuales el usuario controlará y monitoreará a los distintos sensores y actuadores instalados en el hogar.

## 1.2. Tipo de comunicación.

En la actualidad existen varias tecnologías que ofrecen distintas características en la comunicación de sensores y actuadores con el SP, se evaluará la mejor opción de acuerdo a las siguientes características:

- **Comunicación Inalámbrica:** se requiere de esta característica ya que nos permite evitar el cableado en el hogar, lo cual mejora la estética de los distintos dispositivos.
- **Económica:** la idea de integrar la mayor parte de dispositivos del hogar en una nube implica un gasto proporcional al número de dispositivos que se requieran, por lo que se optará por una tecnología económica.
- **Multipunto:** esta característica es importante ya que como se mencionó anteriormente la idea es integrar la mayor parte de dispositivos del hogar.
- **Amplia cobertura:** esta característica es necesaria porque los dispositivos se encuentran dispersos por el hogar y es necesario que la cobertura cubra dichos lugares.
- **Bajo consumo:** este dispositivo funcionará todo el tiempo por lo que se requiere un bajo consumo energético.

## 1.3. Topología de la red.

Para definir la tecnología más idónea es necesario definir el tipo de topología que se acople con la distribución de sensores y actuadores para un hogar inteligente, para esto se tiene distintas opciones como:

**Topología Bus.-** Este tipo de topología se la puede representar como una línea conectada a todos los dispositivos o nodos de la red. En la Figura 1.2 se puede apreciar un ejemplo de este tipo de topología.

**Características:**

- Por lo general, el cable en la topología de bus tiene dos terminales extremos que amortiguan la señal para evitar que se muevan continuamente desde un extremo de la red a la otra.
- Existen dos tipos distintos de topología de bus: lineal y distribuida.
- Una topología de bus lineal implica que sólo hay una línea asociada a dos puntos finales distintos.
- En una topología de bus distribuida puede haber más de un patrón lineal conectado a la red.
- La topología bus a menudo es valorada por su simplicidad y menor costo de implementación.
- Uno de los inconvenientes es su línea central, ya que si la misma falla, toda la red se puede caer. (Computer Hope, 2016).

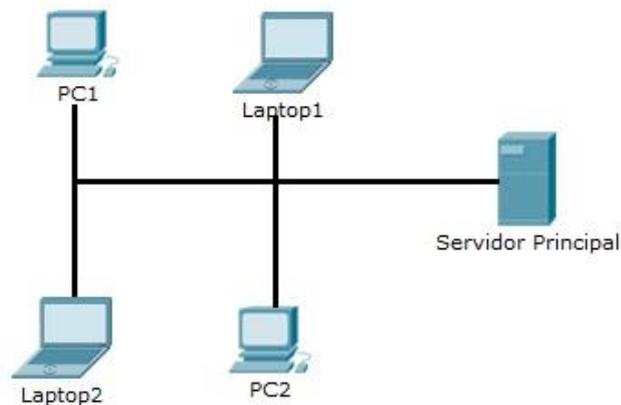


Figura 1.2: Esquema topología Bus

**Topología Árbol.-** Una topología de árbol está formada por múltiples topologías en estrella, que implican una variedad de nodos individuales conectados a un nodo central. En la Figura 1.3 se puede apreciar un ejemplo de este tipo de topología.

**Características:**

- Se la puede definir como una combinación de topologías estrella y bus, donde varios elementos están conectados a través de una sola conexión lateral.
- Cada nodo de un nivel de jerarquía tiene enlaces punto a punto con cada nodo adyacente en su nivel más abajo.
- Cuando se ve de una manera visual, estos sistemas se parecen a una estructura de árbol. (Techopedia, 2013).

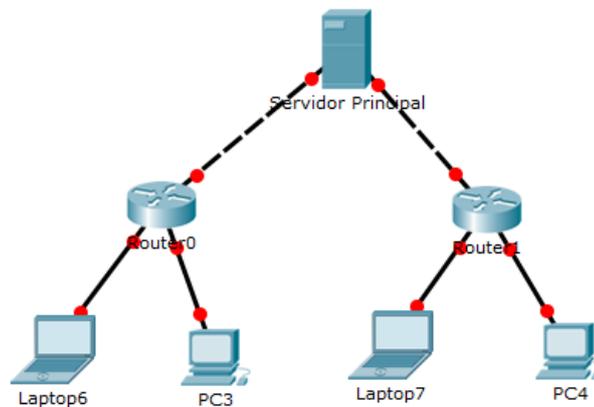


Figura 1.3: Esquema topología Árbol

**Topología Estrella.-** En esta configuración, cada nodo se conecta a un dispositivo de red central, como un *hub*, conmutador o computadora. En la Figura 1.4 se puede apreciar un ejemplo de este tipo de topología.

**Características:**

- El dispositivo de red central actúa como un servidor y los dispositivos periféricos actúan como clientes.
- El sitio central tiene el control de todos los nodos conectados a él.
- El eje central por lo general es un equipo rápido, independiente y es responsable de dirigir todo el tráfico a otros nodos.
- Las principales ventajas de una red en estrella es que un nodo en mal funcionamiento no afecta al resto de la red.

- Sin embargo este tipo de red puede ser propenso a cuello de botella y problemas de fallo en el sitio central (Telecom ABC, 2012).

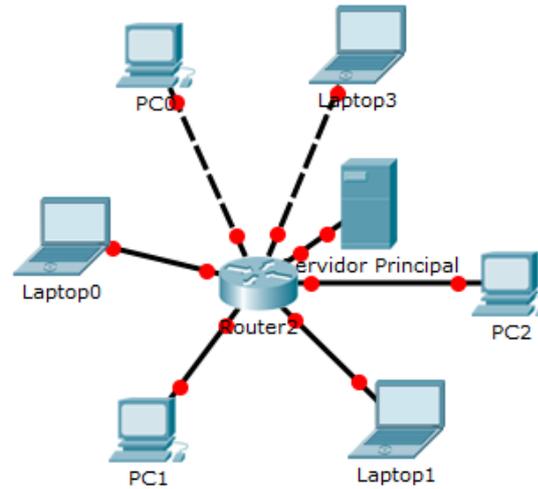


Figura 1.4: Esquema topología Estrella

**Red en anillo.-** Una red en anillo es una red de área local (LAN) en la que los nodos (estaciones de trabajo u otros dispositivos), están conectados en una configuración de bucle cerrado. En la Figura 1.5 se puede apreciar un ejemplo de este tipo de topología.

**Características:**

- La topología de anillo puede resultar óptimo cuando los requisitos del sistema son modestos y las estaciones de trabajo están en puntos dispersos.
- Una rotura en el cable de una red puede resultar en una degradación de velocidad de datos entre pares de estaciones.
- Si dos roturas se producen y no se encuentran en la misma sección de cable, algunas estaciones de trabajo serán cortados de algunos de los otros.
- Cuando la fiabilidad del sistema es una preocupación crítica, una topología de bus o estrella pueden resultar mejor que una red en anillo.
- Si se requiere redundancia, la topología de red de malla puede ser preferible. (Whatis, 2005).

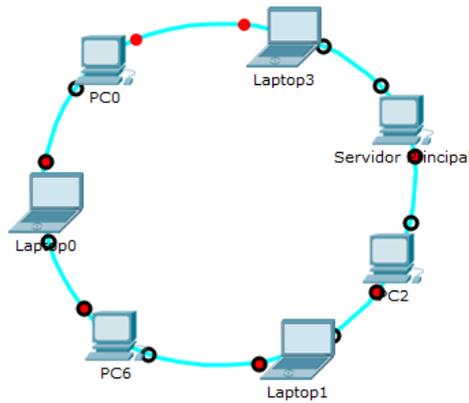


Figura 1.5: Esquema topología Anillo

**Topología de red de malla.-** tiene una configuración de red en la que cada dispositivo está interconectada con los otros. En la Figura 1.6 se puede apreciar un ejemplo de este tipo de topología.

**Características:**

- Esta topología se utiliza comúnmente para las redes inalámbricas.
- En una topología de malla parcial, al menos dos de los equipos de la red cuentan con conexiones a varios otros equipos de esa red.
- Esta es una forma más barata de construir una red que tiene una cierta redundancia en caso de que uno de los equipos primarios o conexiones en la red puedan caer. (Networking Basics, 2014).

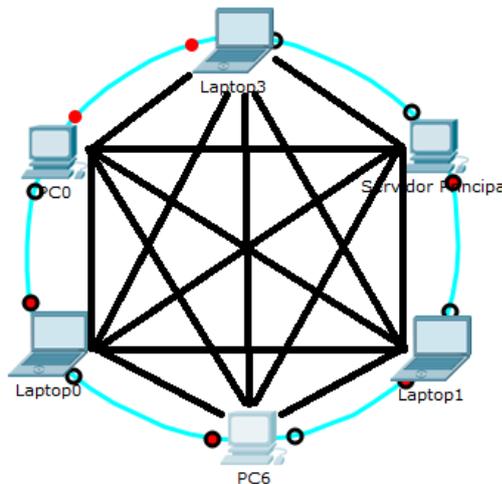


Figura 1.6: Esquema topología Malla

Ahora, entendiendo las distintas topologías con las que se puede trabajar, se realizará una distribución inicial para los distintos dispositivos que estarán conectados con el SP, para publicarlos en la nube; dicha distribución se basará en la posición actual de los interruptores y tomacorrientes que son ubicaciones estáticas dentro del hogar en el que se implantará esta red, como se puede apreciar en la Figura 1.7.



Figura 1.7: Ubicación Interruptores y Tomacorrientes dentro del Hogar

Como se puede apreciar en las Figura 1.7 y según la conexión que se necesita para que cada uno de los dispositivos se conecte con el SP, se optará por utilizar una topología mixta, combinando la topología estrella con la topología de árbol, esta topología nos brindará una mejor conexión de todos los dispositivos con el SP, el tipo de topología que se pretende usar se puede observar en la Figura 1.8:

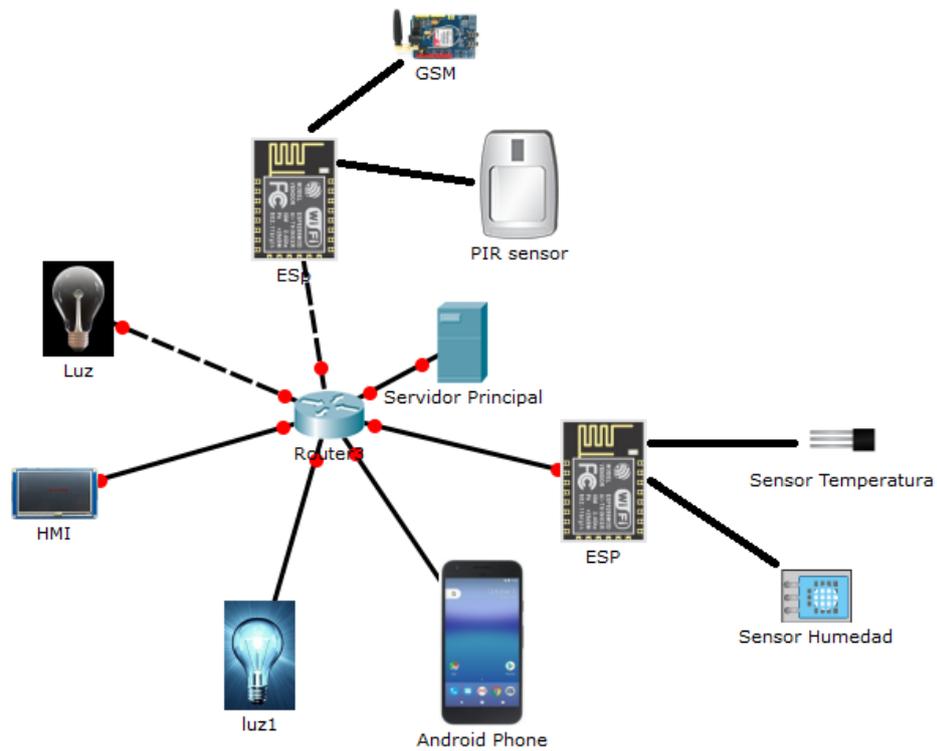


Figura 1.8: Esquema topología mixta (estrella, árbol)

De acuerdo a las características del tipo de comunicación planteado y de la topología elegida, se optará por usar una comunicación WiFi con arquitectura híbrida basada modo *Access Point* y modo estación, la cual nos permite crear un direccionamiento de tal manera que cada dispositivo WiFi pueda comunicarse con el SP<sup>3</sup> y al mismo tiempo servir como repetidor para los dispositivos que no se encuentren dentro de la red Wifi principal.

En el mercado existen distintos tipos de dispositivos que proveen de comunicación WiFi a un microcontrolador, tales como: RS9110-N módulo construido por GHI Electronics, HLK-RM04 el cual pertenece a la empresa Hi-Link, ESP8266-12E de la empresa Espressif, pero no todos pueden funcionar en modo Access Point y modo

---

<sup>3</sup> SP: Servidor Principal

estación al mismo tiempo. El tipo de tecnología que cumple con esta característica es el ESP8266-12E el cual según su hoja de datos, que se puede visualizar en la Tabla 1.1, puede funcionar en modo híbrido y tiene las siguientes características:

Tabla 1.1: Hoja de datos del módulo ESP8266-12E

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

Fuente: (IOT, 2015)

#### 1.4. Servidor Principal

Para elegir el modelo y los requerimientos mínimos del servidor principal es necesario primero desarrollar el programa para analizar las características con las que debe contar el procesador, para que el programa principal funcione sin problema.

#### 1.5. Lenguaje de Programación

Esta etapa es muy importante definirla bien, ya que el lenguaje en el cual esté programado el sistema nos brindara las limitaciones del mismo, es por eso que se optará por elegir un lenguaje de programación que cumpla las siguientes características:

- **Basado en *software* libre.**- Al no pagar ninguna licencia por la adquisición del *software* los costos totales del sistema serán más económicos.
- **Poco consumo de memoria.**- Existen lenguajes de programación en los cuales es necesario adquirir un compilador con licencia para que la construcción del código sea óptimo.
- **Comunidad de desarrolladores.**- Al tener una comunidad grande se puede solventar problemas consultando a la misma, lo cual optimiza el tiempo de programación.

Acorde a las características impuestas anteriormente, se trabajará con Python 2.7, el cual es un *software* de programación libre con una comunidad de desarrolladores muy grande, al ser un *software* libre no es necesario adquirir ninguna licencia para compilar programas de manera óptima.

## 1.6. Tipo de sensores y actuadores

### 1.6.1 Sensores

Un sensor es un dispositivo que detecta y responde a algún tipo de entrada del medio físico. La entrada específica podría ser: luz, calor, movimiento, humedad, presión, o cualquier otro de un gran número de fenómenos medioambientales. La salida es por lo general una señal que se convierte en un valor legible para humanos, que a menudo se posiciona en la ubicación del sensor o se transmite electrónicamente a través de una red para la lectura o procesamiento adicional.

Para que un hogar sea inteligente se necesita información, esta información por lo general será recogida directamente por el hogar con la ayuda de sensores, para utilizar funciones automatizadas. Los sensores son, por tanto, la primera piedra de cualquier sistema de automatización del hogar. Debido a esto, la variedad y calidad de los sensores que se decida instalar determinarán lo que el sistema de automatización sea capaz de hacer y la eficacia con la que opere.

**Sensor de Movimiento:** una de las tecnologías de sensores más populares para los sistemas de automatización doméstico es el detector de movimiento. A menudo, éstos son puestos en múltiples usos, como: la activación de una alarma antirrobo si se detecta movimiento cuando la casa se supone que debe estar vacía, o cambiando automáticamente las luces de encendido y apagado cuando se entra o sale de una habitación. (Hubpages, 2014).

**Sensores de Luz:** Un sensor de luz también puede ser conocido como un fotosensor o fotodiodo. Se utiliza para controlar los niveles de luz ambiental e informar de vuelta a su controlador de automatización del hogar. Esto se utiliza a menudo en combinación con un sensor de movimiento o presencia para cambiar las luces automáticamente cuando alguien entra en una habitación - pero sólo si son necesarios. También pueden ser utilizados para asegurar que las luces de seguridad

sólo funcionan durante la noche, o hacer que la iluminación exterior se enciendan automáticamente al anochecer. (Future Electronics, 2010).

**Sensores de temperatura y humedad:** En combinación con un sensor de humedad, pueden ser utilizados para controlar automáticamente acondicionadores de aire, o incluso para abrir automáticamente una ventana si una habitación comienza a volverse demasiado húmeda. Sensores de temperatura a prueba de agua pueden ser utilizados al aire libre obteniendo datos climáticos en tiempo real.

**Sensores de alarmas contra incendios:** Hay tres tipos principales de sensores de alarma contra incendios que se utilizan en propiedades domésticas (Safelincs, 2016):

- **Óptico / fotoeléctrico:** Este es el tipo más común. Utiliza haces de luz, y la alarma se activa si las partículas de humo interrumpen el haz de luz.
- **Ionización:** Estos detectan partículas ionizadas en el aire, y son más sensibles que los del tipo óptico. Esto puede significar, sin embargo, que es más propenso a falsas alarmas que un sensor óptico. Muchos sistemas modernos usan una combinación de óptico y de ionización.
- **Calor:** detecta temperaturas anómalas.

**Sensores de proximidad:** En algunos sistemas, sensores de proximidad se utilizan en lugar de interruptores. Esto permitirá al usuario simplemente mover su mano sobre un sensor montado en la pared para, por ejemplo, encender las luces, en lugar de tener que pulsar un interruptor. (The Gadget Square, 2013).

**Sensores de contacto:** Los sensores de contacto se utilizan para una variedad de aplicaciones, la más común, en alarmas contra robo. El sensor en sí es básicamente un tipo de interruptor, que envía una señal eléctrica cuando dos superficies hacen contacto. Pueden ser utilizados para monitorear si una puerta o ventana está abierta o cerrada. Sistemas de seguridad de hogares suelen utilizar sensores de contacto en marcos de las ventanas, para disparar una alarma si se abre una ventana después de

que la alarma ha sido activada. (SimpliSafe, 2013). Hay muchas otras aplicaciones también, por ejemplo:

- Apagado automático de la calefacción en una habitación si alguien abre una ventana, para evitar el derroche de energía y así también facilitar el proceso de enfriamiento de la habitación.
- Si se instalan sistemas de puertas accionados, se puede comprobar si las puertas han quedado abiertas y cerrarlas desde una ubicación remota.
- Encendido de la luz del armario cuando se abre la puerta.

**Micrófonos:** Una de las características más agradables que se puede construir en un sistema de automatización del hogar es el control por voz. A pesar de que el reconocimiento de voz ha existido desde hace algún tiempo, aun no se ha construido un sistema de micrófonos ambientales que permita usarlo en una habitación completa.

### 1.6.2 Actuadores

La integración de los actuadores en las redes de sensores, es a menudo considerado como el siguiente paso lógico en la evolución de las redes de sensores inalámbricos, en la automatización de un hogar.

La mayor parte de los actuadores que se usarán en este sistema están ligados directamente con el tipo de dispositivo al que se desee controlar, por ejemplo, para controlar una cafetera es necesario diseñar un circuito que nos permita controlar este electrodoméstico con sus distintas funciones, permitiendo controlarla de manera remota. En la Figura 1.10 se puede ver la estructura para convertir en dispositivos inteligentes a los distintos electrodomésticos del hogar.

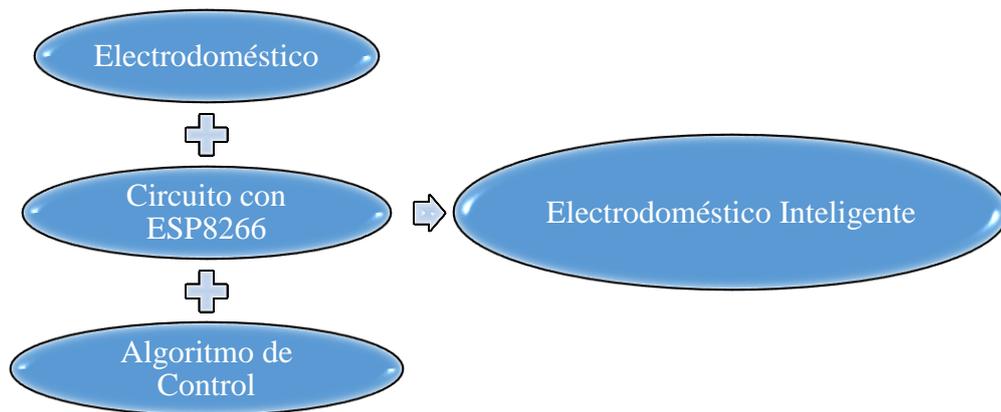


Figura 1.9: Esquema a instalarse en los distintos dispositivos.

### 1.7 Conclusiones

El módulo ESP8266 será el dispositivo principal para conectar a la red principal a la mayor parte de dispositivos inteligentes, así mismo, el SP debe contar con una conexión a WiFi para comunicarse con cada uno de los dispositivos inteligentes, dado que estos dispositivos forman la estructura fundamental del sistema domótico, es importante brindar una mayor robustez tanto al SP como a módulos ESP8266 por medio de:

- Acuses de recibo: necesario para saber que se entregó la información necesaria.
- Temporizadores: esta característica del módulo ESP8266 se utiliza para evitar que el mismo se quede en un bucle infinito, interrumpiendo las acciones que requiera el usuario.
- Reinicio remoto: el cual nos permite reiniciar dichos dispositivos en caso de que el mismo deje de funcionar.

## 2 CAPÍTULO II: DESARROLLO DEL ASISTENTE VIRTUAL (AV<sup>4</sup>).

### 2.1 Introducción

Como se definió en el capítulo anterior para la programación del AV se utilizara Python 2.7, la estructura principal del asistente virtual se basa en la siguiente Figura:

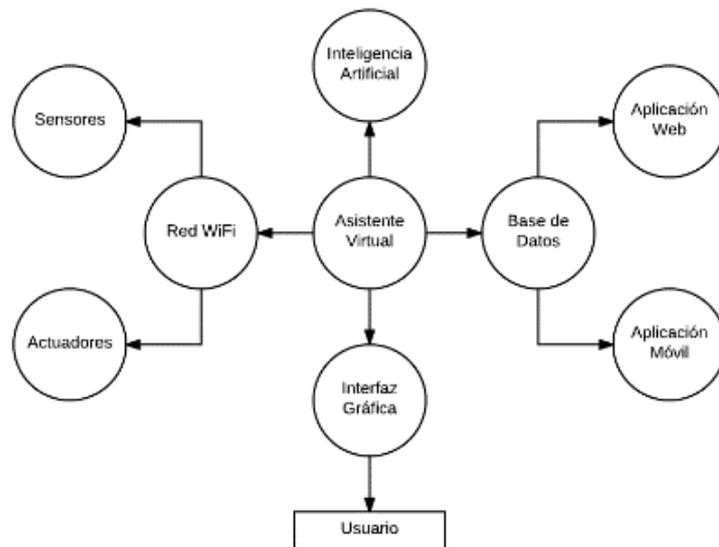


Figura 2.1: Estructura del Asistente Virtual

### 2.2 Interfaz Gráfica

Para el desarrollo de la interfaz gráfica se utilizará la librería Pygame, la cual nos proporciona las herramientas necesarias para crear una interfaz gráfica adaptable, atractiva y sobre todo de fácil manejo para los distintos usuarios.

Lo que se debe definir antes de comenzar con la programación del AV, son las distintas tareas que va a realizar, las cuales se pueden observar en la Figura 2.2:

---

<sup>4</sup> Asistente Virtual

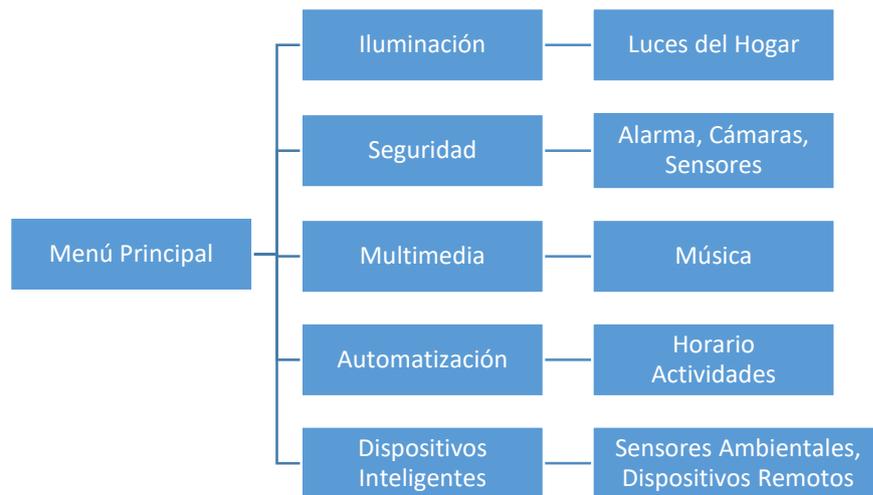


Figura 2.2: Esquema de tareas a realizar por el AV.

Definidas las tareas que va a realizar, se explicará el alcance de cada uno de ellas.

### 2.2.1 Iluminación

Esta opción se basa en la información entregada por la base de datos, para establecer el número de luces que están automatizadas y el nombre de cada una, para generar una matriz de botones como se puede ver en la Figura 2.3, lo cual nos permitirá automatizar más luces, con solo agregar a esta matriz.



Figura 2.3: Matriz de iluminación generada a partir de la base de datos.

### 2.2.2 Seguridad

Esta opción al igual que la anterior se basa en la información entregada por la base de datos para generar una matriz con los distintos sensores como se puede ver en la Figura 2.4. Para realizar la activación de la alarma el asistente primero realiza un sondeo de todos los sensores del hogar para asegurar que todas las puertas y ventanas estén cerradas, de no ser así, el sistema de alarma no se activará, permitiendo al usuario visualizar la ubicación del sensor que no está cerrado.

El sistema de alarma contará con distintas características que brindarán una mayor autonomía y seguridad al hogar:

- **Verificación de sensores en tiempo real.-** Esta característica ya fue explicada anteriormente.
- **Encendido Remoto.-** La activación del sistema de alarma se puede iniciar desde el móvil, desde la página web o simplemente con un mensaje de texto utilizando la red GSM, lo cual entrega mayor autonomía al hogar permitiendo activar el sistema de alarma desde cualquier lugar.
- **Sistema de Alarma silenciosa.-** Cuando el sistema determine un intruso, el asistente realiza una serie de actividades pre-programadas como: encender las luces de la entrada, encender la TV o encender la música, lo cual simula la presencia de personas en el hogar. Paralelamente, el asistente envía una notificación a los *email* y los números precargados al sistema, indicando el sensor que fue activado y adjuntando un video de la detección del intruso. Al tener una interfaz GSM el sistema puede enviar una notificación a la policía indicando la ubicación exacta del hogar.

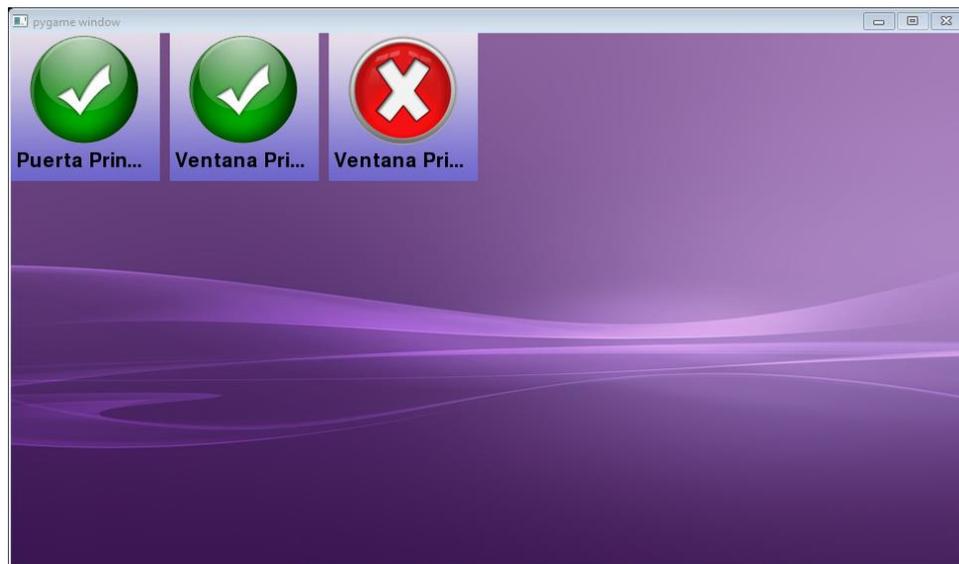


Figura 2.4: Matriz de sensores generada a partir de la base de datos.

### 2.2.3 Multimedia

Esta opción nos permite acceder a la memoria del sistema para extraer los archivos multimedia, con el cual el sistema es compatible, disponiendo al usuario de todos estos archivos para su reproducción como se puede ver en la Figura 2.5.



Figura 2.5: Interfaz gráfica multimedia.

## 2.2.4 Automatización

Esta opción nos permite automatizar las distintas acciones del hogar según un horario, por ejemplo, se puede indicar al sistema que todos los días a las 7 de la mañana encienda las luces de la habitación principal así como la música en esa habitación, el sistema guarda esta información y se muestra al usuario para que pueda editar, borrar o agregar más información al sistema como se puede apreciar en la Figura 2.6.

Además el AV haciendo uso de inteligencia artificial puede agregar tareas automáticamente que se adaptan a la tareas que realiza el usuario de manera cotidiana. Por ejemplo, si los usuarios del hogar todos los días a las 18:00 encienden las luces de la entrada, el sistema pregunta al usuario si desea ingresar dichas tareas al horario de automatización de la misma, esta característica entrega al hogar una mayor autonomía.

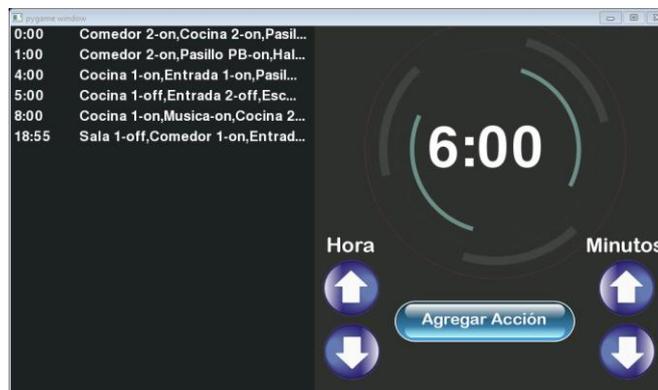


Figura 2.6: Interfaz automatización del hogar.

## 2.2.5 Dispositivos Inteligentes

Esta opción nos permite acceder a los distintos dispositivos que se encuentren dentro del hogar como: lavadora, licuadora, cafetera, equipos de sonido, entre otros. Además permite al usuario revisar un resumen de los sensores ambientales que posea el hogar, permitiendo al usuario saber las condiciones ambientales del hogar en tiempo real.

### 2.3 Base de Datos

Para la elección de la base de datos adecuada se analizará distintas características:

- **Software Libre:** tener esta característica nos permite bajar el precio de implementación del mismo, por el hecho de que no se debe pagar ningún tipo de licencias por el mismo.
- **Multiusuario:** esta característica nos permite que el servidor se conecte a la base de datos de manera local, además de permitir conexiones remotas del mismo para realizar cambios en la base de datos de manera remota.
- **Multihilo:** esta característica es muy importante porque nos permite que los distintos dispositivos o interfaces puedan conectarse y tratar con la base de datos directamente, esta generalización de la base de datos permite actualizar en tiempo real todas las variables del sistema.

MySQL cumple con todas estas características, además de tener una librería en Python para enlazarnos por medio de programación con las distintas tablas que se van a utilizar para el desarrollo del AV. La estructura en la que se basa dicha base de datos se puede apreciar en la Figura 2.7:



Figura 2.7: Estructura Base de Datos.

Para el manejo optimizado de la base de datos se ha creado un programa que sirva como puente para la comunicación del AV con la base de datos que está basado en comando MySQL, a continuación se presenta algunas de esas sentencias para un mejor entendimiento:

Utilizando las funciones de la librería MySQLdb que se puede utilizar en Python, se puede realizar la comunicación con la base de datos basándose en la sintaxis usada en MySQL, como se puede ver en el código anterior para inicializar la conexión es necesario ejecutar el comando “connect” y proporcionar tanto la dirección de la base de datos como la información de usuario y contraseña.

```
import MySQLdb

def init_sql():
    global db
    global cursor
    db = MySQLdb.connect("localhost","root","","asistente" )
    cursor = db.cursor()
```

Figura 2.8: Idle Python27, algoritmo para inicializar base de datos

La sintaxis que se presenta ahora nos permite actualizar cualquier campo existente en la base de datos, el algoritmo está estructurado de manera que solo se debe proporcionar el nombre de la tabla y el campo a seleccionar y obviamente el valor por el que se desea cambiar.

```
def update(tabla,Nombre,Estado):
    sql = "UPDATE %s SET Estado = %i WHERE Nombre LIKE '%s'" % (tabla,Estado,Nombre)
    try:
        cursor.execute(sql)
        db.commit()
    except:
        db.rollback()
```

Figura 2.9: Idle Python27, algoritmo para actualizar datos

Para eliminar cualquier dato se presenta el siguiente algoritmo, al cual se le debe proporcionar el nombre de la tabla y el nombre del dato que se desea eliminar.

```
def eliminar(tabla,Numero):
    sql = "DELETE FROM %s WHERE %i" % (tabla,Numero)
    try:
        cursor.execute(sql)
        db.commit()
    except:
        db.rollback()
```

Figura 2.10: Idle Python27, algoritmo para eliminar datos

Por último, para insertar nuevos datos se presenta el siguiente algoritmo, al cual se le debe proporcionar la información completa de cada dato ingresado según los campos que se tengan en las distintas tablas. Para mayor información acerca de este código revisar anexo 1.

```
def insertar(tabla, Numero, Nombre, Tipo, Estado):
    sql = "INSERT INTO %s(`Numero`, `Nombre`, `Tipo`, `Estado`) VALUES (%i,%s,%s,%i)"
    % (tabla, Numero, Nombre, Tipo, Estado)
    try:
        cursor.execute(sql)
        db.commit()
    except:
        db.rollback()
```

Figura 2.11: Idle Python27, algoritmo para insertar datos

## 2.4 Comunicación Wifi

Como se determinó en el capítulo anterior, la comunicación que tendrá el AV con el hogar y a su vez con el usuario se realizara de una manera inalámbrica por medio de una red WiFi. Los módulos ESP8266 12-E establecerán la comunicación tanto de sensores como de actuadores con el AV. Este tema se desarrollará con mayor detalle en el siguiente capítulo.

## 2.5 Inteligencia Artificial (IA<sup>5</sup>)

Una de las características más importantes de un AV es la capacidad de tomar decisiones basado en distintas reglas o basadas en acciones que se realizan con frecuencia por el usuario, este tipo de características son propias de un Sistema Experto como lo cita (Álvarez, 2004).

### 2.5.1 Sistemas de Inteligencia Artificial.

- **Luz Ambiente:** se optará por un sistema difuso Figura 2.12 el cual nos permite no solo obtener una salida de “nivel\_luz”, para realizar un sistema de control de luz basada en las reglas que se pueden apreciar en la Figura 2.13, sino que nos entrega un valor lingüístico proporcional al nivel de luz ambiental (Figuras 2.14 y 2.15), lo cual ayuda a la interacción del AV con los usuarios del hogar, una explicación más centrada en el tema de la lógica difusa puede ser encontrado en (Vito, 2006)

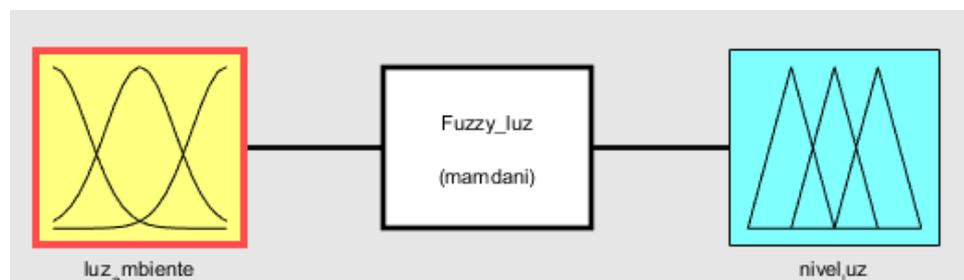


Figura 2.12: Estructura lógica difusa basada en Mandani para sistema de Luz Ambiente.

```

1. If (luz_ambiente is Muy_Bajo) then (nivel_luz is Muy_Alto) (1)
2. If (luz_ambiente is Bajo) then (nivel_luz is Alto) (1)
3. If (luz_ambiente is Medio) then (nivel_luz is Medio) (1)
4. If (luz_ambiente is Alto) then (nivel_luz is Bajo) (1)
5. If (luz_ambiente is Muy_Alto) then (nivel_luz is Muy_Bajo) (1)

```

Figura 2.13: Reglas control difuso de Iluminación extraídas de Matlab.

<sup>5</sup> IA: Inteligencia Artificial.

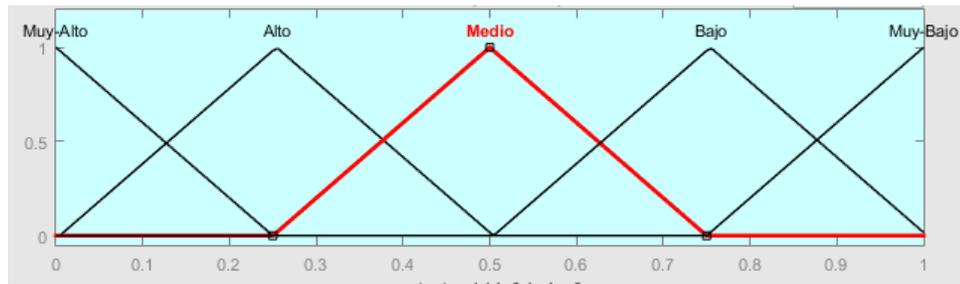


Figura 2.14: Valor lingüístico de Iluminación ambiental extraídas de Matlab.

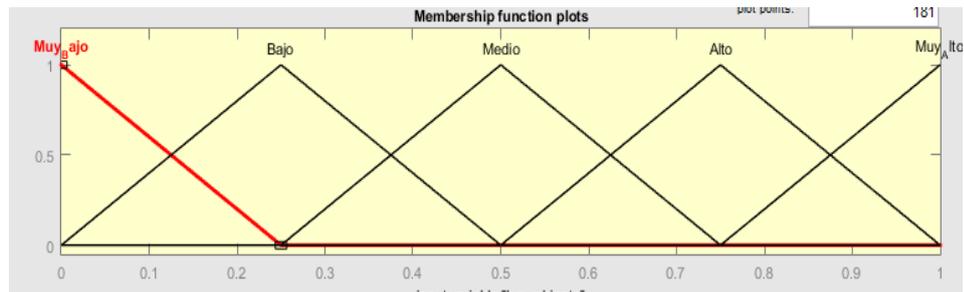
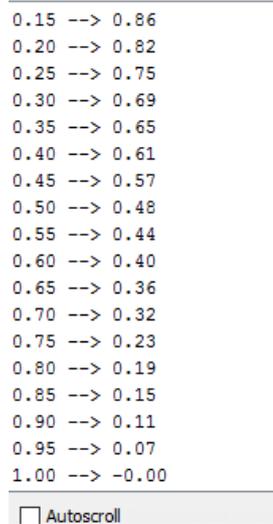


Figura 2.15: Valor lingüístico de iluminación interior extraídas de Matlab.

- Implementación:** este sistema funciona de manera independiente al AV, se implementará la lógica difusa en un microcontrolador, en este caso se hará uso de Arduino. Para realizar su implementación se hará uso del siguiente algoritmo que se puede visualizar en el anexo 6, el cual nos permite realizar un control de la iluminación interna del hogar optimizando el gasto energético ya que se mantiene un equilibrio entre la iluminación externa y la iluminación interna.

Como se puede ver en la Figura 2.16 el sistema difuso funciona de excelente manera, la salida es inversamente proporcional a la entrada proporcionada por el sensor de luz ambiental.



```
0.15 --> 0.86
0.20 --> 0.82
0.25 --> 0.75
0.30 --> 0.69
0.35 --> 0.65
0.40 --> 0.61
0.45 --> 0.57
0.50 --> 0.48
0.55 --> 0.44
0.60 --> 0.40
0.65 --> 0.36
0.70 --> 0.32
0.75 --> 0.23
0.80 --> 0.19
0.85 --> 0.15
0.90 --> 0.11
0.95 --> 0.07
1.00 --> -0.00
```

Autoscroll

Figura 2.16: Monitor Serial IDE Arduino mostrando datos del sistema difuso.

El código completo se puede encontrar en el anexo 2.

**Automatización por Horario:** este sistema está dividido en dos ámbitos: horario y aprendizaje.

- **Horario:** este ámbito está referido directamente a la interacción del usuario con el AV por medio de la interfaz gráfica como se puede apreciar en la Figura 2.6.
- **Aprendizaje:** este ámbito está referido directamente a las acciones más recurrentes que realiza el usuario en el hogar, por ejemplo si el usuario cada que entra a su hogar a las 18:00 enciende la luz de la entrada, enciende la TV y se prepara un café, entonces el AV notificará al usuario si desea agregar dichas tareas al horario de automatización. Este aprendizaje ayudará a los usuarios del hogar con la automatización de tareas y al AV a tener más información acerca de los gustos de los distintos usuarios del hogar, el cuál con ayuda de reconocimiento facial puede llegar a crear un perfil de gustos de cada usuario. Una estructura resumida puede ser observada en la Figura 2.17.

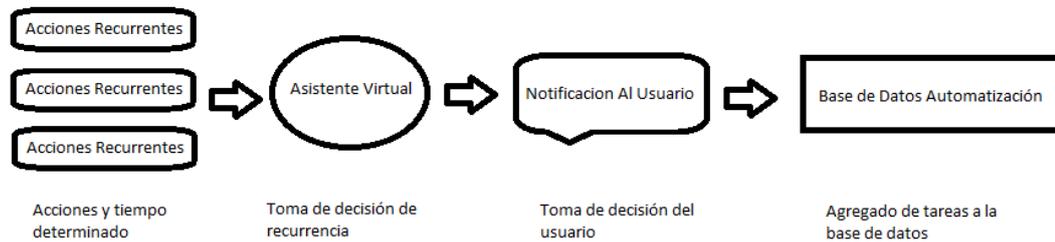


Figura 2.17: Estructura para agregar acciones a la función horario de automatización.

- **Perfil personal:** este sistema consiste en dos etapas importantes: aprendizaje y perfil.
- **Aprendizaje:** en esta etapa el AV primero realizará un entrenamiento facial, que consiste en tomar un número determinado de muestras del usuario para posteriormente, usando algoritmos basados en OpenCV (que se explicará más adelante), crear un patrón que permita reconocer facialmente a cada uno de los usuarios del hogar.
- **Reconocimiento facial:** para el reconocimiento facial se utilizará Open CV 2.4 en conjunto con Python 2.7, dicha librería nos proporciona dos algoritmos: EigenFaces y FisherFaces. Estos dos tipos de algoritmos nos proporcionan un método de reconocimiento de rostro basado en varias muestras, lo cual facilita el reconocimiento ante eventos exteriores tales como cambios en la expresión facial y cambios en la iluminación. (Yang, 2012)
  - **EigenFaces:** Es un algoritmo basado en el Análisis de Componentes Principales (PCA), el principio matemático de este algoritmo se indica detalladamente en (Dev, 2014), por lo que en esta investigación se centrará directamente en la aplicación del algoritmo para la implementación con el AV.

- **FisherFaces:** al igual que el algoritmo indicado anteriormente, este también se basa en PCA, pero integra un análisis discriminante lineal lo cual nos ayuda a diferenciar de mejor manera entre los rostros con facciones parecidas. (Dev, 2014)

Ambos algoritmos tienen una etapa de entrenamiento en el cual el usuario ingresa el número de muestras que el AV utilizará para realizar el reconocimiento. En el anexo 3 se puede encontrar el algoritmo utilizado para el entrenamiento de EigenFaces y en el anexo 4 para FisherFaces. En el anexo 5 se puede analizar el algoritmo combinando los dos tipos de reconocimiento en el cual se elige el que tiene mayor grado de aceptación. A continuación se muestra un flujograma en el cual se puede analizar la etapa de entrenamiento (Figura 2.18) y reconocimiento facial (Figura 2.19).

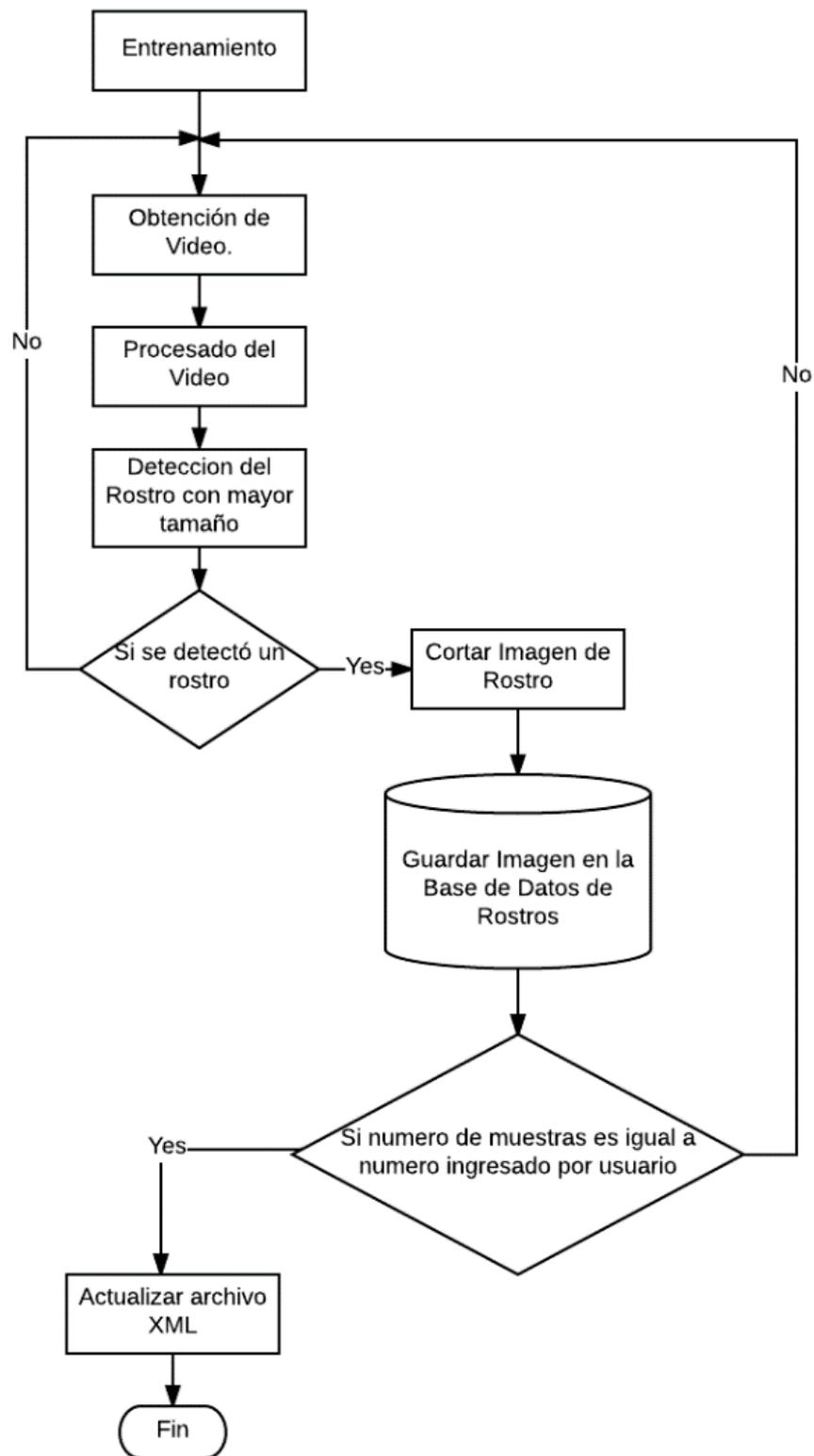


Figura 2.18: Diagrama de Flujo para entrenamiento Facial

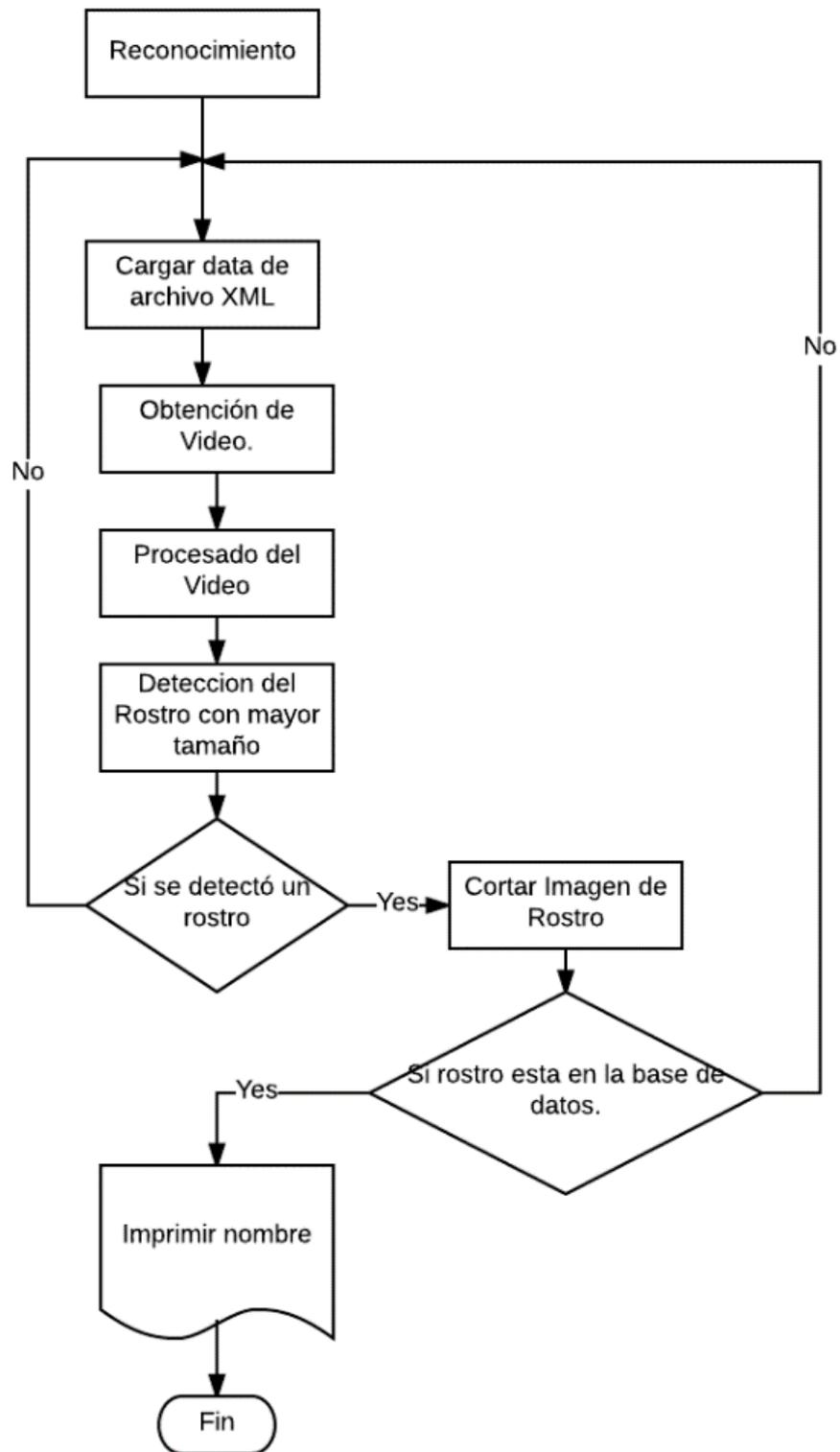


Figura 2.19: Diagrama de Flujo para Reconocimiento Facial.

El sistema integrado en el AV combinará los dos tipos de algoritmos confiando en la respuesta que entregue mayor grado de confiabilidad. Para ello se debe entrenar al AV utilizando ambos algoritmos como se puede ver en la Figura 2.20, dichos algoritmos crean una base de datos de las distintas facciones del rostro con un número determinado de muestras impuestas que para este caso es 100 (Figura 2.21).

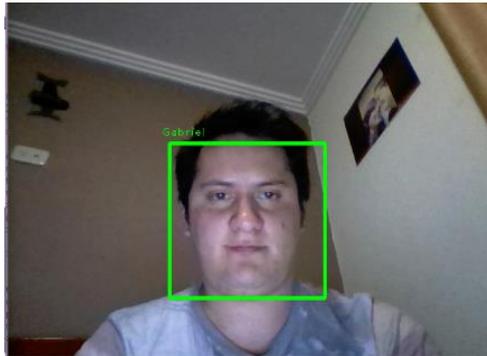


Figura 2.20: Entrenamiento para reconocimiento facial.



Figura 2.21: Base de datos faciales

Dicho entrenamiento se realiza con todos los usuarios del hogar, los cuales nos producen un error igual al 30% en los usuarios que son familia primaria, por lo que se optará por un sistema híbrido entre los dos algoritmos mencionados anteriormente.

**Perfil:** como ya se explicó anteriormente, el AV continuamente hace una recopilación de las distintas tareas que realizan los usuarios para crear un perfil con

el comportamiento de cada uno, para posteriormente automatizar tareas que el usuario realiza durante el día, entregando mayor autonomía al AV con el usuario.

**Aplicaciones con reconocimiento facial:** las ventajas que tiene el sistema al reconocer facialmente a los usuarios del hogar son varias. Por ejemplo, al ingresar al hogar el AV entrega un tiempo para desactivar la alarma antes de que se active la misma, una de las maneras para desactivar la alarma puede estar basado en reconocimiento facial, de manera que cuando el AV reconozca el rostro de cualquier usuario del hogar pueda no solo desactivar la alarma automáticamente sino también automatizar las rutinas que tiene dicho usuario al ingresar al domicilio.

Otra de aplicaciones basadas en reconocimiento facial es, además de crear un perfil del usuario respecto a las tareas que realiza se puede integrar un perfil con las distintas características de la persona como puede ser su peso, contextura, estatura, ritmo cardíaco y demás factores que integrando varios tipos de sensores ayudan a recopilar más información acerca del usuario, creando un historial del usuario mejorando aún más la interacción con el AV.

- **Reconocimiento de voz:** el AV puede realizar varias acciones, la base de datos para el reconocimiento de voz se extiende de manera proporcional al número de tareas que el mismo puede realizar, al extenderse la cantidad de datos la búsqueda toma más tiempo, factor que retarda la interacción del AV con el usuario, por esta razón se hará uso de la Heurística, creando un sistema que a medida que el usuario utilice las distintas tareas, el AV va creando un perfil de recurrencia de cada una de estas, posicionando en los primeros puestos a las tareas más recurrentes, esto ayuda a la interacción con el usuario ya que las tareas más recurrentes se tardarán menos en ser localizadas en la base de datos.
- **Sistema de Riego Inteligente:** el AV al tener la información entregada por el módulo de sensores ambientales, indicado anteriormente, está en la capacidad de analizar dichas variables para actuar sobre un conjunto de vegetación de distintos tipos, por ejemplo el AV analiza el nivel de radiación solar recibida por la

vegetación, para posteriormente saber la cantidad de agua que debe recibir dicha vegetación, para ello, se hará uso de un sistema difuso basado en las reglas visualizadas en la Figura 2.23, el sistema difuso tendrá dos entradas que son características importantes para definir la cantidad de agua que necesita la vegetación, estas características son la Humedad (Figura 2.26) y el nivel de Radiación Solar (Figura 2.25), estas dos variables nos entregarán como salida para el nivel difuso, el tiempo de riego (Figura 2.24), este esquema se ve resumido en la Figura 2.22:

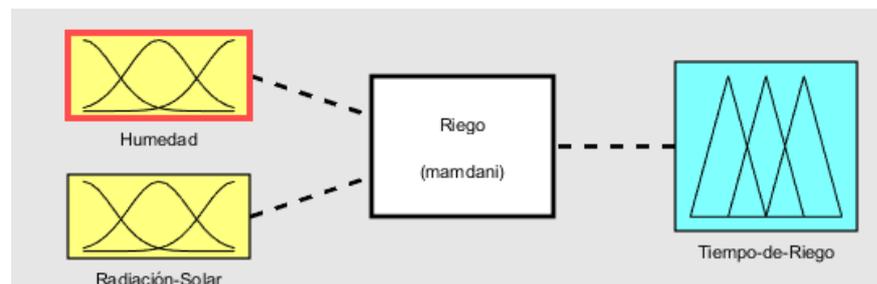


Figura 2.22: Estructura lógica difusa basada en Mandani para sistema de Riego Inteligente.

1. If (Humedad is Muy-Bajo) and (Radiación-Solar is Muy-Bajo) then (Tiempo-de-Riego is Muy-Alto) (1)
2. If (Humedad is Medio) and (Radiación-Solar is Medio) then (Tiempo-de-Riego is Medio) (1)
3. If (Humedad is Bajo) and (Radiación-Solar is Bajo) then (Tiempo-de-Riego is Alto) (1)
4. If (Humedad is Alto) and (Radiación-Solar is Alto) then (Tiempo-de-Riego is Bajo) (1)
5. If (Humedad is Muy-Alto) and (Radiación-Solar is Muy-Alto) then (Tiempo-de-Riego is Muy-Bajo) (1)
6. If (Humedad is Alto) and (Radiación-Solar is Muy-Alto) then (Tiempo-de-Riego is Muy-Bajo) (1)
7. If (Humedad is Medio) and (Radiación-Solar is Muy-Alto) then (Tiempo-de-Riego is Bajo) (1)
8. If (Humedad is Bajo) and (Radiación-Solar is Muy-Alto) then (Tiempo-de-Riego is Medio) (1)
9. If (Humedad is Muy-Bajo) and (Radiación-Solar is Muy-Alto) then (Tiempo-de-Riego is Alto) (1)
10. If (Humedad is Muy-Alto) and (Radiación-Solar is Alto) then (Tiempo-de-Riego is Muy-Alto) (1)
11. If (Humedad is Medio) and (Radiación-Solar is Alto) then (Tiempo-de-Riego is Alto) (1)
12. If (Humedad is Bajo) and (Radiación-Solar is Alto) then (Tiempo-de-Riego is Medio) (1)
13. If (Humedad is Muy-Bajo) and (Radiación-Solar is Alto) then (Tiempo-de-Riego is Bajo) (1)
14. If (Humedad is Muy-Alto) and (Radiación-Solar is Medio) then (Tiempo-de-Riego is Alto) (1)
15. If (Humedad is Alto) and (Radiación-Solar is Medio) then (Tiempo-de-Riego is Medio) (1)
16. If (Humedad is Bajo) and (Radiación-Solar is Medio) then (Tiempo-de-Riego is Bajo) (1)
17. If (Humedad is Muy-Bajo) and (Radiación-Solar is Medio) then (Tiempo-de-Riego is Muy-Bajo) (1)
18. If (Humedad is Muy-Alto) and (Radiación-Solar is Bajo) then (Tiempo-de-Riego is Alto) (1)
19. If (Humedad is Alto) and (Radiación-Solar is Bajo) then (Tiempo-de-Riego is Medio) (1)
20. If (Humedad is Medio) and (Radiación-Solar is Bajo) then (Tiempo-de-Riego is Bajo) (1)
21. If (Humedad is Muy-Bajo) and (Radiación-Solar is Bajo) then (Tiempo-de-Riego is Muy-Bajo) (1)

Figura 2.23: Reglas control difuso extraídas de Matlab.

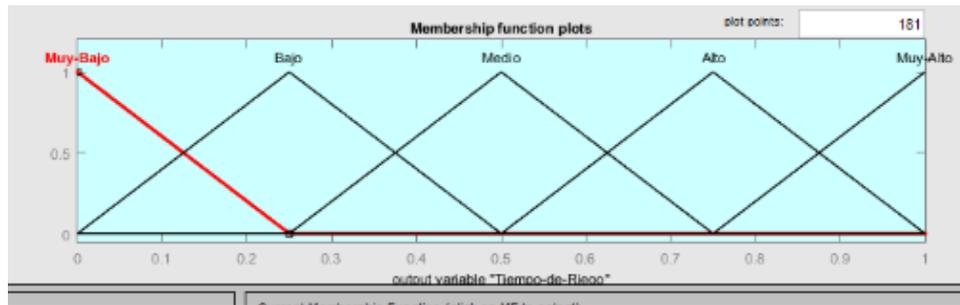


Figura 2.24: Valor lingüístico de Tiempo de Riego extraídas de Matlab. (Salida)

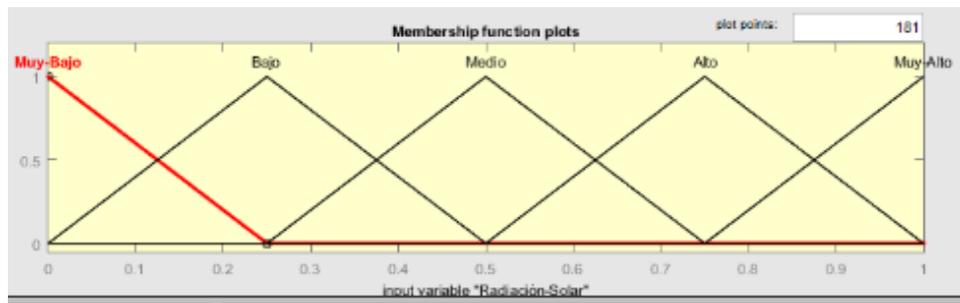


Figura 2.25: Valor lingüístico de Radiación Solar extraídas de Matlab. (Entrada)

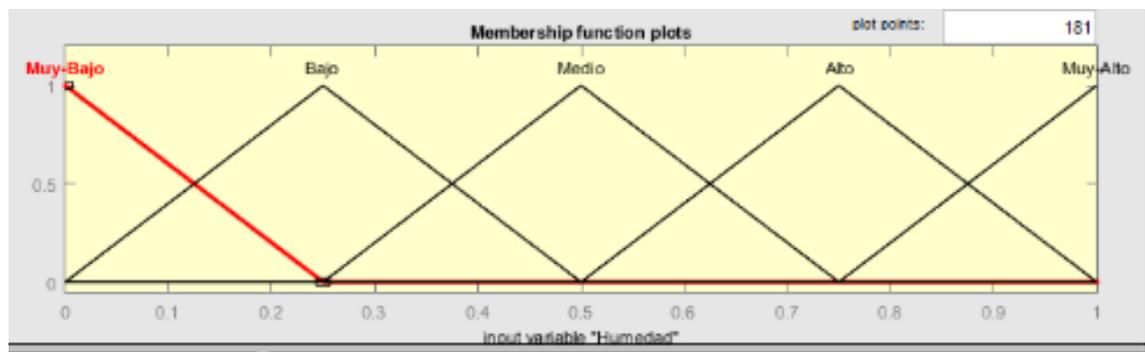


Figura 2.26: Valor lingüístico de Humedad extraídas de Matlab. (Entrada).

- **Sensores para Sistema de Riego:** Para el sistema de riego y de acuerdo a las dos entradas que posee el sistema difuso se hará uso de dos tipos de sensores, un sensor de humedad como se puede ver en la Figura 2.27 y un sensor de radiación solar que para este caso se hará uso de un panel solar pequeño. Esto se debe a que se quiere entregar un mayor grado de autonomía al sistema de riego inteligente. Además el

sistema de Riego Inteligente se volverá autosustentable lo cual coopera con el gasto energético total del hogar. En la Figura 2.28 se puede analizar estructura del sistema.

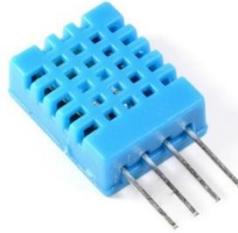


Figura 2.27 Sensor de Humedad DHT11

Fuente: (D-Robotics, 2010)

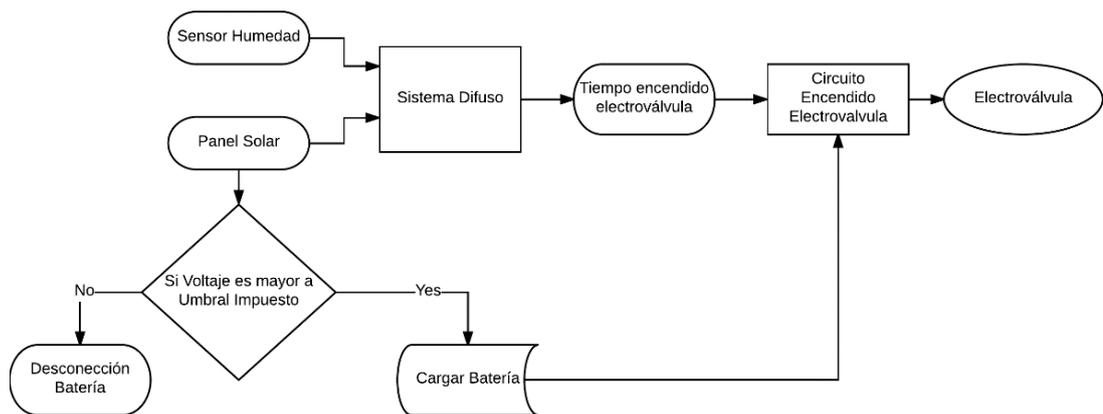


Figura 2.27: Estructura Sistema de Riego Inteligente

### 2.5.2 Reglas para el diseño de un Sistema Experto.

Las principales reglas en las que se va a basar el AV para tomar decisiones son:

- **Iluminación:** Esta regla hace referencia a dos características:

*hora\_actual*: cuya fuente de información principal es la hora del servidor.

*luz\_ambiente*: cuya fuente de información está basada en el sistema difuso de la Figura 2.14, creado a partir de los sensores de iluminación que se encuentran posicionados en los alrededores del hogar.

*if (hora\_actual > 18:00 and luz\_ambiente == "Nivel Bajo")*

*then: (Encender Luces de Entrada)*

- **Horario:** Esta regla se basa en la base de datos de automatización, en el cual el AV analiza la tarea más cercana al horario actual y realiza una comparación con la hora actual en el sistema para realizar la tarea que haya sido agregado a ese horario.

*if (hora\_actual == hora BD automatizacion)*

*then: (Tarea BD automatizacion)*

- **Aprendizaje Horario:** Esta regla está ligada a la regla horario ya que analiza el número de veces que se solicita al AV realizar una determinada tarea y se aumenta una variable que nos indica que dicha tarea es recurrente en un mismo horario. Para ingresar una tarea a la base de datos de automatización se usa la siguiente regla.

*if (frecuencia tarea actual > 5 and horario tarea actual == horario BD)*

*then: (Agregar tarea actual a base de datos de automatización)*

- **Reconocimiento Facial:** Esta regla analiza con visión artificial el rostro del usuario que se presente en frente a una determinada cámara, el AV realiza la comparación con la base de datos facial para saber si reconoce al usuario. Para esto, la siguiente regla es utilizada.

*if (rostro\_actual find base de datos facial > 1)*

*then: (Cargar Perfil Usuario)*

else: (Solicitar datos para entrenamiento con dicho rostro)

- **Base de datos Reconocimiento de voz:** esta regla hace referencia a mejorar el tiempo de búsqueda de las distintas tareas a realizar o en su defecto tener una base de datos con sugerencias por parte del usuario para implementar nuevas tareas, esta comparación se basa en la siguiente regla.

if (tarea\_solicitada find BD tareas > 1 )

then: (frecuencia tarea +=1)

else: (Agregar tarea\_solicitada a sugerencias.)

## 2.6 Conclusiones

La interfaz gráfica es una característica muy importante en este tipo de sistemas ya que es la presentación principal para el usuario, el cual le va permitir interactuar tanto con el asistente virtual como también con el hogar y sus dispositivos inteligentes, por este hecho se ha diseñado un entorno virtual amigable y de fácil manejo.

Cada dispositivo inteligente que funcione con cualquiera de los sistemas difusos planteados anteriormente, enviarán los datos de manera porcentual, es decir, si el sistema difuso tiene como entrada la radiación solar, se realizará un previo análisis para saber cuál es la mayor cantidad de radiación solar que se tiene por día.

### **3 CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DE LA RED DE SENSORES Y ACTUADORES.**

#### **3.1 Introducción**

En el siguiente apartado se presentan los diferentes componentes, *software* y *hardware*, que son necesarios para cada uno de los elementos que forman parte del sistema. También se presenta la tecnología seleccionada para desarrollar cada una de las distintas tareas que realiza el sistema. De la misma manera se presenta un diagrama explicativo para cada *software* utilizado para la programación, tanto de sensores como de actuadores.

#### **3.2 Hardware**

##### **3.2.1 Microcontrolador**

Para interconectar los diferentes dispositivos inteligentes a la red, se utilizó microcontroladores, los cuales son circuitos integrados programables, en los que se ejecuta distintos códigos. En este caso, estos códigos deberán establecer la conexión con el servidor, reconocer al elemento dentro del sistema, así como también enviar y recibir la información pertinente.

Para el sistema se ha escogido el módulo ESP8266-12E (Figura 3.1), el cual incluye un microcontrolador para manejar el protocolo TCP/IP y el *software* necesario para la conexión 802.11, el modelo escogido dispone de entradas/salidas (I/O) digitales y también una entrada analógica al igual que otros microcontroladores, su punto fuerte es disponer de acceso WIFI. A continuación se detalla algunas de sus características más importantes (Shenzen Anxinke Technology, 2017) :

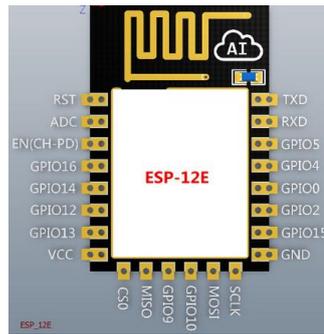


Figura 3.1: ESP8266-12E

Fuente: (Shenzen Anxinke Technology, 2017)

- Procesador interno de 32 bits a 80 MHz y se le puede subir hasta 160Mhz.
- 80K de DRAM.
- 35K de IRAM , memoria rápida para el procesador.
- 1 Mb de memoria Flash para programas.
- Stack TCPIP WIFI a 2.4 Ghz.
- 30 Metros de alcance teórico.
- 16 pines GPIO programables disponibles
- Full TCPIP stack incluido.
- Gestión completa del WIFI con amplificador incluido.
- 802.11 protocol y Wifi Direct (P2P) Soft-AP.
- Regulador y unidad de alimentación incluidos.
- Consumo en reposo <10 mW.
- Soporta el bus SPI.
- 1 entrada analógica
- Alimentación de 3.3v

El módulo ESP8266-12E se puede programar usando varios lenguajes como: Lua, ESPlorer y el IDE de Arduino. Para una mayor comodidad se ha escogido Arduino IDE ya que tiene una gran comunidad de desarrolladores. Se ha elegido el módulo ESP8266-12E debido a su costo, sus medidas, su conexión inalámbrica, así como su

facilidad de programación, ya que se lo puede utilizar en modo estación y punto de acceso al mismo tiempo, como se lo había explicado en capítulos anteriores; además, es compatible con las tarjetas de expansión que han sido diseñadas para Arduino.

### 3.2.2 Servidor

Para ejecutar el servidor y por ende, los diferentes servicios disponibles, se necesita un ordenador en el cual ejecutar los diferentes sistemas y programas, se podría utilizar cualquier ordenador para este trabajo, pero debido a que el asistente virtual tendrá que estar en funcionamiento a lo largo de todo el día, es necesario un computador que tenga un consumo muy bajo.

En el mercado existen varios tipos de ordenadores disponibles, pero el que se ajusta a las necesidades del hogar por su tamaño, precio, capacidad de manejo y facilidad de acceso a sus diferentes componentes es la Raspberry Pi 3, el mismo que se puede apreciar en la Figura 3.2.

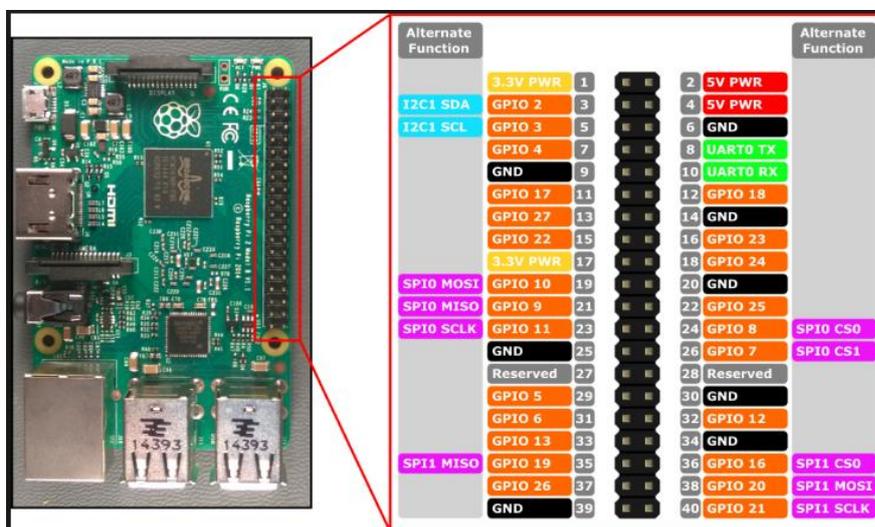


Figura 3.2: Raspberry Pi 3.

Fuente: (Prometec)

Las especificaciones del mini ordenador expuesto anteriormente son (Foundation, 2017):

- Procesador Broadcom BCM2837 a 1,2 GHz de 64 bits con cuatro núcleos ARMv8.
- RAM: 1 GB (compartidos con la GPU)
- Consumo Energetico: 800 mA, (4.0 W) a 5V
- Pines GPIO: 40 pines
- 802.11n Wireless LAN.
- Bluetooth 4.1.
- Bluetooth Low Energy (BLE).
- 4 puertos USB.
- 40 pines GPIO.
- Puerto Full HDMI.
- Puerto Ethernet.
- Conector combo compuesto de audio y vídeo de 3,5 mm.
- Interfaz de la cámara (CSI).
- Interfaz de pantalla (DSI).
- Ranura para tarjetas microSD (ahora push-pull en lugar de push-push).
- Núcleo de gráficos VideoCore IV 3D.
- Dimensiones de placa de 8.5 por 5.3 cm.

Raspberry Pi 3 es un miniordenador de bajo costo que posee los elementos necesarios que lo convierten en la mejor opción, funciona con una alimentación de 5v, lo que es un consumo muy bajo comparado con un computador convencional.

Para el sistema operativo se utilizara Ubuntu Mate, el cual es una distribución Linux basada en Ubuntu diseñado para procesadores ARM (Wimpress & Madhavan, 2017)

que posee controladores para Raspberry Pi 3, este OS<sup>6</sup> se utiliza para implementar el servidor, que como se explicó en el capítulo 2 posee una interfaz gráfica desarrollada en Python, para lo cual es necesario incluir una pantalla táctil de 5 pulgadas (Figura 3.3).

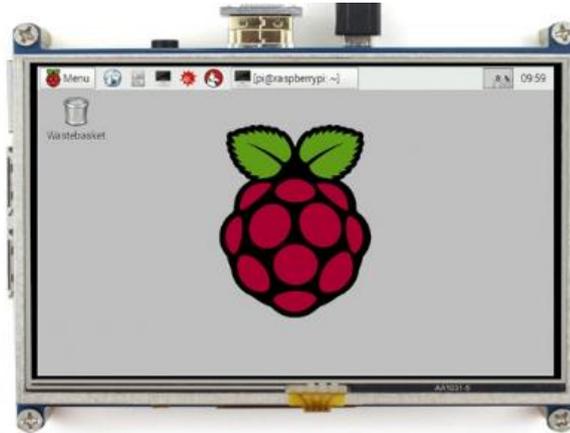


Figura 3.3: Waveshare HDMI TFT Resistiva

Fuente: (Waveshare, 2017).

La pantalla puede ser usada simplemente como un *display* debido a su conexión HDMI, pero también incluye una interfaz resistiva, la cual sirve como entrada táctil para el miniordenador. A continuación, algunas de sus características, además de su interfaz SPI (Waveshare, 2017):

- 800 × 480 de alta resolución, control táctil
- Conexión directa con Raspberry Pi 2 y 3
- HDMI para la visualización
- Control de luz trasera para reducir el consumo de energía.

En la Figura 3.4 se muestra la distribución de pines que posee esta pantalla.

---

<sup>6</sup> OS: Sistema Operativo

PIN NO.	SYMBOL	DESCRIPTION
1, 17	3.3V	Power positive (3.3V power input)
2, 4	5V	Power positive (5V power input)
3, 5, 7, 8, 10, 11, 12, 13, 15, 16, 18, 24	NC	NC
6, 9, 14, 20, 25	GND	Ground
19	TP_SI	SPI data input of Touch Panel
21	TP_SO	SPI data output of Touch Panel
22	TP_IRQ	Touch Panel interrupt, low level while the Touch Panel detects touching
23	TP_SCK	SPI clock of Touch Panel
26	TP_CS	Touch Panel chip selection, low active

Figura 3.4: Pines de conexión a la Raspberry pi

Fuente: (Waveshare, 2017)

En el sitio web del desarrollador de dicha pantalla (Waveshare, 2017), se puede encontrar los controladores correspondientes para ser instalados en Ubuntu Mate. El OS, así como los lenguajes de programación que se encuentran preinstalados en Ubuntu Mate son de código libre.

El lenguaje elegido es Python, que se caracteriza por poseer una gran comunidad de programadores de donde se puede obtener suficiente información acerca de proyectos realizados con la Raspberry Pi. Con este lenguaje se abre uno de los puertos del ordenador para recibir mensajes vía UDP<sup>7</sup>, además de poseer un HMI debido al uso una pantalla táctil.

### 3.2.3 Alarma

La interfaz con el usuario para el sistema de alarma también utilizará un Raspberry Pi 2, con una pantalla táctil capacitiva de 2.8 pulgadas. Este tipo de miniordenador es de una generación menor al antes descrito, por lo que para una mejor fluidez y rapidez al momento de interactuar con la pantalla táctil se ha escogido el sistema

---

<sup>7</sup> UDP: User Datagram Protocol

operativo *Raspbian*, este *software* es distribuido por la propia organización de Raspberry.

Por otra parte, para la pantalla táctil capacitiva se ha escogido al distribuidor y desarrollador de componentes electrónicos Adafruit. El desarrollador en su página web (Adafruit, 2017) proporciona los controladores correspondientes para utilizar la pantalla táctil conjuntamente con el sistema operativo. Al igual que la pantalla anterior, también utiliza el puerto SPI e I2C de la Raspberry, pero el HDMI queda libre para proyectar su contenido en un monitor cualquiera. En la figura 3.5 se muestra la imagen de la pantalla así como su interfaz de conexión:



Figura 3.5: Adafruit TFT Capacitiva

Fuente: (Adafruit, 2017)

Interfaz con Raspberry Pi y características (Adafruit, 2017):

- 320 × 240 de 16 bits y una superposición táctil capacitiva.
- Utiliza los pines: I2C (SDA y SCL), pines SPI (SCK, MOSI, MISO, CE0), así como GPIO 25 y 24, el resto de los pines no se utilizan.

Al igual que con la pantalla anterior, se desarrollará la interfaz en Python, abriendo un puerto para realizar una comunicación con mensajes vía UDP y así realizar una comunicación con el servidor, sensores de movimiento y sensores magnéticos de puertas y ventanas dispuestos en el hogar.

### 3.2.4 GSM

Se trata de un módulo GSM/GPRS compatible con Arduino, que será utilizado conjuntamente con el sistema de alarma del AV. El modulo utilizado es el SIM900 (Figura 3.6), el cuál funciona por medio de comandos AT que son enviados a través del puerto serial del ESP8266-12E. Los comandos AT pueden ser encontrados en la hoja de datos del módulo. El SIM900, posee una salida de audio y una entrada de voz que pueden ser utilizadas cuando se posee una llamada entrante, o se pretenda realizar una llamada de voz.

Además, se pueden enviar y recibir mensajes, leer su contenido y hasta saber qué número de teléfono es el que está llamando con un *caller ID*, todas estas opciones pueden ser configuradas con comandos AT enviados desde el ESP8266-12E.



Figura 3.6: Módulo GSM/GPRS SIM900

Fuente: (HetPro, 2017)

A continuación, algunas de las características del módulo GSM SIM900 (HetPro, 2017):

- Voltaje de operación: 5V~20V
- Comunicación: UART
- Bandas de Frecuencia: 850/900/1800/1900MHz
- Multi-GPRS Slot clase 10/8
- Estación móvil GPRS: Clase B

- Fase GSM:2/2+
- Clase 1: 1W a 1800/1900MHz
- Clase 4: 2W a 850/900MHz
- Servicio de mensajería (cortos): Envío de pequeñas cantidades de datos a través de la red (ascii o primas hexadecimal)
- Pila embebida TCP/UDP: Carga de datos a un servidor web
- Puerto: Serie (Libre selección)
- Altavoz y tomas de auriculares: Si
- Bajo consumo de energía 1.5mA (modo dormir)
- Temperatura de operación  $-40^{\circ}\text{C}\sim+85^{\circ}\text{C}$

### **3.2.5 Dispositivos utilizados en el hogar**

#### **3.2.5.1 Sistema de Alarma**

Los sensores que se utilizarán para el sistema de alarma son de tipo magnético y de presencia. A continuación se presenta las características de los sensores magnéticos:

- Distancia de accionamiento:  $18\text{mm} \pm 6\text{mm}$
- Material: Nd magnética
- Material de la carcasa: ABS
- Hilo conductor: 22 AWG, 300mm de largo
- Color: café
- Peso neto: 130g/4.6 OZ

Cuando las dos piezas están juntas o muy próximas, el sensor se comporta como un interruptor en posición de encendido, al separar las piezas se comporta como un interruptor en posición de apagado, lo cual nos sirve para saber en qué momento las puertas o ventanas se encuentran abiertas o cerradas, y enviar una notificación al servidor.

En lo referente a los sensores de presencia, se van a utilizar sensores PIR (Figura 3.7), que nos entregan una señal digital al momento de detectar movimiento, los mismos estarán colocados a la entrada de la casa, en el pasillo y escaleras, de esta manera se entregará una mayor autonomía al hogar encendiendo y apagando la iluminación en estos espacios de manera automática. Además, se tendrán dos sensores adicionales dispuestos en la sala y el comedor que estarán también conectados con el sistema de alarma para censar la presencia de algún intruso cuando la alarma se encuentra activa.



Figura 3.7: Sensor PIR

Fuente: (PIR, 2017).

Las características de este sensor son las siguientes:

- Sensor piroeléctrico (Pasivo) infrarrojo (También llamado PIR)
- Rango de detección: 3 m a 7 m, ajustable mediante trimmer
- Lente fresnel de 19 zonas, ángulo  $< 100^\circ$
- Salida activa alta a 3.3 V
- Tiempo en estado activo de la salida configurable mediante trimmer
- Consumo de corriente en reposo:  $< 50 \mu\text{A}$
- Voltaje de alimentación: 4.5 VDC a 20 VDC

### 3.2.5.2 Sistema de Riego Inteligente

Como se explicó en el capítulo 2, se utilizará un sensor de humedad, un panel solar y una electroválvula para desarrollar el sistema de riego inteligente. El sensor de humedad es el DHT11, cuyas características son (GeekBot, 2017):

- Alimentación:  $3Vdc \leq Vcc \leq 5Vdc$
- Rango de medición de temperatura: 0 a 50 °C
- Precisión de medición de temperatura:  $\pm 2.0$  °C .
- Resolución Temperatura: 0.1°C
- Rango de medición de humedad: 20% a 90% RH.
- Precisión de medición de humedad: 4% RH.
- Resolución Humedad: 1% RH
- Tiempo de sensado: 1 seg.

La electroválvula escogida es de 12Vdc de ½” (Figura 3.8), nos sirve para controlar el flujo del fluido. Normalmente, la válvula está abierta, y cuando se aplica 12Vdc a los dos terminales, la válvula se cierra bloqueando el paso del agua.



Figura 3.8: Electroválvula

Fuente: (ElectroniLab, 2017)

Por otra parte, el panel solar es también de 12Vdc, con una potencia de 1.5W, y una corriente de 100mA, en la Figura 3.9 se puede apreciar el mismo.



Figura 3.9: Panel Solar

Fuente: (Future, n.d.)

A continuación, el diagrama de flujo del funcionamiento y acceso del sistema de riego:

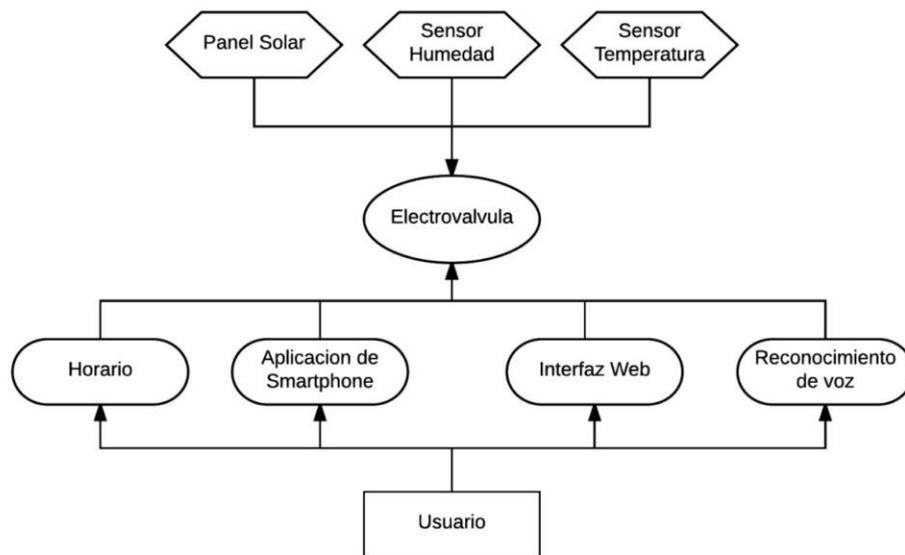


Figura 3.10: Diagrama de flujo del sistema de riego.

### 3.2.5.3 Multimedia

Para realizar las diferentes acciones tales como:

- Controlar encendido y apagado de los televisores, así como también sus demás funciones.
- Manipular las funciones del proyector
- Manejar todas las funciones de un equipo de sonido.

Se ha desarrollado un algoritmo capaz de sustraer y decodificar las señales emitidas por los distintos controles remotos que usen tecnología infrarroja, este algoritmo fue desarrollado en Arduino.

A partir de esta decodificación se podrá manejar los dispositivos antes mencionados a placer con la ayuda de un emisor IR. Estas acciones podrán encontrarse en los horarios de aprendizaje antes mencionados en el capítulo 2, así como en las acciones a escoger en la página web. El receptor IR utilizado es el AX-1838HS (Figura 3.11), que se caracteriza por ser económico, y sencillo de utilizar.



Figura 3.11: Receptor IR

Fuente: (Prometec)

El emisor IR es un led infrarrojo (Figura 3.12) que al igual que el receptor es de muy bajo costo y de fácil implementación. Se utilizan conjuntamente para emitir las señales que han sido decodificadas.



Figura 3.12: Emisor IR

Fuente: (Rocha, 2014)

En la figura 3.13 se presenta el diagrama de flujo en donde se hará uso del emisor infrarrojo:

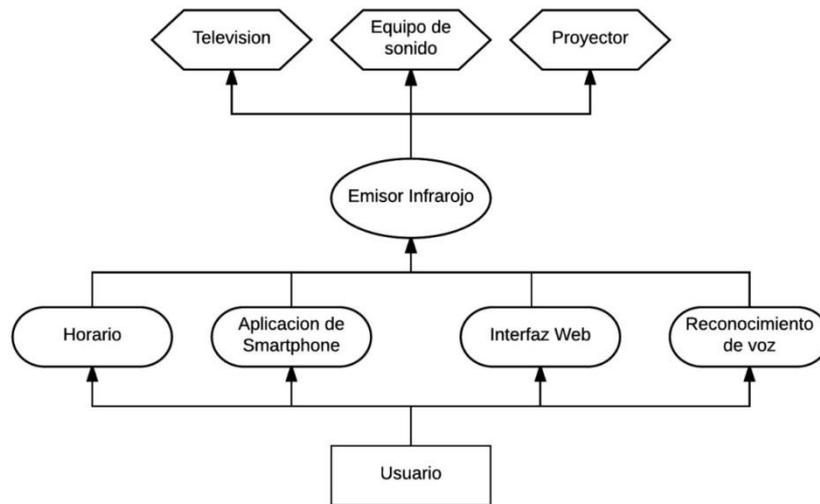


Figura 3.13: Diagrama de flujo activación de dispositivos multimedia

### 3.3 Software

#### 3.3.1 Base de Datos

Como se explicó en el capítulo 2, el sistema escogido es MySQL (Figura 3.14), por ser de código libre y por su fácil interacción con Python. En la base de datos se tendrán 5 tablas, cada una de ellas descritas anteriormente, en donde se podrá

cambiar los estados de la iluminación y sensores, así como la capacidad de editar horarios de acciones a ser realizadas por el AV.

En cada tabla se dispondrá de las direcciones IP de los dispositivos dispuestos en el hogar, así como el uso de campos similares en cada tabla, para relacionarlas dependiendo de la consulta que se realice, o de la información que requiera el AV para ejecutar alguna acción. Además, se dispondrá de una función creada en la base de datos que será capaz de notificar (*trigger*) al AV si algún cambio se ha realizado en el campo de alguna de las tablas.



Figura 3.14: Logo MySQL

Fuente: (Oracle, 21017)

### 3.3.2 Página Web

Para el desarrollo de la página web se utilizarán diferentes tecnologías, tales como:

- PHP
- jQuery
- Javascript
- Ajax

Así como el entorno de desarrollo integrado libre Netbeans, que también se caracteriza por ser de código libre. Este entorno tiene la facilidad de realizar el desarrollo de una página web en diferentes formatos, en este caso se utilizara la

extensión .php, debido a que se necesita la consulta constante de los diferentes campos y tablas de la base de datos antes mencionada. Igualmente, se utilizará Javascript para ejecutar distintos códigos en los navegadores clientes, de este modo logrando obtener datos en tiempo real.

También, se utilizará jQuery, que es un conjunto de librerías JavaScript que han sido diseñadas específicamente para simplificar el desplazamiento de un documento HTML, la animación, la gestión de eventos y las interacciones Ajax (The jQuery Foundation, 2017). Con este último se puede realizar peticiones y respuestas del servidor mediante la transmisión de datos en segundo plano, por lo que no se necesita recargar constantemente la página web sino solo una parte de ella y así mostrar las consultas realizadas al servidor y actuar sobre las mismas. Lo que se logra con estas distintas tecnologías es un mayor dinamismo por parte de la página web.



Figura 3.15: Distintas tecnologías. PHP, Javascript, Ajax, JQuery.

Fuente: (Alvarez, 2009)

Para tener una mayor fluidez tanto en el acceso a la página, como con la interacción que se realiza con la base de datos, se optó por contratar un servicio de hosting, el cual nos entrega el dominio <http://www.smarthousecuena.com>, además de alojar una base de datos en el mismo servicio de hosting y muchas opciones más que pueden ser desarrolladas a futuro como se puede ver en la siguiente Figura.

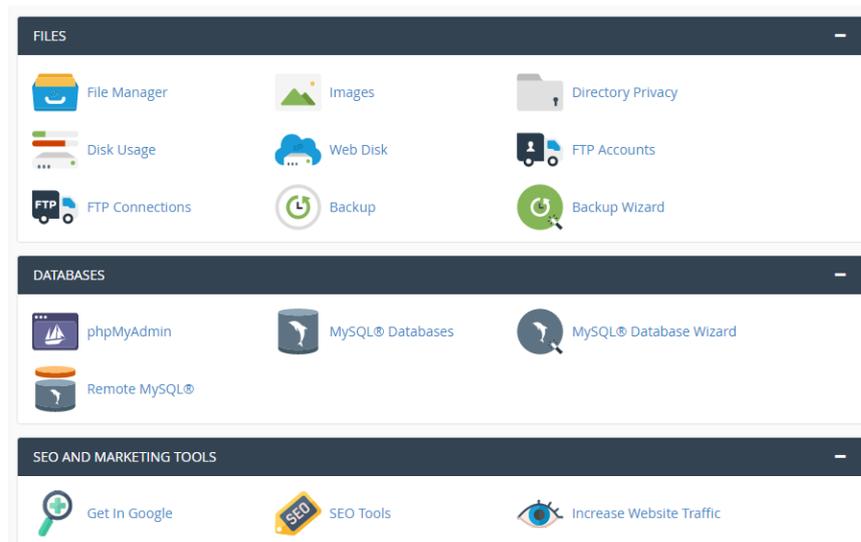


Figura 3.16: Diferentes opciones del servicio de hosting.

En la carpeta “*public\_html*”, dentro de la opción “*File Manager*”, es en donde se ubican los diferentes archivos e imágenes desarrolladas en Netbeans, para publicar la página web en el dominio contratado.

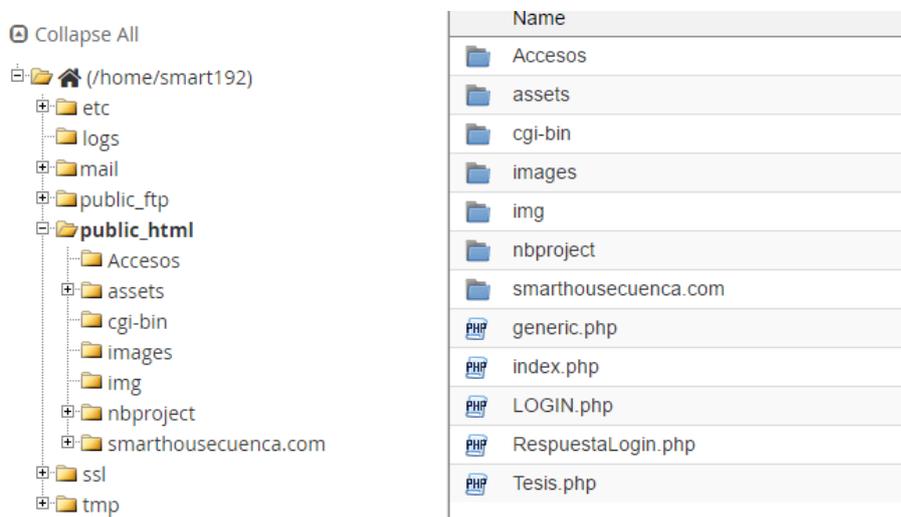


Figura 3.17: Archivos que conforman la página web

Como se puede observar en la Figura 3.16, se tiene el acceso a phpMyAdmin, en donde se encuentra la base de datos antes mencionada que ya implementada contiene las siguientes tablas:

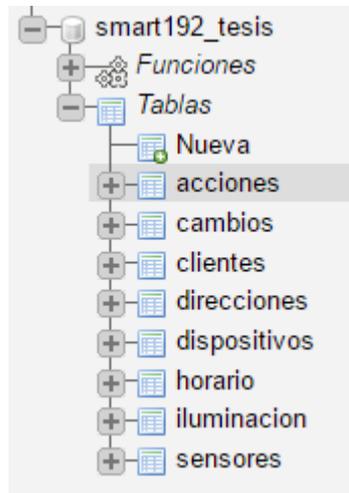


Figura 3.18: Tablas de las que consta la base de datos

La tabla “acciones” posee diferentes tareas tales como: Encender foco principal, Encender proyector, Enviar mail, entre otros, además de sus respectivos códigos que serán utilizados para relacionarlos con las otras tablas. El funcionamiento, así como el uso de las diferentes tablas serán discutidos posteriormente en la implementación de la página web.

### 3.3.3 Android Studio

Android Studio es el IDE oficial para el desarrollo de aplicaciones Android. Se encuentra basado en el lenguaje JAVA y es también de código libre. Se utilizará para el desarrollo de una de las interfaces del usuario con el hogar, en donde se tendrá, al igual que en la página web, el acceso a cada una de las acciones que se pueden efectuar dentro del hogar así como realizar diferentes horarios que el AV tendrá que cumplir.

El acceso a la base de datos se realiza por medio de un *REST Web Service*, el cual nos sirve para intercambiar datos entre aplicaciones de diferentes lenguajes. *REST* se refiere a la transferencia de datos que se centra en el uso de los estándares http (Rodríguez, 2017), es decir, se accede a la información por medio del url del servidor, entregando las variables de la información que se necesita buscar en la base de datos, para posteriormente entregarla a la aplicación, se escogió este tipo de protocolo para mejorar el rendimiento y rapidez de las consultas.

El usuario deberá ingresar con su *login* y *password*, para posteriormente realizar una petición al servidor, donde se encuentra alojada la página web, para que éste a su vez entregue la información en formato json, que es descifrable para la aplicación. De igual manera, para enviar datos desde la aplicación hacia el servidor, se realiza la codificación de los mismos hacia el servidor y estos son cambiados en la base de datos en tiempo real.

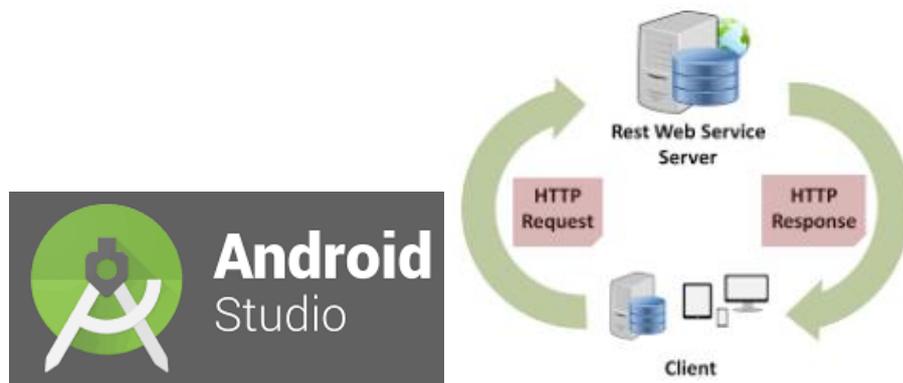


Figura 3.19: Android Studio (Android Studio, 2017), REST Web Service

Fuente: (Rodríguez, 2017)

### 3.3.4 Inteligencia Artificial

Se utiliza el *software* matemático MATLAB para realizar las distintas simulaciones relacionadas a la inteligencia artificial del AV. Anteriormente se dijo que el sistema

utilizará lógica difusa, esto se logra utilizando el toolkit de Matlab FUZZY LOGIC (Lógica Difusa), el cual provee una interfaz gráfica que facilita la creación de una estructura de lógica difusa (Mathworks, 2017). La interfaz permite escoger el tipo de onda a usar (triangular o gaussiano), los límites de cada una de ellas y además las reglas que gobiernan cada acción dependiendo de las variables lingüísticas asignadas. Estos resultados obtenidos en MATLAB se colocan en la programación de Arduino en el módulo ESP8266-12E, logrando así utilizar una herramienta matemática robusta e implementar la misma en un dispositivo de bajo costo.

### **3.3.5 Protocolo de Comunicación**

Para la comunicación a través de la red se ha elegido el protocolo UDP, resultando una mejor elección que TCP debido a que la red puede sufrir una sobre saturación. Además, el protocolo TCP necesita establecer una sesión antes de empezar a enviar mensajes, a diferencia del protocolo UDP que una vez conectado el dispositivo a la red puede empezar a enviar y recibir mensajes. Existen diferentes métodos que se puede encontrar para la conexión de los diferentes dispositivos a la red del hogar.

Uno de ellos se puede efectuar realizando la conexión directa desde la programación de Arduino, desde este punto el usuario no puede realizar ninguna interacción con el módulo ESP8266-12E. Es decir, el modulo se conecta directamente pero después de ello el usuario no podrá acceder al mismo a menos que sea desmontado y grabado nuevamente. Como segunda opción, se tiene el uso de la librería WiFi Manager, también proporcionada por Arduino, con la ayuda de esta librería el usuario podrá tener una interfaz con el módulo WiFi, en la Figura 3.20 se tiene el proceso que se realiza para la conexión del dispositivo a la red del hogar.

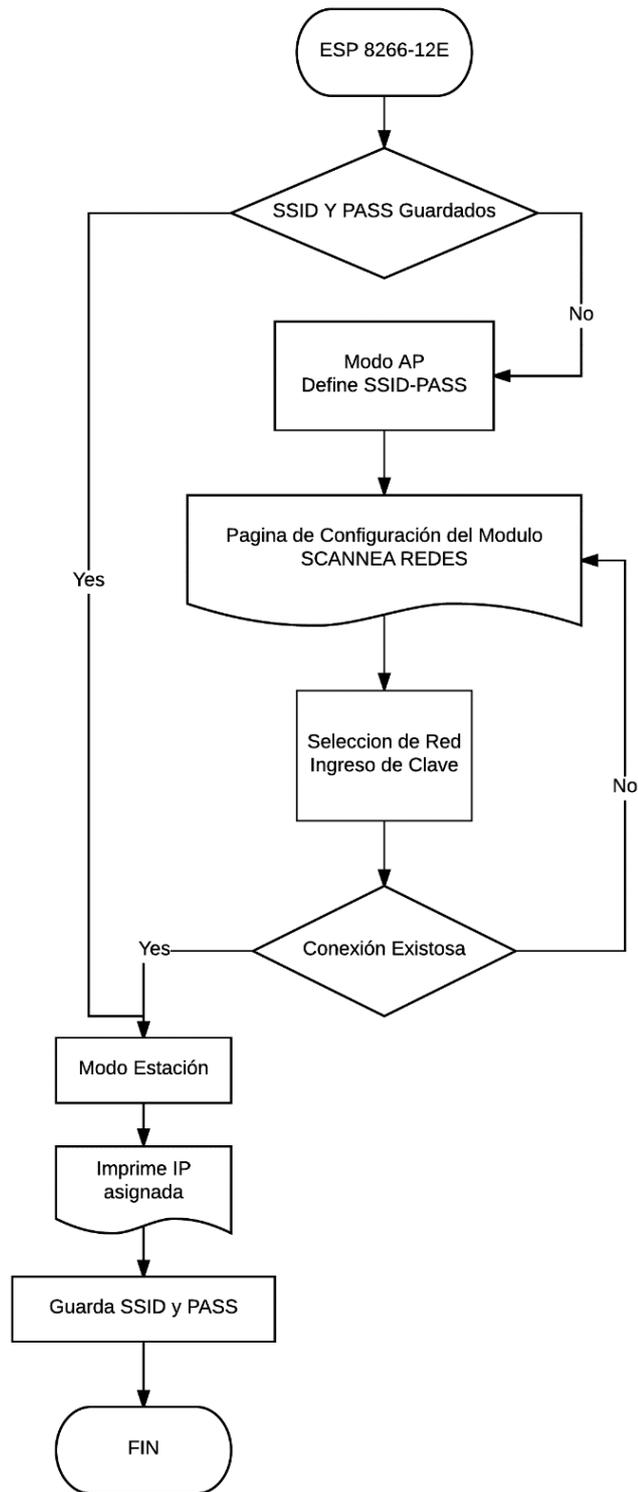


Figura 3.20: Diagrama de flujo del proceso de conexión a la red.

El dispositivo WiFi crea un punto de acceso, en este caso con el nombre “Tesis MenesesPesantez”, en donde el usuario debe conectarse ingresando la clave del punto de acceso. Una vez conectado, se abre la interfaz en la que el usuario tendrá la capacidad de escanear las redes disponibles y conectarse a ella ingresando la clave de la misma. Si el dispositivo tiene una conexión exitosa, el punto de acceso desaparecerá y el módulo WiFi se quedara conectado a la red del hogar. Cabe recalcar que las credenciales antes ingresadas para la conexión a la red quedan grabadas en el dispositivo, por lo que sí existe un apagón el modulo se conectara automáticamente sin necesidad de volver a crear el punto de acceso. Al igual que en el caso anterior, para cambiar la programación, el dispositivo tendrá que ser desmontado y grabado.

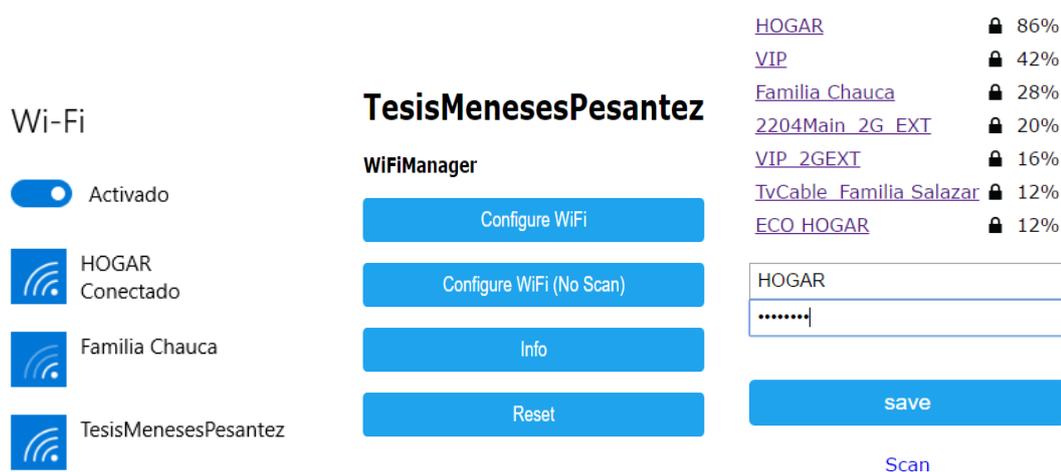


Figura 3.21: Pasos para conexión a la red del Hogar.

Finalmente, el ESP8266-12E acepta la programación OTA (Over The Air), la cual es una manera de programar este dispositivo de forma inalámbrica a través de WiFi. Para realizar este proceso se debe incluir la librería ArduinoOTA en la programación del microcontrolador, así como tener instalado Python 2.7 en el ordenador. Se puede escoger un nombre de *host* para el dispositivo, y después que el mismo se ha conectado a la red se podrá visualizar en el IDE de Arduino de la siguiente manera:

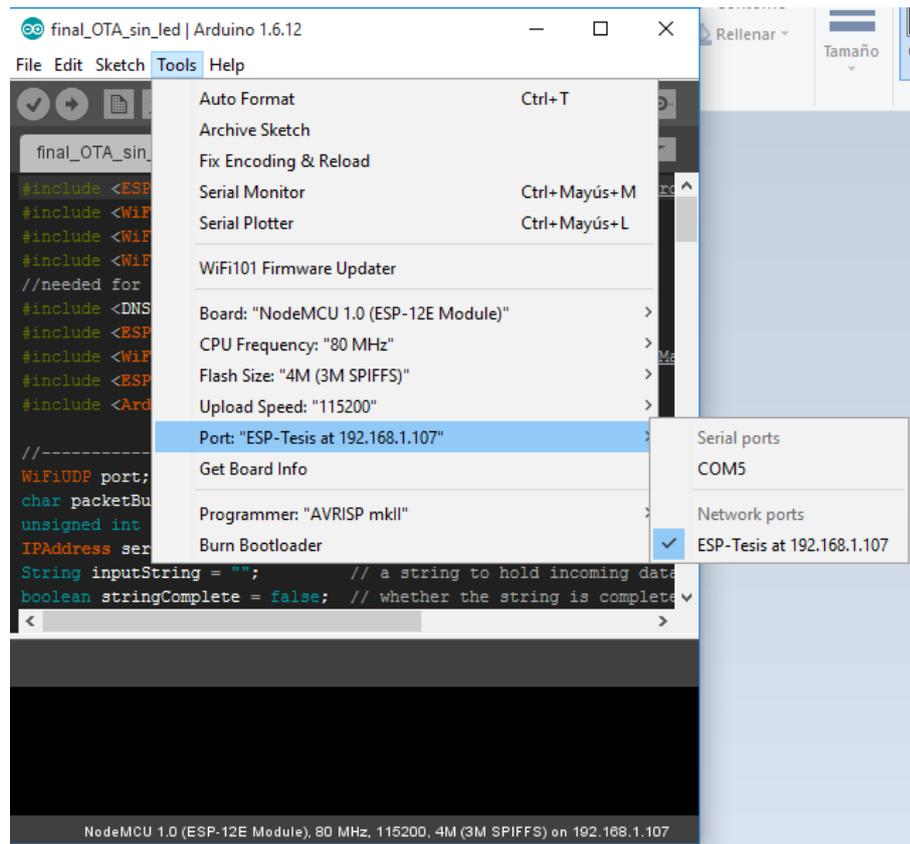


Figura 3.22: Arduino OTA

Como se puede apreciar, una vez conectado el módulo a la red, en el IDE de Arduino se podrá observar en la opción puerto el nombre del host así como la dirección IP entregada por el *router* del hogar, de esta manera no importara en donde se encuentre el dispositivo, no se tendrá que desmontar el mismo para volver a cambiar su programa sino que se grabara inalámbricamente desde el IDE. Esta opción se puede utilizar conjuntamente con las dos anteriores, permitiendo varias opciones de acceso hacia los diferentes dispositivos ubicados en el hogar.

Por otra parte, el servidor será el encargado de recibir los mensajes UDP provenientes de cada nodo, actualizando así los estados de iluminación, sensores y horarios en la base de datos. Ahora, la alarma consta de sensores magnéticos, algunos de ellos cableados a las ventanas y otros inalámbricos con la ayuda de un

módulo WiFi. El Raspberry utilizado para la alarma trabajará conjuntamente con estos sensores así como con el módulo GSM, que será el encargado de enviar un mensaje de texto con la información del sensor que fue activado y una llamada al propietario de la casa.

### **3.4 Implementación de la Red**

De acuerdo a lo especificado en capítulos anteriores, la red de sensores y actuadores que se planea incorporar en el hogar es inalámbrica, conocida también como WSN (Wireless sensor and actuator network). En una red de este tipo, los sensores recogen información sobre el mundo físico, mientras que los actuadores toman decisiones apropiadas sobre el medio ambiente, lo que permite la interacción a distancia. Los sensores distribuidos de una WSN permiten la medición automática de las variables ambientales y el control de algunos aspectos deseados del medio ambiente por medio de sensores y actuadores autónomos o directamente controlables. (Martínez, 2012) Actualmente, constituyen una alternativa tecnológica interesante para el desarrollo de aplicaciones que requiere monitorizar constantemente el estado de cualquier variable relacionada con escenarios de diversos ámbitos. Si las aplicaciones detectan cambios en los valores de dichas variables, pueden activar la ejecución de acciones preventivas que ayuden a restaurar las condiciones normales del entorno monitorizado. (Ngai, 2012)

Para lograr lo antes expuesto se hará uso del módulo WiFi ESP8266 12-E, el cual cumple con los requisitos a ser utilizados por el AV como se detalló en el capítulo 2. La topología de la WSN a ser utilizada se puede observar en la Figura 3.23.

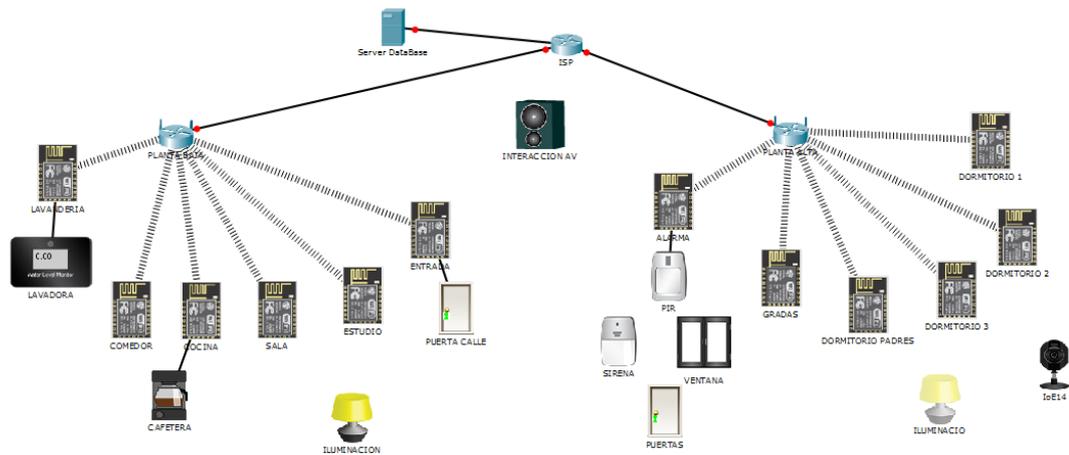


Figura 3.23: Topología de la red WSAN.

En donde se tendrán dos *routers*, uno para cada piso, y un *router* principal el cual será el SP interconectado a internet por el ISP (Internet Service Provider). A cada *router* ubicado en la planta baja y alta, se conectarán varios módulos WiFi colocados en los cajetines de interruptores y tomacorrientes distribuidos en todo el hogar. Para garantizar las direcciones IP para cada módulo WiFi, se procederá a configurar cada *router* con las direcciones MAC de cada dispositivo para asignarles una dirección IP estática y así, mediante la tabla de enrutamiento acceder desde los módulos al servidor. El *router* principal se encontrará conectado al SP, el cual estará encargado de adquirir y guardar toda la información de cada uno de los sensores que se colocaran en el hogar, de esta manera el AV podrá realizar las diferentes acciones programadas. Además, el SP como se explicó en el capítulo anterior poseerá la base de datos de cada perfil de usuario que utilizara el hogar.

Igualmente, se considera la utilización de subredes para dirigir el tráfico hacia, y desde el servidor de una manera más segura y descifrable para los diferentes dispositivos. Se debe configurar también las puertas de enlace de cada *router* (planta alta y baja), para que los dispositivos conectados a estas subredes puedan intercambiar información desde una red a otra y así evitar la pérdida de paquetes. (Bhatt, 2016)

### 3.4.1 Acciones derivadas de los sensores en el hogar

Con la ayuda de sensores inalámbricos en el hogar se puede llegar a formar un entorno monitorizado que mejore la calidad de vida de sus habitantes dependiendo de las necesidades expuestas en dicho entorno. (Gauger, 2011). Por ejemplo, como se había mencionado antes, se puede tener una mayor seguridad en el hogar con la ayuda de sensores PIR (sensor infrarrojo pasivo, Figura 3.24), que ayudan a censar el movimiento en el hogar y así activar diversas acciones pre-programadas en el AV, como es: la grabación del video de un intruso que ingrese al hogar cuando la alarma este activada, además de enviar un correo electrónico al dueño de la vivienda, dar aviso a las autoridades de lo acontecido y también simular la presencia de una persona en el hogar activando luces, encendiendo la música e incluso encender el televisor.

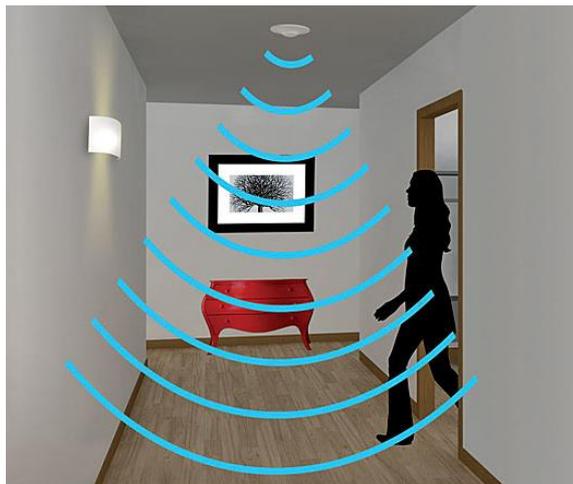


Figura 3.24: Activación sensor PIR

Fuente: (Casadomo, 2017)

Con la ayuda de cámaras, se podrá realizar el reconocimiento facial de cada usuario del hogar permitiendo al AV realizar la toma de decisiones dependiendo del monitoreo que se realice. Por ejemplo, mediante el uso de tratamiento de imágenes se puede detectar el movimiento de una persona en determinada zona del hogar, por lo

que las cámaras serán situadas de tal manera que su campo de visión acapare la mayoría de dicha zona. Posteriormente, el AV tendrá la capacidad de apagar o encender las luces, e incluso preguntar a esa persona si necesita asistencia en alguna forma, y podérselo comunicar a un contacto pregrabado en el AV. Además de esto, se podrá programar el sistema para ser monitorizado remotamente desde una página web, con el afán de observar lo que está ocurriendo en el hogar, y así tener una mayor seguridad al momento de dejar a niños y adolescentes sin supervisión adulta en el hogar.

Por otra parte, una de las características más agradables que se puede construir en un sistema de automatización del hogar es el control por voz. Por lo que con micrófonos se podrá hablar directamente con el AV el cual realizara las peticiones de los usuarios desde encender o apagar luces hasta encender dispositivos inteligentes que se encontraran a disposición del AV cuando el usuario así lo requiera. Por ejemplo, si el usuario pide realizar una llamada a uno de sus contactos registrados en la base de datos, este lo realizará sin ningún inconveniente debido al módulo GSM que se encontrara en el hogar. De igual manera se podrá notificar al usuario de llamadas entrantes o mensajes de texto enviados directamente al hogar. También, si el usuario decide encender su lavadora, cafetera, televisor, radio o incluso tomar una fotografía y difundirla en las redes sociales, este lo realizara debido al extenso alcance que se tiene con la ayuda de dispositivos inteligentes colocados en los diversos dispositivos electrónicos.

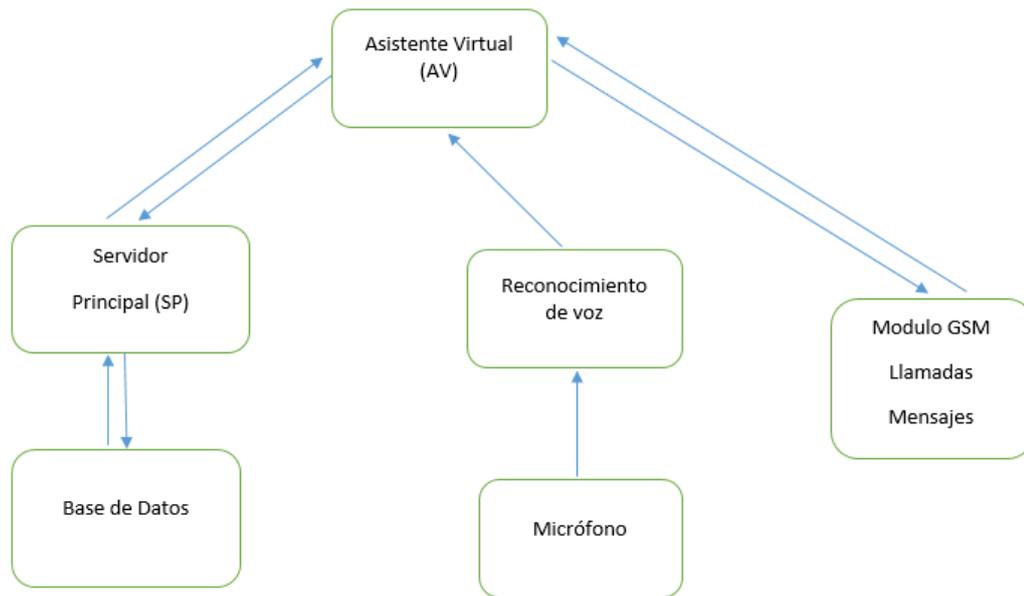


Figura 3.25: Diagrama del acceso al sistema implementado.

### 3.4.2 Actuadores dispuestos en el hogar

Como se expuso anteriormente, la mayoría de las acciones que se realicen en la vivienda se encuentran directamente ligadas a los sensores dispuestos en el mismo. De acuerdo a lo expuesto en el punto anterior, los dispositivos inteligentes tendrán la capacidad de interactuar con el AV, puesto que tanto los sensores como sus acciones derivadas se encuentran entrelazadas con ayuda de la comunicación WiFi antes expuesta. El módulo ESP8266 nos entregara la comunicación necesaria para realizar acciones en cada rincón del hogar de una manera inalámbrica, y así tener un solo sistema inteligente comunicado de manera segura y eficiente para realizar las necesidades de cada usuario. A continuación se tiene el diagrama de flujo del interruptor inteligente:

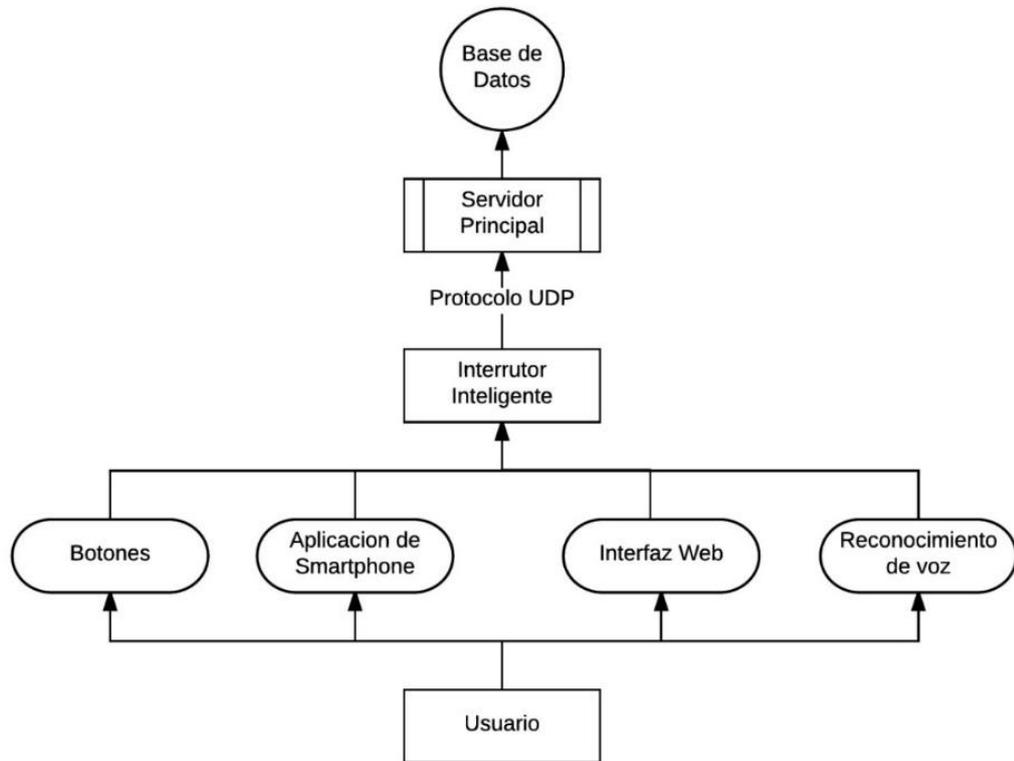


Figura 3.26: Diagrama de flujo interruptor inteligente

Para acoplar los módulos WiFi a cualquier electrodoméstico se realizaron varias pruebas hasta llegar a conformar el circuito apto que pueda realizar tanto la comunicación como las acciones normales de cada dispositivo. A continuación, en la Figura 3.27 se tiene el circuito final realizado que cumple con dichas especificaciones.

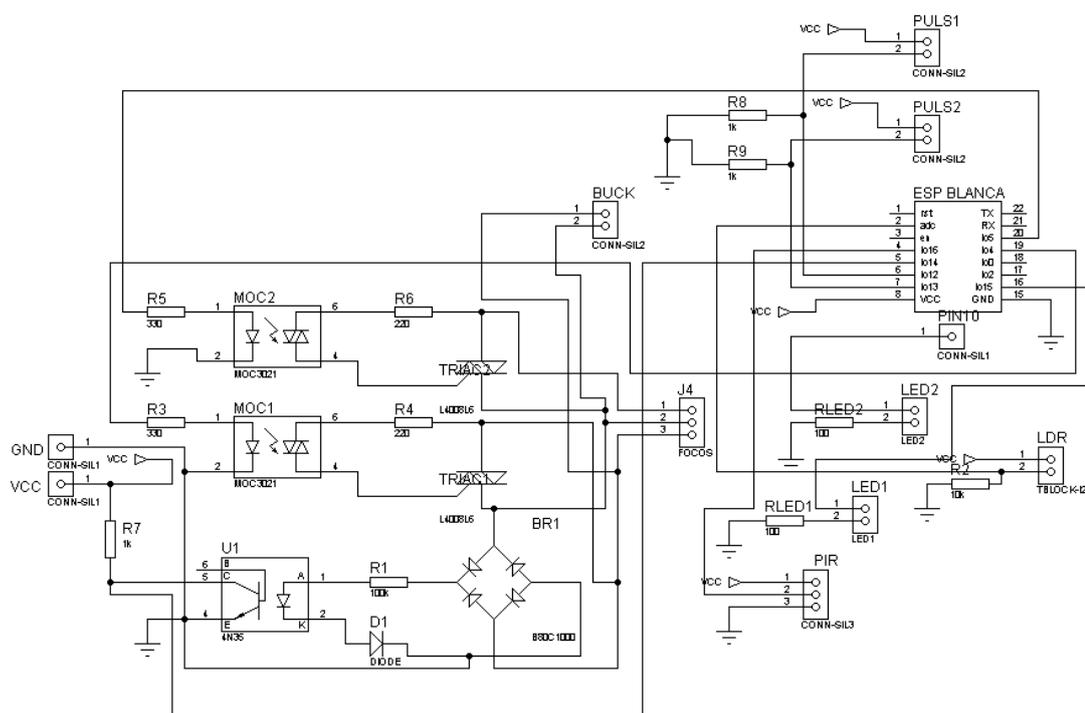


Figura 3.27: Circuito Inteligente

Como se puede apreciar, se hizo uso de un convertor AC-DC el cual regula la tensión de 120V AC a un voltaje de 5V DC y nos entrega hasta 700mA de corriente, lo cual es suficiente para trabajar con el módulo WiFi. Este circuito será colocado dentro de los electrodomésticos con el afán de controlarlos de una manera inalámbrica.

Además, estos circuitos serán también colocados en serie a los focos del hogar en los interruptores para realizar el encendido y apagado de las mismas. Para comenzar, se realiza la rectificación de la onda senoidal con la ayuda de un puente de graetz para así tener la referencia del 100% de la onda, a la salida del puente se tiene una resistencia de 100kΩ para disminuir el voltaje de entrada al aislador óptico (4n33), el mismo que posee un diodo rectificador colocado después del diodo perteneciente al optoaislador y de esta manera proteger al mismo de cualquier corriente parásita perteneciente a la onda. Se utiliza esta numeración específica de integrado debido a que posee un transistor Darlington y por lo tanto la respuesta a la activación del

diodo que lo activa es más efectiva. En el colector de este transistor se coloca una resistencia de  $1k\Omega$  alimentado por los 5 voltios del convertor AC-DC, y el emisor directo a tierra. De la unión entre resistencia y colector se toma la referencia para el pin IO14 del módulo ESP8266, que será programada como una interrupción para así realizar el disparo de los moc3021 y así encender o apagar las luces del hogar.

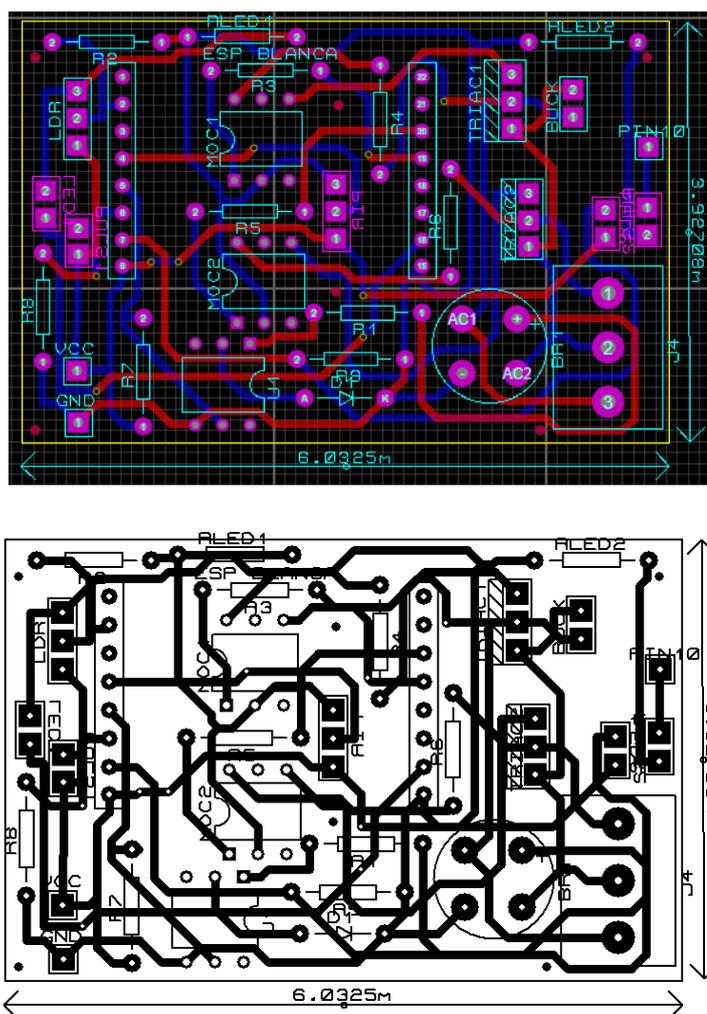


Figura 3.28: PCB del circuito inteligente

En la Figura 3.27 se puede apreciar el PCB diseñado para el circuito expuesto. Aquí se tiene dos opto acopladores (MOC3021) para aislar la carga del sistema de control, los cuales serán disparados por los diferentes pines del módulo ESP8266, en este caso IO4 e IO5. También, se tiene dos triac, los cuales manejan ambos lados de la

onda sinusoidal de 120v y al recibir el disparo en el *gate* estos dejan pasar la corriente y así encender o apagar las diferentes luces de las diferentes habitaciones. También, como se puede apreciar en la Figura 3.27, se tiene además dos pulsantes que ayudaran al usuario a encender o apagar las luces de forma manual o automática (WiFi), dos leds que serán encendidos dependiendo del estado del botón y de la hora del día para así evitar el consumo de la energía en vano, se poseen también dos sensores, uno de presencia PIR, que ya fue explicado anteriormente y un LDR para medir el nivel de luz en ciertas zonas del hogar. Cabe recalcar que el sensor PIR utiliza una entrada digital y el LDR una entrada analógica del módulo ESP8266. Se planteó este diseño con el afán de utilizar cada elemento de acuerdo a como sea necesario dependiendo de su ubicación en el hogar.

Para comprobar el correcto funcionamiento de la medición de la interrupción antes mencionada se realizaron diversas pruebas hasta llegar a concluir que opto acoplador es el más apto para ser usado en dicho circuito. A continuación se tienen dos graficas obtenidas del osciloscopio:

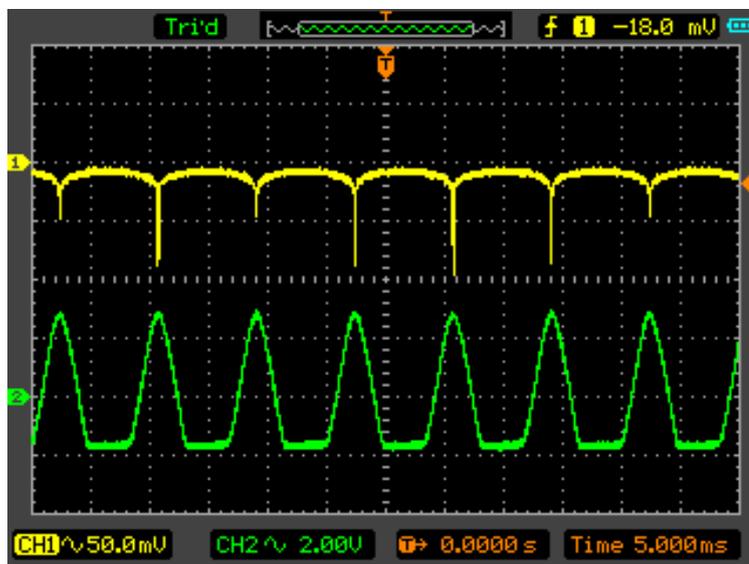


Figura 3.29: Activación del integrado PC817 con respecto a la rectificación a la onda senoidal.

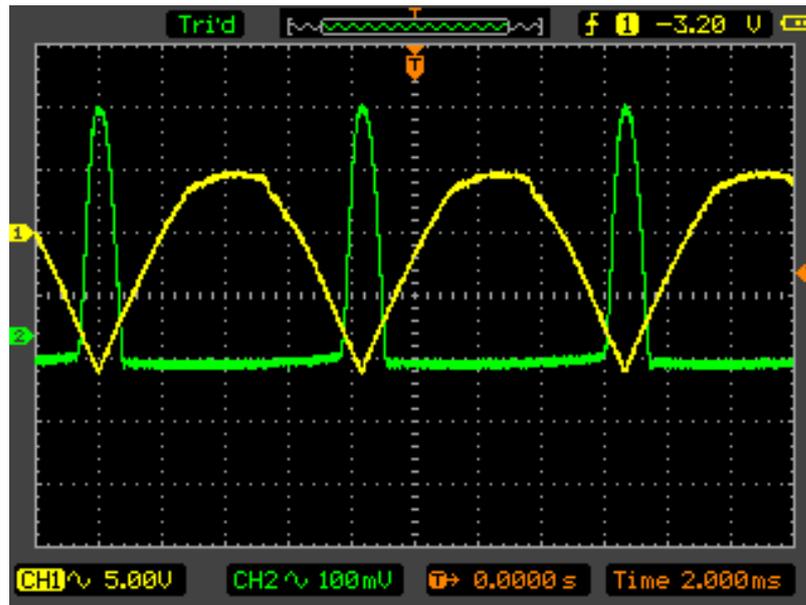


Figura 3.30: Activación del integrado 4n33 con respecto a la rectificación a la onda senoidal.

Como se puede observar en las Figuras 3.29 y 3.30, las gráficas de color verde corresponden a la activación y desactivación del integrado PC817 y 4n33 respectivamente. Como se explicó anteriormente el 4n33, por poseer un transistor Darlington posee una mayor sensibilidad y con ello una mejor referencia al momento de medir la finalización y comienzo de cada onda rectificadas. Lo que se logra con este integrado, es medir en el ESP8266 el momento de cada cruce por cero de la onda senoidal, para así activar la iluminación de forma que se corten ambos lados de la onda entregada por la alimentación del hogar. A continuación, en la Figura 3.31, se aprecia la onda senoidal cortada mediante el disparo del ESP8266 a uno de los MOC3021.

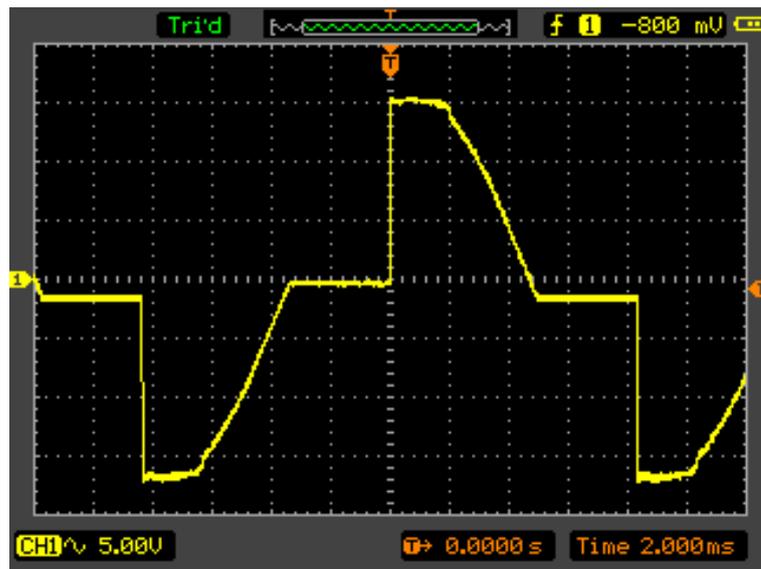


Figura 3.31: Corte de ambos lados de la onda senoidal.

Ahora, se realiza el corte hasta un cierto nivel aceptable tanto para el foco como para el conversor AC-DC. Lo que se logra con este procedimiento es mantener encendido tanto la iluminación como el conversor al mismo tiempo para así alimentar el módulo WiFi y los demás componentes del circuito antes indicados. Este punto se puede entender de mejor manera observando la Figura 3.32.

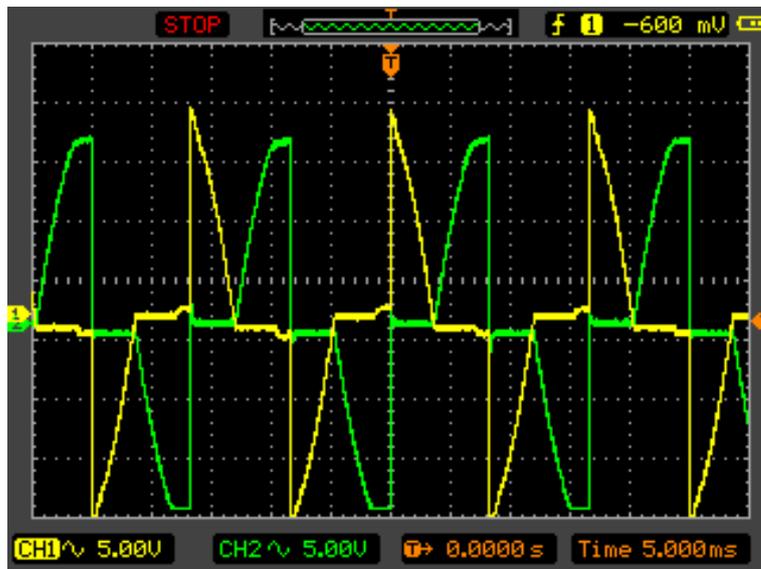


Figura 3.32: Gráfica de la onda en la iluminación así como en el conversor.

### 3.4.3 Alarma Inteligente

El hogar contará con un sistema de alarma comunicado alámbrica e inalámbricamente por medio de la red WSAN, los sensores PIR así como los contactos magnéticos de las ventanas se encontraran intercomunicados con el AV para activar o desactivar la alarma de una forma inalámbrica así como mediante un HMI (Human Machine Interface). Esta interfaz también estará conectada a la red del hogar de manera inalámbrica por medio de un Raspberry Pi 2, aunque funcionará de manera independiente puesto que al tratarse de la seguridad del hogar se requerirá que también posea una batería para su alimentación. Lo que se pretende lograr con un sistema de alarma WiFi es el acceso al mismo de una manera remota y tener la capacidad del monitoreo interno (Figura 3.33), así como la activación o desactivación del mismo desde un dispositivo móvil o una página web. El sistema de la alarma también está interconectada al módulo GSM y a las cámaras ubicadas en el hogar como se explicó en el capítulo 1. A continuación se presenta una gráfica de los dispositivos comunicados que trabajan conjuntamente con la alarma, y sus accesos remotos.

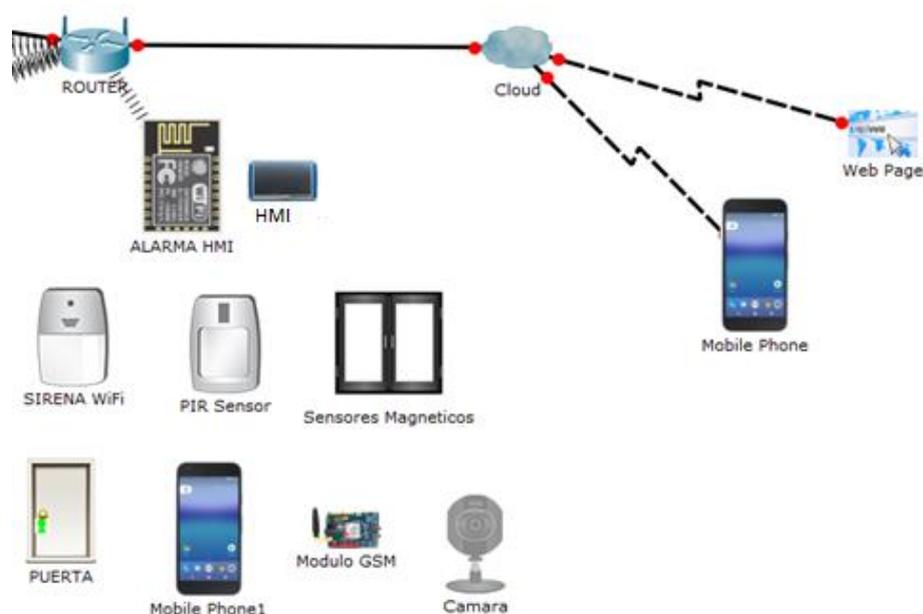


Figura 3.33: Diagrama de la alarma inteligente

A continuación se presenta un diagrama de flujo (Figura. 3.34) del funcionamiento de la alarma inteligente:

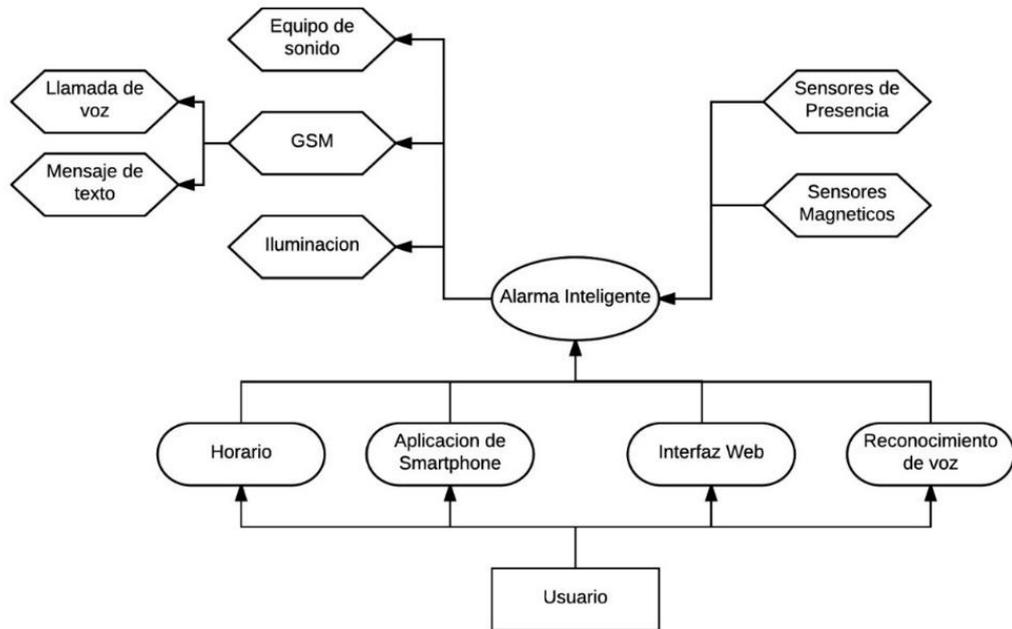


Figura 3.34: Diagrama de flujo alarma inteligente.

### 3.4.4 Módulo GSM

El módulo GSM a utilizarse es el Arduino GSM Module SIM900. Como se explicó anteriormente con la ayuda de este módulo, el AV estará en la capacidad de comunicarse con los usuarios del hogar. Además, nos será de gran ayuda en cuanto a la seguridad del hogar, puesto que el AV podrá enviar un mensaje de texto y realizar una llamada al usuario cuando algún sensor de la alarma sea activado. También, se podrá realizar llamadas desde el hogar hacia contactos previamente registrados, los cuales podrán ser contactados con ayuda del AV. El poseer un dispositivo GSM en el hogar nos permite tener una mayor autonomía puesto que si en algún momento el servicio de internet fallara el AV podría comunicarse con el usuario por medio de un mensaje o una llamada.

### 3.4.5 Acceso Remoto

Para acceder remotamente al hogar se posee de dos sistemas diferentes, el primero mediante una página web y el segundo por medio de una aplicación móvil desarrollada en Android. A continuación se tiene la explicación de ambos sistemas:

#### 3.4.5.1 Página Web

El acceso remoto consta de una página web desarrollada en Netbeans, así como el acceso a la base de datos, desarrollada en phpMyadmin, antes mencionada en el capítulo 2. El usuario puede acceder ingresando su usuario y contraseña, los cuales se encuentran guardados en la base de datos. La página consta de las distintas acciones que se pueden realizar en el hogar: iluminación, horarios, sensores y alarma. Para facilidad del usuario se muestran estas acciones en una sola página a la cual se actualizan los diferentes procesos por medio de JQuery Ajax post, de esta manera se logra una mejor interacción.

Cada uno de los procesos entregados al momento de escoger cada botón posee un acceso a la base de datos dependiendo de la acción escogida. Por ejemplo, en la Figura 3.35 se puede observar el contenido mostrado cuando se escoge la opción Iluminación.



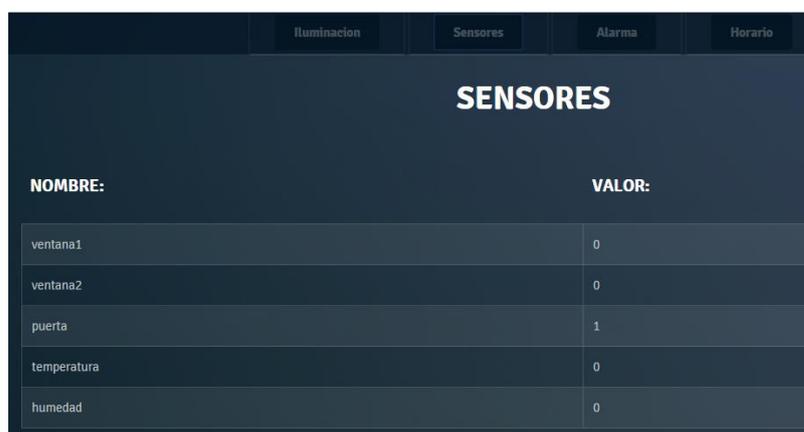
The screenshot shows a web interface with a dark blue background. At the top, the title 'ILUMINACION' is displayed in white. Below the title, a subtitle reads: 'Aquí se muestra cada uno de los dispositivos de iluminación del hogar. Permittiendole acceder a ellos en tiempo real.' Below this, there is a table with three columns: 'Nombre', 'Estado', and 'Estado Actual'. The table contains eight rows of data, each representing a lighting device. The 'Estado' column contains buttons labeled 'Apagar' or 'Encender', and the 'Estado Actual' column contains the current state of the device.

Nombre	Estado	Estado Actual
Gabriel_1	Apagar	Apagado
Gabriel_2	Encender	Encendido
Bruno_1	Apagar	Apagado
Bruno_2	Apagar	Apagado
Cocina_1	Apagar	Apagado
Cocina_2	Apagar	Apagado
Sala_1	Apagar	Apagado
Sala_2	Apagar	Apagado

Figura 3.35: Opción iluminación en página web

En la Figura 3.35, se muestra una tabla que contiene los distintos dispositivos dispuestos en los interruptores del hogar, haciendo referencia a su ubicación. Como en cada interruptor se tiene dos controles de encendido y apagado en la página web también se muestra esta elección. Mediante la opción estado se puede encender o apagar el interruptor, y en el campo estado actual se muestra el cambio realizado en la base de datos, después de haber cambiado el estado del dispositivo.

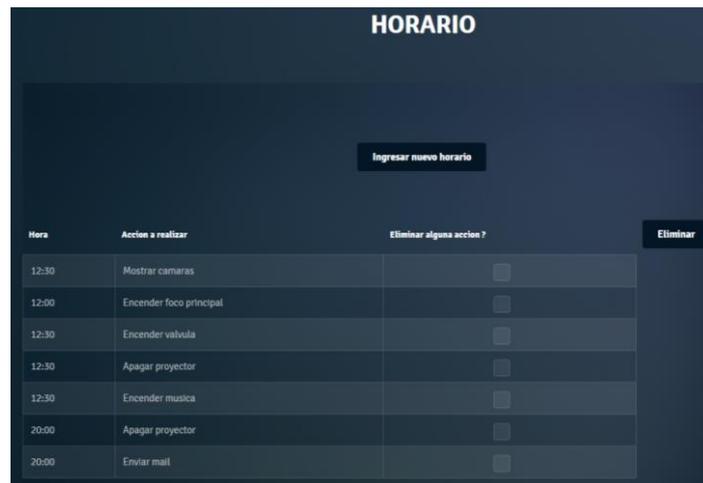
Cuando se escoge la opción sensores, se muestra el nombre y el valor de los sensores dispuestos en el hogar. Los valores se refrescan cada cierto tiempo para así saber cuál es el estado del hogar en tiempo real. Esta opción es informativa para el usuario. (Figura 3.36)



NOMBRE:	VALOR:
ventana1	0
ventana2	0
puerta	1
temperatura	0
humedad	0

Figura 3.36: Opción sensores en página web

Asimismo, en la opción horario, se muestra en una tabla las acciones, así como la hora escogida por el usuario para realizar dicha acción. Además, se le entrega al usuario la facilidad de eliminar las acciones programadas así como adicionarlas (Figura 3.37). Para eliminar las acciones basta con seleccionar el checkbox que se presenta al lado derecho de las acciones, se pueden escoger varias de ellas, las cuales serán también eliminadas de la base de datos.

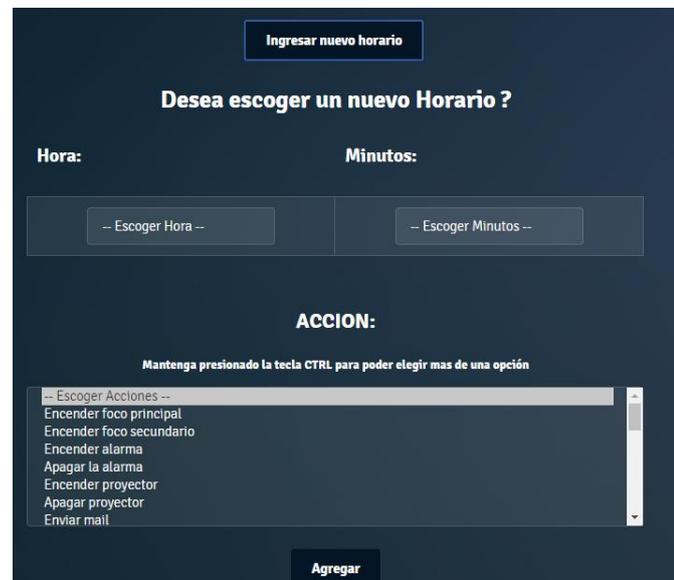


The screenshot shows a dark-themed web interface titled "HORARIO". At the top, there is a button labeled "Ingresar nuevo horario". Below this is a table with the following structure:

Hora	Acción a realizar	Eliminar alguna acción ?	Eliminar
12:30	Mostrar camaras	<input type="checkbox"/>	
12:00	Encender foco principal	<input type="checkbox"/>	
12:30	Encender valvula	<input type="checkbox"/>	
12:30	Apagar proyector	<input type="checkbox"/>	
12:30	Encender musica	<input type="checkbox"/>	
20:00	Apagar proyector	<input type="checkbox"/>	
20:00	Enviar mail	<input type="checkbox"/>	
20:00	Enviar mail	<input type="checkbox"/>	

Figura 3.37: Opción Horario en la página web

Para agregar nuevas acciones, el usuario simplemente deberá escoger el botón “Ingresar nuevo horario”. A continuación se despliega el menú en donde se puede escoger la hora, el minuto y varias acciones a ser realizadas en ese tiempo (Figura 3.38). Una vez escogidos el horario y la o las acciones, éstas se adicionan activando el botón “Agregar”, y posteriormente se muestran en el listado de acciones de la parte inferior (Figura 3.38).



The screenshot shows a form titled "Desea escoger un nuevo Horario?". It contains the following elements:

- A button at the top labeled "Ingresar nuevo horario".
- The title "Desea escoger un nuevo Horario?".
- Two labels: "Hora:" and "Minutos:".
- Two dropdown menus: "-- Escoger Hora --" and "-- Escoger Minutos --".
- A label "ACCION:".
- A note: "Mantenga presionado la tecla CTRL para poder elegir mas de una opción".
- A dropdown menu for actions, currently showing:
  - Escoger Acciones --
  - Encender foco principal
  - Encender foco secundario
  - Encender alarma
  - Apagar la alarma
  - Encender proyector
  - Apagar proyector
  - Enviar mail
- A button at the bottom labeled "Agregar".

Figura 3.38: Opciones entregadas al activar el botón “Ingresar nuevo Horario”

Cabe recalcar que para lograr este fácil acceso a la base de datos se utiliza PHP así como JavaScript, los cuales son de gran utilidad para realizar una página mucho más dinámica y de fácil acceso para el usuario.

### 3.4.5.2 Aplicación Móvil en Android

Como se indicó anteriormente para la programación de la aplicación móvil se ha usado Android Studio. Para empezar es necesario que el usuario se autentifique, con el fin de entregar seguridad a los distintos recursos que proporciona la aplicación, para ello se ha construido una interfaz de fácil manejo, en la cual se debe ingresar los campos de “Usuario” y “Contraseña”, como se muestra en la Figura 3.39



Figura 3.39: Captura de pantalla de la vista principal de la aplicación.

Para revisar si los campos ingresados son correctos, se realiza una consulta a la base de datos, ya que Android Studio no nos presenta una interfaz que nos permita comunicarnos directamente con la base de datos, se utiliza el *Web Service*, como ya se indicó anteriormente, el cual sirve como puente entre la aplicación y la base de datos. En la Figura 3.40 se puede apreciar una estructura de cómo se realiza la conexión con la base de datos.



Figura 3.40: Estructura comunicación Android-Base de Datos.

Luego de haber comprobado que los datos son correctos la aplicación nos permite acceder a los siguientes recursos por medio del servidor:

- Iluminación
- Sensores
- Horario
- Reconocimiento de voz

### **Iluminación**

Para esta categoría como se puede ver en la Figura 3.41, la aplicación genera una matriz de botones que indica en tiempo real el estado de los focos en el hogar. Además, al presionar en cualquiera de ellos la aplicación realiza un cambio en la base de datos, utilizando el puente mencionado anteriormente, ese cambio es detectado por el servidor el cual enciende dicho foco.



Figura 3.41: Captura de pantalla del menú Iluminación.

## Sensores

De la misma manera que el menú de iluminación, este menú nos permite acceder en tiempo real al estado de los distintos sensores del hogar, permitiendo saber si alguna puerta o ventana se encuentra abierta, o en qué habitaciones el sistema detecta personas. En la Figura 3.42 se puede apreciar dicho menú.

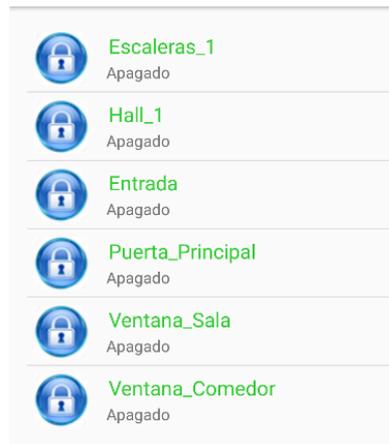


Figura 3.42: Captura de pantalla del menú Sensores

## Horario

En esta categoría se tiene dos pantallas, la primera que se puede visualizar en la Figura 3.43, nos permite revisar, agregar y eliminar las distintas acciones que se encuentran agregadas al sistema, esto permite al usuario manejar el horario con mayor facilidad.



Figura 3.43: Captura de pantalla del menú Horario

La segunda pantalla permite al usuario agregar las distintas actividades que se encuentran registradas en la base de datos, en los diferentes horarios que se elijan. Ya que la aplicación realiza una consulta a la base de datos para mostrar las distintas actividades, no es necesario cambiar la aplicación para agregar más, sino basta con agregarla a la base de datos directamente.



Figura 3.44: Captura de pantalla del menú Actividades

**Reconocimiento de Voz:** Esta sección es muy importante, ya que es una de las interfaces más amigables con el usuario, permitiéndolo acceder a todos los recursos que ofrece el sistema domótico con solo decirlo. En la Figura 3.45 se puede apreciar la aplicación móvil reconociendo el audio, es importante recalcar que otra de las maneras de acceder al reconocimiento de voz es presionando el botón de volumen con los auriculares, entregando una mayor autonomía en el manejo de la aplicación a personas con distintas discapacidades o que tenga bajo conocimiento en el manejo de aplicaciones celulares.

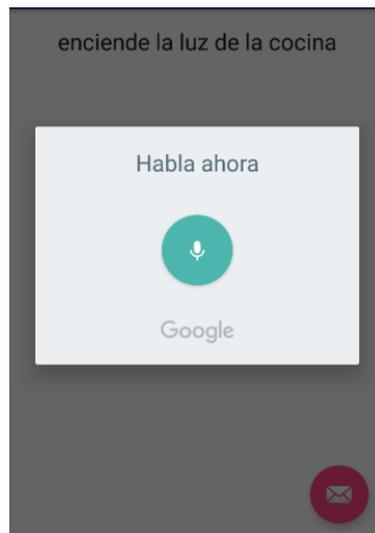


Figura 3.45: Captura de pantalla del menú Reconocimiento de Voz.

### 3.5 Conclusiones

- Brindar varias interfaces de interacción con el sistema domótico, sirve de gran ayuda para los usuarios del hogar, dado que pueden acceder a todos los recursos que brinda el sistema de manera sencilla, para ello se han implementado las siguientes opciones:
  - La aplicación móvil
  - El sitio Web
  - Reconocimiento de Voz
  - Interfaces directas en cada dispositivo
  - Servidor Principal
  - Pantallas Táctiles distribuidas en el hogar.
- El *hardware* utilizado es relativamente económico, pero usando programación se ha entregado mayor robustez a los distintos sistemas implementados, otorgando

mayor confiabilidad a los datos entregados por los sensores, y mayor utilidad a los distintos dispositivos que se encuentran instalados en el hogar.

- Desarrollar *hardware* que proporcione interfaces de manejo fácil para los distintos usuarios del hogar, es primordial, por lo que se ha optado en realizar menús sencillos, de fácil manejo y sobre todo adaptables a los distintos usuarios del sistema.

## 4. CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA IMPLEMENTADO.

### 4.1 Introducción

Luego de haber construido y programado todos los dispositivos y sistemas, es momento de realizar la implementación de los mismos en el hogar para realizar las pruebas y corrección de errores.

#### 4.1.1 Iluminación inalámbrica

Para realizar la instalación de los interruptores primero es necesario aislar todos los componentes que pueden provocar un corto-circuito como se puede ver en la Figura 4.1.



Figura 4.1: Aislamiento de los distintos componentes del interruptor.

Luego de aislar los componentes se conecta el cable perteneciente a la fase conjuntamente con los cables que provienen de los focos, los mismos que serán manipulado por el usuario de manera directa o inalámbrica. Se monta el circuito en el cajetín y en la carcasa principal como se puede ver en la Figura 4.2.



Figura 4.2: Implementación del interruptor inteligente.

Luego de haber realizado los procesos anteriormente expuestos, se sujetan con tornillos al cajetín y el interruptor inteligente queda completamente instalado como se puede ver en la Figura 4.3.



Figura 4.3: Interruptor inteligente instalado.

### 4.1.2 Portero Inteligente

Para convertir un portero eléctrico normal en un portero inteligente es, necesario la instalación de un módulo ESP8266 en el mismo, el cual nos permite controlar la apertura del portón eléctrico de manera remota e inalámbrica, así como también sensar si el portón se encuentra abierto o cerrado. Para ello se procede a realizar la instalación en el portero eléctrico como se puede ver en la Figura 4.4.



Figura 4.4: Instalación del módulo ESP8266 al portero eléctrico.

### 4.1.3 Sistema de alarma inteligente

Para realizar la implementación del sistema de alarma inteligente primero es necesario instalar todos los sensores magnéticos y de presencia, para ello se evitará el cableado en el hogar evitando un cambio en la estética de la casa, en las Figuras 4.5 y 4.6, se puede ver la instalación de los sensores.



Figura 4.5: Instalación del sensor magnético en la puerta principal.



Figura 4.6: Instalación del sensor de presencia en la entrada principal.

Algunos dispositivos funcionan como dispositivos multipropósito los cuales nos ayudan a optimizar el espacio que ocupa cada uno de ellos, así como la necesidad de instalar una fuente de alimentación por cada dispositivo, en la Figura 4.7 se puede ver un dispositivo que funciona como sensor de presencia e interruptor inteligente.



Figura 4.7: Dispositivo multipropósito, sensor presencial e interruptor inteligente.

Una vez instalado los sensores distribuidos en las habitaciones, puertas y ventanas se procede a implementar el sistema de alarma inteligente, el mismo que está dotado por un módulo GSM y un módulo WiFi, los cuales se instalarán dentro de una caja negra como se puede ver en la Figura 4.8, ésta se ubica en un lugar secreto del hogar para evitar que un intruso pueda desconectarla.



Figura 4.8: Sistema de alarma inteligente.

#### 4.1.4 Sistema de riego inteligente

Para realizar la implementación del sistema de riego inteligente es necesario primero realizar la instalación del panel solar tomando en cuenta las zonas que reciben más radiación solar durante el día, dicho panel cargará una batería de 12 voltios como se puede ver en la Figura 4.9.



Figura 4.9: Sistema de riego inteligente

Este sistema se controla de manera inalámbrica por el usuario por lo que es necesario incluir un módulo ESP8266 en el mismo, una vez construido el sistema se procede a realizar la instalación de la electroválvula como se puede apreciar en la Figura 4.10.



Figura 4.10: Sistema de riego

El agua será suministrada a las plantas por medio de una manguera, previamente perforada, que se encuentra distribuida a los largo de todas las plantas, la cual por medio del sistema difuso predice el volumen de agua que debe suministrar a las plantas.

#### 4.2 Costo de la implementación.

A continuación se proporcionará los costos exactos de los distintos componentes, módulos, accesorios y demás usados para la implementación del sistema inteligente, es importante destacar que los valores que se presentan en la Tabla 4.1 brindan una idea clara del costo que tendría el sistema completo, sin tomar en cuenta el costo de las horas de diseño, construcción y programación de cada dispositivo.

Tabla 4.1: Costos del sistema inteligente

Cantidad	Dispositivo	Precio Unitario	Precio Total
10	Interruptor Inteligente	19.71	197.1
1	Módulo ESP8266	4	4
2	Triac BT-136	0.7	1.4

2	MOC3021	0.9	1.8
1	Puente Rectificador	0.5	0.5
1	Diodo 1N4001	0.2	0.2
1	Optoacoplador 4N35	0.45	0.45
11	Resistencias(330, 100,220,100k ohm)	0.17	1.87
1	Placa Fibra de Vidrio	1	1
2	Borneras	0.15	0.3
1	Sensor PIR	0.9	0.9
2	Pulsantes Grandes	0.15	0.3
2	Led Blanco	0.07	0.14
1	Fuente BUCK (110v. AC - 5v. CD)	6	6
1	Carcasa Interruptor	0.85	0.85
<b>1</b>	<b>Alarma Inteligente</b>	<b>161</b>	<b>161</b>
1	Módulo ESP8266	4	4
1	Raspberry Pi 2	70	70
1	Pantalla Capacitiva Adafruit 2.8"	45	45
1	Fuente 12 V - 2 A.	6	6
1	Módulo GSM	12	12
1	Chip GSM	3	3
3	Sensor Magnético	1.5	4.5
5	Sensor PIR	0.9	4.5
1	Batería 12V	12	12
<b>1</b>	<b>Sistema de Riego Inteligente</b>	<b>154.4</b>	<b>154.4</b>
1	Batería 12V	12	12
1	Electroválvula	10	10
1	Varios Plastigama	3	3
1	Manguera Distribuidora	5	5
1	Relee	0.5	0.5
1	Módulo ESP8266	4	4
1	Carcasa Negra	4	4
1	Sensor Humedad y Temperatura DHT11	0.9	0.9
<b>1</b>	<b>Portón Inteligente</b>	<b>4.9</b>	<b>4.9</b>
1	Módulo ESP8266	4	4
1	Cables Hembra-Macho	1	1
1	Sensor Magnético	1.5	1.5

1	<b>Servidor Principal</b>	6.5	6.5
1	Raspberry Pi 3.	80	80
1	Pantalla resistiva de 5"	50	50
1	Fuente Alimentación	8	8
1	Memoria 16GB clase 10	8	8
<hr/>			
1	<b>Dispositivos Varios</b>	152.5	152.5
3	Emisor Infrarrojo	0.9	2.7
1	Receptor Infrarrojo	1	1
2	Cámara USB	12	24
2	Micro Procesador ONION	10	20
1	Estaño	5	5
1	Componentes Electrónicos Varios	15	15
<hr/>			
<b>TOTAL</b>			<b>\$ 676.4</b>

### 4.3 Pruebas del sistema.

Luego de la implementación de todo el sistema es necesario someterlo a varias pruebas con el fin de analizar el comportamiento del sistema ante distintos estímulos exteriores y sobre todo su adaptabilidad con los distintos miembros del hogar en el que está instalado.

#### 4.3.1 Pruebas de tiempos de respuesta

La velocidad de comunicación entre los distintos sensores y actuadores con el servidor principal y viceversa es una de las características más importantes para una mejor adaptación de las personas del hogar con el sistema implementado, para ello se realizaron las siguientes pruebas, el cual se basa en el esquema de la Figura 4.11.

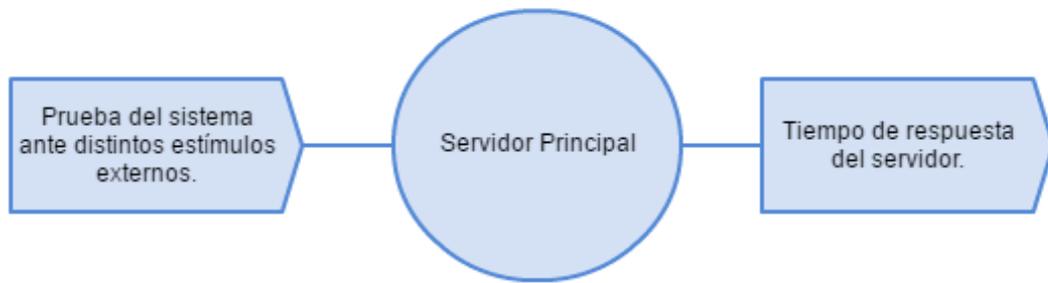


Figura 4.11: Esquema de pruebas para el servidor principal.

#### 4.3.1.1 Tiempo de respuesta del servidor.

Utilizando la librería “times” de Python la cual nos permite medir el tiempo de ejecución de las distintas tareas, se procederá a medir el tiempo de respuesta del servidor ante distintas peticiones:

- **Escenario 0: Petición interno del servidor (Figura 4.12).**

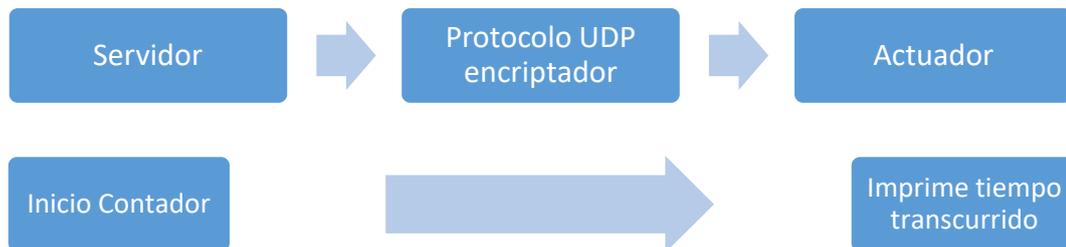


Figura 4.12: Estructura del escenario 0 para prueba de tiempo de respuesta del servidor.

Utilizando la estructura anteriormente planteada se obtiene un tiempo de respuesta de 0.032 segundos equivalentes a 32 milisegundos como se puede ver en la Figura 4.13.

```
===== RESTART: D:\Desktop\Tesis\Servidor_UDP.py =
Tiempo Transcurrido:
0.0320000648499
(13L, 0L, 'off_proyector', 'Gabriel')
13:0
```

Figura 4.13: Shell de Python para visualización de tiempo de respuesta.

- **Escenario 1: Petición desde uno de los dispositivos móviles (Figura 4.14).**

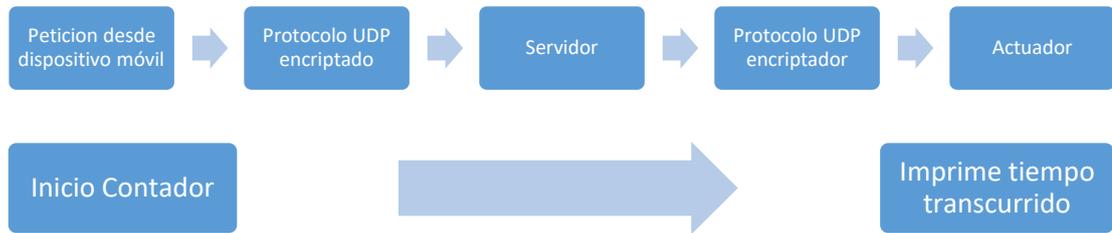


Figura 4.14: Estructura del escenario 1 para prueba de tiempo de respuesta del servidor.

Utilizando la estructura anteriormente planteada se obtiene un tiempo de respuesta de 0.311 segundos equivalentes a 311 milisegundos como se puede ver en la Figura 4.15.

```

===== RESTART: D:\Desktop\Te
(13L, 0L, 'off_proyector', 'Gabriel')
13:0
Entro
off1
192.168.1.146
5050
Tiempo Transcurrido:
0.311000108719
  
```

Figura 4.15: Shell de Python para visualización de tiempo de respuesta.

- **Escenario 2: Petición desde un lugar remoto por medio de la página (Figura 4.16).**

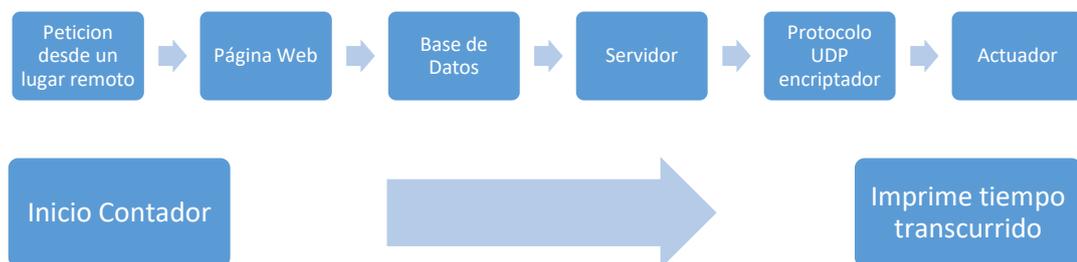


Figura 4.16: Estructura de escenario 2 para prueba de tiempo de respuesta del servidor.

Utilizando la estructura anteriormente planteada se obtiene un tiempo de respuesta de 0.402 segundos equivalentes a 402 mili Segundos como se puede ver en la Figura. 4.17.

```

===== RESTART: D:\Desktop\Te
(13L, 0L, 'off_proyector', 'Gabriel')
13:0
on#Gabriel_1
Tiempo Transcurrido:
0.401999950409

```

Figura 4.17: Shell de Python para visualización de tiempo de respuesta.

#### 4.3.1.2 Tiempo de respuesta de los distintos actuadores y sensores.

Para medir los tiempos de respuesta de los distintos dispositivos se analizará el tiempo en cada microcontrolador utilizando la sentencia “millis()” la cual nos entrega el tiempo en milisegundos que ha transcurrido para realizar cualquier tarea, dicho tiempo será almacenado y enviado por UDP al servidor en Python.

- **Escenario 0: Petición de encendido/apagado a interruptor (Figura 4.18).**

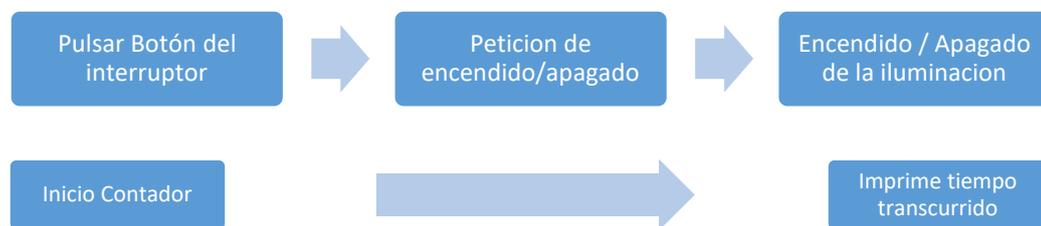


Figura 4.18: Estructura del escenario 0 para prueba de tiempo de respuesta del interruptor.

Utilizando la estructura anteriormente planteada se obtiene un tiempo de respuesta de 0.017 segundos equivalentes a 17 milisegundos como se puede ver en la Figura 4.19.

```
===== RESTART: D:\Desktop\Te
Tiempo Transcurrido:
17 mS
(13L, 0L, 'off_proyector', 'Gabriel')
13:0
```

Figura 4.19: Shell de Python para visualización de tiempo de respuesta.

- **Escenario 1: Petición de activación/desactivación de la alarma (Figura 4.20).**

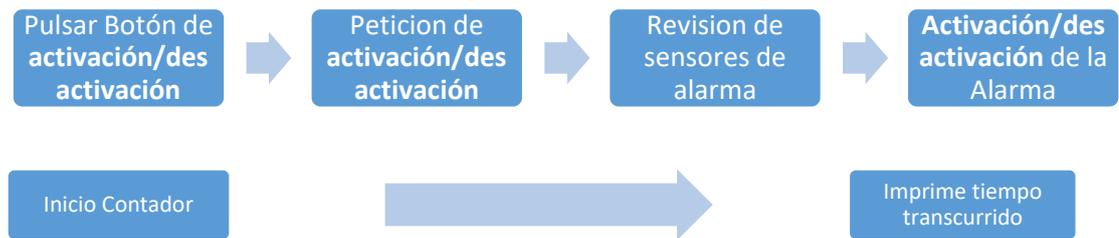


Figura 4.20: Estructura del escenario 1 para prueba de tiempo de respuesta del interruptor.

Utilizando la estructura anteriormente planteada se obtiene un tiempo de respuesta de 0.05745 segundos equivalentes a 57.45 milisegundos como se puede ver en la Figura 4.21.

```
===== RESTART:
Tiempo Transcurrido:
57,45 mS
```

Figura 4.21: Shell de Python para visualización de tiempo de respuesta.

- **Escenario 2: Petición de activación/desactivación del sensor de la puerta principal (Figura 4.22).**

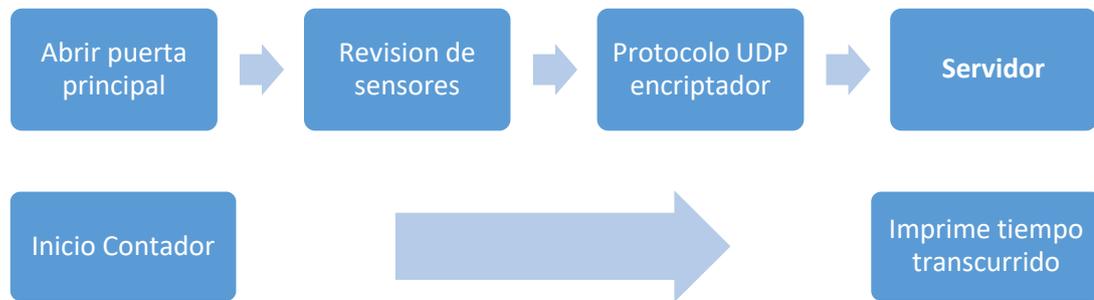


Figura 4.22: Estructura del escenario 1 para prueba de tiempo de respuesta del interruptor.

Utilizando la estructura anteriormente planteada se obtiene un tiempo de respuesta de 0.1756 segundos equivalentes a 175.6 milisegundos como se puede ver en la Figura 4.23.

```

===== REST!
Tiempo Transcurrido:
0.1756
  
```

Figura 4.23: Shell de Python para visualización de tiempo de respuesta.

#### 4.3.1.3 Tiempo de respuesta del reconocimiento de voz en Android.

Para medir el tiempo que toma Android en convertir una frase en texto se utilizará la opción Stop-Watch que nos proporciona Android Studio el cual nos devuelve el tiempo de ejecución de cada acción realizada en el teléfono, como el tiempo que tarda esta conversión está ligado al número de palabras que tiene la frase, se usará una frase promedio de prueba: “Asistente enciende la Luz del Comedor”

- **Escenario 0: Petición de encendido/apagado a interruptor (Figura 4.24).**



Figura 4.24: Estructura del escenario 0 para prueba de tiempo de respuesta del interruptor.

Utilizando la estructura anteriormente planteada se obtiene un tiempo de respuesta de 0.538 segundos equivalentes a 538 milisegundos como se puede ver en la Figura 4.25 en la cual se puede notar restando los valores en los cuales fue iniciado y finalizado el proceso de reconcomiendo.

```
logcat Monitors →  
03-24 11:26:17.099 10361-10489/com.example.gabri.tesis  
03-24 11:26:17.099 10361-10489/com.example.gabri.tesis  
03-24 11:26:17.637 10361-10645/com.example.gabri.tesis  
03-24 11:26:17.637 10361-10645/com.example.gabri.tesis  
|  
ODO Android Monitor Terminal
```

Figura 4.25: Monitor de Android para visualización de tiempo de respuesta.

### 4.3.2 Prueba del sistema ante distintos estímulos externos.

#### 4.3.2.1 Prueba de rendimiento del sistema.

Para realizar esta prueba se enviará peticiones al servidor desde todos los dispositivos que se encuentran instalados en el hogar con un tiempo de demora entre peticiones que será analizada en cada escenario, cada petición será respondida por el servidor con el número de petición que se ha enviado para así saber si le llegaron todas las peticiones y de igual manera si las mismas fueron respondidas, se realizara 4 pruebas con el fin de probar la respuesta del servidor variando los tiempos y el número de peticiones:

- **Escenario 0: 100 peticiones desde 10 dispositivos distintos con un tiempo de demora de 500 ms.**

Como se puede ver en la Figura 4.26 para el escenario 0 la respuesta del servidor es la siguiente:

Peticiones Receptadas: 1000 de 1000 peticiones enviadas, efectividad 100%

Respuesta a peticiones: 1000 de 1000 peticiones recibidas, efectividad 100%

```
===== RESTART: D:/Desktol
Se ha recibido 1000 peticiones
Se ha respondido 1000 peticiones
>>> |
```

Figura 4.26: Shell de Python para visualización respuesta a peticiones

- **Escenario 1: 100 peticiones desde 10 dispositivos distintos con un tiempo de demora de 50 ms.**

Como se puede ver en la Figura 4.27 para el escenario 1 la respuesta del servidor es la siguiente:

Peticiones Receptadas: 925 de 1000 peticiones enviadas, efectividad 92.5%

Respuesta a peticiones: 925 de 925 peticiones recibidas, efectividad 100%

```

===== RESTART: D:/Desktop,
Se ha recibido 925 peticiones
Se ha respondido 925 peticiones
>>> |

```

Figura 4.27: Shell de Python para visualización respuesta a peticiones

- **Escenario 2: 100 peticiones desde 25 dispositivos distintos con un tiempo de demora de 500 ms.**

Como se puede ver en la Figura 4.28 para el escenario 2 la respuesta del servidor es la siguiente:

Peticiones Receptadas: 2500 de 2500 peticiones enviadas, efectividad 92.5%

Respuesta a peticiones: 2500 de 2500 peticiones recibidas, efectividad 100%

```

===== RESTART: D:/Desktop
Se ha recibido 2500 peticiones
Se ha respondido 2500 peticiones
>>> |

```

Figura 4.28: Shell de Python para visualización respuesta a peticiones

- **Escenario 3: 100 peticiones desde 25 dispositivos distintos con un tiempo de demora de 50 ms.**

Como se puede ver en la Figura 4.29 para el escenario 3 la respuesta del servidor es la siguiente:

Peticiones Receptadas: 2100 de 2500 peticiones enviadas, efectividad 84%

Respuesta a peticiones: 2100 de 2100 peticiones recibidas, efectividad 100%

```

===== RESTART: D:/Desktc
Se ha recibido 2100 peticiones
Se ha respondido 2100 peticiones
>>> |

```

Figura 4.29: Shell de Python para visualización respuesta a peticiones

#### **4.3.2.2 Prueba de robustez del sistema.**

Debido a que todos los dispositivos y sistemas que están instalados en el hogar funcionan de manera independiente al servidor principal, la mejor manera de probar la robustez del mismo es analizar su comportamiento ante distintos estímulos externos, los cuales se analizarán con escenarios.

- **Escenario 0: Desconexión de la red Wifi**

Al perderse la conexión con la base de datos el sistema pasa a “modo offline”, en el cual el servidor deja de realizar consultas a la base de datos por lo tanto el acceso remoto al hogar queda inhabilitado, de esta manera el funcionamiento del sistema dentro del hogar no cambia, esto permite que la conexión con la base de datos solo sea necesario para consultas remotas.

- **Escenario 1: Desconexión de la red eléctrica.**

Como se mencionó anteriormente todos los sistemas funcionan de manera independiente por lo que cuando exista una desconexión de la red eléctrica los sistemas que pasaran a “modo batería” son: la alarma por el hecho de que la misma debe funcionar todo el tiempo y los distintos sensores que componen la alarma, ya que el consumo del sistema de alarma es bajo se puede agregar más dispositivos a la conexión de batería como luces de guía para que el hogar no se quede completamente a oscuras.

### **4.4 Ventajas y Desventajas del Sistema**

#### **4.4.1 Ventajas:**

- La creación de un sistema inteligente utilizando alternativas de bajo costo es factible si se puede entregar mayor robustez al sistema por medio de programación.

- Los dispositivos contruidos, programados e implementados en esta tesis son de fácil despliegue y de configuración amigable para cualquier usuario que posea un dispositivo con WiFi.
- Otra de las grandes ventajas que posee el sistema es que cada dispositivo tiene abierto el modo de programación OTA, el cual nos permite actualizar, mejorar y reconstruir el *software* de manera remota sin necesidad de estar presentes en dicho lugar, lo cual entrega autonomía en el monitoreo de cada producto.
- El ahorro energético es una de las características que posee este sistema, ya que al estar proporcionado de sensores en toda la casa puede realizar un monitoreo continuo de todas las habitaciones y así saber que habitación esta con la luz encendida sin tener presencia de algún usuario.
- El sistema de alarma inteligente nos proporciona un monitoreo constante de los distintos sensores que se encuentran repartidos por la casa, además cuando algún intruso activa la alarma, este sistema está en la capacidad de enviar notificaciones al celular por medio de la red GSM con mensajes de texto y llamadas o por medio de la red WiFi utilizando notificaciones basadas en la aplicación en Android.
- Cada sistema implementado en esta tesis funciona de manera independiente por lo que no es necesario adquirir el servidor principal para que los distintos sistemas como interruptores inalámbricos, alarma inteligente, riego inteligente y demás funcionen de manera correcta.
- El servidor principal nos entrega una conexión remota desde cualquier parte del mundo con la casa, lo cual permite interactuar con los distintos sistemas y dispositivos instalados en cualquier momento, brindando un mayor control y monitoreo a todos los usuarios del hogar.

- El sistema domótico permite integrar cualquier dispositivo que no sea inteligente al sistema, por medio de la integración de un pequeño circuito que permita controlar sus funciones principales.
- Se puede enlazar con casi cualquier dispositivo con acceso a internet.
- El sistema de riego inteligente promueve en el usuario un uso más responsable y óptimo del agua, realizando el riego de las plantas de manera autónoma y automatizada.
- El reconocimiento de voz entrega mayor comodidad al momento de interactuar con el sistema, siendo una interfaz de comunicación con el asistente muy ventajoso para personas con distintas discapacidades.
- El asistente está dotado de un diccionario que le permite entender con mayor facilidad las distintas peticiones que realice el usuario, lo cual evita que el usuario tenga que aprender frases exactas para interactuar con la casa, brindando mayor flexibilidad al momento de interactuar con el asistente.
- Al tener un sistema de reconocimiento de voz que no hace uso de conexión a internet para realizar la traducción de voz a texto, nos brinda mayor velocidad de respuesta y mayor robustez al sistema ya que el reconocimiento se puede realizar en cualquier momento.
- Como consecuencia de todos los anteriores apartados se consigue un nivel de confort muy superior provocando que la calidad de vida de los usuarios del hogar aumente considerablemente.

#### **4.4.2 Desventajas:**

- Al ser un sistema relativamente nuevo, en su aplicación se pueden experimentar fallos en los sistemas, fallos en la comunicación, etc.

- Se puede dar un entorpecimiento del usuario, dependiendo del grado de automatización del sistema, por lo que siempre se optará por construir e implementar sistemas ergonómicos y de fácil manejo.
- Al ser el internet su principal medio de comunicación se expone al sistema a ser invadido por algún *software* malicioso contribuyendo a la manipulación y monitoreo sin permiso por el usuario principal.

#### 4.5 Conclusiones

- Como se puede apreciar en la Figura 4.30 el tiempo máximo de respuesta del asistente virtual es aproximadamente medio segundo, siendo un tiempo aceptable para una comunicación fluida con el usuario.

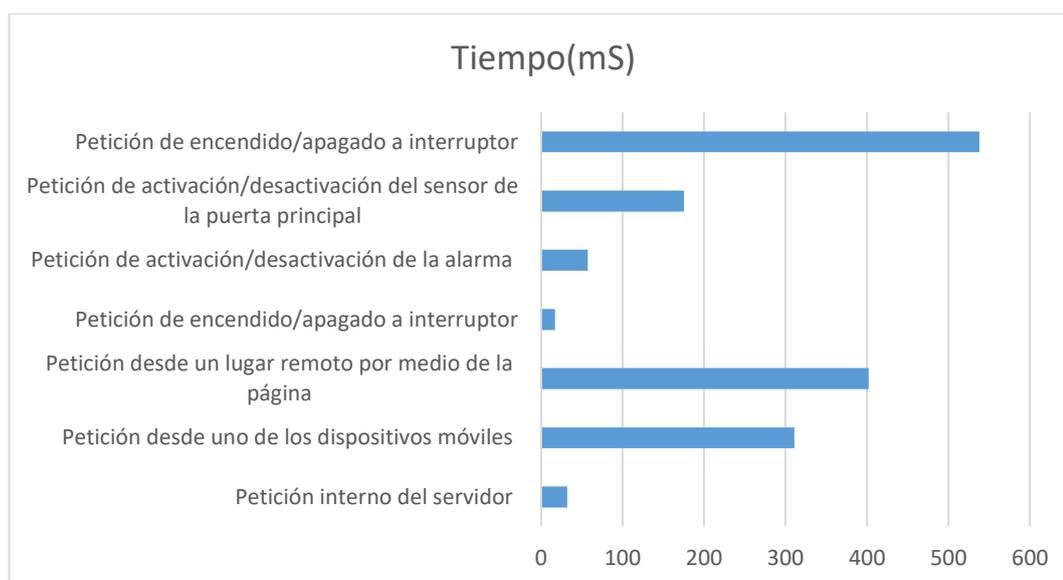


Figura 4.30: Resumen tiempos de respuesta del servidor.

- En la Figura 4.31 se puede apreciar el rendimiento del sistema ante distintos estímulos, se puede apreciar que existe una sobresaturación del sistema en el Escenario 1 en el cual se tiene un rendimiento de 92.5%, esta baja en el

rendimiento es causado por la poca asignación de memoria que tiene asignado predeterminadamente el servidor.

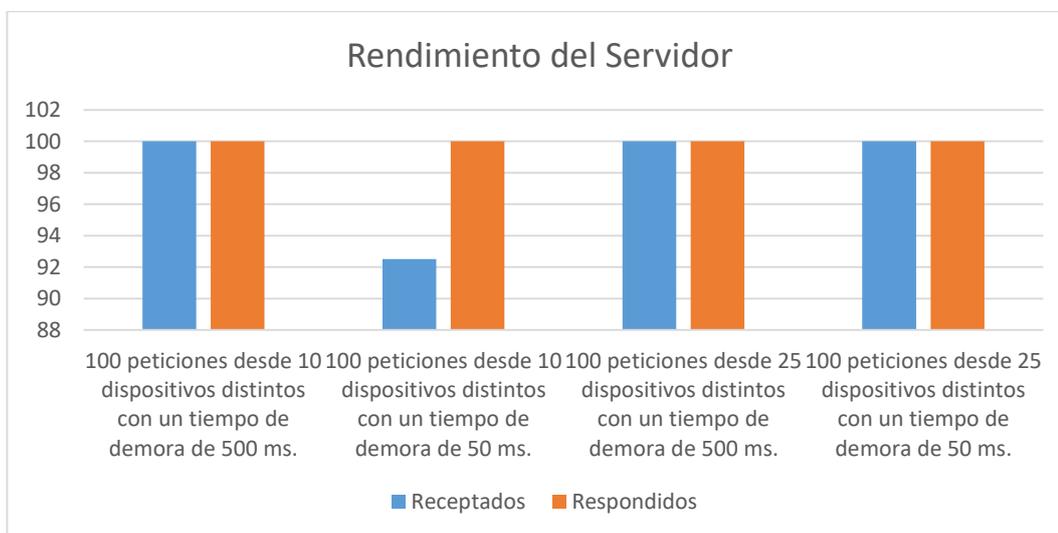


Figura 4.31: Resumen de rendimiento del servidor.

- En la Figura 4.32 se muestra el consumo energético en KWH, en el cual se puede apreciar claramente que la implementación del sistema conjuntamente con sensores que permiten un uso optimizado de la energía eléctrica, funcionan de manera correcta provocando un ahorro energético del 20.53% entre el mes de diciembre hasta el mes de marzo.

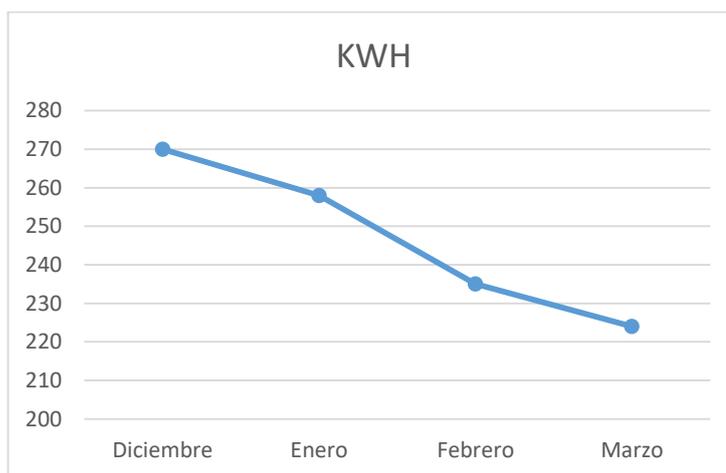


Figura 4.32: Gráfico de consumo energético desde que se implementó el sistema

## CONCLUSIONES

Luego de implementar el sistema en su totalidad y de realizar las distintas pruebas para determinar los alcances del sistema inteligente, se puede concluir lo siguiente:

- El sistema implementado cumple con los distintos objetivos planteados.
- La investigación acerca de los distintos sistemas inteligentes forja una gran base al momento de elegir la tecnología que se adapte más a las expectativas planteadas.
- El AV(Asistente Virtual) programado está en la capacidad de interactuar con los distintos usuarios del hogar, utilizando varias interfaces visuales, permitiendo al usuario manipular todos los recursos o dispositivos implementados.
- La red de sensores coopera conjuntamente con el AV para mantener informado a cada usuario de la casa sobre las distintas acciones físicas o climáticas que sucedan en el hogar, además, el AV utiliza la información proporcionada por los sensores para realizar tareas automatizadas como el riego del jardín.
- Todos los actuadores implementados en este trabajo de graduación, se encuentran a disposición del usuario, el cual puede acceder a ellas por medio de las distintas interfaces de comunicación.
- El AV permite al usuario manipular el hogar de manera remota a través de la aplicación Web o de la aplicación móvil.
- El comportamiento en conjunto de los distintos sistemas funciona de manera correcta y óptima, permitiendo la implementación de más sistemas o dispositivos a la red principal.
- El tipo de comunicación elegido fue Wifi, para el cual se utilizó el módulo ESP8266-12E; una de las ventajas, que cabe mencionar, es el bajo costo del mismo, permitiendo integrar un mayor número de dispositivos a la red planteada.
- Una de las grandes cualidades del sistema es la integración del control de electrodomésticos a la red principal, con el fin de manejarlos de manera

automatizada; esta integración se realiza en cualquier tipo de electrodoméstico, evitando tener que cambiarlo para integrarlo a la red.

- Entregar mayor robustez por medio de programación es una característica muy importante a la hora de implementar los distintos dispositivos, ya que un sistema estable proporciona una mayor confianza para los distintos usuarios que manipulan los recursos ofrecidos.
- Por el hecho de que el sistema se implementó en una casa habitada, es necesario que los cambios en la infraestructura del hogar sean mínimos, guardando siempre su estética, razón por la que se eligió una tecnología inalámbrica, la implementación de los distintos dispositivos no varía la estética del hogar ya que se evita el cableado en su mayor parte.
- En el hogar en el cual se implementó el sistema habitan personas de varias edades, desde niños hasta personas adultas, por lo cual se ha implementado distintas interfaces para utilizar los distintos recursos del hogar, entre las interfaces más importantes se tiene:
  - Aplicación Móvil.
  - Aplicación Web
  - Reconocimiento de Voz
  - Servidor Principal
  - Pantallas Táctiles distribuidas en el hogar.
  - Interfaces manuales en cada dispositivo (botones, pulsantes, interruptores, etc.)
- La interfaz gráfica es una característica muy importante en este tipo de sistemas ya que es la presentación principal para el usuario, el cual le va permitir interactuar tanto con el asistente virtual como también con el hogar y sus

dispositivos inteligentes, por este hecho se ha diseñado un entorno virtual amigable y de fácil manejo.

- La creación de un sistema inteligente utilizando alternativas de bajo costo es factible utilizando algoritmos que garanticen el óptimo desempeño del mismo.
- Los dispositivos construidos, programados e implementados en esta tesis son de fácil despliegue y de configuración amigable para cualquier usuario que posea un dispositivo con WiFi.
- El sistema posee en todos sus dispositivos un modo de programación OTA, el cual nos permite actualizar, mejorar y reconstruir el *software* de manera remota sin necesidad de estar presentes en dicho lugar, lo cual entrega autonomía en el monitoreo de cada producto.
- Al poseer el valor de todos los sensores instalados en el hogar, el sistema promueve un uso eficaz de la energía eléctrica, realizando un monitoreo continuo de todas las habitaciones y así determinar que habitación está con la luz encendida sin tener presencia de algún usuario.
- El sistema de alarma inteligente proporciona al AV un monitoreo constante de los distintos sensores que se encuentran repartidos por la casa, y esto sumado a las distintas notificaciones GSM, indicadas en capítulos anteriores, entrega una mayor seguridad al hogar.
- Cada sistema implementado en esta tesis funciona de manera independiente, por lo que no es necesario adquirir el SP(Servidor Principal) para que los distintos sistemas inteligentes como: interruptores inalámbricos, alarma, riego y demás aplicaciones funcionen correctamente.
- El SP nos entrega una conexión remota desde cualquier parte del mundo con la casa, lo cual permite interactuar con los distintos sistemas y dispositivos

instalados en cualquier momento, brindando un mayor control y monitoreo a todos los usuarios del hogar.

- El sistema domótico permite integrar cualquier dispositivo que no sea inteligente al sistema, por medio de la integración de un pequeño circuito que permita controlar sus funciones principales.
- Se puede enlazar con casi cualquier dispositivo con acceso a internet.
- El sistema de riego inteligente promueve en el usuario un uso más responsable y óptimo del agua, realizando el riego de las plantas de manera autónoma y automatizada.
- El reconocimiento de voz entrega mayor comodidad al momento de interactuar con el sistema, siendo una interfaz de comunicación con el asistente muy ventajoso para personas con distintas discapacidades.
- Todas las características utilizadas para brindar un mayor nivel de confort en el hogar, se pueden adaptar a una institución pública para ofrecer a personas con discapacidad visual una alternativa para moverse dentro de las instalaciones, por lo que se ha optado por presentar una investigación conjuntamente con docentes de la Escuela de Ingeniería Electrónica de la Universidad del Azuay, con el fin de implementar este sistema, promoviendo un mayor nivel de accesibilidad al establecimiento.

## RECOMENDACIONES

Las recomendaciones en este apartado se basan en la experiencia de haber construido, programado e implementado un sistema domótico en un hogar habitado.

- Se recomienda realizar un estudio con más detalle del sistema de reconocimiento de rostro, utilizando distintas técnicas de clasificado basado en redes neuronales, permitiendo al AV tener una mejor desempeño en la comunicación con cada usuario del hogar.
- Para mejorar el rendimiento del circuito interruptor es necesario la integración de un microcontrolador extra, el cual debe ocuparse para realizar el control de voltaje a la salida de cada foco, independizando al módulo ESP8266-12E para que realice solamente la comunicación con el servidor.
- Es recomendable realizar una investigación acerca del protocolo RTSP<sup>8</sup>, el cual permite realizar una transmisión en tiempo real directamente en la página Web, permitiendo al usuario tener un mejor monitoreo remoto del hogar.
- Para un mejor desempeño del acceso remoto, se recomienda implementar una comunicación UDP de la página Web con el servidor principal, permitiendo una conexión más fluida entre el usuario remoto y el hogar.
- Para un mejor desempeño de la red wifi principal es recomendable realizar un programa que enlace al servidor principal con la configuración principal del *router*, permitiendo al AV tener un mejor control de los distintos dispositivos de la red.
- Al abrir un puerto de comunicación entre la página web y el SP, el sistema queda propenso a ataques o virus, para lo cual es necesario mejorar la seguridad y encriptación en la comunicación.

---

<sup>8</sup> RTSP: Real Time Streaming Protocol

## REFERENCIAS

- Adafruit, E. I. (2017). *Adafruit*. Obtenido de <https://www.adafruit.com/>
- Álvarez, J. L. (2004). Aplicación de un Sistema Experto para el desarrollo de Sistema Evaluador del modelo Capability Maturity Model (CMM) niveles dos y tres. *Universidad de las Américas Puebla*.
- Alvarez, M. A. (05 de Junio de 2009). *desarrolloweb.com*. Obtenido de <https://desarrolloweb.com>
- Android Studio. (2017). Obtenido de <https://developer.android.com/studio/index.html?hl=es-419>
- Bhatt, A. &. (2016). Cost effective digitization of home appliances for home automation with low-power WiFi devices. *Advances in Electrical, Electronics, Information, Communication and Bio-Informatics*.
- Casadomo. (2017). Obtenido de <https://www.casadomo.com/2015/10/08/nuevos-detectores-de-movimiento-de-legrand-para-el-ahorro-energetico>
- Computer Hope*. (2016). Obtenido de <http://www.computerhope.com>
- Dev, O. (27 de Oct de 2014). *Docs opencv*. Obtenido de [http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\\_tutorial.html#algorithmic-description](http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#algorithmic-description)
- D-Robotics. (30 de 07 de 2010). DHT11 Humidity & Temperature Sensor. UK.
- ElectroniLab. (2017). Obtenido de <https://electronilab.co/tienda/electrovalvula-valvula-selenoide-agua-12v-dc-12/>
- Fritz, R. (2016). *about tech*. Obtenido de <http://compnetworking.about.com>
- Fundation, R. P. (2017). *Raspberry Pi*. Obtenido de <https://www.raspberrypi.org/>
- Future Electronics*. (2010). Obtenido de <http://www.futureelectronics.com>
- Future, B. S. (s.f.). *chinasolar-panel*. Obtenido de <http://www.chinasolar-panel.com/es/>

- Gauger, M. M. (2011). Prototyping sensor-actuator networks for home automation. In Proceedings of the workshop on Real-world wireless sensor networks. *ACM*.
- GeekBot. (2017). Obtenido de <http://www.geekbotelectronics.com/dht11-sensor-de-humedad-y-temperatura/>
- HetPro. (2017). *HetPro*. Obtenido de <http://hetpro-store.com/TUTORIALES/sim900-gsm-shieldarduino/>
- Hubpages. (4 de Septiembre de 2014). Obtenido de Hubpages: <http://hubpages.com>
- IOT, E. S. (2015). *Espressif Systems*. Obtenido de <http://bbs.espressif.com/>
- Juan David Gutiérrez, W. R. (2004). Lógica difusa como herramienta para la bioindicación de la calidad del agua con macroinvertebrados acuáticos en la sabana de bogotá. *Instituto de Ciencias Naturales, Universidad Nacional de Colombia*.
- Martínez, D. B. (2012). Redes de Sensores y Actuadores Inalámbricas: Una Caracterización y Caso de Estudio para Aplicaciones Médicas en Espacios Cerrados. *niversidad Rovira i Virgili de Tarragona, España*.
- Mathworks. (2017). Obtenido de <https://www.mathworks.com/products/fuzzy-logic.html>
- Networking Basics*. (12 de Seotiembre de 2014). Obtenido de <http://www.networking-basics.net>
- Ngai, E. C. (2012). Reliable reporting of delay-sensitive events in wireless sensor-actuator networks. *2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems*.
- Oracle. (21017). *MySQL*. Obtenido de <https://www.mysql.com/>
- PIR, P. (2017). Obtenido de <http://www.prometec.net/sensor-pir/>
- Prometec. (s.f.). *Prometec*. Obtenido de <http://www.prometec.net/rpi-gpio/>
- RAE. (2014). *Real Academia Española*. Obtenido de <http://www.rae.es/>
- Rocha, S. (30 de Abril de 2014). *De todo un Seba*. Obtenido de <http://alumnos.informatica.utem.cl/~srocha>

- Rodriguez, A. (2017). *IBM Developers Work*. Obtenido de <https://www.ibm.com/developerworks/library/ws-restful/>
- Safelincs*. (2016). Obtenido de <http://www.safelincs.co.uk>
- Shenzen Anxinke Technology. (2017). ESP-12E WiFi Module.
- SimpliSafe*. (24 de Julio de 2013). Obtenido de <http://simplisafe.com>
- Techopedia*. (2013). Obtenido de <https://www.techopedia.com>
- Telecom ABC*. (2012). Obtenido de <http://www.telecomabc.com>
- The Gadget Square*. (2013). Obtenido de <http://thegadgetsquare.com>
- The jQuery Foundation. (2017). Obtenido de <https://jquery.com/>
- Vito, C. E. (2006). Introducción al razonamiento aproximado: Lógica Difusa. *Revista Argentina de Medicina Respiratoria*.
- Waveshare. (2017). *waveshare*. Obtenido de <http://www.waveshare.com/>
- Whatis*. (Septiembre de 2005). Obtenido de <http://whatis.techtarget.com>
- Wimpress, M., & Madhavan, R. (2017). *ubuntu-mate*. Obtenido de <https://ubuntu-mate.org/raspberry-pi/>
- Yang, M.-H. (2012). Kernel EigenFaces vs Kernel Fisherfaces: Face Recognition Using Kernel Methods. *IEEE International Conferencie on Automatic Face and Gesture Recognition*.

**ANEXOS.****Anexo 1: Base de Datos**

```
import MySQLdb
```

```
def update(tabla,Nombre,Estado):
```

```
    sql = "UPDATE %s SET Estado = %i WHERE Nombre LIKE '%s'" %  
(tabla,Estado,Nombre)
```

```
    try:
```

```
        cursor.execute(sql)
```

```
        db.commit()
```

```
    except:
```

```
        db.rollback()
```

```
def eliminar(tabla,Numero):
```

```
    sql = "DELETE FROM %s WHERE %i" % (tabla,Numero)
```

```
    try:
```

```
        cursor.execute(sql)
```

```
        db.commit()
```

```
    except:
```

```
        db.rollback()
```

```
def insertar(tabla, Numero, Nombre, Tipo, Estado):
```

```
sql = "INSERT INTO %s(`Numero`, `Nombre`, `Tipo`, `Estado`) VALUES
(%i,%s,%s,%i)" % (tabla, Numero, Nombre, Tipo, Estado)
```

```
try:
```

```
    cursor.execute(sql)
```

```
    db.commit()
```

```
except:
```

```
    db.rollback()
```

```
def leer(tabla):
```

```
    cursor.execute("SELECT * FROM %s;" % (tabla))
```

```
    rows = cursor.fetchall()
```

```
    #for row in rows:
```

```
        # print row
```

```
    return rows
```

```
def Consultar(Valor,Nombre):
```

```
    cursor.execute("SELECT `IP`, `Tabla` FROM `direcciones` WHERE %s LIKE
'%s'" % (Valor, Nombre))
```

```
    rows = cursor.fetchall()
```

```
    return rows
```

```
def Consultar_Nombre(Valor,Nombre):
```

```
    cursor.execute("SELECT `Nombre`, `Tabla` FROM `direcciones` WHERE %s
LIKE '%s'" % (Valor, Nombre))
```

```
rows = cursor.fetchall()

return rows

def init_sql():

    global db

    global cursor

    db = MySQLdb.connect("localhost","root","","asistente" )

    cursor = db.cursor()

def insertar_horario(hora,data):

    sql = "INSERT INTO `horario` (`hora`, `accion`) VALUES ('%s', '%s')" % (hora,
data)

    try:

        cursor.execute(sql)

        db.commit()

    except:

        db.rollback()

def consultar_horario(hora):

    cursor.execute("SELECT `accion` FROM `horario` WHERE hora LIKE '%s'" %
(hora))

    rows = cursor.fetchall()

    return rows

def eliminar_horario(hora):
```

```
sql = "DELETE FROM horario WHERE hora LIKE '%s'" % (hora)

try:

    cursor.execute(sql)

    db.commit()

except:

    db.rollback()

def update_horario(hora,data):

    print hora

    print data

    sql = "UPDATE horario SET accion = '%s' WHERE hora LIKE '%s'" % (data,
hora)

    try:

        cursor.execute(sql)

        db.commit()

    except:

        db.rollback()
```

## **Anexo 2: Lógica Fuzzy Iluminación.**

```
#include <FuzzyRule.h>

#include <FuzzyComposition.h>

#include <Fuzzy.h>
```

```
#include <FuzzyRuleConsequent.h>

#include <FuzzyOutput.h>

#include <FuzzyInput.h>

#include <FuzzyIO.h>

#include <FuzzySet.h>

#include <FuzzyRuleAntecedent.h>

float cont, dist ;

// Step 1 - Instantiating an object library

Fuzzy* fuzzy = new Fuzzy();

void setup(){

  Serial.begin(9600);

  // Step 2 - Creating a FuzzyInput distance

  FuzzyInput* luz = new FuzzyInput(1);// With its ID in param

  // Funciones de Pertenencia

  FuzzySet* Muy_Bajo = new FuzzySet(-0.25, 0, 0, 0.25);

  luz->addFuzzySet(Muy_Bajo);

  FuzzySet* Bajo = new FuzzySet(0, 0.25, 0.25, 0.5);

  luz->addFuzzySet(Bajo);

  FuzzySet* Medio = new FuzzySet(0.25, 0.5, 0.5, 0.75);

  luz->addFuzzySet(Medio);
```

```
FuzzySet* Alto = new FuzzySet(0.5, 0.75, 0.75, 1);

luz->addFuzzySet(Alto);

FuzzySet* Muy_Alto = new FuzzySet(0.75, 1, 1, 1.25);

luz->addFuzzySet(Muy_Alto);

fuzzy->addFuzzyInput(luz);

FuzzyOutput* salida = new FuzzyOutput(1);

// Funciones de Pertenencia

FuzzySet* Muy_Bajo1 = new FuzzySet(-0.25, 0, 0, 0.25);

salida->addFuzzySet(Muy_Bajo1);

FuzzySet* Bajo1 = new FuzzySet(0, 0.25, 0.25, 0.5);

salida->addFuzzySet(Bajo1);

FuzzySet* Medio1 = new FuzzySet(0.25, 0.5, 0.5, 0.75);

salida->addFuzzySet(Medio1);

FuzzySet* Alto1 = new FuzzySet(0.5, 0.75, 0.75, 1);

salida->addFuzzySet(Alto1);

FuzzySet* Muy_Alto1 = new FuzzySet(0.75, 1, 1, 1.25);

salida->addFuzzySet(Muy_Alto1);

fuzzy->addFuzzyOutput(salida);

FuzzyRuleAntecedent* ifLuzMuyBajo = new FuzzyRuleAntecedent();

ifLuzMuyBajo->joinSingle(Muy_Bajo);
```

```

FuzzyRuleConsequent* thenSalidaMuyAlto = new FuzzyRuleConsequent();

thenSalidaMuyAlto->addOutput(Muy_Alto1);

FuzzyRule* fuzzyRule01 = new FuzzyRule(1, ifLuzMuyBajo, thenSalidaMuyAlto);

fuzzy->addFuzzyRule(fuzzyRule01);

FuzzyRuleAntecedent* ifLuzBajo = new FuzzyRuleAntecedent(); // Instantiating an
Antecedent to expression

ifLuzBajo->joinSingle(Bajo); // Adding corresponding FuzzySet to Antecedent
object

FuzzyRuleConsequent* thenSalidaAlto = new FuzzyRuleConsequent(); //
Instantiating a Consequent to expression

thenSalidaAlto->addOutput(Alto1);// Adding corresponding FuzzySet to
Consequent object

// Instantiating a FuzzyRule object

FuzzyRule* fuzzyRule02 = new FuzzyRule(2, ifLuzBajo, thenSalidaAlto);

fuzzy->addFuzzyRule(fuzzyRule02); // Adding FuzzyRule to Fuzzy object

FuzzyRuleAntecedent* ifLuzMedio = new FuzzyRuleAntecedent(); // Instantiating
an Antecedent to expression

ifLuzMedio->joinSingle(Medio); // Adding corresponding FuzzySet to Antecedent
object

FuzzyRuleConsequent* thenSalidaMedio = new FuzzyRuleConsequent(); //
Instantiating a Consequent to expression

thenSalidaMedio->addOutput(Medio1);// Adding corresponding FuzzySet to
Consequent object

```

```
// Instantiating a FuzzyRule object
```

```
FuzzyRule* fuzzyRule03 = new FuzzyRule(3, ifLuzMedio, thenSalidaMedio);
```

```
fuzzy->addFuzzyRule(fuzzyRule03); // Adding FuzzyRule to Fuzzy object
```

```
FuzzyRuleAntecedent* ifLuzAlto = new FuzzyRuleAntecedent(); // Instantiating an
Antecedent to expression
```

```
ifLuzAlto->joinSingle(Alto); // Adding corresponding FuzzySet to Antecedent
object
```

```
FuzzyRuleConsequent* thenSalidaBajo = new FuzzyRuleConsequent(); //
Instantiating a Consequent to expression
```

```
thenSalidaBajo->addOutput(Bajo1); // Adding corresponding FuzzySet to
Consequent object
```

```
// Instantiating a FuzzyRule object
```

```
FuzzyRule* fuzzyRule04 = new FuzzyRule(4, ifLuzAlto, thenSalidaBajo);
```

```
fuzzy->addFuzzyRule(fuzzyRule04); // Adding FuzzyRule to Fuzzy object
```

```
FuzzyRuleAntecedent* ifLuzMuyAlto = new FuzzyRuleAntecedent(); //
Instantiating an Antecedent to expression
```

```
ifLuzMuyAlto->joinSingle(Muy_Alto); // Adding corresponding FuzzySet to
Antecedent object
```

```
FuzzyRuleConsequent* thenSalidaMuyBajo = new FuzzyRuleConsequent(); //
Instantiating a Consequent to expression
```

```
thenSalidaMuyBajo->addOutput(Muy_Bajo1); // Adding corresponding FuzzySet to
Consequent object
```

```
// Instantiating a FuzzyRule object
```

```
FuzzyRule* fuzzyRule05 = new FuzzyRule(5, ifLuzMuyAlto, thenSalidaMuyBajo);  
fuzzy->addFuzzyRule(fuzzyRule05);  
  
cont=0;  
  
dist=0;  
  
Serial.print("Entrada");  
  
Serial.print(" --> ");  
  
Serial.println("Salida");  
  
}  
  
void loop(){  
  
dist = cont;  
  
fuzzy->setInput(1, dist);  
  
fuzzy->fuzzify();  
  
float output = fuzzy->defuzzify(1);  
  
Serial.print(cont);  
  
Serial.print(" --> ");  
  
Serial.println(output);  
  
delay(250);  
  
cont = cont + 0.05;  
  
}
```

**Anexo 3: Entrenamiento EigenFaces**

```
import numpy as np

import cv2

import sys

import os

FREQ_DIV = 5 #frequency divider for capturing training images

RESIZE_FACTOR = 4

NUM_TRAINING = 100

class TrainEigenFaces:

    def __init__(self):

        cascPath = "haarcascade_frontalface_default.xml"

        self.face_cascade = cv2.CascadeClassifier(cascPath)

        self.face_dir = 'face_data'

        self.face_name = 'Gabriel'

        self.path = os.path.join(self.face_dir, self.face_name)

        if not os.path.isdir(self.path):

            os.mkdir(self.path)
```

```
self.model = cv2.createEigenFaceRecognizer()

self.count_captures = 0

self.count_timer = 0

def capture_training_images(self):

    video_capture = cv2.VideoCapture(0)

    while True:

        self.count_timer += 1

        ret, frame = video_capture.read()

        inImg = np.array(frame)

        outImg = self.process_image(inImg)

        cv2.imshow('Video', outImg)

        # When everything is done, release the capture on pressing 'q'

        if cv2.waitKey(1) & 0xFF == ord('q'):

            video_capture.release()

            cv2.destroyAllWindows()

            return

    def process_image(self, inImg):

        frame = cv2.flip(inImg,1)

        resized_width, resized_height = (112, 92)

        if self.count_captures < NUM_TRAINING:
```

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

gray_resized = cv2.resize(gray, (gray.shape[1]/RESIZE_FACTOR,
gray.shape[0]/RESIZE_FACTOR))

faces = self.face_cascade.detectMultiScale(

    gray_resized,

    scaleFactor=1.1,

    minNeighbors=5,

    minSize=(30, 30),

    flags=cv2.cv.CV_HAAR_SCALE_IMAGE

)

if len(faces) > 0:

    areas = []

    for (x, y, w, h) in faces:

        areas.append(w*h)

    max_area, idx = max([(val,idx) for idx,val in enumerate(areas)])

    face_sel = faces[idx]

    x = face_sel[0] * RESIZE_FACTOR

    y = face_sel[1] * RESIZE_FACTOR

    w = face_sel[2] * RESIZE_FACTOR

    h = face_sel[3] * RESIZE_FACTOR

    face = gray[y:y+h, x:x+w]
```

```

face_resized = cv2.resize(face, (resized_width, resized_height))

img_no = sorted([int(fn[:fn.find('.')]) for fn in os.listdir(self.path) if fn[0]!='.']+[0])[-1] + 1

if self.count_timer%FREQ_DIV == 0:

    cv2.imwrite('%s/%s.png' % (self.path, img_no), face_resized)

    self.count_captures += 1

    print "Captured image: ", self.count_captures

    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 3)

    cv2.putText(frame, self.face_name, (x - 10, y - 10),
cv2.FONT_HERSHEY_PLAIN, 1,(0, 255, 0))

elif self.count_captures == NUM_TRAINING:

    print "Training data captured. Press 'q' to exit."

    self.count_captures += 1

return frame

def eigen_train_data(self):

    imgs = []

    tags = []

    index = 0

    for (subdirs, dirs, files) in os.walk(self.face_dir):

        for subdir in dirs:

```

```

img_path = os.path.join(self.face_dir, subdir)

for fn in os.listdir(img_path):

    path = img_path + '/' + fn

    tag = index

    imgs.append(cv2.imread(path, 0))

    tags.append(int(tag))

    index += 1

(imgs, tags) = [np.array(item) for item in [imgs, tags]]

self.model.train(imgs, tags)

self.model.save('eigen_trained_data.xml')

print "Training completed successfully"

return

if __name__ == '__main__':

    trainer = TrainEigenFaces()

    trainer.capture_training_images()

    trainer.eigen_train_data()

    print "Type in next user to train, or press Recognize"

```

#### **Anexo 4: Entrenamiento FisherFaces**

```
import numpy as np
```

```
import cv2

import sys

import os

FREQ_DIV = 5 #frequency divider for capturing training images

RESIZE_FACTOR = 4

NUM_TRAINING = 100

class TrainFisherFaces:

    def __init__(self):

        cascPath = "haarcascade_frontalface_default.xml"

        self.face_cascade = cv2.CascadeClassifier(cascPath)

        self.face_dir = 'face_data'

        self.face_name = 'Gabriel'#sys.argv[1]

        self.path = os.path.join(self.face_dir, self.face_name)

        if not os.path.isdir(self.path):

            os.mkdir(self.path)

        self.model = cv2.createFisherFaceRecognizer()

        self.count_captures = 0

        self.count_timer = 0

    def capture_training_images(self):

        video_capture = cv2.VideoCapture(0)
```

```

while True:

    self.count_timer += 1

    ret, frame = video_capture.read()

    inImg = np.array(frame)

    outImg = self.process_image(inImg)

    cv2.imshow('Video', outImg)

    # When everything is done, release the capture on pressing 'q'

    if cv2.waitKey(1) & 0xFF == ord('q'):

        video_capture.release()

        cv2.destroyAllWindows()

        return

def process_image(self, inImg):

    frame = cv2.flip(inImg,1)

    resized_width, resized_height = (112, 92)

    if self.count_captures < NUM_TRAINING:

        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        gray_resized = cv2.resize(gray, (gray.shape[1]/RESIZE_FACTOR,
gray.shape[0]/RESIZE_FACTOR))

        faces = self.face_cascade.detectMultiScale(

            gray_resized,

            scaleFactor=1.1,

```

```

minNeighbors=5,

minSize=(30, 30),

flags=cv2.cv.CV_HAAR_SCALE_IMAGE

)

if len(faces) > 0:

    areas = []

    for (x, y, w, h) in faces:

        areas.append(w*h)

    max_area, idx = max([(val,idx) for idx,val in enumerate(areas)])

    face_sel = faces[idx]

    x = face_sel[0] * RESIZE_FACTOR

    y = face_sel[1] * RESIZE_FACTOR

    w = face_sel[2] * RESIZE_FACTOR

    h = face_sel[3] * RESIZE_FACTOR

    face = gray[y:y+h, x:x+w]

    face_resized = cv2.resize(face, (resized_width, resized_height))

    img_no = sorted([int(fn[:fn.find('.')]) for fn in os.listdir(self.path) if fn[0]!='.'
]+[0])[-1] + 1

    if self.count_timer%FREQ_DIV == 0:

        cv2.imwrite('%s/%s.png' % (self.path, img_no), face_resized)

        self.count_captures += 1

```

```
print "Captured image: ", self.count_captures

cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 3)

cv2.putText(frame, self.face_name, (x - 10, y - 10),
cv2.FONT_HERSHEY_PLAIN, 1,(0, 255, 0))

elif self.count_captures == NUM_TRAINING:

    print "Training data captured. Press 'q' to exit."

    self.count_captures += 1

return frame

def are_enough_faces(self):

    existingFaces = 0

    for (subdirs, dirs, files) in os.walk(self.face_dir):

        for subdir in dirs:

            existingFaces += 1

    if existingFaces > 1:

        return True

    else:

        return False

def fisher_train_data(self):

    imgs = []

    tags = []
```

```
index = 0

for (subdirs, dirs, files) in os.walk(self.face_dir):

    for subdir in dirs:

        img_path = os.path.join(self.face_dir, subdir)

        for fn in os.listdir(img_path):

            path = img_path + '/' + fn

            tag = index

            imgs.append(cv2.imread(path, 0))

            tags.append(int(tag))

        index += 1

(imgs, tags) = [np.array(item) for item in [imgs, tags]]

self.model.train(imgs, tags)

self.model.save('fisher_trained_data.xml')

print "Training completed successfully"

return

if __name__ == '__main__':

    trainer = TrainFisherFaces()

    trainer.capture_training_images()

    if trainer.are_enough_faces():

        trainer.fisher_train_data()
```

```
print "Type in next user to train, or press Recognize"
```

```
else:
```

```
print "Type in next user to train, or press Recognize"
```

### **Anexo 5: Reconocimiento de Rostro utilizando FisherFaces y EigenFaces.**

```
import face_recog_eigen as fre
```

```
import face_recog_fisher as frf
```

```
import numpy as np
```

```
import cv2
```

```
import Image
```

```
recognizerf = frf.RecogFisherFaces()
```

```
recognizerf.load_trained_data()
```

```
recognizer = fre.RecogEigenFaces()
```

```
recognizer.load_trained_data()
```

```
video_capture = cv2.VideoCapture(0)
```

```
while True:
```

```
    ret, frame = video_capture.read()
```

```
    inImg = np.array(frame)
```

```
    outImg, face_names, conf = recognizer.process_image(inImg)
```

```
    outImg1, face_names1, conf1 = recognizerf.process_image(inImg)
```

```
try:  
  
    print str(face_names[0])+'+str(conf[0])+'  
  
    print str(face_names1[0])+'+str(conf1[0])+'  
  
except:  
  
    pass  
  
cv2.imshow('Video', outImg)  
  
if cv2.waitKey(1) & 0xFF == ord('q'):  
  
    video_capture.release()  
  
    cv2.destroyAllWindows()
```

## **Anexo 6: Sistema lógica difusa para optimizar de la iluminación interior**

FuzzyRule.h

FuzzyComposition.h

Fuzzy.h

FuzzyRuleConsequent.h

FuzzyOutput.h

FuzzyInput.h

FuzzyIO.h

FuzzySet.h

FuzzyRuleAntecedent.h

Ahora para recrear las funciones de pertenencia descritas anteriormente y realizadas en Matlab se utilizó la siguiente sintaxis.

**Entrada: Luz Ambiente**

```
FuzzyInput* luz = new FuzzyInput(1);

FuzzySet* Muy_Bajo = new FuzzySet(-0.25, 0, 0, 0.25);

luz->addFuzzySet(Muy_Bajo);

FuzzySet* Bajo = new FuzzySet(0, 0.25, 0.25, 0.5);

luz->addFuzzySet(Bajo);

FuzzySet* Medio = new FuzzySet(0.25, 0.5, 0.5, 0.75);

luz->addFuzzySet(Medio);

FuzzySet* Alto = new FuzzySet(0.5, 0.75, 0.75, 1);

luz->addFuzzySet(Alto);

FuzzySet* Muy_Alto = new FuzzySet(0.75, 1, 1, 1.25);

luz->addFuzzySet(Muy_Alto);

fuzzy->addFuzzyInput(luz);
```

**Salida: Iluminación Interior**

```
FuzzyOutput* salida = new FuzzyOutput(1);

FuzzySet* Muy_Bajo1 = new FuzzySet(-0.25, 0, 0, 0.25);

salida->addFuzzySet(Muy_Bajo1);

FuzzySet* Bajo1 = new FuzzySet(0, 0.25, 0.25, 0.5);
```

```
salida->addFuzzySet(Bajo1);
```

```
FuzzySet* Medio1 = new FuzzySet(0.25, 0.5, 0.5, 0.75);
```

```
salida->addFuzzySet(Medio1);
```

```
FuzzySet* Alto1 = new FuzzySet(0.5, 0.75, 0.75, 1);
```

```
salida->addFuzzySet(Alto1);
```

```
FuzzySet* Muy_Alto1 = new FuzzySet(0.75, 1, 1, 1.25);
```

```
salida->addFuzzySet(Muy_Alto1);
```

```
fuzzy->addFuzzyOutput(salida);
```

Ahora para crear las reglas a las que se rige nuestro sistema difuso se usara el siguiente algoritmo:

```
FuzzyRuleAntecedent*ifLuzMuyBajo=new FuzzyRuleAntecedent();
```

```
ifLuzMuyBajo->joinSingle(Muy_Bajo);
```

```
FuzzyRuleConsequent*thenSalidaMuyAlto=new FuzzyRuleConsequent();
```

```
thenSalidaMuyAlto->addOutput(Muy_Alto1);
```

```
FuzzyRule* fuzzyRule01 = new FuzzyRule(1, ifLuzMuyBajo, thenSalidaMuyAlto);
```

```
fuzzy->addFuzzyRule(fuzzyRule01);
```