

**Universidad Católica de Santa María**  
**Facultad de Ciencias e Ingenierías Físicas y**  
**Formales**  
**Escuela Profesional de Ingeniería de Sistemas**



**PROTOTIPO DE UN SISTEMA DE VIGILANCIA Y DETECCIÓN DE  
FUGA DE GAS, UTILIZANDO RASPBERRY PI Y ARDUINO PARA  
MEJORAR LA SEGURIDAD EN VIVIENDAS UNIFAMILIARES**

Tesis presentada por la bachiller:

**Pérez Palomino, Ximena Isabel**

para optar el título profesional de:

**Ingeniera de Sistemas**

Especialidad en Sistemas de Información

Asesora:

**Mg. Rosas Paredes, Karina**

Arequipa – Perú

2019

FACULTAD DE CIENCIAS E INGENIERIAS FISICAS Y FORMALES  
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

INFORME DICTAMEN DE BORRADOR TESIS

VISTO

El Borrador de TESIS titulado:

"Sistema de vigilancia y detección de gas, con Raspberry Pi y Arduino para mejorar la seguridad en viviendas unifamiliares"

Presentado por (el) (la) (los) Bachiller (es):

Ximena Isabel Pérez Palomino

Nuestro dictamen es:

Procedente

OBSERVACIONES: Se cambia título por:

"Prototipo de un sistema de vigilancia y detección de fuga de gas, utilizando Raspberry Pi y Arduino para mejorar la seguridad en viviendas unifamiliares"

Arequipa, 4 de setiembre de 2019



1160



1221

## PRESENTACIÓN

Sr. Director de la Escuela Profesional de Ingeniería de Sistemas

Sres. Miembros del Jurado Dictaminador

De conformidad con las disposiciones del reglamento de Grados y Títulos de la Escuela Profesional de Ingeniería de Sistemas, pongo a vuestra consideración el presente trabajo de investigación titulado: **“SISTEMA DE VIGILANCIA Y DETECCIÓN DE FUGA DE GAS, UTILIZANDO RASPBERRY PI Y ARDUINO PARA MEJORAR LA SEGURIDAD EN VIVIENDAS UNIFAMILIARES”**, el mismo que de ser aprobado me permitirá optar el título Profesional de Ingeniería de Sistemas.

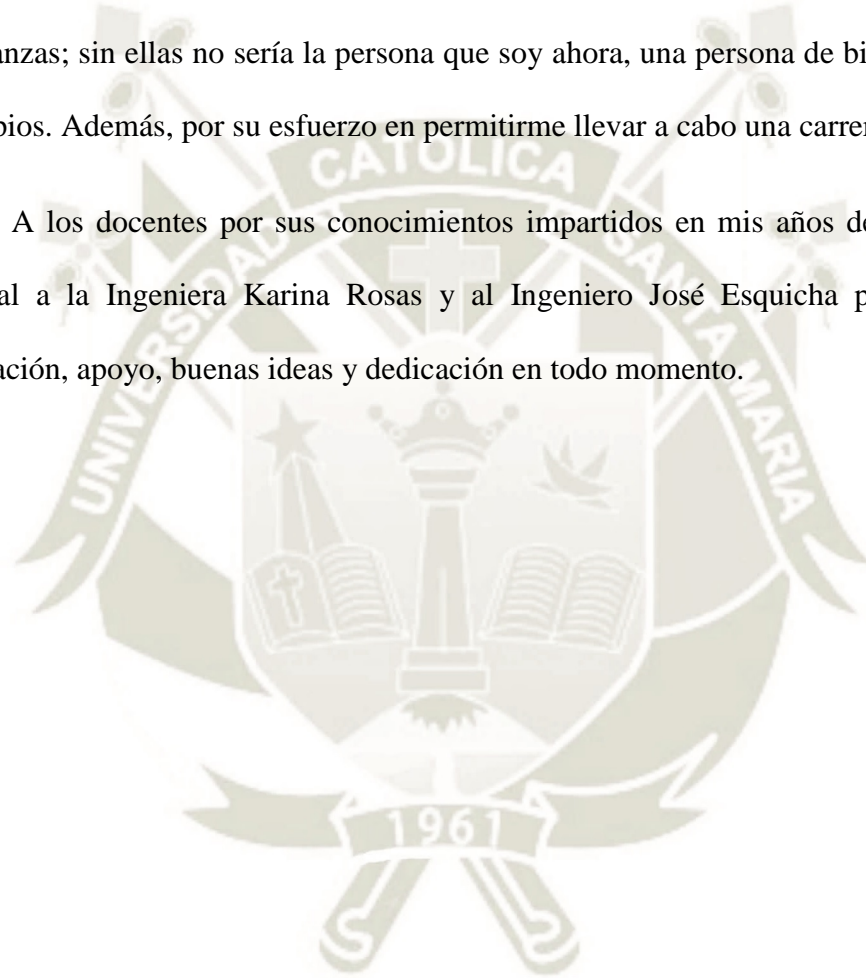
Pérez Palomino, Ximena Isabel

## AGRADECIMIENTOS

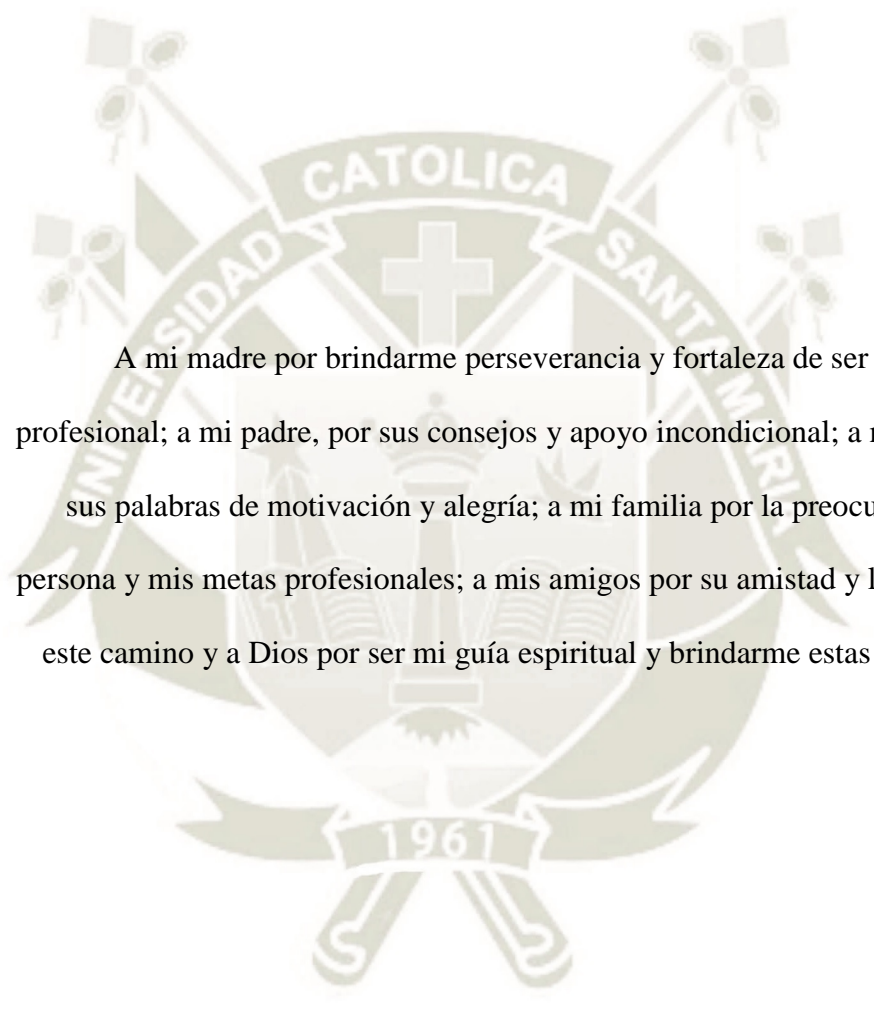
Primeramente agradecer a Dios por guiarme en este camino y darme la fuerza en llegar a la meta.

A mis padres por su apoyo y comprensión incondicional, por sus consejos y sus enseñanzas; sin ellas no sería la persona que soy ahora, una persona de bien, con valores y principios. Además, por su esfuerzo en permitirme llevar a cabo una carrera profesional.

A los docentes por sus conocimientos impartidos en mis años de estudiante y en especial a la Ingeniera Karina Rosas y al Ingeniero José Esquicha por sus consejos, motivación, apoyo, buenas ideas y dedicación en todo momento.



## DEDICATORIA



A mi madre por brindarme perseverancia y fortaleza de ser mejor persona y profesional; a mi padre, por sus consejos y apoyo incondicional; a mi hermano, por sus palabras de motivación y alegría; a mi familia por la preocupación hacia mi persona y mis metas profesionales; a mis amigos por su amistad y la ayuda dada en este camino y a Dios por ser mi guía espiritual y brindarme estas oportunidades y bendiciones.

## RESUMEN

La criminalidad ha aumentado en la ciudad de Arequipa, sobretodo el robo e ingreso indebido a la propiedad, por ello varias empresas brindan sistemas de seguridad. Algunas empresas de seguridad dan un servicio complejo, difícil de manipular y a un precio elevado, lo cual lo vuelve un servicio no apto para el público en general. Además, cuando el usuario quiere incorporar otros equipos al sistema, el precio también aumenta.

En este trabajo se desarrolla un prototipo de un sistema de vigilancia y detección de fuga de gas, el cual considera que la seguridad del hogar no tan solo rige en alertar sobre la presencia de un intruso sino en generar un ambiente de confianza. Por ello, el sistema advierte a los miembros del hogar si hay gases tóxicos que podrían generar un incendio o explosión. Las alertas por ingreso de un intruso o presencia de gas se dan mediante mensajes de texto, y al detectar a un intruso, se toma una imagen, la cual se envía al correo electrónico del propietario. El sistema cuenta con un panel de control, un sistema de reconocimiento de voz y una aplicación móvil.

El sistema de vigilancia y detección se implementa en un microprocesador Raspberry Pi 3 Modelo B, que es el encargado de administrar sensores, actuadores y periféricos del sistema. Además, utiliza un microcontrolador Arduino UNO para conectar el módulo GSM/GPR que es el encargado del envío de mensajes de texto de alerta.

Las herramientas utilizadas en este proyecto son: librerías Python, la plataforma de desarrollo Firebase para la conexión del sistema y la aplicación móvil, y la plataforma Wit.ai para el reconocimiento de comandos de voz, para brindar seguridad al hogar.

*Palabras clave: seguridad, Raspberry Pi, Arduino UNO, módulo GSM/GPR, comandos de voz.*

## ABSTRACT

Crime has increased in the city of Arequipa, especially theft and improper entry to property, so several companies provide security systems. Some security companies provide a complex service, difficult to handle and at a high price, which makes it a service not suitable for the general public. In addition, when the user wants to incorporate other equipment into the system, the price also increases.

In this work, a prototype of security system and gas detection was developed, which considers that the security of the home not only governs to alert on the presence of an intruder but also to generate an atmosphere of trust. Therefore, the system warns household members if there are toxic gases that could cause a fire or explosion. Alerts for entry of an intruder or presence of gas are given by text messages, and when detecting an intruder, an image is taken, which is sent by email to the owner. The system has a control panel, a voice recognition system and a mobile application.

The gas detection and security system is implemented in a Raspberry Pi 3 Model B microprocessor, which is responsible for managing sensors, actuators and peripherals of the system. In addition, it uses an Arduino UNO microcontroller to connect the GSM / GPR module that is responsible for sending alert text messages.

The tools used in this project are: Python libraries, the Firebase development platform for the connection of the system and the mobile application, and the Wit.ai platform for voice command recognition, to provide home security.

*Keyword: security, Raspberry Pi, Arduino UNO, GSM/GPR module, voice commands.*

## INTRODUCCIÓN

Actualmente, en la ciudad de Arequipa la delincuencia ha aumentado, según el secretario técnico del Consejo Regional de Seguridad Ciudadana (Coresec) en 2016 eran 16 mil 664 delitos comparado con 2007 con 9 mil 188 delitos (Vilca, 2017). Arequipa se encuentra entre las cuatro regiones con percepción de inseguridad ciudadana más alta, según el Instituto Nacional de Estadística e Informática (INEI) (2019) en su boletín semestral de septiembre 2018 a febrero 2019, el 93,2% de la población cree que será víctima de algún acto delictivo. En el caso de robo de viviendas, las cifras fueron el 3,6% de viviendas fueran afectadas por robo y un 9,0% de viviendas por intento de robo. El ingreso indebido y robo a las viviendas va en incremento y pueden llegar a cifras alarmantes.

Además, existen otros riesgos que pueden afectar el hogar, como son los incendios. Según el Instituto Nacional de Defensa Civil (INDECI) (2017) las principales causas de incendio son: fuga de gas y cortocircuitos. Los cuales se inician de manera instantánea o gradual, y se podrían evitar si se toman medidas de prevención. Los incendios pueden provocar pérdidas de vidas humanas y/o daños materiales.

Varias empresas brindan sistemas de seguridad para el hogar, estos sistemas tienen costos elevados y si se desea agregar otras características y equipos el precio aumenta aún más. Algunos sistemas de seguridad son difíciles de usar o manejar para el consumidor, y algunos son configurados con sonidos muy parecidos que podrían generar confusión.

En este proyecto se desarrolla un prototipo de un sistema de vigilancia y detección de fuga de gas utilizando Raspberry Pi, Arduino UNO, un micrófono, una cámara web y sensores para mejorar la seguridad en el hogar. Con la idea de desarrollar un sistema de vigilancia y detección de fuga de gas con varias características y de menor costo. El sistema consiste principalmente, en alerta al propietario del ingreso no autorizado de un intruso por



medio de mensajes de texto y se toma una foto que se envía por correo electrónico. En caso de detectar gas se alertara por medio de un mensaje de texto enviado por el sistema y mensajes en la pantalla del hogar. El sistema cuenta con un panel de control, un sistema de reconocimiento de voz y una aplicación móvil que permite gestionar y controlar el sistema.

El proyecto está constituido por dos capítulos en los cuales se trataran los siguientes puntos:

En el capítulo 1, Descripción del proyecto; define los objetivos, los alcances y limitaciones; en fundamentos teóricos, se resalta proyectos similares a este y algunas ideas planteadas en la sección de Estado del Arte (Antecedentes del proyecto) y se trabajan conceptos importantes y necesarios para entender el proyecto (bases teóricas del proyecto). Técnicas y herramientas que se utilizaron para el desarrollo del proyecto y mencionar aspectos relevantes del desarrollo.

En el capítulo 2, Documentación técnica del proyecto, se desarrolla el plan de proyecto informático que explica las actividades y el tiempo en que se realizaron y el estudio de viabilidad del proyecto para ver el costo y equipos que se utilizaran. Se realiza un análisis de comparación de los componentes a utilizar; Especificación del diseño del sistema de vigilancia y detección de fuga de gas, de la aplicación móvil y de la maqueta; y Documentación técnica del sistema de vigilancia y detección de fuga de gas y aplicación móvil. Pruebas y Resultados, para comprobar el funcionamiento del prototipo por medio de pruebas de ejecución como: funcionalidad, seguridad, usabilidad, velocidad y rendimiento. Desarrollo de manuales de usuario para el sistema de vigilancia y detección de fuga de gas y la aplicación móvil.

Finalmente, se definen las conclusiones y recomendaciones de la tesis.

## ÍNDICE DE CONTENIDOS

RESUMEN.....	vi
ABSTRACT.....	vii
INTRODUCCIÓN .....	viii
INDICE DE FIGURAS.....	xiii
INDICE DE TABLAS .....	xvi
CAPÍTULO 1: DESCRIPCIÓN DEL PROYECTO.....	1
1.1.    Objetivos.....	1
1.1.1.    Objetivo General .....	1
1.1.2.    Objetivos Específicos.....	1
1.2.    Alcances y Limitaciones .....	2
1.2.1.    Alcances .....	2
1.2.2.    Limitaciones.....	2
1.3.    Fundamentos teóricos.....	2
1.3.1.    Estado del Arte (Antecedentes del proyecto).....	2
1.3.2.    Bases teóricas del proyecto .....	11
1.3.2.1.    Automatización .....	11
1.3.2.2.    Internet de las Cosas (IOT) .....	11
1.3.2.3.    Domótica .....	12
1.3.2.4.    Sensores.....	13
1.3.2.4.1.    Sensor de movimiento PIR HC-SR501 .....	14
1.3.2.4.2.    Sensor de temperatura y humedad Dht11.....	15
1.3.2.4.3.    Sensor de gas MQ4 .....	16
1.3.2.5.    Actuadores.....	17
1.3.2.5.1.    Diodo emisor de luz (Led).....	17
1.3.2.5.2.    Módulo Relé (Foco) .....	18
1.3.2.5.3.    Micro Servo 9g SG90.....	19
1.3.2.6.    Periféricos.....	20
1.3.2.6.1.    Teclado matricial 4x4 (KeyPad).....	20
1.3.2.6.2.    Pantalla LCD 1602 (16x2) .....	21
1.3.2.6.3.    Piezo buzzer (zumbador).....	21
1.3.2.6.4.    Micrófono.....	22
1.3.2.6.5.    Cámara USB (Webcam).....	23
1.3.2.7.    Arduino UNO.....	24
1.3.2.8.    Raspberry Pi 3 Modelo B.....	25
1.3.4.2.1.    Características de los GPIO.....	27

1.3.2.9.	Módulo Shield GSM/GPRS .....	28
1.3.2.10.	Firebase .....	30
1.3.2.11.	Software de reconocimiento de voz .....	31
1.3.2.12.	Cliente de correo electrónico MuTT .....	32
1.3.2.13.	Sistema Electrónico.....	33
1.4.	Técnicas y Herramientas .....	33
1.4.1.	Comparación de Metodología Tradicional vs Ágil .....	33
1.4.1.1.	Metodología de programación extrema (XP) .....	34
1.4.1.1.1.	Fase de planeamiento .....	35
1.4.1.1.2.	Fase de diseño .....	35
1.4.1.1.3.	Fase de codificación .....	36
1.4.1.1.4.	Fase de pruebas .....	36
1.4.1.1.5.	Fase de implementación .....	37
1.4.2.	Herramientas .....	37
1.5.	Aspectos Relevantes del Desarrollo .....	39
CAPITULO 2: DOCUMENTACIÓN TÉCNICA .....		44
2.1.	Plan del Proyecto Informático.....	44
2.1.1.	Planificación temporal del proyecto.....	44
2.1.2.	Estudio de Viabilidad del Proyecto .....	48
2.1.2.1.	Descripción de Productos y Servicios .....	48
2.1.2.2.	Consideraciones tecnologías .....	49
2.1.2.3.	Características del producto .....	49
2.1.2.4.	Estrategias de marketing .....	50
2.1.2.4.1.	Alta disponibilidad .....	50
2.1.2.4.2.	Seguridad.....	51
2.1.2.4.3.	Escalabilidad .....	51
2.1.2.4.4.	Análisis de datos.....	51
2.1.2.4.5.	Costo reducido.....	52
2.1.2.5.	Organización y recursos humanos.....	52
2.1.2.6.	Proyecciones Financieras .....	52
2.2.	Especificación de requisitos de software (Elicitación y Análisis).....	53
2.2.1.	Especificación de requisitos .....	53
2.2.2.	Análisis.....	54
2.2.2.1.	Comparación microprocesadores Raspberry Pi.....	54
2.2.2.2.	Comparación microcontroladores Arduino .....	56
2.2.2.3.	Comparación sensores .....	57
2.2.2.4.	Comparación software de reconocimiento de voz.....	60

2.2.2.5.	Comparación gestor de base de datos.....	61
2.3.	Especificación de diseño .....	62
2.3.1.	Diseño del sistema de vigilancia y detección de fuga de gas .....	62
2.3.2.	Diseño de la aplicación móvil .....	73
2.3.3.	Diseño de la maqueta .....	74
2.4.	Documentación técnica de programación .....	77
2.4.1.	Documentación del sistema de vigilancia y detección de fuga de gas .....	77
2.4.1.1.	Documentación del sistema en Raspberry Pi 3 Modelo B .....	77
2.4.1.2.	Documentación del sistema en Arduino Uno R3 .....	84
2.4.2.	Documentación aplicación móvil.....	86
2.5.	Pruebas de Ejecución .....	92
2.5.1.	Pruebas de funcionalidad .....	92
2.5.2.	Pruebas de seguridad.....	103
2.5.3.	Pruebas de usabilidad .....	104
2.5.4.	Pruebas de velocidad y rendimiento.....	106
2.6.	Manuales de usuario.....	113
CONCLUSIONES .....		114
RECOMENDACIONES .....		116
REFERENCIAS .....		118
ANEXOS.....		126
Anexo A. Glosario de Términos .....		126
Anexo B. ENCUESTA 1: Sistema de vigilancia y detección de fuga de gas .....		131
Anexo C. ENCUESTA 2: Satisfacción del sistema de vigilancia y detección de fuga de gas.....		139
Anexo D. Fichas de Validación del sistema de vigilancia y detección de fuga de gas para mejorar la seguridad en el hogar.....		143
Anexo E. Manual de Usuario: sistema de vigilancia y detección de fuga de gas y Manual de usuario: aplicación móvil. ....		145

## INDICE DE FIGURAS

Figura 1. Diagrama de bloques: Estructura de un sensor .....	13
Figura 2. Sensor PIR HC-SR501.....	14
Figura 3. Sensor de temperatura y humedad Dht11 .....	16
Figura 4. Sensor de gas MQ4 .....	17
Figura 5. Representación de un Led.....	17
Figura 6. Módulo Relé .....	18
Figura 7. Servomotor 9g SG90 .....	19
Figura 8. Teclado Matricial 4X4 .....	20
Figura 9. Zumbador.....	22
Figura 10. Adaptador de Sonido USB.....	23
Figura 11. Placa Arduino UNO.....	25
Figura 12. Microprocesador Raspberry Pi 3 Modelo B .....	27
Figura 13. Módulo GPRS/GSM.....	30
Figura 14. Sistema Electrónico .....	33
Figura 15. Diagrama Metodología de Programación Extrema.....	35
Figura 16. Modelo de una red neuronal recurrente para reconocimiento de voz .....	39
Figura 17. Ingreso de entidades a Wit.ai .....	40
Figura 18. Entidades en Wit.ai .....	40
Figura 19. Base de datos en Firebase .....	41
Figura 20. Administrador de usuarios de Firebase .....	42
Figura 21. Dashboard usuarios activos .....	43
Figura 22. Dashboard interacción de usuarios .....	43
Figura 23. Diagrama de Gantt de los procesos del proyecto.....	44
Figura 24. Diagrama de Gantt fase de análisis.....	45
Figura 25. Diagrama de Gantt fase de diseño .....	45
Figura 26. Diagrama de Gantt fase de implementación .....	46
Figura 27. Diagrama de Gantt fase de implementación II.....	47
Figura 28. Diagrama de Gantt fase de pruebas .....	47
Figura 29. Diagrama de Gantt fase instalación/evaluación.....	47
Figura 30. Diagrama de bloques del sistema de vigilancia y detección de fuga de gas .....	63
Figura 31. Diagrama de flujo sistema de seguridad .....	67
Figura 32. Diagrama de flujo sistema de detección de fuga de gas .....	69
Figura 33. Diagrama de flujo ingreso de contraseña.....	70
Figura 34. Diagrama de flujo sistema de reconocimiento de voz .....	72
Figura 35. Diagrama de secuencia aplicación móvil.....	74
Figura 36. Representación de la maqueta.....	76
Figura 37. Módulos necesarios para programación en Raspberry Pi .....	77
Figura 38. Credenciales y declaración variables Firebase .....	78
Figura 39. Declaración de puertos GPIO Raspberry Pi.....	78
Figura 40. Funciones teclado Matricial 4X4 .....	79
Figura 41. Conexión con Arduino y función enviar mensaje.....	80
Figura 42. Función contraseña_alarma .....	80
Figura 43. Función activar_alarma.....	81
Figura 44. Función escuchar (reconocimiento de voz) .....	81
Figura 45. Código en grabar.sh (bash) .....	82
Figura 46. Código de acceso Wit.ai .....	82
Figura 47. Función principal Main.....	83
Figura 48. Opción de ingreso por teclado .....	83
Figura 49. Método setup Arduino .....	84

Figura 50. Método loop Arduino.....	85
Figura 51. Método iniciar (enviar SMS) .....	85
Figura 52. Dependencias para conectar con Firebase .....	86
Figura 53. Obtener una instancia de FirebaseAuth .....	87
Figura 54. Método de autenticación del usuario .....	87
Figura 55. Método para crear un nuevo usuario.....	88
Figura 56. Método para encender o apagar el sistema .....	88
Figura 57. Método para visualizar el estado del sistema de vigilancia y detección.....	89
Figura 58. Método para cerrar sesión.....	89
Figura 59. Función del botón “ok” para cambiar la contraseña .....	90
Figura 60. Método para visualizar temperatura y humedad.....	90
Figura 61. Variables necesarias para elegir la palabra clave.....	91
Figura 62. Función del botón generar .....	91
Figura 63. Función del botón “ok” para guardar palabra clave en base de datos.....	91
Figura 64. Método para verificar formato de correo electrónico .....	92
Figura 65. Función del botón ok para guardar correo electrónico .....	92
Figura 66. Ingreso de contraseña.....	93
Figura 67. Mensaje de texto "Intruso detectado" .....	94
Figura 68. Mensaje de correo electrónico con la imagen de un intruso .....	94
Figura 69. Alarma activada por la aplicación móvil .....	95
Figura 70. Mensaje de Auxilio.....	95
Figura 71. Alarma de presencia de gas activada por la aplicación móvil .....	96
Figura 72. Prueba T2 con encendedor.....	96
Figura 73. Mensaje de texto "Presencia de gas".....	97
Figura 74. Opciones sistema de reconocimiento de voz .....	98
Figura 75. Comando de voz "encender luz" .....	98
Figura 76. Prototipo del sistema de vigilancia y detección de fuga de gas .....	99
Figura 77. Panel de Firebase para administrar cuentas .....	104
Figura 78. Correo electrónico con el link para cambiar contraseña .....	104
Figura 79. Uso del CPU sin ejecutar ningún programa.....	107
Figura 80. Uso de CPU en el pico más alto .....	107
Figura 81. Uso de CPU ejecutando el programa.....	108
Figura 82. Uso de memoria sin ejecutar ningún programa .....	109
Figura 83. Uso de memoria RAM en el pico más alto de uso de CPU .....	109
Figura 84. Mayor uso de memoria RAM en el transcurso de uso del programa.....	109
Figura 85. Uso de memoria RAM al ejecutar el programa .....	110
Figura 86. Temperatura de Raspberry Pi 3 Modelo B .....	111
Figura B 1. Resultado de la primera pregunta Encuesta 1 .....	131
Figura B 2. Resultado de la segunda pregunta Encuesta 1.....	132
Figura B 3. Resultado de la tercera pregunta Encuesta 1 .....	133
Figura B 4. Resultado de la cuarta pregunta Encuesta 1 .....	133
Figura B 5. Resultado de la quinta pregunta Encuesta 1 .....	134
Figura B 6. Resultado de la sexta pregunta Encuesta 1.....	134
Figura B 7. Resultado de la séptima pregunta Encuesta 1 .....	135
Figura B 8. Resultados de la octava pregunta Encuesta 1 .....	136
Figura B 9. Resultado de la novena pregunta Encuesta 1 .....	136
Figura B 10. Resultado de la décima pregunta Encuesta 1 .....	137
Figura B 11. Resultados de la décimo primera pregunta Encuesta 1 .....	137
Figura B 12. Resultados de la décimo segunda pregunta Encuesta 1 .....	138

Figura C 1. Resultados de la primera pregunta Encuesta 2 .....	139
Figura C 2. Resultados de la segunda pregunta Encuesta 2 .....	140
Figura C 3. Resultados de la tercera pregunta Encuesta 2 .....	140
Figura C 4. Resultados de la cuarta pregunta Encuesta 2 .....	141
Figura C 5. Resultados de la quinta pregunta Encuesta 2 .....	141
Figura C 6. Resultados de la sexta pregunta Encuesta 2 .....	142
Figura E 1. Opciones de Ingreso por teclado .....	146
Figura E 2. Pantalla "Ingreso de opción" .....	146
Figura E 3. Pantalla "Opción: Activar Alarma" .....	146
Figura E 4. Pantalla "Ingrese contraseña" .....	146
Figura E 5. Pantalla "Ingreso correcto" .....	146
Figura E 6. Pantalla "Alarma Activara en:" .....	146
Figura E 7. Pantalla "Alarma Activada" .....	146
Figura E 8. Pantalla "Opción: Desactivar Alarma" .....	146
Figura E 9. Pantalla "Correcto Alarma desactivada" .....	146
Figura E 10. Pantalla "Opción Activar alarma incendio" .....	146
Figura E 11. Pantalla "Alarma incendio activada" .....	146
Figura E 12. Pantalla "Opción: Desactivar alarma incendio" .....	146
Figura E 13. Pantalla: "Alarma incendio desactivada" .....	146
Figura E 14. Pantalla "Opción: tem/hume" .....	146
Figura E 15. Pantalla "temperatura y humedad" .....	146
Figura E 16. Pantalla "Opción: Lo escucho" .....	146
Figura E 17. Pantalla "Dijo: activar alarma" .....	146
Figura E 18. Pantalla "Pronuncie palabra clave" .....	146
Figura E 19. Pantalla "Intento 1" .....	146
Figura E 20. Mensaje de Auxilio .....	146
Figura E 21. Pantalla "Incorrecto" .....	146
Figura E 22. Pantalla "Intentos superados" .....	146
Figura E 23. Pantalla "Sistema bloqueado" .....	146
Figura E 24. Pantalla "Han Transcurrido" .....	146
Figura E 25. Pantalla "Alarma Activada" .....	146
Figura E 26. Mensaje de texto "Intruso detectado" .....	146
Figura E 27. Correo electrónico con la imagen del intruso .....	146
Figura E 28. Pantalla "Puerta se cerrara en: " .....	146
Figura E 29. Carpeta para guardar las imágenes .....	146
Figura E 30. Pantalla "Presencia de gas" .....	146
Figura E 31. Pantalla "Posible presencia de fuego y humo" .....	146
Figura E 32. Mensaje de texto Presencia de gas detectada .....	146
Figura E 33. Pantalla "No es una opción" .....	146
Figura E 34. Aplicación móvil "Inicio sesión" .....	146
Figura E 35. Ingreso de nuevo usuario .....	146
Figura E 36. Aplicación móvil: Menú principal .....	146
Figura E 37. Aplicación Móvil: Ingreso nueva contraseña .....	146
Figura E 38. Aplicación móvil: Visualizar temperatura y humedad .....	146
Figura E 39. Aplicación Móvil: cambiar palabra clave .....	146
Figura E 40. Pantalla Presencia de Gas .....	146

## INDICE DE TABLAS

Tabla 1. Elementos básicos de la domótica.....	12
Tabla 2. Tipos de Interfaz de Control .....	13
Tabla 3. Características del sensor PIR.....	14
Tabla 4. Características del sensor Dht11 .....	15
Tabla 5. Características de un micro servomotor .....	19
Tabla 6. Memorias del Microcontrolador ATmega 328P .....	25
Tabla 7. Especificaciones Raspberry Pi 3 Modelo B .....	26
Tabla 8. Especificaciones del Módulo Shield GSM/GPRS .....	29
Tabla 9. Componentes del Módulo GSM/GPRS.....	29
Tabla 10. Software de reconocimiento de voz .....	32
Tabla 11. Comparación de Metodologías Tradicionales vs Ágil .....	34
Tabla 12. Componentes del sistema .....	50
Tabla 13. Comparación de Modelos Raspberry Pi.....	55
Tabla 14. Comparación entre modelos Arduino .....	57
Tabla 15. Comparación entre sensores MQ-x.....	58
Tabla 16. Comparación entre sensores de temperatura y humedad DHT .....	58
Tabla 17. Comparación entre sensores de movimiento.....	59
Tabla 18. Comparación software de reconocimiento de voz .....	60
Tabla 19. Pines del microprocesador Raspberry Pi.....	64
Tabla 20. Pines del microcontrolador Arduino .....	65
Tabla 21. Matriz de trazabilidad de pruebas .....	99
Tabla 22. Matriz de pruebas sistema de vigilancia .....	101
Tabla 23. Matriz de pruebas sistema de detección de fuga de gas .....	102
Tabla 24. Matriz de pruebas reconocimiento de voz.....	102



## CAPÍTULO 1: DESCRIPCIÓN DEL PROYECTO

### 1.1. Objetivos

#### 1.1.1. Objetivo General

Implementar un prototipo de un sistema de vigilancia y detección de fuga de gas, con reconocimiento de voz que envía alertas por medio de mensaje de texto e imagen por correo electrónico para mejorar la seguridad en viviendas unifamiliares, utilizando Raspberry Pi y Arduino.

#### 1.1.2. Objetivos Específicos

1. Desarrollar un prototipo de un sistema de vigilancia y detección de fuga de gas que brinde seguridad a viviendas unifamiliares con tecnología Open Hardware y Open Source Software, mediante sensores, actuadores y periféricos (cámara, micrófono, etc.).
2. Identificar la herramienta que permite realizar reconocimiento de voz por medio de un micrófono.
3. Diseñar una aplicación móvil que permita realizar acciones en el sistema de vigilancia y detección de fuga de gas, desde un lugar diferente de donde se ha implementado el prototipo.
4. Desarrollar un prototipo funcional a pequeña escala en el cual se pueda demostrar el funcionamiento del sistema propuesto.
5. Verificar que el prototipo se ejecute de manera correcta con seguridad, velocidad y rendimiento adecuados.

## **1.2. Alcances y Limitaciones**

### **1.2.1. Alcances**

- Realizar un sistema de vigilancia y detección de fuga de gas con Open Hardware y Open Source Software.

### **1.2.2. Limitaciones**

- Para el desarrollo del proyecto la autora realizará un prototipo a escala para realizar las pruebas respectivas.
- Para el prototipo, se considera construcciones de un solo piso con una puerta principal.
- La aplicación móvil servirá para dispositivos con sistema operativo Android.
- El sistema requiere de una constante conexión a Internet para enviar imágenes al correo electrónico.
- El sistema requiere de un chip de red telefónica con saldo mínimo de cinco soles para enviar mensajes de texto.

## **1.3. Fundamentos teóricos**

### **1.3.1. Estado del Arte (Antecedentes del proyecto)**

En la actualidad, se ha dado un creciente desarrollo en proyectos relacionados con domótica, automatización del hogar e Internet de las Cosas, para implementarlos han surgido microprocesadores y microcontroladores que son opciones rentables y fáciles de adquirir. Por ello, se propone un sistema de vigilancia y detección de fuga de gas con reconocimiento de voz, que por medio de una aplicación móvil se puede gestionar y controlar remotamente, desarrollado con el microprocesador Raspberry Pi y el microcontrolador Arduino.

En los últimos años, el reconocimiento de voz ha tenido un gran crecimiento, la mayoría de empresas de tecnología como Amazon, Google, Microsoft, etc., lo utilizan para

desarrollar asistentes personales o brindar otros servicios. Al consumidor le parece atractivo e intuitivo, lo que genera un crecimiento acelerado para realizar más productos con esta característica. Los siguientes proyectos utilizan reconocimiento de voz con Raspberry Pi.

Google cuenta con su propio servicio de reconocimiento de voz, Google Cloud Speech API, que lo utiliza en sus productos. Gyulyustan y Enkov (2017), implementaron un sistema de reconocimiento de voz con un microprocesador Raspberry Pi, el micrófono de una cámara web USB y la API de Google. El sistema funciona cuando detecta voz, se conecta a la API de Google, la cual convierte el audio en texto, el texto se imprime y según la línea de comando ejecuta una acción. La API de Google no es gratuita cobra por el número de palabras que convierte, por ello el sistema a desarrollar utiliza otro sistema de reconocimiento de voz.

Muchos sistemas de asistente personal no son amigables con las personas que tienen alguna discapacidad, limitación física o no pueden operar una computadora. Khong Fu y Samson (2017), realizaron un sistema de asistente personal (PAL) para este conjunto de personas, con Raspberry Pi 2, un micrófono USB, parlantes y Jasper (software de reconocimiento de voz). El asistente personal por medio de comandos de voz realiza las siguientes funciones: búsqueda de información en internet, buscar correos electrónicos no leídos, programar eventos en el calendario, etc. El sistema presenta algunos inconvenientes por utilizar Jasper como lentitud en el reconocimiento de voz por los acentos o el ruido, los autores plantean mejorarlo utilizando un STT (Speech to text) como Wit.ai que presenta mayor precisión. Por medio de este paper surge la idea de utilizar Wit.ai como sistema de reconocimiento de voz, por ser bueno en precisión y ser gratuito.

Younis, Ijaz, Randhawa y Ijaz (2017) observaron que no había sistemas de domótica, especializados en pacientes, discapacitados, personas mayores o que viven solas. Por ello,

realizaron un proyecto que se compone de una sección base y de una sección remota, que se comunican por una conexión inalámbrica ZigBee. La sección base se compone de un micrófono, una Raspberry Pi 2 Modelo B y el software de reconocimiento de voz Sphinx-4. En la sección base se ingresa un comando de voz que se procesa en Raspberry Pi por el software de reconocimiento de voz y genera una secuencia binaria (comando). La sección remota tiene un microcontrolador de bajo nivel que recibe por medio de ZigBee (inalámbrico) el comando generado en la sección base, para realizar alguna acción en los dispositivos. Es un sistema de automatización de bajo costo y bajo consumo, fácil de instalar y con un sistema de reconocimiento de voz eficiente. Se diferencia con el sistema a desarrollar en la utilización Sphinx-4 como sistema de reconocimiento de voz, debido a que solo funciona con el idioma inglés y se quiere un sistema de reconocimiento de voz en español.

Desarrollar aplicaciones móviles o webs permiten tener control y monitoreo remoto de sistemas de automatización del hogar convirtiéndolos en sistemas más accesibles. Daramas, Pattarakitsophon, Eiumtrakul, Tantidham y Tamkittikhun (2016), implementaron un sistema de automatización doméstica para la detección de intrusos (HIVE) con una base de datos en la nube (Firebase). Funciona cuando algún sensor (PIR, interruptor magnético o sensor de peso) se activa y envían una señal, generando una reacción en los actuadores como un sonido (zumbador) o prendiendo una luz (led) en advertencia. Los sensores y actuadores son inalámbricos utilizan módulos XBee que se conectan por medio del protocolo ZigBee. El sistema cuenta con una aplicación en Android, que por medio de Firebase se conecta con el sistema, lo que permite que el usuario final pueda monitorear, configurar y recibir notificaciones de forma remota. Este proyecto es parecido al sistema a desarrollar por la base de datos que utiliza, pero se diferencian por los sensores y actuadores que utiliza y el funcionamiento del sistema de seguridad.

Combinar Automatización del hogar con Internet de las Cosas genera sistemas inteligentes que interactúan con el entorno sin la interacción humana. Mary, Harsha y Reddy (2017), realizaron un sistema sostenible con el medio ambiente que utiliza Internet de las Cosas para controlar los dispositivos del hogar a través de un dispositivo con conexión a Internet. El sistema cuenta con una aplicación en Android que a través de la base de datos en la nube (Firebase) envía comandos o revisa los datos de los sensores que están conectados al microprocesador Raspberry Pi o al microcontrolador Arduino. El sistema cuenta con los sensores de: Sensores magnéticos y de movimiento (PIR) (seguridad); Sensor de temperatura y humedad (DHT11) para detectar detectan el nivel de humedad y temperatura de una habitación; y Sensor de humedad de suelo para medir los niveles de humedad (auto riego). Es un sistema que aprovecha las ventajas del medio ambiente y es de bajo costo. Es un sistema complejo con varias funciones, el sistema a desarrollar solo se centra en realizar un sistema automatizado de seguridad.

Li, Yen, Lin, Tung y Huang (2018), diseñaron un sistema de Internet de las Cosas llamado JustIoT, que cuenta con una base de datos Firebase en tiempo real, una aplicación web, un microcontrolador Arduino (hardware y software) y un servidor inteligente MQTT. JustIoT, es un sistema de casa inteligente en el cual se puede monitorear y administrar desde una página web la humedad, temperatura y brillo de habitaciones del hogar. La página web la desarrollaron con la tecnología front-end Angular, la conectaron con una base de datos en la nube (Firebase), que permite la conexión con un servidor MQTT que envía comandos y obtiene datos de los sensores que están conectados a Arduino. El sistema se diseñó con fines educativos para promover el desarrollo de aplicaciones IoT en estudiantes. Este sistema se diferencia del sistema a desarrollar en la en la utilización de un servidor MQTT, el sistema propuesto utiliza Raspberry Pi el cual obtiene y envía datos de los sensores.

La automatización del hogar se encuentra en constante crecimiento los propietarios desean seguridad, ahorro de energía y un lugar con varias comodidades. Hadwan y Reddy (2016), diseñaron un sistema integrado de bajo costo para el seguimiento de luces, temperatura, alarma y otros electrodomésticos. El sistema se compone de varios sensores, cuando alguno cambia de señal sea el de temperatura o PIR, la data es enviada por Raspberry PI al servidor, se guarda los datos en una base de datos y se muestran en la página web. Arduino permite tener un control de los dispositivos electrónicos. La principal ventaja de implementar este sistema con Raspberry PI es la integración de la red con sensores inalámbricos, el servidor web y el servidor de base de datos. Es un sistema de automatización del hogar que incluye varios puntos, el sistema a desarrollar se centra en la seguridad, no tal solo en el ingreso de intrusos sino en detectar gas como medida de prevención y cuenta con una aplicación móvil para el control y monitoreo.

Otro proyecto de automatización del hogar, que se define por elegir Raspberry PI por su alto rendimiento y Arduino por sus pines de entradas y salidas, es un sistema de control y permiso desarrollado por Kumar y Pati (2016). El sistema se compone de: una Raspberry Pi 2, que da funciones para el sistema de permiso de visitantes; y de un Arduino Mega 2560, que procesa la señal de entrada y salida hacia los dispositivos del hogar. El sistema de control permite monitorear y supervisar por medio de una pantalla táctil e interruptores eléctricos varios electrodomésticos (la luz, la ventilación, la refrigeración, etc.). El sistema de permiso, concede la interacción segura entre el visitante y el propietario por medio de mensajes, en caso no haya nadie en el hogar, el visitante puede dejar un mensaje de voz que se guarda con una imagen con hora y fecha. El sistema usa Raspberry Pi y Arduino al igual que el sistema a desarrollar, se diferencian en que el sistema advierte de la presencia de un visitante y no cuando ingresa un intruso al hogar.

El siguiente proyecto, también implementa la automatización del hogar, se diferencia del anterior porque solo usa Raspberry Pi 3 y tiene como prioridad a las personas mayores o discapacitados. Vaidya et al. (2017), realizaron un sistema de control de electrodomésticos, monitoreo y seguridad. El sistema permite controlar la luminosidad de la luz, la ventilación, y la iluminación (cortinas) por medio de Raspberry Pi, según el comando ingresado por la aplicación móvil ya sea de forma táctil o de voz. Cuenta con un sistema de reconocimiento de rostros que por medio de una cámara USB permite el ingreso de algunas personas, se puede acceder a la cámara por la aplicación móvil. Por medio de la aplicación y la web se visualiza en tiempo real el estado de los aparatos electrónicos del hogar lo que permite tener un mejor control de la energía y los datos se guardan periódicamente para tener un historial de uso. Es un proyecto que incluye varios factores de automatización del hogar, el sistema a desarrollar solo trabaja seguridad y detección, no tendrá reconocimiento de rostros.

Desarrollos de automatización del hogar con Internet de las cosas (IoT) está en constante crecimiento y una de las razones es el incremento de la tecnología. Raghunath y Anil Kumar (2017), realizaron un modelo de hogar inteligente con sistemas de seguridad y automatización, utilizando un microprocesador Raspberry Pi 2. El sistema de seguridad permite al propietario visualizar por la página web videos en vivo y darle movimiento a la cámara para captar imágenes de diferentes vistas. También, puede observar una imagen del visitante, confirmar su identidad y dar o denegar el ingreso de forma remota. Por otro lado, el sistema de automatización controla luces, ventilación, temperatura y humedad. El microprocesador se encarga de realizar el procesamiento y enviar señales de respuesta como encender o apagar dispositivos según los valores que captan los sensores. En varios aspectos se diferencia del proyecto a realizar, uno de ellos es visualizar videos en vivo, ya que el sistema a realizar solo toma imágenes al detectar una presencia.

Otra opción es utilizar solo el microcontrolador Arduino, en proyectos de automatización del hogar, ya que es flexible y económico, tiene un buen número de entradas digitales y analógicas, y el software es de código abierto. Soliman, Alahmadi, Maash y Elhabib (2017), elaboraron un proyecto que se compone de una computadora personal (PC) como servidor que aloja una plataforma de gestión LabView y un microcontrolador Arduino. Con el sistema se puede monitorear, controlar y acceder, a los dispositivos, ya sea automáticamente en respuesta de alguna señal de algún sensor relacionado o por medio del usuario. El sistema gestiona tres electrodomésticos: gestión de la temperatura, ahorro de energía lumínica y cámara de seguridad con un sensor ultrasónico. Todos los sistemas pueden ser administrados por el usuario manualmente a través de la plataforma LabView. Se comprobó que es un sistema simple, rentable y flexible. Se diferencia del proyecto a desarrollar por utilizar una plataforma que no se utiliza en el sistema y usar solo un microcontrolador Arduino.

Otro proyecto que se desarrolla con Arduino, automatización del hogar e Internet de las cosas (IoT), es realizado Wadhvani, Singh, Singh y Dwivedi (2018). El sistema que proponen cuenta con varios sensores como: un sensor magnético para la seguridad de la puerta, que al ser manipulado suena el zumbador; un sensor de llama que detecta fuego en la casa y envía una alerta por medio de un zumbador; y un sensor LDR, detectan la presencia o nivel de luz, para tener un control automático de las luces, apagar o prender según el ambiente. Cuenta con una plataforma THINKSPEAK que se encuentra en la nube, en la cual el usuario visualiza el estado de los dispositivos que cambian según los valores que captan los sensores. Es una propuesta menos costosa con la idea de implementar IoT en el área doméstica, que permite el control de los dispositivos por medio de teléfonos inteligentes. Este proyecto se diferencia del proyecto a desarrollar por utilizar solo un microcontrolador



Arduino y una plataforma especial, en cambio el proyecto a desarrollar necesita de un microprocesador Raspberry PI.

Kumar, Natraj y Jacob (2017) realizaron un sistema de seguridad para el hogar que se identifica por tener reconocimiento fácil y de movimiento. Se compone de un sensor PIR, una cámara Pi, un microcontrolador Raspberry Pi y OpenCV para el procesamiento de imágenes. Cuando ocurre un movimiento que el captado por el sensor PIR, la cámara toma una imagen, se le advierte al propietario por medio de un mensaje de texto con un link donde se visualiza la imagen. El software de reconocimiento Haar Cascade detecta las caras de una imagen y las compara con un archivo de datos XML. Cuando detecta una cara y se encuentra en el archivo, se le permite el ingreso, sino se alerta al propietario y se le niega el ingreso. Es un sistema que está diseñado para detectar rostros con precisión y en tiempo real. Se diferencia con el sistema a desarrollar en el uso de un sistema de reconocimiento fácil y de movimiento.

El tema de seguridad es importante en varios ambientes y no tan solo en el hogar, como la industria, la automovilística, la salud, los bancos y finanzas, etc. Un sistema de alarma de una bóveda de un Banco dependía del factor humano, para mejorar la seguridad y automatizarlo, Ray y Uddin (2016), llevaron a cabo un sistema que costa de cuatro sensores (sonido, movimiento, laser y gas), una cámara IP y un módulo GSM. Los sistemas funcionan de manera independiente pero se integran en un sistema automatizado. Cuando alguno de los sensores detecta algo, la alarma se activa y enviara un mensaje de texto de alerta. El sistema es gestionado por el microcontrolador Arduino y el módulo GSM para el envío de mensajes de texto. El sistema es monitoreado desde cualquier parte del mundo por medio de la cámara IP. Es una solución segura, automatizada y de supervisión remota. Al sistema se le puede agregar otros niveles de seguridad, diferentes sensores y otros equipos de seguridad.

Es un sistema dedicado a bancos, aun así algunos sensores y el módulo GSM es utilizado en el sistema a desarrollar.

Otra manera de aplicar seguridad al hogar es advertirle al propietario de un posible incendio o una fuga de gas. Dhruvajyoti y Debashis (2016), realizaron un sistema de alarma con sensores de fuego automatizado para el hogar y la industria de bajo costo. El sistema se compone de varios sensores: de fuego, temperatura, de gas y de humo. Con la finalidad de generar un ambiente lo más seguro posible. El microcontrolador con la ayuda de los sensores está en constante verificación, en caso de emergencia se envía un mensaje a través del módulo GSM, que está conectado con el usuario, la policía y los bomberos. Es un sistema desarrollado con tecnologías emergentes que tienen gran acogida, porque son fáciles de usar y de bajo costo. El proyecto es dedicado a la seguridad pero en el ámbito de fuga de gas y posibles incendios, es parecido al sistema a desarrollar en advertir al propietario si hay una fuga de gas.

En Filipinas los incendios son un problema recurrente y el mayor número se produce en los hogares. Los bomberos no saben con exactitud en qué lugar del hogar ocurre el incendio, y reciben varias llamadas que son falsas. Por ello, Abaya et al. (2016), realizaron una tecnología integrada con sensores que reconocen humo y temperatura causada por incendios, con un microcontrolador Arduino, módulo GSM y envió inalámbrico de datos. El sistema envía los datos a la estación de bomberos a un sistema de mapas interactivo que permite determinar en tiempo real donde ocurre un incidente y la ubicación exacta. El dueño será alertado por medio de un mensaje de texto. Implementar el mecanismo permitirá a los bomberos atender el incendio de forma oportuna e inmediata, evitando lesiones, muertes y destrucción de la propiedad. Además, permite que las estaciones de bomberos tengan una comunicación efectiva y se mantengan informados de las ocurrencias. El sistema le avisa a

la estación de bomberos de posibles incendios, lo cual lo hace diferente al sistema a desarrollar que solo le advierte al propietario.

Anteriormente, se describieron varios proyectos que utilizan tecnologías emergentes, automatización, Internet de las Cosas, microprocesadores y microcontroladores con el ideal de brindar comodidad, seguridad, monitoreo, control y servicios a los usuarios. Ninguno es completamente igual al sistema a desarrollar, tienen algunos aspectos similares como el uso de Raspberry Pi, el uso de Arduino, el uso de sensores, el uso de una base de datos en la nube (Firebase), el uso de Wit.ai como sistema de reconocimiento de voz y el módulo GSM que sirve para controlar el envío de mensajes de texto y avisar al propietario cuando ocurre una alerta. Cabe resaltar que la estructura del proyecto y la lógica que se utiliza es propia. Además, en el Perú la inseguridad es alta y el costo de un sistema de alarma es elevado, un gran número de personas no pueden acceder a esos servicios. Para resolver este problema se plantea un sistema de vigilancia y detección de fuga de bajo costo y bajo consumo, que sea fácil de utilizar e implementar, y que tenga un sistema de reconocimiento de voz eficiente.

### **1.3.2. Bases teóricas del proyecto**

#### **1.3.2.1. Automatización**

La automatización es la aplicación de máquinas o procedimientos automáticos que requieren la mínima intervención del ser humano en la realización de un proceso, debe de ser un proceso flexible que se ajuste a los cambios (Martínez, 2017). Los Tipos de automatización son: fija, programable y flexible. Las ventajas de la automatización son: menos tareas manuales, eliminar trabajos repetitivos, mejorar la calidad y productividad.

#### **1.3.2.2. Internet de las Cosas (IOT)**

Desde que surgió Internet se han conectado millones de dispositivos. Los avances en tecnología convirtieron la red en un sistema inalámbrico que puede conectar todo, desde computadoras de escritorio hasta teléfonos inteligentes y tablets. IoT es un concepto que se

refiere a la interconexión digital de los objetos cotidianos (desde maquinaria industrial hasta lavadoras) con Internet, convirtiéndolos en objetos inteligentes, a través de herramientas que les permitan ampliar las funciones que son capaces de realizar (Muñoz, 2017).

El beneficio que trae consigo IoT es la posibilidad de automatizar las tareas que requieren mano de obra o llevan demasiado tiempo realizar y simplificar los procesos relacionados (Kranz, 2017).

### 1.3.2.3. Domótica

La domótica es la instalación e integración de varias redes y dispositivos electrónicos en el hogar, que permite la automatización de actividades cotidianas y el control local o remoto de la vivienda o del edificio inteligente (Huidobro y Millán, 2010). Este concepto surge con el fin de: ahorrar energía, mejorar la seguridad y dar el mayor confort a los dueños. Balbi (2016) considera que la domótica brinda una respuesta a todo, centralizando, automatizando y optimizando el funcionamiento integral de las viviendas.

Un sistema de domótica consta de 3 elementos básico: sensores, módulo de control y actuadores (Tabla 1):

Tabla 1. Elementos básicos de la domótica  
*Elementos básicos de la domótica*

Elementos Básicos de la Domótica		
Sensores	Módulo de Control	Actuadores
Elementos que recogen datos del entorno (temperatura, humedad, etc.)	Gestiona sensores y actuadores.	Dispositivos que transmiten órdenes a través de la red, a instancias del usuario o a través de programación horaria (Maestre, 2015).
	Centro del sistema	
Envían los datos al módulo de control.	Cuando un sensor capta un valor, se envía al módulo de control para realice alguna acción en consecuencia.	

Fuente: (Maestre, 2015)

Para que el usuario interactúe con el módulo de control puede ser a través de diferentes interfaces de control (Tabla 2).

Tabla 2.  
*Tipos de Interfaz de Control*

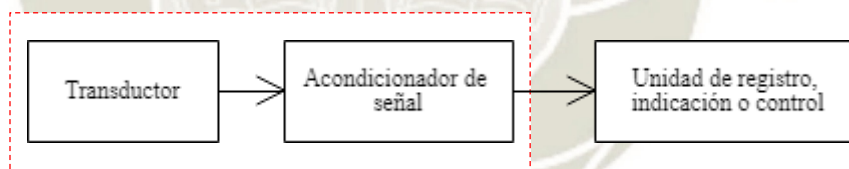
Tipos de Interfaz de Control			
Interfaz local	Interfaz de voz	Interfaz de mensajes móviles	Interfaz Web
Se compone de teclado y pantalla en el mismo lugar donde se lleva a cabo el sistema domótica	Por medio de un dispositivo con micrófono se puede programar o saber el estado del edificio.	Utiliza la red GSM, se puede alertar enviando un SMS.	Por medio de un servidor web se puede configurar o conocer el estado del edificio.

Fuente: Elaboración propia

Mediante la domótica se puede simplificar algunas acciones que se realizan en el hogar para proveer mayor comodidad y confort. Para la seguridad del hogar: detectores de movimiento, sistemas de video vigilancia, conexión automatiza con centrales de alarma, etc., contribuyen a hacer que el domicilio sea más seguro, especialmente cuando estamos ausentes (Moro, 2011).

#### 1.3.2.4.Sensores

Dispositivo de entrada que provee una salida manipulable de la variable física medida (Corona, Abarca y Mares, 2014).



Sensor

Figura 1. Diagrama de bloques: Estructura de un sensor

Fuente: Elaboración propia

Se cuenta con diferentes tipos de sensores, dependiendo de la señal enviada pueden ser digitales o análogos. Sensor digital: señales que adoptan dos posibles valores el máximo 1 y el mínimo 0, ejemplo: interruptores y pulsadores. Sensor análogo: envían una señal dentro de un rango de valores y permiten realizar diferentes acciones en función del valor enviado, ejemplo: regulador de luminosidad (Martín, 2011).

#### 1.3.2.4.1. Sensor de movimiento PIR HC-SR501

Son sensores que detectan movimiento, son pequeños, asequibles, fáciles de utilizar, no utilizan mucha electricidad y no se deterioran.

Torrente (2013), indica que el sensor PIR (“Pyroelectric passive InfraRed sensors”) detecta la radiación de algún cuerpo caliente como un humano o un animal que pasa por el rango de detección. Para detectar el movimiento uno de los sensores capta el cuerpo que atraviesa la zona de detección, poniéndose en estado positivo; al momento en que el cuerpo abandona el área ocurre lo contrario, el segundo sensor lo capta y cambia a un estado negativo. Detecta los cambios de temperatura.

El sensor utiliza dos potenciómetros y un jumper que permite modificar sus parámetros y adaptarlo a las necesidades: sensibilidad de detección, tiempo de activación y respuesta ante detecciones repetitivas (Punto Flotante, 2017). Cuenta con varias características que se especifican en la Tabla 3.

Tabla 3. Características del sensor PIR

Características del sensor PIR

Característica	Valor
Voltaje de alimentación	De 5 a 12 VDC
Consumo promedio	<1 miliamperio
Rango de distancia	De 3 a 7 metros ajustable
Delay	Ajustable de 3s a 5 min
Angulo de detección	Cono de 110°
Temperatura de operación	-15° a +70°C

Fuente: (Punto Flotante, 2017)



Figura 2. Sensor PIR HC-SR501

Fuente: (Punto Flotante, 2017)

1.3.2.4.2. Sensor de temperatura y humedad Dht11

Sensor digital básico y de bajo costo, se compone de dos elementos uno de medición de humedad de tipo resistivo y otro de medición de temperatura NTC, ambos se encuentran estrictamente calibrados generando precisión en la obtención de los datos (Sunrom, 2012). En la tabla 4 se muestra las características del sensor Dht11.

Tabla 4.  
*Características del sensor Dht11*

Características Sensor Dht11				
Parámetros	Condiciones	Mínimo	Típico	Máximo
Humedad				
Resolución		1%RH	1%RH 8 Bit	1%RH
Repetitividad			+ - 1%RH	
Exactitud	25 °C		+ - 4%RG	
	0-50 °C			+ -5%RH
Intercambiabilidad	Totalmente intercambiable			
Rango de medición	0 °C	30%RH		90%RH
	25 °C	20%RH		90%RH
	50 °C	20%RH		80%RH
Tiempo de respuesta (segundos)	1/e (63%) 25°C, 1 m/s Air	6 S	10 S	15 S
Temperatura				
Resolución		1 °C	1 °C	1 °C
		8 Bit	8 Bit	8 Bit
Repetitividad			+ - 1 °C	
Exactitud		+ - 1 °C		+ - 2°C
Rango de medición		0 °C		50 °C
Tiempo de respuesta (segundos)	1/e (63%)	6 S		30 S

Fuente: (Adafruit, 2018)

La memoria del sensor, tiene programas que contienen los coeficientes de calibración, el sensor los utiliza como procesos de detección de señal interna. Son de tamaño pequeño, utilizan un bajo consumo de energía y la transmisión de la señal puede llegar hasta 20 metros (D-Robotics UK ,2010).

Es fácil conectarse a este sensor con un microcontrolador debido a que se compone de 4 pines (Adafruit, 2018): un pin de VCC: se conecta a 3.3 – 5V de

potencia; uno de salida de datos; un pin no conectado (No se usa) y uno de conexión a tierra (GND).



Figura 3. Sensor de temperatura y humedad Dht11  
Fuente: (D-Robotics UK, 2010)

#### 1.3.2.4.3. Sensor de gas MQ4

El sensor MQ4 detecta emisiones de gases combustibles como: metano, propano y butano. En un rango de sensibilidad de: 300 a 10000 ppm. Es de bajo costo, de larga vida de funcionamiento y se usa en diferentes aplicaciones como: detector de fugas de gas doméstico, detector de gas combustible industrial y detector de gas portátil (Hanwei Electronics, 2016).

Los sensores de la serie MQ utilizan un pequeño calentador interior con un sensor electroquímico para medir diferentes tipos de combinaciones de gases (Vaduva, 2018). Para detectar el gas combustible, el sensor de gas MQ4 se compone de un material sensible SnO<sub>2</sub> (Dióxido de estaño). SnO<sub>2</sub> tiene menor conductividad en el aire limpio, cuando la concentración de gas aumenta, la conductividad del sensor es más alta. Cuando hay un cambio de conductividad determina que detecto concentración de gas (Hanwei Electronics, 2016). El sensor MQ4 funciona con 5V estables y 150 mA o hasta 250 mA. Para mediciones estables por lo menos debe de calentar por un minuto.

El modelo cuenta con cuatro pines: un pin VCC 5V, un pin conexión a tierra (GND), un pin DO (salida digital) TTL digital 1 y 0 (0.1 y 5V) y un pin AO (salida analógica) 0.1 – 0.3 V (limpio), mayor concentración alrededor de 4V.





Figura 4. Sensor de gas MQ4  
Fuente: (E-Radionica, 2016)

### 1.3.2.5. Actuadores

Dispositivo que recibe una orden de algún regulador o controlador para activar un elemento final, es capaz de transformar energía hidráulica, neumática o eléctrica (Torrente, 2013). Ejemplos: un motor (energía mecánica), una bombilla (energía lumínica), un altavoz (energía acústica), etc.

#### 1.3.2.5.1. Diodo emisor de luz (Led)

Dispositivo semiconductor (diodo) con dos cables, un ánodo (cable largo) y un cátodo (cable corto que corresponde a la conexión a tierra (GND)). Para que el led emita luz, tiene que pasar una mayor cantidad de voltaje por el ánodo que por el cátodo (Margolis, 2011).

La intensidad para el funcionamiento óptimo de un LED es de 15 mA, puede variar según el color y material del cual está hecho, se debe utilizar una resistencia en serie al LED para limitar la intensidad de corriente que lo atraviesa y evitar que se dañe (Torrente, 2013). Es bueno consultar las especificaciones del fabricante para conocer intensidades y tensiones máximas que puede soportar. Con los LED se puede desarrollar varias aplicaciones como: iluminación interior y exterior, paneles LED al aire libre, reemplazar lámparas incandescentes, etc.

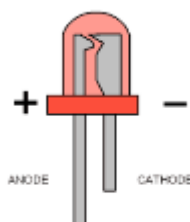


Figura 5. Representación de un Led  
Fuente: (Build Electronic Circuits, 2014)

### 1.3.2.5.2. Módulo Relé (Foco)

El relé es un dispositivo electromagnético que funciona como un interruptor controlado por un circuito eléctrico (Valero, 2016). Se compone de una bobina que al circular una pequeña corriente (3.3V, 5V o 12V) genera un campo magnético, que a su vez hace mover una placa metálica abriendo o cerrando un circuito eléctrico independiente por el cual circula un voltaje superior (220V) (del Valle, 2016). Permitiendo controlar cualquier aparato electrónico como: motores, bombas, climatizadores, iluminación o cualquier dispositivo electrónico (Llamas, 2016).

El módulo relé, se compone de dos circuitos: un circuito primario de baja tensión, se conecta con un microcontrolador que le da la señal de encendido y apagado, y el circuito secundario que es el interruptor encargado de encender o apagar la carga (Llamas, 2016).

Para conectar un módulo relé con el microcontrolador Arduino se utilizan tres pines: GND conexión a tierra, VCC conexión al voltaje 5V y un pin digital de señal (prendido o apagado) (del Valle, 2016). Para la conexión de la corriente y el dispositivo electrónico, el relé tiene tres salidas que funciona como pares, el del medio con los otros dos. Un par es normalmente cerrado (NC), permanecen conectados si no se le envía una señal a la bobina (circuito cerrado). El otro par es normalmente abierto (NA), cuando se envía una señal a la bobina, el circuito se cierra. Se conecta una salida normalmente abierta a la bombilla y otra a la alimentación (del Valle, 2016).

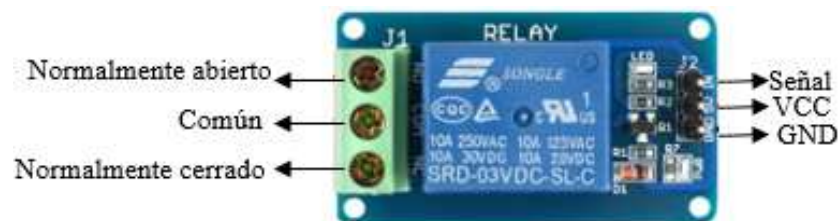


Figura 6. Módulo Relé  
Fuente: Elaboración propia

### 1.3.2.5.3. Micro Servo 9g SG90

El servomotor es un motor que se limitan a la velocidad pero es más fuerte en el torque. El eje no gira libremente, se le debe enviar una señal de control para indicarle la cantidad de grados que tiene que rotar (Torrente, 2013). Se pueden utilizar en juguetes, timones, ascensores, palancas, etc.

El servomotor costa de tres partes: el motor, el tren de engranajes y el circuito de control. Cuando se aplica voltaje al motor, gira el tren de engranajes y el centro del servomotor. El tren de engranajes se compone de un potenciómetro que cuando gira, varia el voltaje y la resistencia. El circuito de control se encarga de comparar los valores que le envían (señales de entrada) con el del potenciómetro, si estos son iguales el motor se detiene y mantiene la posición, si son diferentes hace un cambio (Huang y Runberg, 2017). En la Tabla 5 se especifican las características de un micro servomotor.

Tabla 5. Características de un micro-servomotor  
*Características de un micro servomotor*

Características de Micro servomotor	
Característica	Valor
Peso	9 g
Dimensión	22.2 x 11.8 x 31 mm aprox.
Torque	1.8 kgf cm
Velocidad de operación	0.1 s/60 grados
Tensión de funcionamiento	4.8 V (~5V)
Ancho de banda muerta	10 $\mu$ s
Rango de temperatura	0 ° C – 55 ° C

Fuente: (Huang y Runberg, 2017)



Figura 7. Servomotor 9g SG90  
Fuente: (Huang y Runberg, 2017)

### 1.3.2.6. Periféricos

Unidades o dispositivos de hardware auxiliares o independientes conectados a una unidad de procesamiento, por los cuales se puede comunicar con el exterior (Crespo, 2016).

#### 1.3.2.6.1. Teclado matricial 4x4 (Keypad)

Keypad es un teclado numérico que se compone de botones, pulsadores conectados entre sí. Las teclas se programan a través de un microcontrolador, funcionando de tal manera que cuando alguna es presionada se enviará una señal de tipo binario, generando una reacción establecida en el circuito (Torrente, 2013).

Los teclados usan una combinación de cuatro filas y cuatro columnas para proporcionar estados de botón al microcontrolador. Cada tecla se compone de un botón que está conectado un extremo a una fila y el otro extremo a una columna (Parallax, 2011). Para saber que botón se presiona, el microcontrolador analiza los estados de cada una de las cuatro columnas (pines 1 - 4) ya sea baja (0) o alta (1), luego revisa el estado de las cuatro filas (pines 5 – 8) (Figura 13). Dependiendo del estado de la matriz, se puede decir que botón se presiona.

El teclado matricial tiene un diseño delgado, una interfaz fácil de usar y una excelente relación entre precio y rendimiento. Cuenta con una temperatura de funcionamiento de 0° a 50 ° C. Sus dimensiones son: teclado 2.7 x 3.0 in (6.9 x 7.6 cm) y cable: 0.78 x 3.5 in (2.0 x 8.8 cm).

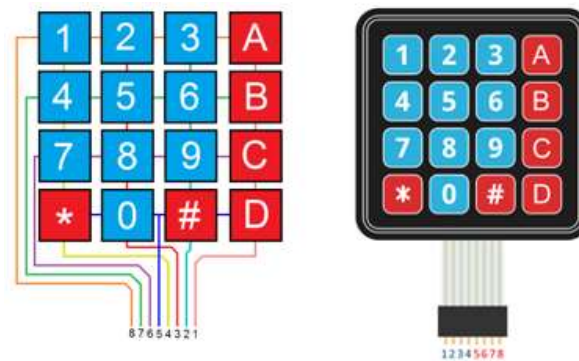


Figura 8. Teclado Matricial 4X4  
Fuente: (Luis Llamas, 2016)

#### 1.3.2.6.2. Pantalla LCD 1602 (16x2)

Pantalla de cristal líquido “Liquid Crystal Displays” (LCD) para mostrar mensajes. El modelo 16x2 es un LCD de caracteres que muestra texto en ASCII. El texto se muestra en dos filas, y tiene un espacio de 16 caracteres (8 bits) por fila. La pantalla LCD es económica, fácil de programar, puede mostrar caracteres especiales o personalizados, animaciones, etc.

La pantalla LCD se compone de un COB (Chip-on-Board) y de un controlador SPL780D1. La temperatura de funcionamiento esta entre  $-10^{\circ}\text{C}$  a  $+60^{\circ}\text{C}$  y la de almacenamiento entre  $-20^{\circ}\text{C}$  a  $+70^{\circ}\text{C}$ . Funciona con un voltaje de alimentación de 5V.

La pantalla LCD funciona por medio de dos registros: comando y datos. El registro de comando almacena una instrucción que se ejecuta en la pantalla LCD, para realizar una tarea predefinida como: iniciar, borrar la pantalla, dar una posición al cursor, etc. Y el registro de datos almacena los datos que se muestran en la pantalla LCD (Kushagra, 2012).

#### 1.3.2.6.3. Piezo buzzer (zumbador)

El zumbador piezoeléctrico es como un pequeño altavoz que genera un tono cuando se le aplica un voltaje (varias frecuencias) a los dos cables, por medio de clics muy rápidos. Se compone de un disco de metal redondo y un cristal (piezoeléctrico). Cuando el cristal se deforma por presencia de voltaje, el disco genera una vibración dentro del cilindro (Huang y Runberg, 2017).

El zumbador se compone de dos patas, una positiva (larga) y otra negativa (corta). La negativa se debe conectar a suelo (GND) y la pata positiva se debe conectar a un pin del microcontrolador (Bosu y Choudhuri, 2017).



Figura 9. Zumbador

Fuente: (Bosu y Choudhuri, 2017)

#### 1.3.2.6.4. Micrófono

Los micrófonos son dispositivos que transforman energía acústica en energía eléctrica. Permiten el registro, almacenamiento, transmisión y procesamiento electrónico de las señales de audio (Creaciones informáticas aplicadas, 2012). Los micrófonos permiten la comunicación entre la persona y la computadora por medio de comandos.

El micrófono se encarga de transformar las vibraciones que presenta la presión acústica sobre su capsula, en energía eléctrica. La presión acústica es producida por el sonido. La energía provocada por las ondas sonoras genera un movimiento ondulatorio en el aire, provocando variaciones en la presión del aire, así es como el micrófono detecta el sonido (variaciones de presión) (Creaciones informáticas aplicadas, 2012).

El micrófono para conectarse con una computadora cuenta con una salida Jack. El conector Jack se caracteriza por tener uno o dos anillos en la punta separados por aros de plástico con aislante negro, el primer aro es para conexiones mono (TS) y el segundo es para estero o balanceadas (TRS) (García, 2016). El microprocesador Raspberry Pi no cuenta con una entrada Jack. Por ello, se utiliza un adaptador de sonido, que convierte una entrada Jack a una salida USB. El adaptador es una tarjeta de sonido, que permite la conexión USB y cuenta con entradas de audio y sonido (micrófono y auriculares). El adaptador de sonido es compatible con varios sistemas operativos como: Windows, Linux, IOS, etc.



Figura 10. Adaptador de Sonido USB  
Fuente: (A y L Systems, 2018)

#### 1.3.2.6.5. Cámara USB (Webcam)

Una Webcam es una cámara que capta imágenes que se pueden visualizar por una página web, aplicación de escritorio o en la pantalla de la computadora a la cual esté conectada por un cable USB o de forma inalámbrica.

Saxena (2016) recomienda una serie de características que se deben tener en cuenta para adquirir una cámara web: Frecuencia de cuadros: la cámara web capta fotogramas, la velocidad ideal es de 30 por segundo; Resolución de video (calidad de grabación): las de baja resolución son de 640 por 480 píxeles, requieren menos ancho de banda y recursos del sistema; la definición es más alta de 720, la calidad puede ser como un DVD pero la conexión es lenta y la resolución de imagen fija: las cámaras web deben tener de capacidad de foto de 10 MP o más, para capturar una imagen que tenga buena calidad de 3072 por 2304 píxeles.

La Webcam que se utiliza en el proyecto es una Logitech C170, tiene las siguientes características: Configuración fácil plug-and-play, tanto en computadoras de escritorio como portátil; Video llamadas de calidad VGA, 640 x 480 píxeles. Cuenta con el software de cámara Logitech Vid™ HD; Fotos de 5 megapíxeles (software mejorado); Cuenta con un micrófono incorporado con tecnología RightSound (reductor de ruido) y grabación de video XGVA (1024 x 768) (Logitech®, 2011).

### 1.3.2.7. Arduino UNO

Arduino es una placa de hardware libre, que se compone de un microcontrolador reprogramable y una serie de pines hembra donde se puede conectar sensores y actuadores; utiliza software libre y multiplataforma que se debe instalar en el ordenador para escribir, verificar y cargar en la memoria del microcontrolador a través de una conexión USB; tiene un lenguaje de programación libre, que implementa elementos parecidos a otros lenguajes de programación (Torrente, 2013).

El microcontrolador se programa usando Arduino Programming Language (basado en Wiring) y el Arduino Development Environment (basado en Processing) (Arduino, 2015). El lenguaje Arduino se basa en los lenguajes de programación C/C++.

Se puede programar el microcontrolador Arduino para leer entradas de sensores (luz en un sensor, presionar un botón, etc.) y convertirlo en una salida (activar un motor, encender un LED, etc.), con el ideal de realizar proyectos que mejoren la forma de vida.

Arduino Uno R3 es la versión que se utiliza en el proyecto, es un modelo USB de la placa Arduino. La placa se compone de un microcontrolador del modelo ATmega328P de la marca Atmel. La placa se compone de 14 pines digitales de entrada/salida y 6 tienen la característica de salida PWM, 6 entradas analógicas, un conector de alimentación, una interface USB, un encabezado ICSP y un botón de reinicio (Arduino, 2014). La placa puede usar energía de una conexión USB o de una fuente de alimentación externa, funciona con un voltaje de 5V, el voltaje de entrada recomendado es de 7 a 12 V y el límite es de 6 a 20 V (Arduino, 2014). El microcontrolador ATmega 328P cuenta con tres memorias que permiten ingresar programas y ejecutarlos (Tabla 6).



Tabla 6.  
*Memorias del Microcontrolador ATmega 328P*

Memorias en el Microcontrolador ATmega328P		
Memoria flash	Memoria SRAM	Memoria EEPROM
Se almacena permanentemente el programa hasta que se dé una reescritura.	Memoria volátil, donde se guardan los datos que se necesitan para crear o manipular el programa en ese instante y correctamente.	Memoria persistente, así se apague el microcontrolador los datos se almacenan para el próximo reinicio.
Capacidad es de 32KB	Capacidad de 2KB	Capacidad de 1KB

Fuente: Elaboración propia

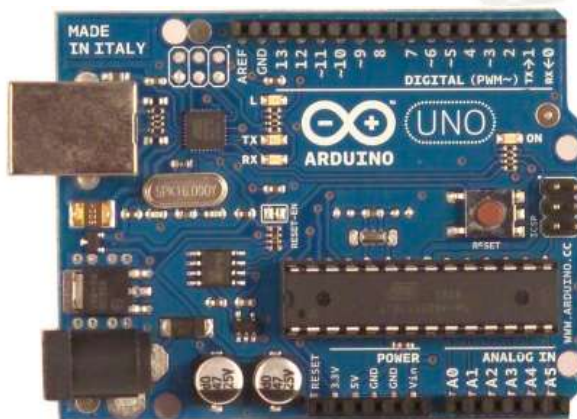


Figura 11. Placa Arduino UNO

Fuente: (Arduino, 2014)

### 1.3.2.8. Raspberry Pi 3 Modelo B

Raspberry Pi es una placa que se compone de un procesador, un chip gráfico y memoria RAM, lo que la hace un miniordenador (ordenador de placa única) (Pascual, 2013). El proyecto de realizar miniordenadores como Raspberry Pi, surge para enseñar informática y brindar una opción menos costosa.

La placa Raspberry Pi se compone de un procesador Broadcom BCM2837 system-on-chip (SoC), una unidad central de procesamiento, una unidad de gráfico, una unidad de audio y hardware de comunicaciones, que se encuentran integrados debajo del chip de memoria 256 MB en el centro de la placa (Upton y Halfacree, 2016). El chip contiene una unidad de procesamiento central ARM (CPU) y una unidad de procesamiento gráfico Videocore 4 (GPU), ambos comparten la memoria (McManus y Cook, 2013). Utiliza ARM por ser una opción de bajo consumo energético. Raspberry Pi puede funcionar con una fuente

de alimentación de 5V 1A y producir un menor calor residual (Upton y Halfacree, 2016).

Para la conectividad cuenta con Wifi 802.11n y Bluetooth 4.1 (de bajo consumo). En la

Tabla 7 se visualiza las especificaciones del microprocesador Raspberry Pi 3 Modelo B.

Tabla 7.  
*Especificaciones Raspberry Pi 3 Modelo B*

Modelo Raspberry Pi 3 Modelo B	
Características	
Procesador	Chip Broadcom BCM2387 ARM Cortex-A53 de cuatro núcleos a 1.2 GHz (64 bits)
GPU	Co-procesador Multimedia Dual Core Video Core IV®. Proporciona Open GL ES 2.0, acelerado por hardware Open VG y decodificación de alto perfil 1080p30 H.264. Capaz de 1Gpixel / s, 1.5Gtexel / s o 24GFLOPs con filtrado de texturas e infraestructura DMA
Memoria RAM	1GB SDRAM 400 MHz
Sistema Operativo	Boots desde la tarjeta Micro SD, ejecutando una versión del sistema operativo Linux o Windows 10 IoT.
Toma de corriente / Voltaje	Toma Micro USB 5V, 2.5 <sup>a</sup>
Conectividad	
LAN inalámbrica 802.11b/g/n (Wi-fi)	Protocolo: WEP, WPA WPA2, algoritmos AES-CCMP (clave máxima longitud de 256 bits). Rango máximo de 100 metros.
Bluetooth 4.1 (Bluetooth Classic y LE) / IEEE 802.15	Algoritmo de cifrado simétrico Advanced Encryption Standard (AES) con clave de 128 bits. Rango máximo de 50 metros.
Conectores (Puertos)	
Puerto Ethernet	Toma Ethernet 10/100 BaseT
USB	4 puertos
Video	HDMI (rev 1.3 y 1.4) RCA compuesto (PAL y NTSC)
Audio	Salida de audio jack de 3,5 mm HDMI Conector USB 4 x USB 2.0
Conectores GPIO	40 pines de 2,54 mm (100 mil): cinta de 2x20 27 pines GPIO, las demás son líneas de suministro +3.3 V, +5 V y GND
Conector de cámara	Interfaz serie de cámara MIPI de 15 pines (CSI-2)
Puerto tarjeta de memoria	Micro SDIO

Fuente: Raspberry Pi Foundation (2016)



Figura 12. Microprocesador Raspberry Pi 3 Modelo B  
Fuente: (Raspberry Pi, 2017)

En la Raspberry Pi se puede programar en Python y Java. Python, es el lenguaje de programación que se utiliza en los proyectos de Raspberry Pi, tiene la ventaja de utilizar los pines GPIO, para conectar el mundo digital con el mundo físico (Abellán, 2018).

#### 1.3.4.2.1. Características de los GPIO

En el proyecto se utilizan varios pines de entrada/salida GPIO (General Purpose Input/Output) de la placa, para conectar sensores, actuadores y periféricos. El Modelo Raspberry Pi 3 Modelo B tiene un encabezado de entrada/salida de 40 pines GPIO, de los cuales 27 son puertos GPIO, UART, I2C, SPI, fuentes de 3.3V y 5V y GND (Raspberry Pi Foundation, 2016). Raspberry Pi 3 Modelo B cuenta con los buses de comunicación: Bus serie UART (Receptor/Trasmisor Asíncrono Universal), proporciona una interfaz en serie de dos cables, un cable lleva la transmisión y el otro la señal de transmisión, sirve para comunicarse con un dispositivo que envía datos a la Raspberry Pi; Bus I2C (inter-circuitos integrados), se compone de dos pines, uno proporciona la señal SDA (Serial Data Line) y el otro la señal SCL (Serial clock), y permite la comunicación entre múltiples circuitos integrados; y Bus SPI (Interfaz de Periférico Serie), es un bus de cuatro hilos con múltiples líneas Chip Select, realiza una comunicación con más de un dispositivo de destino (Upton y Halfacree, 2016).

Se debe de considerar el nivel de tensión y corriente que se va a conectar a la Raspberry Pi para no causarle daño. Solo se puede conectar tensiones compatibles de 3.3 V (no 5V) y pueden sacar hasta 16 mA (Moya, 2016). Los pines GPIO pueden tener dos estados: alto (1- High) se les proporcionan un voltaje positivo de 3.3V y bajo (0-Low) no reciben energía son iguales a tierra o a 0V (Upton y Halfacree, 2016).

### **1.3.2.9. Módulo Shield GSM/GPRS**

GSM (Sistema Global de Comunicaciones Móviles) es un sistema estándar de red telefónica de segunda generación (2G), la comunicación se produce completamente en digital (Vialfa, 2017). Soporta la transmisión de voz, la transferencia de datos, el intercambio de mensajes de texto y servicios de romming. Las redes GSM usan TDMA (Acceso Múltiple por División de tiempo), funciona brindando ranuras de tiempo a los teléfonos para que se comuniquen, alternando la comunicación en intervalos (Matus, 2018). Cuenta con un rendimiento máximo de 9.6 kbps, y una velocidad de transferencia y descarga inferior de 100 Kbps.

GPRS (General Packet Radio Service) es un sistema de transmisión de datos empleando la red de telefonía celular GSM (Cobo, 2016). Permite el acceso a Internet y otros tipos de comunicación, es la versión 2.5G. Proporciona cobertura inalámbrica, una velocidad de transferencia de 56 a 114 kbps y 30 SMS por minuto. Utiliza el protocolo IP para identificar al dispositivo y darle ingreso a Internet y el protocolo WAP para ingresar a varias aplicaciones inalámbricas mediante Internet.

El módulo Shield GSM/GPRS que se usa en el proyecto es desarrollado por Tinysine Electronics. Es una tarjeta compacta de comunicación inalámbrica. Es como un teléfono celular, ya que utiliza un SIM900. Se configura y controla vía UART usando comandos AT. La placa sirve para el servicio de mensajes cortos, llamadas (audio) y servicio GPRS.

El módulo GSM/GPRS se conecta directamente sobre el microcontrolador sin utilizar cables. Debajo de la placa se encuentran patas de aluminio con la misma forma de las entradas del microcontrolador para conectarlo sobre él. Tiene 12 GPIOs, 2 PWMs y un ADC. Es compatible con microcontroladores que sean parecidos a Arduino UNO. Se necesita de una librería GSM de Tinyshop para programar el módulo GSM/GPRS para cualquier función. La librería se compone de varias librerías, una de ellas permite enviar y recibir mensajes. Esta librería solo funciona para este modelo en común. En la Tabla 8 se muestran las especificaciones del Módulo Shield GSM/GPRS y en la Tabla 9 se indican los botones y luces que presenta el módulo.

Tabla 8. Especificaciones del Módulo Shield GSM/GPRS  
*Especificaciones del Módulo Shield GSM/GPRS*

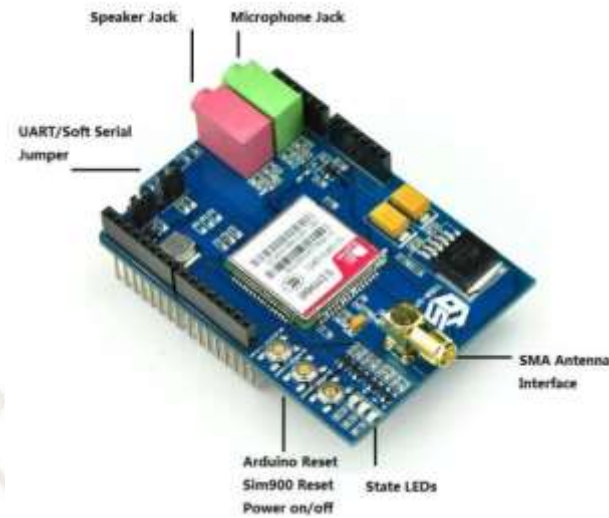
Módulo Shield GSM/GPRS	
Especificaciones	
Banda cuádruple 850/900/1800/1900 MHz: funcionaría en redes GSM en todos los países del mundo.	GPRS multi-slot clase 10/8 Estación móvil GPRS clase B
Cumple con la fase GSM 2/2 +	Clase 4 (2W @ 850 / 900MHz) Clase 1 (1W @ 1800 / 1900MHz)
Control a través de comandos (GSM 07.07, 07.05 y SIMCOM Comandos AT mejorados)	
Selección de puerto serie gratuito	
Compatible con RTC con Super Cap	
Función de encendido / apagado y restablecimiento compatible con la interfaz Arduino	
Consumo de 1.5 mA (susp)	

Fuente: (TinySine, 2014)

Tabla 9. Componentes Módulo GSM/GPRS  
*Componentes Módulo GSM/GPRS*

Componentes Módulo GSM/GPRS	
LED	Botones
Potencia del Shield (P): indica el estado de alimentación	Botón de encendido SIM900: se ejecuta automáticamente cuando se enciende la placa
Potencia SIM900 (S): indica el estado de alimentación del SIM900	Botón restablecer SIM900: se presiona por un segundo para reiniciar conexión.
Estado de red (N) indica el estado de la red	Botón Restablecer Arduino: reinicia la placa Arduino

Fuente: (TinySine, 2014)



*Figura 13.* Módulo GPRS/GSM  
Fuente: (TyniSine, 2014)

#### 1.3.2.10. **Firestore**

Firestore es una plataforma back-end-as-a-service de desarrollo móvil en la nube de Google, disponible para Android, iOS, web (Peña, 2018). Cuenta con una interfaz de programación de aplicaciones para guardar y sincronizar datos en la nube en tiempo real.

Alguno de los servicios que ofrece son: Base de datos en tiempo real (Realtime): permite guardar todos los datos necesarios de la aplicación, es NoSql basada en documentos JSON (árbol JSON) (Talavera, 2017); Autenticación: cuenta con un servicio de inicio de sesión, dando la facilidad de gestionar un usuario y contraseña; Almacenamiento: permite guardar archivos de usuarios; Informes sobre fallos: se utiliza para detectar errores que aparezcan en la aplicación y poder erradicarlos a tiempo; Monitor de rendimiento: esta herramienta se encarga de medir el rendimiento de la aplicación, para detectar posibles problemas de carga y prevenir cualquier problema, y Analítica: cuenta con un panel de control, tiene una solución gratuita para todo tipo de medidas (usuarios, eventos, conexiones, etc.) (Zamora, 2016).

Los lenguajes compatibles con Firestore son: JavaScript (Frontend como Node), iOS (Swift y Objective C), Android (Java y Kotlin), Python y aplicaciones Java y C++ y Unity.

Firestore es una plataforma que permite la conexión de una aplicación móvil Android con un sistema en Raspberry Pi, siendo el intermediario de la comunicación y entregando datos siempre actualizados. En este caso la aplicación móvil funcionaría como un sistema de control.

#### **1.3.2.11. Software de reconocimiento de voz**

Para el 2020, ComScore asegura que la mitad de búsquedas en Internet se realizarán mediante servicios de voz, con mayor incremento en países en desarrollo, por el bajo nivel de alfabetización que presentan, enfatizando que es más fácil hablar un idioma que escribirlo (Sanz, 2018). Las grandes compañías como: Google, Apple (Siri), Microsoft (Cortana) y Amazon (Alexa), utilizan reconocimiento de voz en sus asistentes virtuales.

El reconocimiento de voz es un tipo de inteligencia artificial. Mediante el lenguaje humano se establece una comunicación entre el hombre y el ordenador o dispositivo inteligente, es capaz de descifrar la información que contiene la voz humana (InVoxMedical, 2018). Algunos sistemas de reconocimiento de voz tratan de identificar fonemas o palabras dentro de un diccionario o vocabulario limitado (Ávila, 2016).

Para que el reconocimiento de voz sea bueno se debe de considerar un amplio vocabulario o entrenarlo solo para palabras que se utilizarán en el sistema. Además, se debe de usar una buena entrada como un micrófono. A continuación, se describe software de reconocimiento de voz, son motores Speech-To-Text (STT), que graban un audio y lo transforman en texto, que se pueden utilizar en Raspberry Pi (Tabla 10).

Tabla 10.  
*Software de reconocimiento de voz*

Software de reconocimiento de voz		
Google Cloud Speech Api	Pocket Sphinx	Wit.ai
Por medio de redes neuronales convierte audio en texto.	Es un decodificador de voz de código abierto. (Jasper, 2016).	Es una plataforma de código abierto y gratuito, desarrollado en la nube.
Reconoce 120 idiomas y variantes.	La tasa de reconocimiento no es buena y tiene muchas dependencias.	Realiza procesamiento de lenguaje natural y tiene un alto nivel de reconocimiento.
Características: procesar audio sin ruido, sin perder calidad y cuenta con filtro para el contenido inapropiado.	El software funciona con tres documentos: un modelo acústico que define como se oye la palabra, un modelo de lenguaje y un diccionario de pronunciación.	El software funciona cuando los comandos de voz del usuario son enviados a los servidores de la compañía, quien los procesa y trasforma a formato JSON y devuelve un resultado (Espeso, 2014).
Se puede utilizar con Python o Java, para ambos se cuenta con librerías y código.	Funciona en el sistema operativo Raspbian, el que utiliza Raspberry Pi, con el lenguaje de programación Python.	Disponible para varios sistemas operativos, hardware y lenguajes de programación: Android, iOS, Node.js, Raspberry Pi, Rudy, Python o C.
Requiere de una conexión a Internet. Tiene un nivel gratuito de 0 a 60 min de reconocimiento de voz, después cobra de 0.006 dólares por 15 segundos.	No requiere de conexión a Internet para funcionar.	Como es un servicio en la nube, requiere de una continua conexión a Internet.

Fuente: Elaboración propia

### 1.3.2.12. Cliente de correo electrónico MuTT

Es un correo electrónico que se utiliza desde la terminal de Linux, no tiene entorno gráfico y es basado en texto. Es ligero por la cantidad mínima de recursos que consume. Se puede implementar en cualquier sistema operativo (multiplataforma) y es programado en C.

Es un agente de usuario de correo, por ello, necesita de un agente de transferencia de correo (MTA) para enviar y recibir correos electrónicos. Puede utilizar smtp url para la configuración y soporta la mayoría de formatos de correo electrónico (mbox y Maildir) y



protocolos (POP3, IMAP), que permiten leer el correo electrónico o correos de sistemas remotos (Marble, 2004).

### 1.3.2.13. Sistema Electrónico

Es un conjunto de circuitos que interactúan entre sí para generar un resultado. Se compone de: sensores que captan información del exterior y la transforman en señal eléctrica; circuitos internos procesan la señal eléctrica, que depende de los componentes de hardware y el código (programa) que tenga grabado en el hardware; actuadores son los que transforman la señal eléctrica, actuando en el exterior; y fuente de alimentación brinda la energía necesaria para que funcione el sistema y sus componentes (Torrente, 2013). Estos sistemas pueden ser: un sistema de iluminación en el hogar, un sistema de seguridad, un sistema de riego, etc.

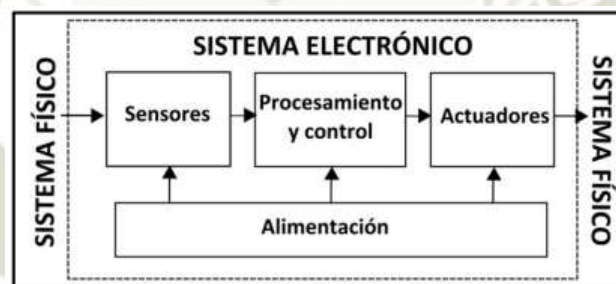


Figura 14. Sistema Electrónico  
Fuente: (Torrente, 2013)

## 1.4. Técnicas y Herramientas

### 1.4.1. Comparación de Metodología Tradicional vs Ágil

Para desarrollar un proyecto se debe elegir una metodología adecuada, en la cual se consideren los tiempos, cambios y la relación con el cliente. Se comparó las metodologías tradicionales (metodología de la cascada) con las metodologías ágiles (metodología de programación extrema XP) (Tabla 11).

Tabla 11.  
Comparación de Metodologías Tradicional vs Ágil

Metodología tradicional	Metodología Ágil
Predictivo	Adaptativos
Proceso rígido	Proceso flexible
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación
Cierta resistencia a los cambios	Preparados para cambios durante el proyecto
Grupos Pequeños <10	Grupos grandes

Fuente: Elaboración propia

Después de observar las características de ambas metodologías, se eligió desarrollar el proyecto con una metodología Ágil, por ser flexible cuando se da un cambio en cualquier etapa del proyecto, presenta una buena comunicación con el cliente, el grupo de desarrollo es pequeño lo cual brinda una mejor coordinación, y cuenta con retroalimentación continua y efectiva.

En metodologías Ágiles las más conocidas son Scrum y programación extrema (XP). Scrum es una metodología ágil que se utiliza cuando el grupo de desarrollo tiene que realizar varios proyectos a la vez, en un tiempo determinado y requieren de una buena coordinación. Al ser el proyecto de un solo desarrollo se optó por realizarlo con la metodología de programación extrema (XP), por ser ágil y flexible frente a cambios que se den en el desarrollo, cuenta con retroalimentación continua y el desarrollo se caracteriza por equipos pequeños de programadores que se centran en un desarrollo, con la idea de implementar y programar lo más rápido posible.

#### **1.4.1.1. Metodología de programación extrema (XP)**

La metodología de programación extrema (XP) se compone de cinco fases: fase de planeamiento, fase de diseño, fase de codificación, fase de pruebas y fase de lanzamiento. Para comenzar se determina los requerimientos necesarios para realizar el proyecto, seguidamente se desarrolla la metodología.

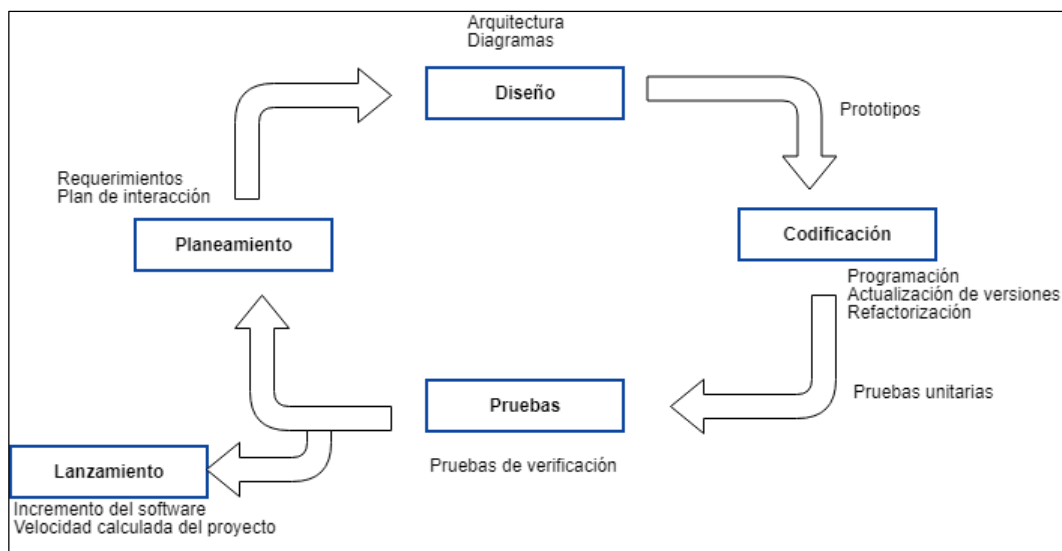


Figura 15. Diagrama Metodología de Programación Extrema  
Fuente: Elaboración propia

#### 1.4.1.1.1. Fase de planeamiento

La primera fase es la de planeamiento, donde se organizan los requerimientos del cliente y el beneficio que obtendrá a partir de estos. Se realiza una primera retroalimentación para definir las necesidades del cliente y si cumplen con los requerimientos solicitados. En esta fase se define el desarrollo de las tareas y el tiempo en que se van a desarrollar.

#### 1.4.1.1.2. Fase de diseño

En la fase de diseño se determina la estructura interna del software. Se descomponen y organizan los elementos que se pueden desarrollar por separado. Se diseñan tres partes del sistema: el sistema de vigilancia y detección de fuga de gas, el diseño de la aplicación móvil y el diseño de una maqueta (prototipo).

El diseño del sistema de vigilancia y detección, tiene como prioridad una interfaz interactiva, opciones fáciles de ingresar y mensajes fáciles de entender y visualizar. El diseño de la aplicación móvil, requiere de una interfaz interactiva y

fácil de usar, permitiendo tener control del sistema. El diseño de la maqueta representa el hogar de un usuario, para demostrar que es un prototipo funcional.

#### 1.4.1.1.3. Fase de codificación

En la fase de codificación se divide en dos partes: la parte de programación del sistema de vigilancia y detección y la parte de programación de la aplicación móvil.

La parte de la programación del sistema de vigilancia y detección se divide en dos partes: la programación del microcontrolador Arduino y la del microprocesador Raspberry Pi. En el microcontrolador Arduino se desarrolla el control del servomotor para la puerta, encendido/apagado del foco y el envío de mensajes de texto para alertar al propietario (módulo GSM). En el microprocesador Raspberry Pi se implementará algoritmos para que funcionen los sensores, actuadores y periféricos. También, se desarrollará el envío de imágenes y la conexión con la aplicación (base de datos Firebase) y el microcontrolador Arduino. Además, de implementar el software de reconocimiento de voz, sockets e hilos.

Para la programación de la aplicación móvil se realiza código para hacer validación de usuarios, ingreso de información y tener un control de la alarma.

La fase de codificación se caracteriza por generar código de calidad y realizar la constante eliminación de código duplicado o ineficiente.

#### 1.4.1.1.4. Fase de pruebas

Durante la codificación del proyecto se dan pruebas continuas de verificación, con el fin de realizar un desarrollo efectivo. Al finalizar el desarrollo del proyecto se realizará una serie de pruebas: pruebas de funcionalidad del sistema, pruebas de

seguridad (claves), pruebas de interactividad con el usuario (usabilidad) y pruebas de velocidad y rendimiento.

#### 1.4.1.1.5. Fase de implementación

Después de realizar todas las pruebas necesarias, el sistema en si se implementara en una maqueta como prototipo, con el fin de demostrar el funcionamiento y verificar que funciona en situaciones que podrían ocurrir en el hogar.

### 1.4.2. Herramientas

A continuación, se describen las herramientas utilizadas en el ciclo de vida de desarrollo de este sistema.

Se realizaron encuestas al público mayor de edad que puede adquirir un sistema de vigilancia y detección. Los requisitos fueron determinados por la desarrolladora, observando que en el mercado existen sistemas de alarma inteligente que tienen un precio elevado, son complejas o dependen de un pago al mes. Según estos puntos se realizaron las preguntas, para saber si al público le interesa adquirir el sistema.

En la fase de diseño se realizaron diagramas de secuencia, diagramas de flujo tanto para el sistema como para la aplicación móvil, para tener una idea de cómo sería el desarrollo y si cumplen con lo necesario.

Las herramientas IDE (entorno de desarrollo integrado) que se utilizaron para la programación del sistema fueron: IDE de Arduino para la programación del microcontrolador Arduino que utiliza un lenguaje parecido a C++; Para la programación de Raspberry Pi que utiliza el sistema operativo Raspbian se utilizó Thonny un IDE de Python amigable, que permite programar, tiene un depurador que ayuda a corregir los errores, tiene ingreso a la propia consola de Python, permite visualizar variables e inspeccionar objetos.

Para la aplicación móvil se utilizó Android Studio, que tiene un emulador (dispositivo virtual), se puede programar en Java y XML, y un editor de diseño fácil de utilizar. Se utilizó Android Studio por ser la plataforma que brinda mejores características para desarrollar la aplicación móvil y permite la conexión con la base de datos en la nube (Firebase).

Se utiliza una base de datos en la nube (Firebase) que permite la conexión de la aplicación con el sistema. La aplicación envía o recibe la información de la base de datos y la base de datos se comunica con el sistema. También, el sistema envía o recibe información de la base de datos. Firebase es una base de datos en tiempo real, permite realizar autenticación, brinda almacenamiento y da analítica del uso de la aplicación y conexiones. Por estas características es que se trabajó con esta base de datos, ya que permite una fluida comunicación entre la aplicación y el sistema.

Para el sistema de reconocimiento de voz se utiliza la plataforma Wit.ai, por que brinda un servicio gratuito, tiene buena calidad de transformación de audio a texto y se puede implementar en Raspberry Pi porque tiene código en Python que permite la conexión con el servicio.

Se realizaron una serie de pruebas: la prueba de funcionalidad para saber si el sistema cumple con determinados requerimientos; las pruebas de seguridad para visualizar el ingreso de una contraseña y el inicio de sesión por medio de la aplicación móvil; las pruebas de usabilidad para verificaron cómo funciona el sistema y si los usuarios lo adquirirían, y las pruebas de rendimiento y velocidad que por medio de comandos en Linux se constató que el uso de CPU, temperatura, memoria sean aptas para la Raspberry Pi y no genere algún error de hardware.

### 1.5. Aspectos Relevantes del Desarrollo

Para el reconocimiento de voz, se utilizó Wit.ai como un proveedor de servicios. Es una plataforma de tecnología abierta que reconoce palabras y discursos (lenguaje natural), y los transforma al formato Json (JavaScript Object Notation). Desarrollada para aplicaciones que necesitan convertir oraciones en datos estructurados.

Para realizar este proceso de reconocimiento, Wit.ai utiliza una red neuronal recurrente (RNN) por proyecto. Una RNN es una red de aprendizaje que el estado actual influye en el estado siguiente. Es útil en el proceso de reconocimiento de habla porque cada letra influye en la probabilidad de la siguiente. Esta red es alimentada por un espectrograma de segmentos de audio de 20-40 ms. Por cada porción de 20ms se transforma una letra, al finalizar se termina con una secuencia de letras. Después, se elimina los espacios en blanco y se combinan las letras para formar una palabra o una frase. Al pasar el archivo de audio por Wit.ai, se obtiene un archivo Json con el contenido del comando.

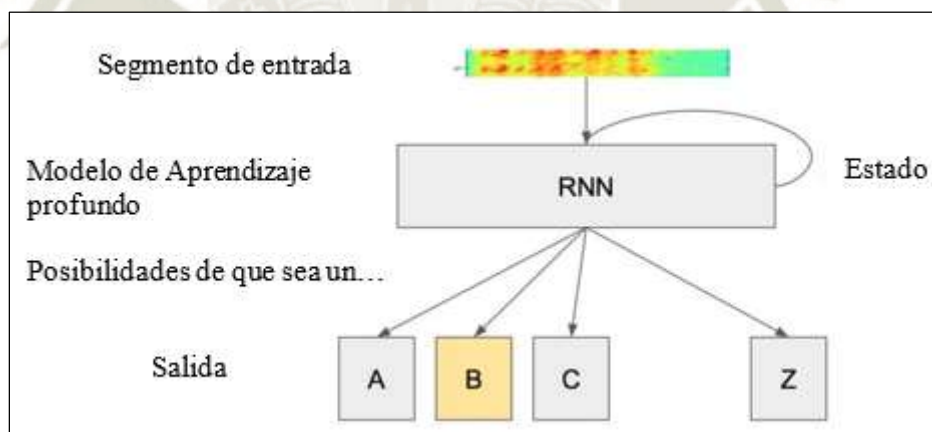


Figura 16. Modelo de una red neuronal recurrente para reconocimiento de voz

Fuente: (Thomas Eberman, 2015)

Wit.ai cuenta con una plataforma en la nube, a la cual se accede por una página web. En la plataforma se indica cuáles son las entidades que se van a utilizar. Las entidades son los comandos que se ingresaran al sistema. Los comandos que se enseñaron son: activar alarma, desactivar alarma, activar gas, desactivar gas, encender luz, apagar luz y palabras

claves. Se le enseña a entender una frase por partes como: activar alarma la comprende como dos entidades, activar como encendido (on) y alarma como otra entidad.



Figura 17. Ingreso de entidades a Wit.ai  
Fuente: Elaboración propia

Además, en Wit.ai se puede observar por medio del historial (logs) la salida que genera al realizar el cambio de audio a un archivo Json. El archivo Json es la salida de la red neuronal, se compone de entidades, valores y el nivel de confianza. El nivel de confianza es el valor de lo que entendió la red neuronal en relación a las entidades ingresadas.

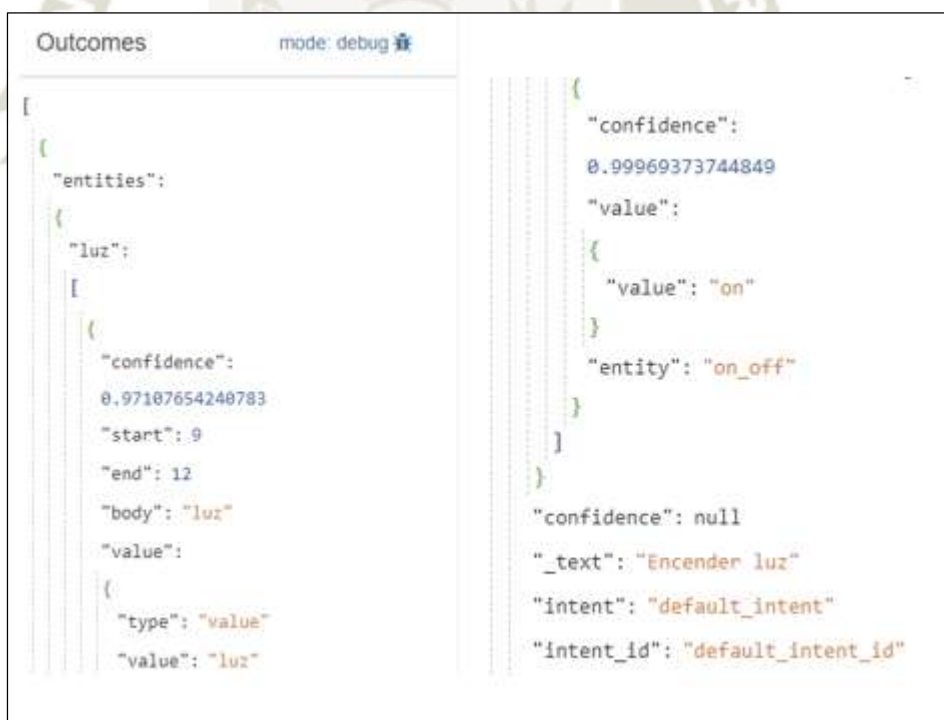


Figura 18. Entidades en Wit.ai.  
Fuente: Elaboración propia

Para conectar el sistema de vigilancia y detección con la aplicación móvil, se utilizó la plataforma Firebase como puente. Firebase es una plataforma de desarrollo móvil que permite desarrollar apps multiplataforma (Android, iOS, etc). Firebase cuenta con varios



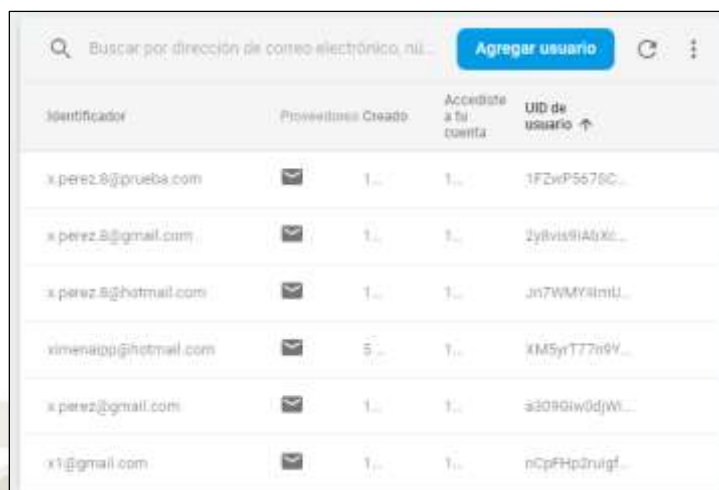
servicios en el proyecto se utilizaron: la base de datos en tiempo real, autenticación y dashboards.

En la base de datos en tiempo real (Realtime Database) de Firebase, los datos se almacenan como objetos Json. La base de datos tiene forma de un árbol Json, sin tablas ni registros. Cuando se agrega datos al árbol Json, se convierten en nodos de la estructura. La base de datos que se creó en Firebase es pequeña pero necesaria para los datos que se utilizan en el sistema de vigilancia y detección. Son diez datos que se estructuran como un árbol, con un padre como “home” (raíz) que tiene hijos como ramas.

```
rpi-demo-658dd
├── home
│   ├── alarma: false
│   ├── codfamilia: "1234"
│   ├── contrasena: "1234"
│   ├── correo: "x.perez.8@gmail.com"
│   ├── estado_temp: false
│   ├── humedad: 18
│   ├── incendio: false
│   ├── palclave: "TECNOLOGICO"
│   └── temperatura: 23
└── message: "Hola mundo"
```

*Figura 19.* Base de datos en Firebase  
Fuente: Elaboración propia

Para la autenticación Firebase permite varios métodos como: por correo/contraseña, por teléfono, por una cuenta de google, por una cuenta de Facebook, por una cuenta en twitter o por una cuenta de GitHub. Se utilizó la autenticación por correo/contraseña por ser el método más usado por los usuarios, que prefieren ingresar con un correo de cualquier proveedor. En la plataforma se visualiza los usuarios creados, la fecha de creación, la fecha del último acceso y un identificador de usuario. El administrador de la plataforma en la sección de autenticación puede: agregar otros usuarios, restablecer la contraseña, inhabilitar la cuenta y borrar la cuenta.



Identificador	Proveedor	Creado	Accede a la cuenta	UID de usuario
x.perez.8@prueba.com	📧	1...	1...	1F2xP5678C...
x.perez.8@gmail.com	📧	1...	1...	2y8vis9IAqXc...
x.perez.8@hotmail.com	📧	1...	1...	3n7WMyAlmIt...
x.perez.8@gmail.com	📧	5...	1...	4M5yT77n9V...
x.perez.8@gmail.com	📧	1...	1...	5309Giw0dJW...
x1@gmail.com	📧	1...	1...	6CpFHp2nuqf...

Figura 20. Administrador de usuarios de Firebase  
Fuente: Elaboración propia

En la parte de analytics se visualizan varios dashboards que muestran: los usuarios activos, en que parte de la aplicación interactúan más los usuarios (pantalla), cual es el porcentaje de retención de usuarios y de qué país es la audiencia de la aplicación. Por ser una aplicación pequeña solo se tomara en cuenta para la administración de la aplicación los usuarios activos y en que pantalla interactúa mayormente el usuario.

En el gráfico se puede visualizar el número de usuarios activos, estos datos son de diciembre 2018 a enero de 2019, fecha en que fueron realizadas las pruebas para registrar nuevos usuarios. En este rango de 28 días los usuarios activos fueron 27 para comprobar que el registro de nuevos usuarios funcione correctamente, y en un rango de siete días fue de un usuario activo. Este dashboard le permite al administrador observar la cantidad de usuarios que utilizan la aplicación en un rango personalizado, de un mes (30 o 28 días) y de siete días.

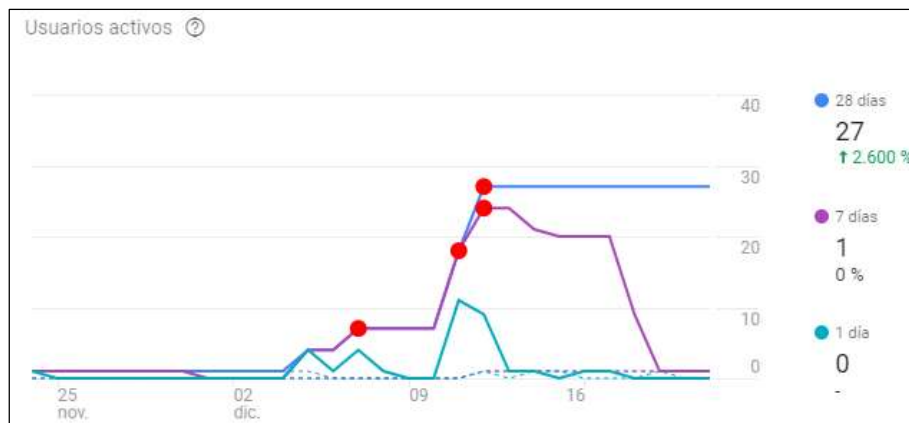


Figura 21. Dashboard usuarios activos  
Fuente: Elaboración propia

En el siguiente gráfico se puede observar cual es la pantalla en que los usuarios tienen más interacción con la aplicación móvil. Los datos que se visualizan son en el mismo rango que el gráfico anterior (diciembre 2018 a enero 2019). Se puede visualizar que el usuario interactúa mayor tiempo en la pestaña de menú principal (MainActivity), la segunda pantalla es la de iniciar sesión (LoginActivity) para que los usuarios pueden ingresar a la aplicación y la tercera pantalla es la de la información de la aplicación. Este dashboard le permite al administrador observar cual es el tiempo y pantalla en que los usuarios emplean más tiempo.

Clase de pantalla	% del total	Tiempo medio
MainActivity	32,73 % ↓ 47,8 %	0 mi...17 s ↑ 38,2 %
LoginActivity	23,19 % ↑ 30 %	0 mi...17 s ↑ 58,2 %
Ayuda_Activity	13,47 % ↑ 1.1... %	0 mi...33 s ↑ 1.578 %
temperatura_Activity	6,33 % ↑ 1.0... %	0 mi...27 s ↑ 2.620,7 %
correo_Activity	5,66 % ↑ 249 %	0 mi...27 s ↑ 803,8 %
cambiarco..._Activity	4,38 % ↑ 17... %	0 mi...16 s ↑ 435,3 %
RegisterActivity	4,2 % ↑ 15... %	0 mi...34 s ↑ 1.062,2 %
incendio_activity	3,01 % ↓ 7,2 %	0 mi...15 s ↑ 160,4 %

Figura 22. Dashboard interacción de usuarios  
Fuente: Elaboración propia

## CAPITULO 2: DOCUMENTACIÓN TÉCNICA

### 2.1. Plan del Proyecto Informático

#### 2.1.1. Planificación temporal del proyecto

Para la realización del proyecto en desarrollo se realizó un diagrama de Gantt en el cual se podrá identificar las tareas para realizar y las fechas en las cuales se desarrollaran cada una de estas, cada tarea cuenta con sub tareas. A continuación se determinaran los principales procesos del proyecto en desarrollo:

- Análisis
- Diseño
- Implementación
- Pruebas
- Instalación / Evaluación

Dichas tareas serán mostradas en el siguiente diagrama de Gantt, además de la secuencia en la que se realizó cada proceso. También, se determina la duración de cada proceso según el número de dedicación efectiva indicada en horas/hombre. Cabe recalcar que el proyecto tendrá una duración de 270 días.

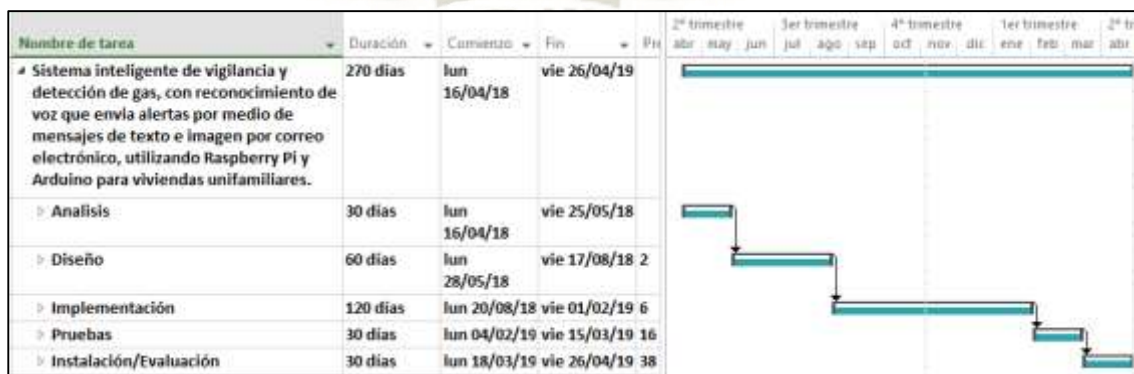


Figura 23. Diagrama de Gantt de los procesos del proyecto  
Fuente: Elaboración propia

A continuación, se muestra el plan de actividades de cada uno de los procesos determinados para el proyecto. Primeramente se muestra el plan de actividades de la fase de Análisis, en la cual se detalla las subtarear, se indica el tiempo de duración de cada una y la secuencia que se debe de seguir. La duración de esta fase será de 30 días.

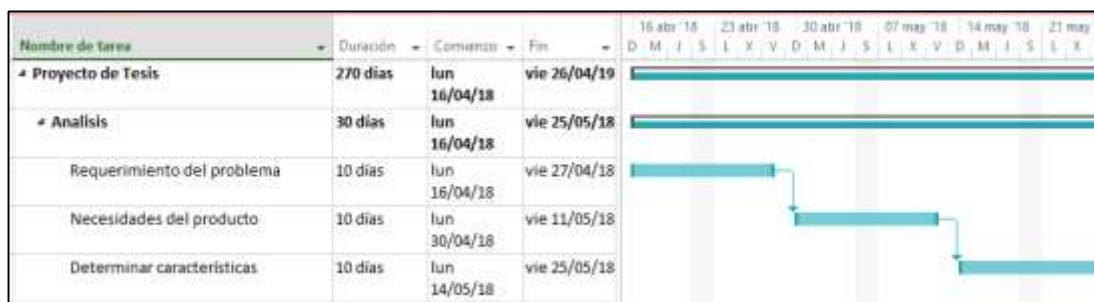


Figura 24. Diagrama de Gantt fase de análisis  
Fuente: Elaboración propia

Después de la fase de análisis, se realizó el planeamiento de la fase de diseño del sistema, el cual se divide en tres subtarear: diseño del sistema de vigilancia y detección, diseño de la aplicación móvil y diseño de la maqueta. Se determinó las tarear de cada subtarea, las fechas de inicio y fin y la secuencia de desarrollo de cada tarea. Esta fase tendrá una duración de 60 días de trabajo.

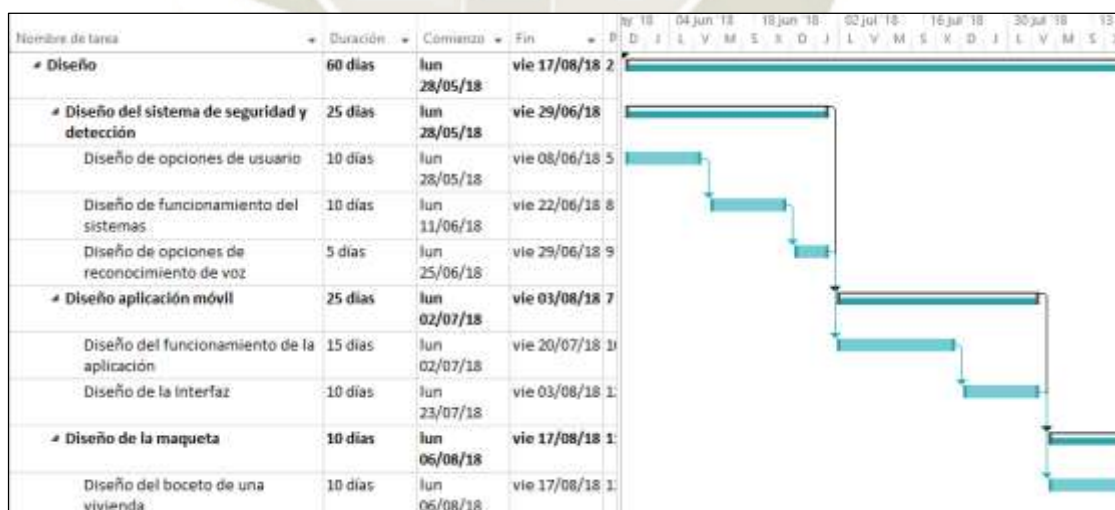


Figura 25. Diagrama de Gantt fase de diseño  
Fuente: Elaboración propia

Posteriormente se realizó el planeamiento de la fase de implementación, la cual se divide en cuatro subprocesos: desarrollo del sistema en Raspberry Pi, desarrollo del sistema en Arduino, desarrollo de la base de datos Firebase y el desarrollo de la aplicación móvil. Esta fase es la más importante del proyecto, ya que en esta fase se hace el desarrollo directo del sistema, la programación de cada proceso necesario para la realización del proyecto. Cada subproceso cuenta con tareas, se les asigna un inicio y fin y se determina la secuencia a realizar. La fase de implementación es la más larga con una duración de 120 días.

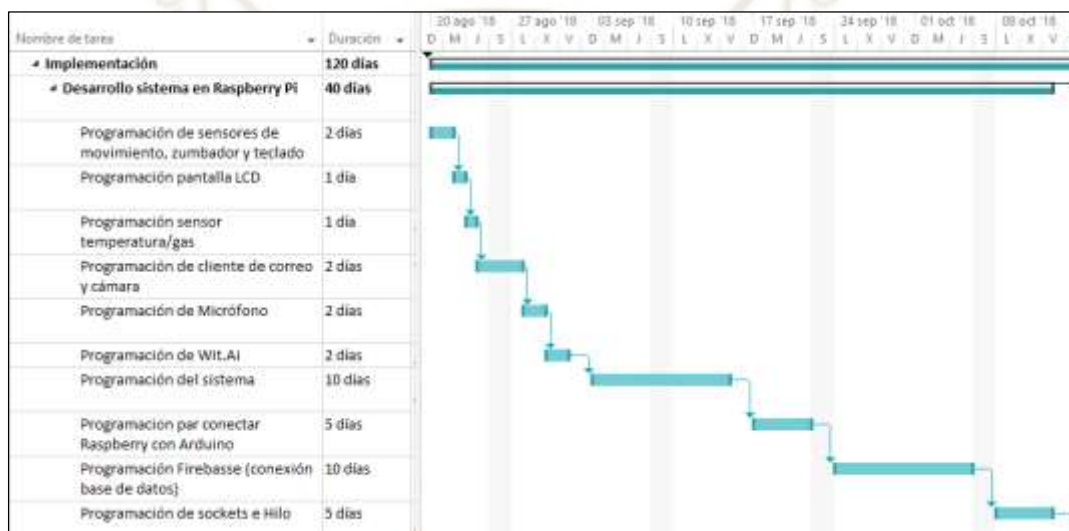


Figura 26. Diagrama de Gantt fase de implementación  
Fuente: Elaboración propia

La fase de implementación se divide en cuatro subprocesos: desarrollo del sistema en Raspberry Pi con una duración de 40 días, desarrollo del sistema en Arduino con una duración de 20 días, desarrollo de la base de datos Firebase con una duración de 20 días y desarrollo de la aplicación móvil con una duración de 40 días.

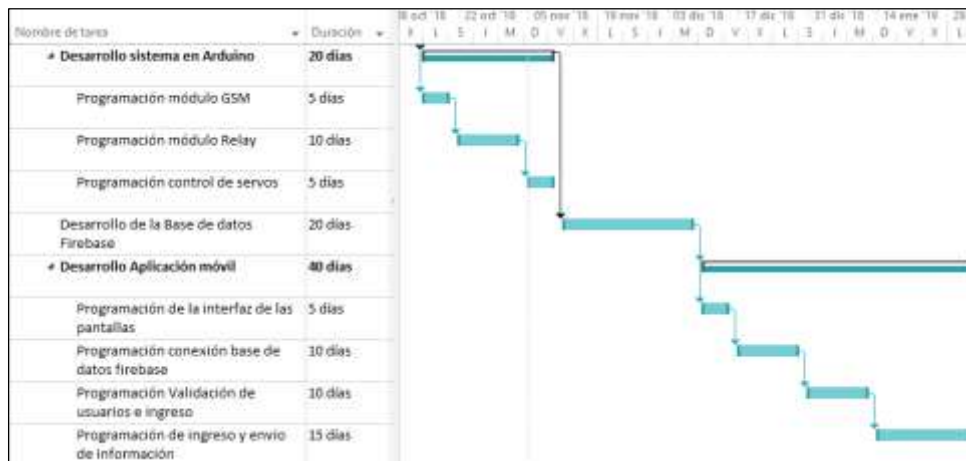


Figura 27. Diagrama de Gantt fase de implementación II  
Fuente: Elaboración propia

Después de la fase de implementación y desarrollo, se desarrolla la planeación del cronograma de la fase de pruebas. Se indican subtarefas, se les asignan las fechas necesarias de inicio y fin y la secuencia en la que se desarrollan. Se determinó que la fase de pruebas del proyecto tendrá una duración de 30 días.

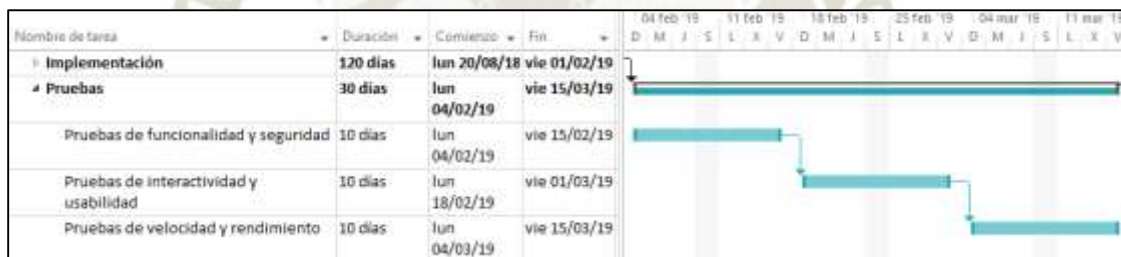


Figura 28. Diagrama de Gantt fase de pruebas  
Fuente: Elaboración propia

Para finalizar, se determinó el cronograma para la fase de Instalación y Evaluación del sistema. Se identificaron las subtarefas que tiene el proceso, se indicó la duración de cada una de ellas y su duración de inicio a fin. Esta fase tendrá una duración de 30 días.



Figura 29. Diagrama de Gantt fase instalación/evaluación  
Fuente: Elaboración propia

## 2.1.2. Estudio de Viabilidad del Proyecto

### 2.1.2.1. Descripción de Productos y Servicios

El sistema de vigilancia y detección de fuga de gas se caracteriza por advertir al usuario de un posible ingreso de un intruso o presencia de gas, enviando alertas por mensaje de texto y la imagen del intruso por correo electrónico.

Utiliza sensores, actuadores y periféricos que facilitan el desarrollo del sistema, son de fácil uso e implementación en el hogar u otro ambiente.

El sistema realiza envío de mensajes de texto por ser el método de comunicación más simple, ya que, cuando ocurre un evento natural (terremoto, Tsunami, etc.) u otro evento, no sufre fallas o caídas. Además, no depende de una conexión de Internet.

El sistema captura imágenes de un tiempo determinado cuando detecta la presencia de un intruso, no hace grabaciones en tiempo real por considerar el espacio que ocupa y que el usuario necesita de la imagen precisa y no buscar el momento exacto para documentarlo.

Cuenta con una aplicación móvil que permite comunicarse con el sistema en tiempo real, activarlo y desactivarlo. Además, permite tener un control y configuración de correo electrónico, contraseña, palabra clave, etc. Para que el usuario pueda gestionar el sistema como desee. No permite tomar una imagen por tema de seguridad, por si alguien adquiere el celular de forma ilícita.

Al contar con un sistema de reconocimiento de voz, lo hace un sistema novedoso, permitiendo al usuario interactuar de una forma diferente con el sistema. En la actualidad, un mayor número de dispositivos implementan esta función como los televisores, celulares, relojes, etc.

Además, el sistema permite el ingreso de opciones por medio de un teclado para ingresar a las opciones de reconocimiento de voz, visualización de temperatura y humedad,



activar o desactivar la alarma o detección de fuga de gas. Presenta una interfaz amigable con el usuario y mensajes interactivos por medio de una pantalla LCD.

#### 2.1.2.2. Consideraciones tecnológicas

- La programación de Raspberry Pi se da en el lenguaje de programación Python, se desarrolla en el sistema operativo Raspbian de Linux.
- Una memoria de más de 16 GB para configurar el sistema operativo en la Raspberry Pi.
- La programación de Arduino se da con un lenguaje parecido a C++.
- La programación de la aplicación se desarrolla en Android Studio, lenguajes de programación Java y XML.
- Un módulo GSM para el envío de mensajes de texto.
- Configuración de un cliente de correo electrónico Mutt.
- Conexión Wifi. (funciona sin conexión con menos funciones).
- Utiliza base de datos en la nube: Firebase.
- Conexión con Wit.ai para el reconocimiento de voz en línea.

#### 2.1.2.3. Características del producto

Es un sistema de vigilancia y detección de fuga de gas elaborado para familias unifamiliares, que habiten en un ambiente de un piso con una puerta principal. Cuenta con los siguientes componentes y su respectivo precio (Tabla 12):

Tabla 12.  
*Componentes del sistema*

Componentes	Precio
Raspberry Pi 3 Modelo b	150.00
Arduino UNO	35.00
Módulo GSM	50.00
Cámara web Logitech	75.00
Micrófono iBlue	18.00
Adaptador de sonido	14.00
Micro SD 16 GB	27.00
Chip Movistar	12.00
02 Sensor PIR	16.00
KeyPad teclado	15.00
01 Micro servo 900g	10.00
LCD led 16x2	15.00
Protoboard mini	4.00
Protoboard estándar	8.00
Módulo Relay	10.00
Sensor de gas	13.00
Sensor de temperatura	9.00
Led	1.00
Foco	10.00
Buzzer	2.00
Cables	
37 Cables macho-hembra	11,10
08 Cables macho-macho	2.40
Cable rojo 10 m	2.00
Cable negro 10 m	2.00
Cable amarillo 10 m	2.00
Cable verde 10 m	2.00
Estaño + pasta	5.00
Otros gastos	430.00
Total (redondeado):	950.00

Fuente: Elaboración propia

#### 2.1.2.4. Estrategias de marketing

##### 2.1.2.4.1. Alta disponibilidad

El sistema de vigilancia y detección de fuga de gas necesita de una conexión a Internet para algunas funciones como: comunicación con la aplicación móvil y los comandos de reconocimiento de voz. Cuando no hay conexión a Internet se puede usar el sistema con características reducidas, envía mensajes de texto y toma una imagen que se guarda en una carpeta en la Raspberry Pi que se puede observar desde

el sistema operativo. El sistema no depende de una conexión a Internet para cumplir con su propósito.

#### 2.1.2.4.2. Seguridad

Para la seguridad del sistema se utiliza una contraseña de cuatro caracteres entre números y letras (0-9, a-d), para activar o desactivar cualquiera de los sistemas, esta contraseña esta codificada en la base de datos en la nube. En el caso de activar por medio de reconocimiento de voz, el sistema le solicita al usuario una palabra clave que le permita activar o desactivar cualquiera de los sistemas.

La base de datos en la nube cuenta con un servicio de autenticación que determina qué tipo de datos puede visualizar un usuario determinado. La base de datos al ser desarrollada por Google, es el encargado de procesar los datos, determinando que están bajo el control del cliente y Firebase actúa como “controlador de datos”. Los servicios de Firebase encriptan datos en tránsito con HTTPS y datos de clientes aislados de manera lógica. Firebase cumple con certificaciones de seguridad y privacidad: ISO 27001, ISO 27017, ISO 27018, SOC 1, SOC 2 y SOC 3.

#### 2.1.2.4.3. Escalabilidad

Al sistema se le puede agregar más hardware (sensores, actuadores, periféricos), pero se debe de considerar la cantidad de voltaje que puede soportar, sin realizar cortocircuitos o quemar las placas Raspberry Pi o Arduino.

#### 2.1.2.4.4. Análisis de datos

Firebase permite realizar análisis de datos, se visualizan dashboards que presentan el performance de la aplicación, el número de ingresos a la aplicación, el número de descargas que se hicieron a la base de datos y el almacenamiento.

#### 2.1.2.4.5. Costo reducido

Los servicios de Firebase y Wit.ai se utilizan en su versión gratuita por parte de ambos no habría una cuota a pagar. Para los mensajes de texto se utiliza un chip movistar prepago, se recomienda realizar una recarga de cinco soles al mes. Otro costo es adquirir todos los componentes necesarios para el sistema. Todos los componentes utilizados se describen en la Tabla 12, con su respectivo precio. Como se puede observar son elementos de bajo costo y se caracteriza por su largo tiempo de vida.

#### 2.1.2.5. Organización y recursos humanos

- Desarrollador y arquitecto de aplicaciones en Raspberry Pi (Python), Android (Java y XML) y Arduino (lenguaje parecido a C++).
- Consultor especialista en desarrollo del sistema.

#### 2.1.2.6. Proyecciones Financieras

El sistema de vigilancia y prevención es una opción económica comparada con otros sistemas de alarma. Cuenta con un costo accesible, ya que otras empresas cobran entre S/ 1500 a S/ 3000 y las características que presentan son similares.

El usuario que desee adquirir el sistema se le cobrara el valor aproximado de S/ 950.00 un costo único, que depende del tamaño del espacio donde se desea instalar. A mayor sea el espacio por longitud de los cables el precio podría aumentar entre 10 o 20 soles. El precio incluye todos los componentes necesarios para el sistema, la instalación y un manual de inducción al sistema. No incluye los 5 soles de saldo que necesita el chip que envía los mensajes de texto.

Por cada sistema vendido se gana alrededor de S/ 400, la programación del sistema es la misma, amoldándola al cliente e instalándola donde se desee.

## 2.2. Especificación de requisitos de software (Elicitación y Análisis)

### 2.2.1. Especificación de requisitos

Para determinar los requisitos de software para el proyecto, se realizó encuestas a posibles usuarios del sistema. La encuesta lleva como nombre: “Encuesta: Sistema de vigilancia y detección de fuga de gas”. Para hacer la elección de las personas encuestadas, se utilizó el método de conveniencia. Por ello, las personas encuestadas son allegadas al encuestador como familiares, amigos, amigos de los familiares (personas entre 40 y 70 años), etc. El tamaño de la muestra es de 52 personas.

Al realizar la encuesta los resultados fueron los siguientes:

- El mayor rango de personas encuestadas tiene entre 30 y 50 años y representan el 44.2% de 52 personas.
- Se consideró hacer un prototipo funcional que represente una casa de un solo piso con una puerta principal por ser el modelo de construcción más representante entre las personas encuestadas con un 46.2%.
- De las personas encuestadas el 55.8% NO cuenta con un sistema de alarma convencional, lo cual los convierten en posibles clientes.
- El 51,9% de personas encuestadas indican que alguna vez ingresaron a su propiedad sin su consentimiento y un 63,5% indicó que fue víctima de robo en su propiedad, ambas respuestas representan el nivel de inseguridad en la muestra. La cual es más del 50% de las personas encuestadas, un porcentaje medio a alto.
- El 61,5% ha escuchado hablar sobre los sistemas de alarma inteligente.
- Las personas encuestadas están dispuestas a pagar entre 800 y 1000 soles por un sistema con estas características.

- El 84,6% respondió afirmativamente que adquiriría un sistema con estas características.
- El 75% prefiere pagar una sola vez por un sistema de seguridad que depender de una empresa y pagar una cuota mensual.

Los siguientes resultados determinan los requerimientos del sistema de vigilancia y detección de fuga de gas:

- Controlar el sistema de vigilancia y detección de fuga de gas por medio de un panel de control y aplicación móvil.
- Realizar notificaciones por mensajes de texto y aplicación móvil.
- Agregar otra característica al sistema: detectar la presencia de gas e incendio y alertar a algún familiar por medio de un botón de emergencia.

## **Anexo B. ENCUESTA: Sistema de vigilancia y detección de fuga de gas.**

### **2.2.2. Análisis**

Después de observar los requerimientos del sistema, se hizo un análisis de: microprocesadores, microcontroladores, sensores, software de reconocimiento de voz y gestores de base de datos, necesarios para el sistema.

#### **2.2.2.1. Comparación microprocesadores Raspberry Pi**

Para el desarrollo del sistema se compararon los principales y habituales modelos de Raspberry Pi. Los modelos son: Raspberry Pi 1 B+, Raspberry Pi 2 Modelo B y Raspberry Pi 3 Modelo B (Tabla 17). Después de comparar los modelos se eligió el modelo Raspberry Pi 3 Modelo B, por tener cuatro puertos USB, mejor conectividad Bluetooth 4.1 y Wi-fi 802.11n (Low Energy), mejor potencia de cálculo con un SOC Broadcom BCM2837, un procesador ARMv8 de cuatro núcleos a 1.2GHz de 64 bits y memoria RAM de 1 GB.

Tabla 13.  
*Comparación de Modelos Raspberry Pi*

Comparación Modelos Raspberry Pi			
Modelo	Raspberry Pi 1 B+	Raspberry Pi 2 B	Raspberry Pi 3 B
Fecha	15 Feb 2012	1 Feb 2015	29 Feb 2016
Precio	130 soles	150 soles	150 soles
SOC	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837
Core	ARM 1176JFZ-S	ARM Cortex-A7	ARMv8 64-bit
N° Cores	1	4	4
GPU	VideoCore IV	250Mhz VideoCore IV	400MHz VideoCore IV
CPU Clock	700 MHz	900 MHz	1.2 GHz
RAM	512 MB	1 GB	1 GB
Memoria	Micro SD	Micro SD	Micro SD
USB	2	4	4
Ethernet	Si	Si	Si
Wi-Fi	No	No	Si
Bluetooth	No	No	Si
HDMI	Si	Si	Si
GPIO	40	40	40
Consumo	700 mA	820 mA	1400 mA

Fuente: (Luis Llamas, 2016)

Para utilizar el microprocesador Raspberry se necesita instalar en una memoria microSD un sistema operativo, los principales para IoT son:

- Raspbian: sistema operativo desarrollado específicamente para Raspberry Pi. Basado en la distribución de Linux denominada Debian, es libre. En el sistema operativo viene preinstalado software de programación, educativo y de uso general (navegador web, Libre Office, etc.). Permite aprovechar el máximo potencial del microprocesador.
- Ubuntu Mate: Es equivalente a Raspbian, pero en la distribución de Linux Ubuntu. Es un sistema operativo de escritorio. Es asequible a usuarios adaptados a recursos limitados y necesita de una tarjeta microSD de 6GB.
- Windows 10 IoT Core: es una versión especial de Windows para Raspberry Pi. Es una plataforma de desarrollo para que los programadores experimenten

con dispositivos conectados a Internet. Se necesita otro ordenador con Windows 10 y Visual Studio.

El sistema operativo con el cual se va a trabajar es Raspbian, por ser específico para Raspberry Pi. Además, los tiempos de espera de correr una aplicación son menores y puede realizar diferentes tareas a la vez, a diferencia de Ubuntu Mate que presenta latencia y programas pesados. En comparación con Windows 10 IoT, no tiene muchas librerías para los componentes que requiere el sistema, por ser un sistema operativo reciente.

En Raspbian se puede programar en Python o Java. Para el desarrollo del proyecto se eligió Python, debido a que hay más librerías, mayor desarrollo en comunidad (foros), y es más fácil de utilizar los pines GPIO. En cambio Java es más pesado de ejecutar y no hay mucho desarrollo.

#### **2.2.2.2. Comparación microcontroladores Arduino**

Para el proyecto se consideró usar una placa Arduino para conectar los componentes que no pueden funcionar a 3.3v en la placa Raspberry Pi y para acoplar el módulo GSM/GPRS para el envío de mensajes de texto. Las placas de Arduino que se observaron fueron: Arduino Uno R3, Arduino Mega y Arduino Leonardo (Tabla 14).



Tabla 14.  
*Comparación entre modelos Arduino*

Comparación entre modelos Arduino			
Modelo	Arduino Uno	Arduino Mega 2560	Arduino Leonardo
Microcontrolador	AVR ATmega 328 8 bits	AVR ATmega 2560 8 bits	AVR ATmega 32u4 8 bits
Frecuencia	16 Mhz	16 Mhz	16 Mhz
Memoria RAM	2 KiB	8 KiB	2.5 KiB
Memoria EEPROM	1 KiB	4 KiB	1 KiB
Memoria FLASH	32 KiB	128 o 256 KiB	32 KiB
Pines digitales entrada/salida	14/14	54/54	20/20
Tensión/corriente pines digitales	5v 40 mA	5v 40 mA	5v 40 mA
Pines analógicos entradas/salidas	6/0	16/0	12/0
Tensión/resolución pines analógicos	5v 10 bits (1024 valores)	5v 10 bits (1024 valores)	5v 10 bits (1024 valores)
Pines de interrupción externa	2	6	2
Pines PWM	6	15	7
Conexiones Serial/ UART	1	4	1
Conexiones I2C/TWI	1	1	1
Precio	79 soles	150 soles	60 soles

Fuente: Elaboración propia

Se comparó Arduino Uno con Arduino Mega, se tuvo en cuenta el tamaño y el número de pines que se va a utilizar. Arduino Mega es más grande que Arduino UNO. Para el proyecto se necesitan siete pines digitales, con los 14 pines digitales de Arduino Uno se cumple con las expectativas. Y El precio de Arduino Mega es el doble de un Arduino Uno, lo que se quiere es una solución económica. También, se comparó Arduino Uno con Arduino Leonardo, este modelo es nuevo. Por ello, no es compatible con algunos shields (módulos) y librerías. El modelo Arduino Uno es más comercial. Por estas razones se eligió trabajar con la placa Arduino Uno.

### 2.2.2.3. Comparación sensores

- Sensor de detección de gas: Para que sea un sistema de detección de fuga de gas se necesita de un sensor que detecte gas en un ambiente. Se comparó sensores MQ-x

(Tabla 15) los cuales pueden detectar gas o humo. Se eligió el sensor de gas MQ-4 por ser la opción más económica, comercial y que detecta gas metano y gas natural. Este sensor se instalara en la cocina para advertir al usuario de una fuga de gas que podría generar un incendio.

Tabla 15.  
*Comparación entre sensores MQ-x*

Comparación entre sensores MQ-x				
Modelo	MQ-2	MQ-3	MQ-4	MQ-5
Sustancia detectadas	Metano, butano, GLP, humo	Alcohol, Etanol, humo	Metano, gas natural comprimido (GNP)	Gas natural (propano), GLP
Calentador	5V	5V	5V	5V
Peso	10 g	10 g	10 g	10 g
Precio	15 soles	18 soles	13 soles	20 soles

Fuente: Elaboración propia

- Sensor de temperatura y humedad: Para medir temperatura y humedad se necesita de un sensor. Se compararon sensores para esta función de la familia DHT (Tabla 16). Se eligió el sensor DHT11, ya que soporta ciclos más rápidos de lectura, menores ciclos de operación y es de menor precio. El sensor dht11 se instalara en la cocina, para advertir al usuario cuando hay una temperatura mayor de 40° (posible incendio).

Tabla 16.  
*Comparación entre sensores de temperatura y humedad DHT*

Comparación entre sensores de temperatura y humedad DHT		
Modelo	DHT11	DHT22
Alimentación	3.3V a 5V	3.3V a 5V
Corriente máxima	2.5mA durante la conversión	2.5mA durante la conversión
Lectura de humedad	+/- 5% de precisión	+/- 2% a 5% de precisión
Lector de temperatura	+/- 2 °C de precisión	+/- 0.5 °C de precisión
Capaz de medir humedad	20% a 80%	0% a 100%
Capaz de medir temperatura	0 a 50 °C	-40°C a 125 °C
Tasa de muestreo	1 Hz (cada segundo)	0.5 Hz (cada 2 segundos)
Dimensiones	15.5mm x 12mm x 5.5 mm	15.1 mm x 25 mm x 7.7mm
Precio	9 soles	13 soles

Fuente:

Elaboración propia

- Sensor de movimiento: Por ser un sistema de seguridad, se caracteriza por la detección de intrusos en el hogar, se necesita de algún sensor que cumpla con esta función. Por ello, se comparó el sensor PIR y el sensor ultrasónico (Tabla 17). Se escogió el sensor PIR por el espacio que se necesita proteger (sala-comedor), por ser un espacio cerrado y de visión directa hacia la puerta principal. Se pondrá dos sensores PIR para que detecten el movimiento y el ingreso de algún intruso. El precio de un sensor ultrasónico es dos sensores PIR, que es más económico.

Tabla 17.  
*Comparación entre sensores de movimiento*

Comparación entre sensores que detectan movimiento		
Modelo	PIR	Ultrasónico
Objeto detectable	Detección afectada por materiales/colores del objeto.	Detección no afectada por materiales/colores del objeto.
Distancia de detección	Max. 1000 mm 3.94"	Max. 10 m 32.8"
Precisión	Alta	Baja
Velocidad de respuesta	Rápida	Lenta
Rango de medición	Pequeño	Grande
Espacio	Cerrados, techos altos, con bastante flujo de aire y zonas de visión directa.	Espacios donde la visión no es posible y requieren un mayor nivel de sensibilidad.
Precio	8 soles	15 soles

Fuente: Elaboración propia

#### 2.2.2.4. Comparación software de reconocimiento de voz

Hay varios softwares de reconocimiento de voz que se pueden utilizar en Raspberry Pi. Algunos de ellos son: Google Cloud Speech API, Pocket Sphinx y Wit.ai. Se compararon los tres softwares como se visualiza en la Tabla 18.

Tabla 18. Comparación software de reconocimiento de voz  
*Comparación software de reconocimiento de voz*

Comparación software de reconocimiento de voz			
Software	Google Cloud Speech API	Pocket Sphinx	Wit.ai
Conexión a Internet	Si	No	Si
Tiene un costo	Si	No	No
Código abierto	No	Si	No
Nivel de Reconocimiento	Alto	Medio	Alto
Idioma	80 a 110 idiomas	Ingles / español por librerías	Varios (Inglés, Español, etc.)

Fuente: Elaboración propia

El sistema de reconocimiento de Google (Cloud Speech API) cobra por el servicio de reconocimiento de voz de 0.006 dólares por 15 segundos, en cambio, Pocket Sphinx y Wit.ai son softwares sin costo. Por otro lado, Pocket Sphinx el nivel de reconocimiento es medio a bajo no reconoce muy bien las palabras, al momento de probarlo no reconocía las

palabras que se decían, en cambio, Wit.ai tiene mejor reconocimiento y una interfaz interactiva con el desarrollador. El desarrollador indica las palabras que se van a ingresar, en caso se ingrese nuevas, también las reconoce y las considera como posibles entidades.

Se eligió como software de reconocimiento de voz a Wit.ai por la facilidad de uso, el nivel de reconocimiento de voz (entiende lo que dice el usuario) y no tiene costo.

#### 2.2.2.5. Comparación gestor de base de datos

El sistema requiere de una base de datos en tiempo real que permita que la aplicación móvil se comunique con el sistema. Algunas opciones son: Firebase, MySql y Thingspeak.

- **Firestore:** es una plataforma que contiene una base de datos en la nube para aplicaciones en tiempo real. Es multiplataforma, para Android, IOS, C++ y Node.js. No utiliza web services, api rest o configuración del servidor.
- **MySQL:** es un gestor de base de datos relacionales (RDBMS) para transacciones y datos relacionales. Es de código abierto. Hay varios proveedores de nube que admiten una versión administrada. Con el servidor que más trabaja es Apache y el entorno de desarrollo PHP. Se puede trabajar con phpMyAdmin en el cual se gestiona MySQL gráficamente por Internet.
- **Thingspeak:** es una plataforma IoT. Que permite recoger y almacenar datos de sensores en la nube y desarrollar aplicaciones. Permite visualizar y analizar datos en MATLAB. Desventajas: poca documentación en la red, falta de funcionalidades y menor cantidad de hardware compatible.

Thingspeak y Mysql son plataformas de gestión de base de datos interesantes pero solo reciben y muestran los datos que envían los sensores. Se eligió trabajar con Firestore, porque no solo presenta una base de datos en tiempo real, sino, brinda otros servicios como: autenticación, hosting, almacenamiento en la nube, supervisión del rendimiento de la app o

web y Google Analytics. Ventajas: fácil migración de datos, validación de varios tipos de datos, no requiere de mucho conocimiento, es fácil de usar, no tiene costo y es de rápido desarrollo. La versión de Firebase que se utilizó es Spark (gratuita) que incluye 100 conexiones simultáneas, 1GB de almacenamiento, 10 GB de transferencia, 1GB de descarga al día y dominio personalizado y SSL (Secure Sockets Layer). Firebase es una amplia plataforma que permite la comunicación entre el sistema que se desarrolla en Raspberry Pi y la aplicación móvil en Android.

### **2.3. Especificación de diseño**

Después, de considerar los requisitos, necesidades y características que necesita un sistema de vigilancia y detección. Se procede al diseño del sistema, se utiliza el funcionamiento de alarmas clásicas como ejemplo, brindándole una interfaz intuitiva e interactiva con el cliente. Esta fase se divide en tres partes: el diseño del sistema de vigilancia y detección, el diseño de la aplicación móvil y el diseño de una maqueta como prototipo.

#### **2.3.1. Diseño del sistema de vigilancia y detección de fuga de gas**

Son varios los componentes que se utilizan en el desarrollo del sistema, se organizan según las características y voltaje que requieren para funcionar. Algunos componentes necesitan una conexión de 5V, se conectan a Arduino, ya que Raspberry Pi solo maneja 3.3V en sus pines, menos en los de voltaje y conexión a tierra (GND). El microprocesador Raspberry Pi tiene más poder de procesamiento por eso permite utilizar software de reconocimiento de voz e implementar la conexión a una base de datos en la nube. Se realizó un diagrama de bloques de la conexión de los componentes con el microprocesador Raspberry Pi 3 Modelo B y el microcontrolador Arduino Uno.

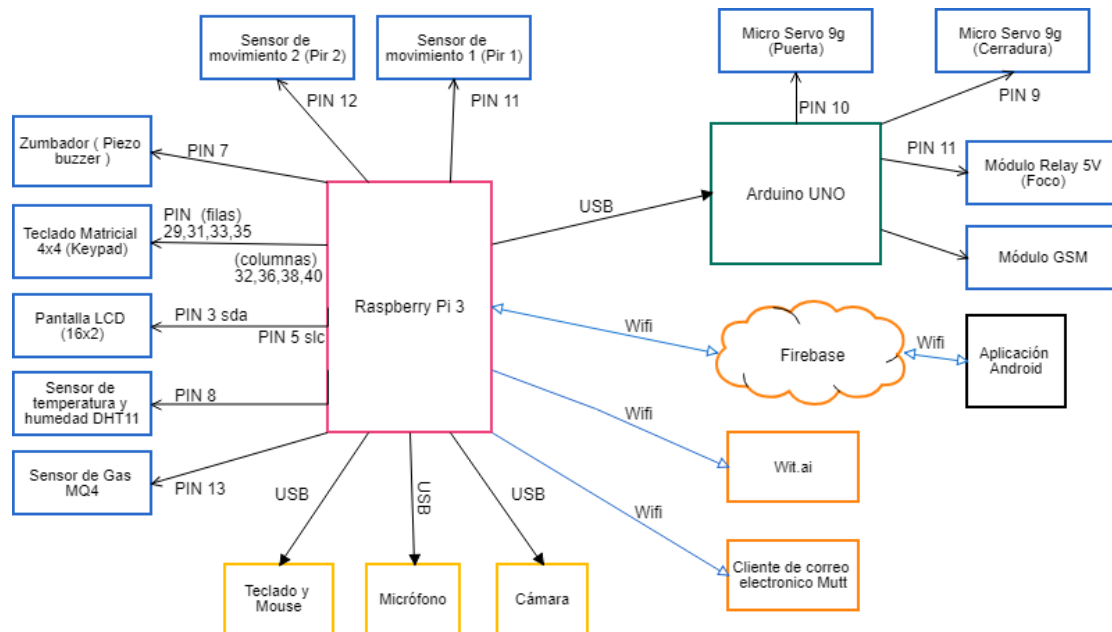


Figura 30. Diagrama de bloques del sistema de vigilancia y detección de fuga de gas  
Fuente: Elaboración propia

Raspberry Pi 3 Modelo B es un microprocesador que cuenta con 40 pines de entrada/salida en los cuales se conectan sensores, actuadores y periféricos. En la tabla 19, se explica que pines se utilizaron y que se conecta en cada uno de ellos. Los pines 3 y 5 son utilizados para conectar una pantalla LCD que necesita de la conexión SDA y SCL. Se utiliza dos salidas de voltaje 5V y dos conexiones a tierra (GND). En el pin 7 se conecta el buzzer (zumbador), en los pines 8, 11, 12 y 13 se conectan sensores. Para la conexión del teclado matricial 4x4 se necesitan de cuatro pines (29, 31, 33, 35) para las filas y 4 pines (32, 36, 38, 40) para las columnas.

Tabla 19.  
*Pines del microprocesador Raspberry Pi*

Pines del microprocesador Raspberry Pi					
Sensor / Conexión	Nombre	Pin #	Pin #	Nombre	Sensor / Conexión
	<b>3.3 V PWR</b>	1	2	<b>5V PWR</b>	Conexión positiva
SDA LCD	<b>GPIO 2</b> <b>SDA1 I2C</b>	3	4	<b>5V PWR</b>	Conexión positiva
SCL LCD	<b>GPIO 3</b> <b>SCL1 I2C</b>	5	6	<b>GND</b>	Conexión a tierra
Piezo buzzer	<b>GPIO 4</b>	7	8	<b>GPIO 14</b>	Sensor de temperatura
	<b>GND</b>	9	10	<b>GPIO 15</b>	
Sensor PIR 1	<b>GPIO 17</b>	11	12	<b>GPIO 18</b>	Sensor PIR 2
Sensor de gas	<b>GPIO 27</b>	13	14	<b>GND</b>	Conexión a tierra
	<b>GPIO 22</b>	15	16	<b>GPIO 23</b>	
	<b>3.3 V PWR</b>	17	18	<b>GPIO 24</b>	
	<b>GPIO 10</b>	19	20	<b>GND</b>	
	<b>GPIO 9</b>	21	22	<b>GPIO 25</b>	
	<b>GPIO 11</b>	23	24	<b>GPIO 8</b>	
	<b>GND</b>	25	26	<b>GPIO 7</b>	
	<b>ID_SD</b>	27	28	<b>ID_SC</b>	
1 fila teclado	<b>GPIO 5</b>	29	30	<b>GND</b>	
2 fila teclado	<b>GPIO 6</b>	31	32	<b>GPIO 12</b>	5 columna teclado
3 fila teclado	<b>GPIO 13</b>	33	34	<b>GND</b>	
4 fila teclado	<b>GPIO 19</b>	35	36	<b>GPIO 16</b>	6 columna teclado
	<b>GPIO 26</b>	37	38	<b>GPIO 20</b>	7 columna teclado
	<b>GND</b>	39	40	<b>GPIO 21</b>	8 columna teclado

**Leyenda**

<b>Power +</b>	<b>UART</b>
<b>GND</b>	<b>SPI</b>
<b>I2C</b>	<b>GPIO</b>

Fuente: Elaboración propia

Además, Raspberry Pi 3 Modelo B cuenta con cuatro puertos USB. En los cuales se conecta: en el primer puerto se conectan el teclado y mouse, en el segundo puerto el microcontrolador Arduino, en el tercer puerto el micrófono con un adaptador de puerto Jack a USB y en el cuarto puerto la cámara USB.

El microcontrolador Arduino cuenta con cinco pines de poder (3.3V, 5V y GND), con seis pines analógicos y con catorce pines digitales (Tabla 24). Cada pin tiene un voltaje de 5V, por ello se conecta un servomotor y un módulo Relay (foco). El módulo GSM se



conecta sobre el Arduino y utiliza los pines Tx y Rx (8 y 7) para la comunicación y envió de mensajes de texto.

Tabla 20.  
*Pines del microcontrolador Arduino*

Pines del microcontrolador Arduino					
Sensor/Conexión	Función	Pin	Pin	Función	Sensor/Conexión
		<b>RESET</b>	<b>GND</b>		
	Power	<b>3.3 V</b>	<b>13</b>	Pines digitales	Led
Conexión positiva		<b>5 V</b>	<b>12</b>		
Conexión a Tierra		<b>GND</b>	<b>11</b>		Relay
		<b>GND</b>	<b>10</b>		Servo puerta
		<b>VIN</b>	<b>9</b>		
	Pines analógicos	<b>A0</b>	<b>8</b>		Tx GSM
		<b>A1</b>	<b>7</b>		Rx GSM
		<b>A2</b>	<b>6</b>		
		<b>A3</b>	<b>5</b>		
		<b>A4</b>	<b>4</b>		
		<b>A5</b>	<b>3</b>		
			<b>2</b>		
			<b>1</b>	Tx	
			<b>0</b>	Rx	

Fuente: Elaboración propia

Después, de organizar los componentes y la capa física del sistema, se diseña la capa lógica que determina el funcionamiento del sistema y se desarrolla según las características que necesita el sistema. Se divide en dos partes según el agente que utiliza el sistema, cliente y administrador (desarrollador). El cliente es quien gestiona y controla la alarma, observa temperatura y humedad en tiempo real, utiliza comandos de voz, alerta a un familiar si se encuentra en peligro y por medio de la aplicación móvil puede gestionar y controlar el sistema de forma remota. Por otro lado el administrador del sistema se encarga de: consultar y verificar el desarrollo de la base de datos en la nube, implementar el software de reconocimiento Wit.ai, observar el performance de la aplicación, el desarrollo y la actualización del sistema.

Para que el sistema sea interactivo con el usuario, permite el ingreso de varias opciones por medio del panel de control (teclado) como se puede observar en el diagrama

de secuencia, según la opción, presenta mensajes en una pantalla LCD. Es un teclado alfanumérico con números del 0 al 9 y letras de la “A” a la “D”. Las opciones que se puede ingresar son: opción A, activar la alarma completa (vigilancia y detección de fuga de gas); opción B, desactivar la alarma completa; opción 0, enviar un mensaje de emergencia a algún familiar; opción C, activar solo detección de fuga de gas; opción D, desactivar detección de fuga de gas; opción \*, muestra temperatura y humedad cuando los sistemas de vigilancia y detección no están activados; y opción #, sirve para activar la opción de reconocimiento de voz, el sistema espera el ingreso de una opción por medio de comandos de voz.

El sistema de seguridad funciona cuando la alarma esta activada sea por ingresar la opción A, por comando de voz (“activar alarma”) o por la aplicación móvil. Es una alarma que detecta presencia de intrusos y detecta presencia de gas, es un sistema completo. El sistema de seguridad cuando detecta una presencia por medio de los sensores PIR (infrarrojo), suena el buzzer alarmando al intruso, envía un mensaje de texto al propietario, prende un foco según la hora (de 6:00 pm a 4:00 am), toma una imagen, la guarda y la envía a un correo electrónico determinado, y avisa que se cerraran las puertas como método de protección. Para que el sistema no este enviando mensajes muchas veces o enviando correos electrónicos, cuenta con un contador de tiempo que al pasar treinta segundos de detectar una presencia abre las cerraduras. Si, después de ese lapso de tiempo el sistema detecta una presencia, realiza el procedimiento descrito con anterioridad.

El módulo GSM es el que envía los mensajes de texto, funciona como un celular con una operadora móvil en este caso Movistar. Enviar mensajes de texto es un método que no necesita de internet y si ocurre un desastre natural, y las líneas se colapsan, el mensaje igual llegara a un receptor con latencia de algunos minutos.

El sistema funciona con y sin internet. Sin una conexión a internet el sistema no puede enviar la imagen al correo electrónico, el usuario no puede manejar el sistema por la aplicación móvil y no funciona el sistema de reconocimiento de voz. Pero si guarda la imagen adquirida en una carpeta en la misma Raspberry Pi y las demás características del sistema funcionan con normalidad.

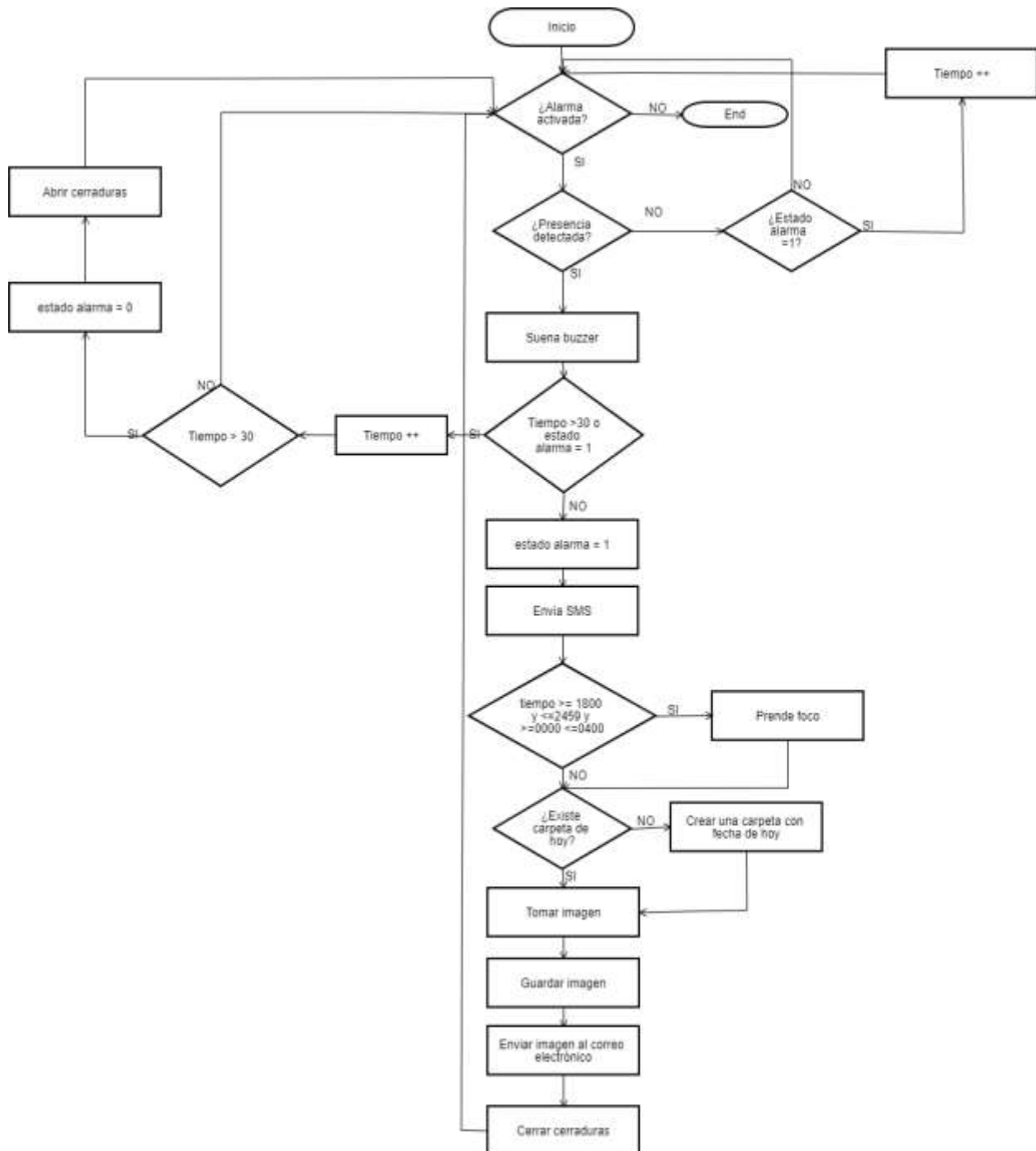
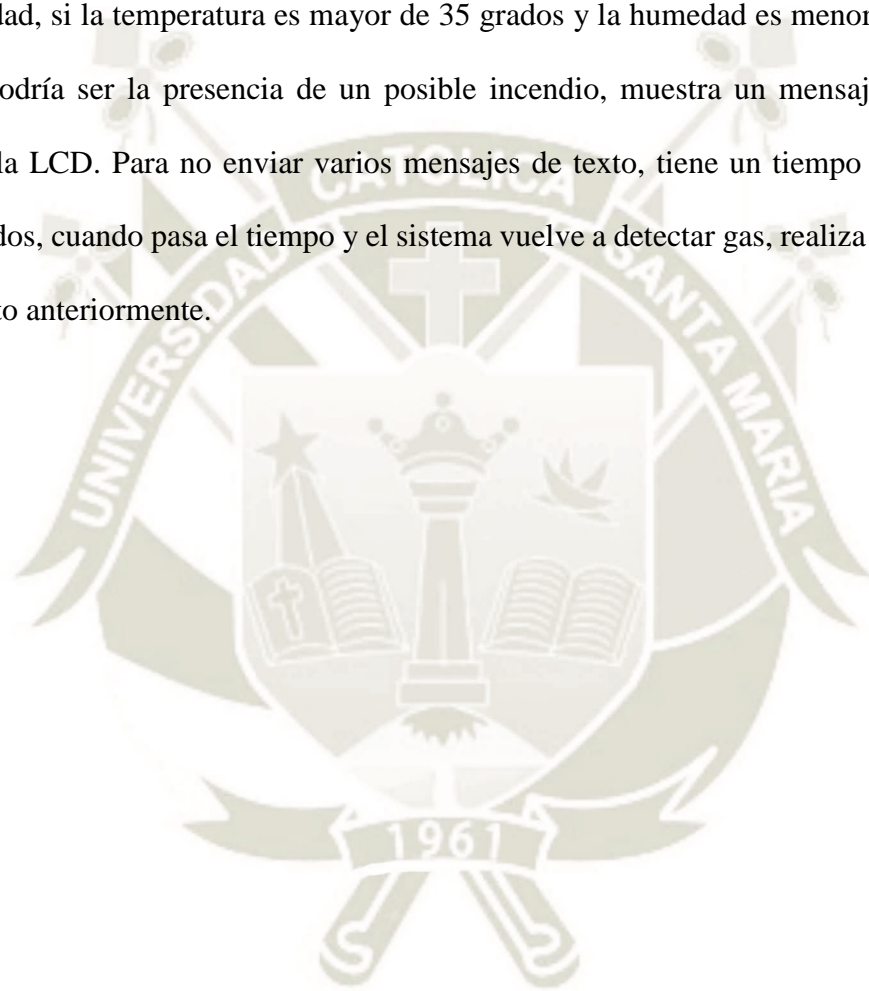


Figura 31. Diagrama de flujo sistema de seguridad  
Fuente: Elaboración propia

El sistema de detección de fuga de gas se activa por las opciones de: ingresar la opción C por medio del panel de control, decir un comando de voz (“activar gas”), por medio de la aplicación móvil o en conjunto con el sistema de seguridad. El sistema de detección de fuga de gas funciona cuando el sensor de gas MQ4, siente la presencia de gas metano, suena el buzzer y envía un mensaje de texto de alerta. El sensor dht11, mide la temperatura y humedad, si la temperatura es mayor de 35 grados y la humedad es menor de 10 grados, lo cual podría ser la presencia de un posible incendio, muestra un mensaje de alerta en la pantalla LCD. Para no enviar varios mensajes de texto, tiene un tiempo de control de 15 segundos, cuando pasa el tiempo y el sistema vuelve a detectar gas, realiza el procedimiento descrito anteriormente.



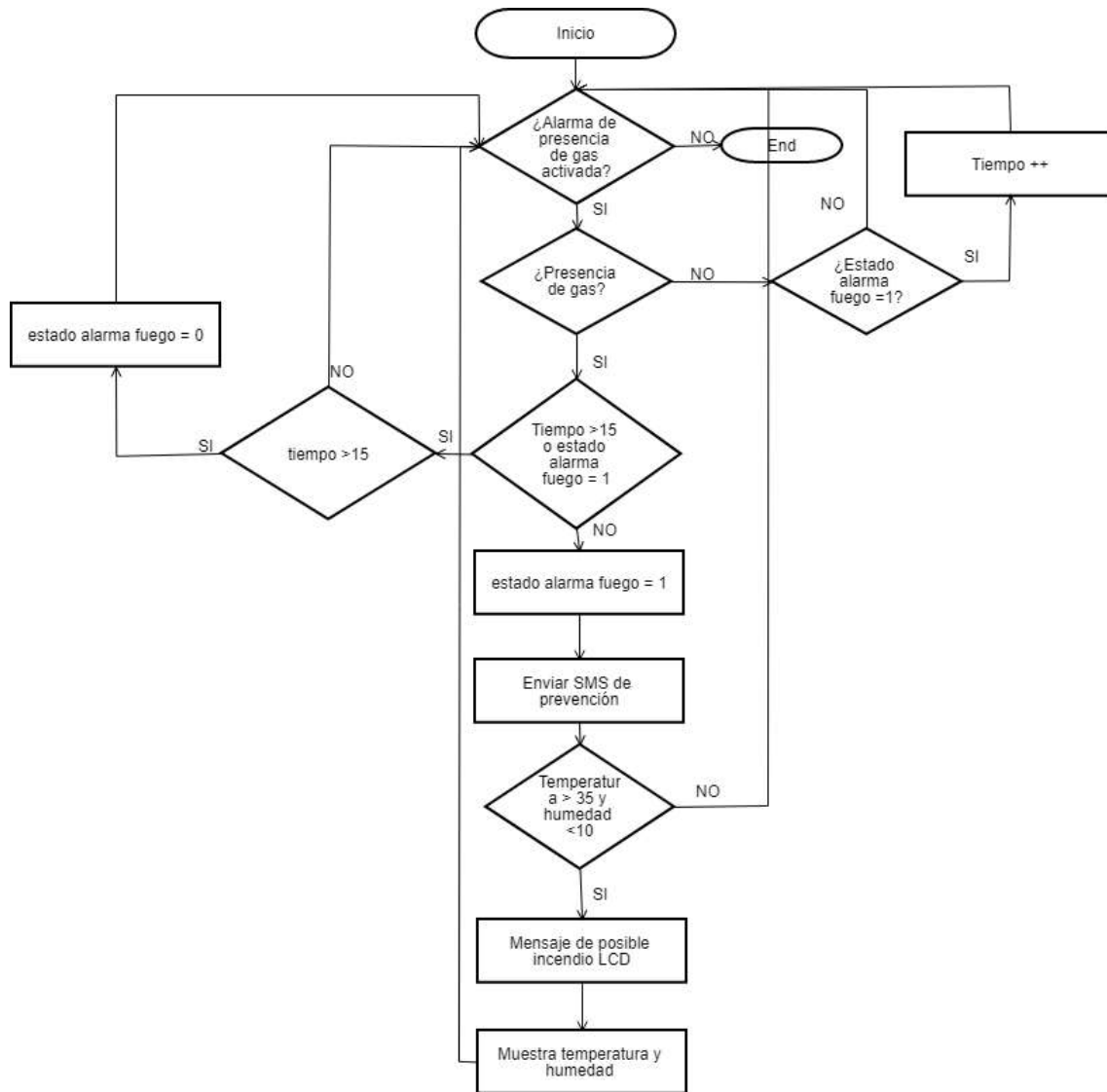


Figura 32. Diagrama de flujo sistema de detección de fuga de gas  
Fuente: Elaboración propia

Para activar o desactivar las alarmas de seguridad o detección de fuga de gas, se ingresa una opción por el teclado correspondiente a la acción que se va a realizar. En cualquiera de las opciones, se le solicita al usuario que ingrese una contraseña. Para ingresar una contraseña el usuario cuenta con tres intentos, si al tercer intento la contraseña sigue siendo incorrecta, el sistema se bloquea por 20 segundos. Cuando se le pide una contraseña al usuario y no ingresa nada o se demora en ingresar, el sistema espera veinte segundos, sino vuelve al menú principal.

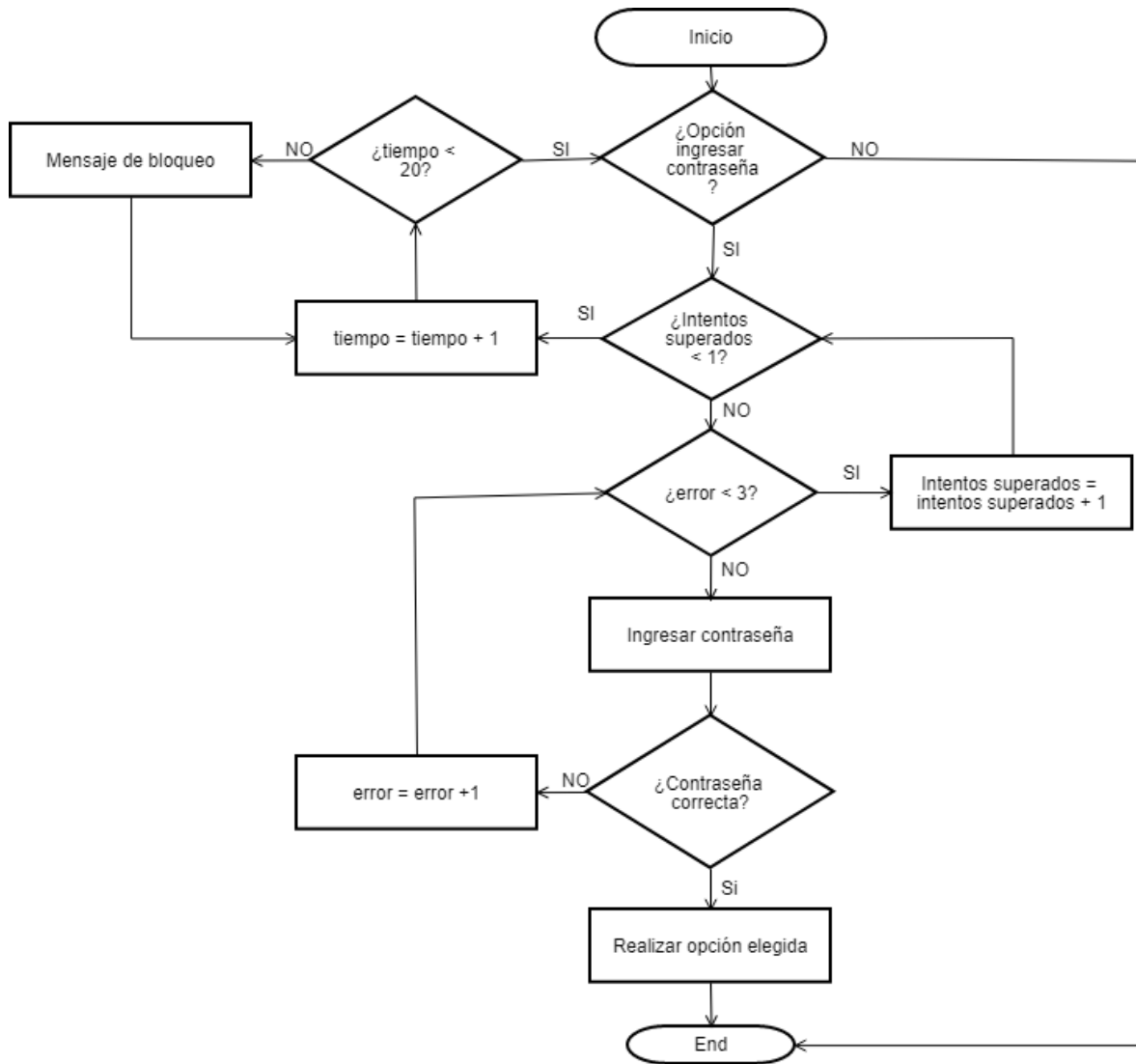


Figura 33. Diagrama de flujo ingreso de contraseña  
Fuente: Elaboración propia

Para activar el sistema de reconocimiento de voz se utiliza la opción #. Los comandos que se pueden decir son: “activar alarma”, “desactivar alarma”, “activar gas”, “desactivar gas”, “dime temperatura”, “dime humedad”, “encender luz” y “apagar luz”. Como método de prevención, si alguna alarma estuviera activada, solo se pueden usar los comandos de “desactivar” según la alarma correspondiente. Para las opciones de: “activar alarma”, “desactivar alarma”, “activar gas” y “desactivar gas”, se le pedirá al usuario que diga la palabra clave para que la opción se haga valida. Para ingresar una palabra clave el usuario cuenta con tres intentos, si no ingresan una palabra clave valida, el sistema regresa al menú principal.

El sistema de reconocimiento funciona con Wit.ai un software de reconocimiento de voz, que cuenta con el servicio “Speech to Text” (voz a texto). Es un servicio que necesita una conexión a Internet, ya que cuenta con servidores en la nube, ahí es donde transforma el audio en texto utilizando una red neuronal recurrente (RNN).

El sistema de reconocimiento de voz funciona cuando el usuario ingresa la opción #. El sistema le indica al usuario decir un comando, el cual es grabado por el micrófono en formato wav, lee el archivo de audio, realiza una cabecera de petición a Wit.ai con los permisos de acceso y envía una petición http con la cabecera y el archivo de audio a Wit.ai. El archivo de audio que se envió a Wit.ai regresa en formato Json (convertido de audio a texto en los servidores de Wit.ai), el archivo Json se convierte en texto y se guarda en un archivo. Después, se verifica si es que el archivo tiene texto o no, si tuviera texto se transforma de mayúsculas a minúsculas para que no haya errores y se compara el texto obtenido con alguna de las opciones, si no hay una opción válida se vuelve al menú principal. En caso no hubiera texto, se indica al usuario que no se detectó voz y regresa al menú principal.

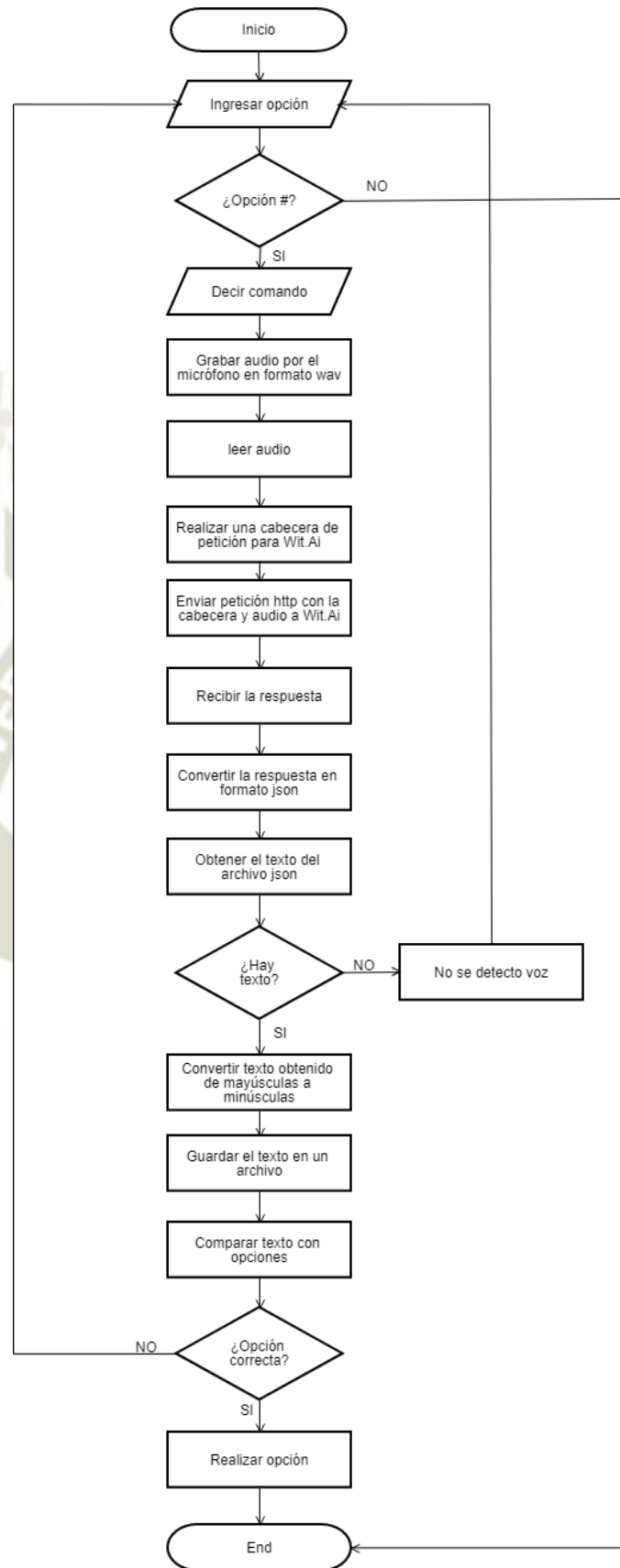


Figura 34. Diagrama de flujo sistema de reconocimiento de voz  
Fuente: Elaboración propia



El sistema de reconocimiento de voz necesita de una conexión a Internet, por ello el sistema revisa constantemente la conexión. Entonces, si el usuario quisiera ingresar a la opción de reconocimiento de voz y no hubiera conexión a Internet, el sistema no dejaría ingresar a esta opción controlando el error de conexión con los servicios de Wit.ai.

### **2.3.2. Diseño de la aplicación móvil**

La aplicación móvil funciona como un sistema de control de los datos que son necesario para que funcione la alarma. Los datos que se pueden gestionar por la aplicación móvil son: la contraseña del sistema que se ingresa por medio del panel de control (activar o desactivar); palabra clave para activar o desactivar el sistema por medio de comandos de voz y el correo electrónico al cual se envía la imagen si hay presencia de un intruso.

Por medio de la aplicación móvil se visualiza el estado de la alarma (activada o desactivada), se puede activar o desactivar los sistemas y permite observa la temperatura y humedad del hogar de forma remota.

Para que la aplicación haga cambios en el sistema se comunica con la base de datos en la nube Firebase, cuando se realiza algún cambio, la base de datos envía los nuevos datos al sistema. La base de datos Firebase sirve como un intermediario entre la comunicación de la aplicación móvil y el sistema de vigilancia y detección de fuga de gas.

En el diagrama de secuencia se observa las actividades que realizan el usuario y el administrador. Por ejemplo, cuando el usuario quiere realizar algún cambio o visualizar la información, tiene que validar sus datos como usuario e ingresar una contraseña. Después, ingresar a la opción que desea, si hacer un cambio o solo visualizar la información, los datos se obtienen o se cambian en la base de datos en la nube. Las tareas del administrador son

ingresar a la aplicación para verificar que esté funcionando óptimamente, que los datos se ingresen, modifiquen y se consulten correctamente y las opciones sean claras para el usuario.

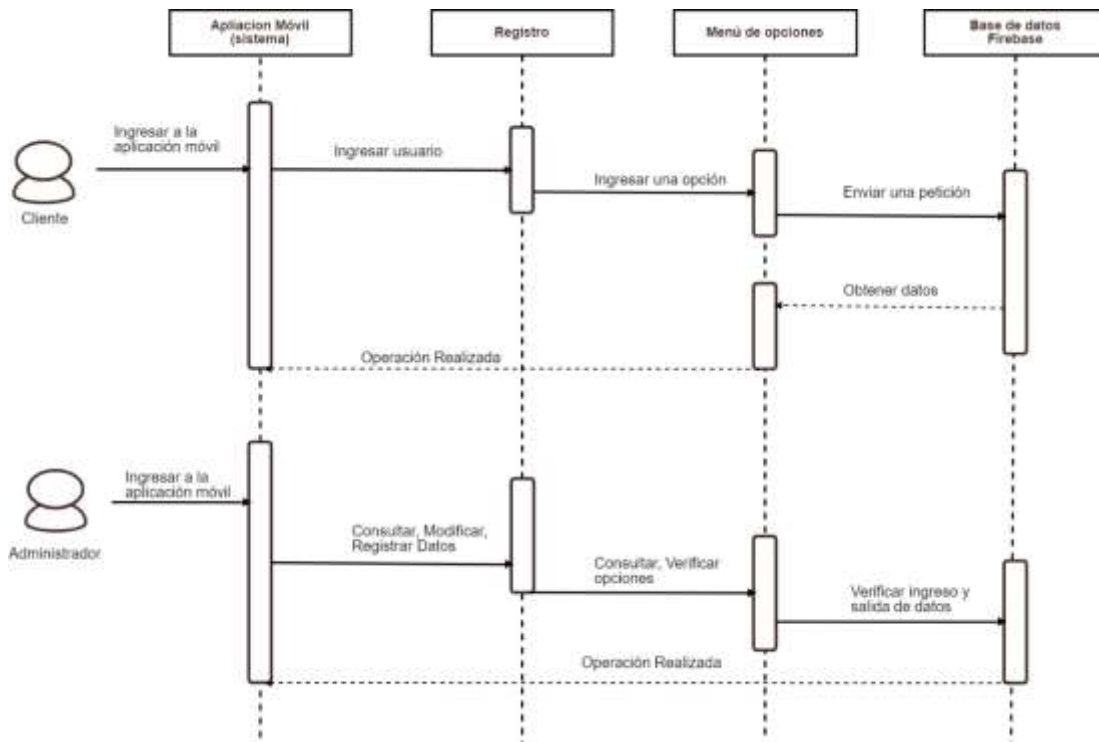


Figura 35. Diagrama de secuencia aplicación móvil  
Fuente: Elaboración propia

### 2.3.3. Diseño de la maqueta

El diseño de la maqueta (prototipo) representa el hogar de un usuario, es un departamento de un piso para una sola familia (unifamiliar), ya que últimamente las personas prefieren vivir en departamento y cuenta con una sola puerta principal. Cuenta con cinco ambientes: sala-comedor, cocina, dormitorio principal y dos dormitorios. También, cuenta con tres baños, uno para invitados y dos para los cuartos.

En el prototipo se instalarán los sensores, actuadores y periféricos que componen el sistema de vigilancia y detección de fuga de gas. En el área de la sala-comedor por ser la que colinda con la puerta principal se pondrán los dos sensores PIR que detectan el movimiento por medio de infrarrojos, así si detectan la presencia de un intruso suena una alarma (buzzer) que se instala en la cocina para que se escuche en todo el departamento. En esa área, también se instalará una cámara con una visión cercana a la puerta y a la mayor parte del espacio. En

la sala, se pondrá un foco de 3 watts que se activa a partir de las 6:00 pm para generar un ambiente adecuado para tomar las imágenes de noche.

En una parte cercana a la puerta se pondrá la pantalla LCD, el teclado matricial y el micrófono, para facilitar el ingreso y la interacción con el usuario. Como medida de seguridad al detectar una presencia, el sistema le indica al usuario que las puertas se cerraran en tres segundos, para eso se implementa un servo para cerrar la puerta, cuando se termine el tiempo de 30 segundos sin presencia se abre el pestillo de la puerta o si se desconecta la alarma ocurrirá el mismo proceso.

En la cocina se instala dos sensores uno de detección de presencia de gas y otro de temperatura y humedad, para alertar al usuario de un posible incendio o fuga de gas.

El sistema de vigilancia y detección de fuga de gas, muestra su funcionamiento en la maqueta demostrando que es un prototipo funcional y se puede implementar en un ambiente real, brindando seguridad, control y gestión de un sistema para el hogar.

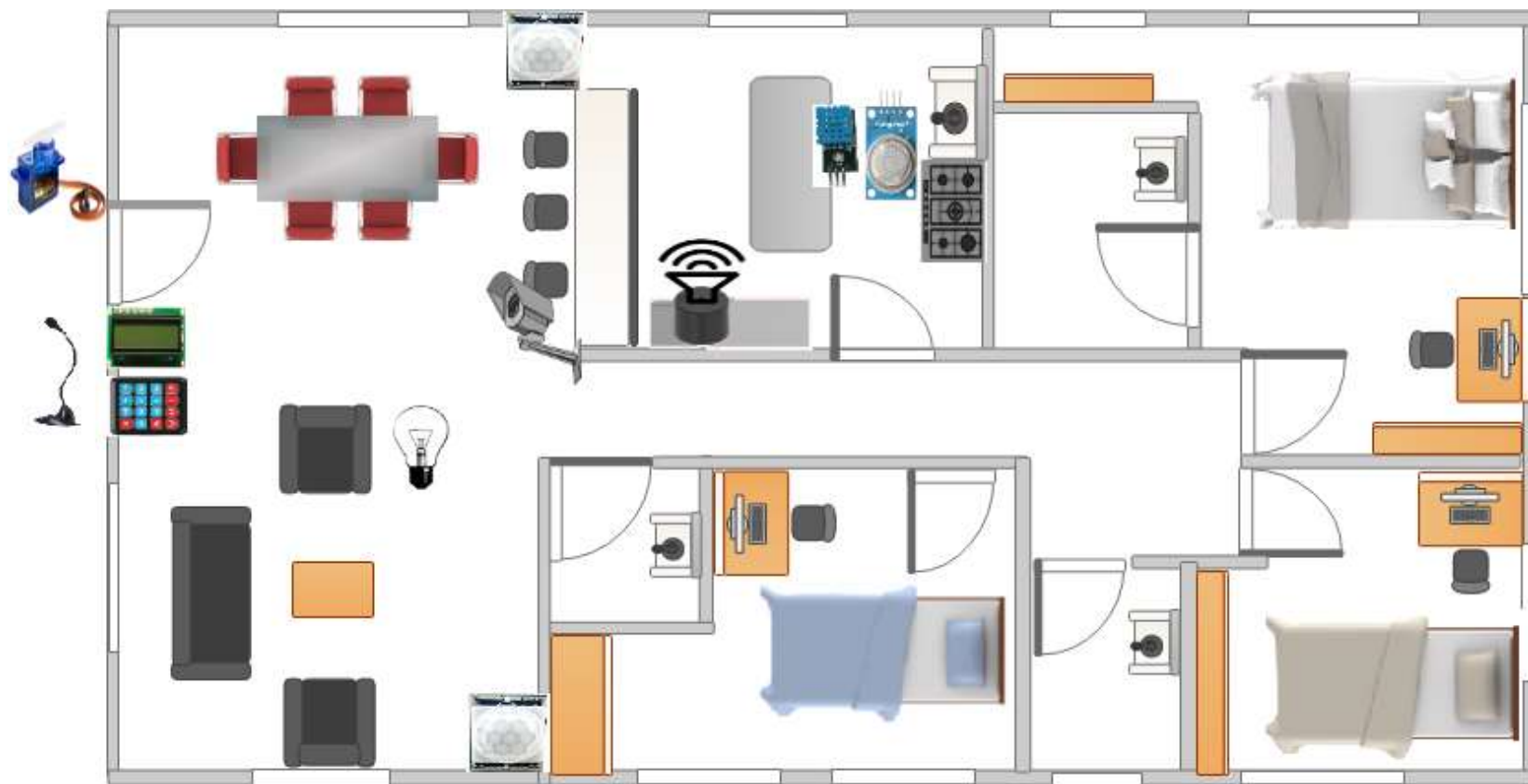


Figura 36. Representación de la maqueta  
Fuente: Elaboración propia

## 2.4. Documentación técnica de programación

La codificación del proyecto se divide en dos partes: codificación del sistema de vigilancia y detección de fuga de gas, y codificación de la aplicación móvil.

### 2.4.1. Documentación del sistema de vigilancia y detección de fuga de gas

Para desarrollar el sistema de vigilancia y detección de fuga de gas se programó una parte en el microprocesador Raspberry Pi y otra parte en el microcontrolador Arduino. En el Raspberry Pi se utilizó el IDE (entorno de desarrollo integrado) Thonny de Python y en Arduino en su IDE, en un lenguaje parecido a C++.

#### 2.4.1.1. Documentación del sistema en Raspberry Pi 3 Modelo B

En el microprocesador Raspberry Pi se programa la mayor parte del sistema de vigilancia y detección de fuga de gas. Es donde se lleva a cabo el funcionamiento de los sensores, actuadores y periféricos.

Para llevar a cabo la programación de todas las funciones del sistema se necesitan módulos. Algunos de los módulos son para utilizar sensores y actuadores (pantalla LCD y sensor Dht11). Otros módulos son necesarios para tener control del sistema y del tiempo. Y algunos otros son para conectar con la base de datos Firebase o para trabajar con hilos y sockets.

```
import I2C_LCD_driver #modulo para LCD con I2C
import RPi.GPIO as GPIO #modulo para utilizar los pines
import Adafruit_DHT #modulo para sensor de temperatura y humedad dht11
import time #modulo de funciones con relacion al tiempo
import os #modulo funciones dependientes del sistema operativo (manipular rutas, abrir archivos)
import subprocess #generar nuevos subprocesos
import io #modulo para gestionar las interfaces de python
import sys #modulo que proporciona acceso a algunas variables utilizadas o mantenidas por el interprete
import signal #modulo para usar manejadores de señal en python
import serial #modulo para encapsular el acceso al puerto serial
import os.path as path #modulo para archivos de nombres largos (rutas largas)
import firebase_admin #modulo de conexion con firebase
from firebase_admin import credentials #credenciales de firebase
from firebase_admin import db #base de datos de firebase
from threading import Thread #modulo que contruye subprocesos de nivel superior (hilos)
import socket #modulo que proporciona acceso a la interfaz de socket
```

Figura 37. Módulos necesarios para programación en Raspberry Pi

Fuente: Elaboración propia

Al trabajar con la base de datos Firebase se necesita de un archivo Json de credenciales que se descarga de la página del proyecto de Firebase, el cual permite la

conexión, facilita la obtención y envío de datos. La base de datos trabaja con una estructura de árbol (padre-hijos). Para obtener los datos se especifica cuáles son los hijos que se van a utilizar. Se declara las variables que se van a utilizar con referencia al dato de la base de datos y a su raíz principal.

```
#firebase
#archivo de credenciales firebase
cred = credentials.Certificate('/home/pi/Desktop/cred.json')
#archivo para iniciar database firebase
firebase_admin.initialize_app(cred, { })
#variables en la bd arbol de referencia
REF_HOME = 'home'
REF_CORREO = 'correo'
REF_PALCLAVE = 'palclave'
REF_ALARMA = 'alarma'
REF_ALAR_IN = 'incendio'
REF_ESTADOTEMP = 'estado_temp'
REF_HUMEDAD = 'humedad'
REF_TEMPERATURA = 'temperatura'
REF_CONTRASENA = 'contrasena'
#variables de referencia de la bd
refHome = db.reference(REF_HOME) #raiz principal de la bd
refcorreo = refHome.child(REF_CORREO) #hijo de la raiz principal
refpalclave = refHome.child(REF_PALCLAVE)#hijo de la raiz principal
refalarma = refHome.child(REF_ALARMA)#hijo de la raiz principal
refalar in = refHome.child(REF_ALAR_IN)#hijo de la raiz principal
refestadoTemp = refHome.child(REF_ESTADOTEMP)#hijo de la raiz principal
refhumedad = refHome.child(REF_HUMEDAD)#hijo de la raiz principal
reftemperatura = refHome.child(REF_TEMPERATURA)#hijo de la raiz principal
refcontrasena = refHome.child(REF_CONTRASENA)#hijo de la raiz principal
```

Figura 38. Credenciales y declaración variables Firebase

Fuente: Elaboración propia

Los sensores utilizados en el proyecto son: dos sensores PIR, un sensor de temperatura Dht11 y un sensor de gas. Como actuador está el buzzer. Para trabajar con ellos se declaró el tablero de pines de Raspberry Pi como GPIO.BOARD para trabajar con su número de pin. Los pines que usan los sensores son declarados como entrada o salida según la funcionalidad que necesitan.

```
#Variables de sensores
pir_sensor = 11 #sensor pir1 pin11
pir2_sensor = 12 #sensor pir2 pin12
buzzer = 7 #buzzer pin7
tipo_sensor temp = 11 #tipo del sensor DHT11
pin_temp = 14 #puerto GPIO14 pin8
gas = 13 #puerto pin13 GPIO27
GPIO.setmode(GPIO.BOARD) #para usar los pines por su número (pin1)
GPIO.setwarnings(False) #desactivar advertencias
GPIO.setup(buzzer,GPIO.OUT)#declara el pin que usa el buzzer como una salida
GPIO.setup(pir_sensor,GPIO.IN)#declara el pin que usa el pir como una entrada
GPIO.setup(pir2_sensor,GPIO.IN)#declara el pin que usa el pir como una entrada
GPIO.setup(gas,GPIO.IN)#declara el pin que usa el gas como una entrada
mylcd = I2C_LCD_driver.lcd() #objeto de la clase LCD
```

Figura 39. Declaración de puertos GPIO Raspberry Pi

Fuente: Elaboración propia

La configuración del teclado matricial 4x4 es diferente. Se declaró la variable `MATRIX` como matriz bidimensional con los valores del teclado en físico. Después, se definió los pines que utiliza el teclado, cuatro pines como filas y cuatro pines como columnas. Para finalizar se declara las columnas como salidas y las filas como entradas, ambas se nombran en estado encendido. Funciona cuando se presiona una fila y columna y hay un cambio de estado que hace referencia al carácter que se encuentra en la intercepción.

```
#parametros teclado
MATRIX = [[1,2,3,'A'], #matriz bidimensional
          [4,5,6,'B'],
          [7,8,9,'C'],
          ['*',0,'#','D']]

ROW = [29,31,33,35] #pines de ingreso como filas
COL = [32,36,38,40] #pines de ingreso como columnas

for j in range(4):
    GPIO.setup(COL[j],GPIO.OUT) #nombra a los puertos de la matriz COL, GPIO como output (salidas)
    GPIO.output(COL[j],1) #nombra a los puertos en estado high

for i in range(4):
    #nombra a los puertos de la matriz como entradas input
    GPIO.setup(ROW[i],GPIO.IN,pull_up_down = GPIO.PUD_UP)
    #entradas en alto(1), pull up resistencia
```

Figura 40. Funciones teclado Matricial 4X4  
Fuente: Elaboración propia

Algunos componentes como el módulo Relay y el servomotor se implementan en Arduino porque trabajan con corriente de 5v. Arduino se conecta con Raspberry Pi por un puerto USB por medio de una conexión serial. Trabajar la conexión como un puerto serial permite enviar y recibir datos digitales, utilizando dos conectores para la comunicación RX (recepción) y TX (trasmisión). Al momento de conectar Arduino con Raspberry Pi, Arduino se identifica como un dispositivo de comunicación USB del subtipo modelo de control abstracto (ACM), y la comunicación configurada entre ambos dispositivos tiene una velocidad de 9600 bps. La función `mensaje_Arduino` sirve para enviar datos de Raspberry Pi a Arduino, la cual envía una letra a Arduino. Según la letra enviada, Arduino realiza una acción para la cual fue programado.

```

arduino = serial.Serial('/dev/ttyACM0',9600) #Puerto donde se encuentra el Arduino

def mensaje_arduino(letra_arduino):
    arduino.write(letra_arduino) #Mandar un comando hacia Arduino
    time.sleep(0.2)
    
```

Figura 41. Conexión con Arduino y función enviar mensaje  
Fuente: Elaboración propia

La función contraseña\_alarma se utiliza cuando el usuario quiere activar o desactivar el sistema de vigilancia o el sistema de detección de fuga de gas por medio del teclado. Al presionar cualquiera de esas opciones se le pide al usuario ingresar una contraseña. Si, el usuario al ingresar cualquier carácter se demora más de cinco segundos, el sistema regresa al menú principal. El usuario tiene tres opciones de ingresar la contraseña correcta, sino el sistema se bloquea por 20 segundos. La contraseña se obtiene de la base de datos cuando hay conexión a Internet, en caso no hubiera conexión se obtiene de un archivo de texto cifrado. Al obtener la contraseña de la base de datos y del archivo de texto se descodifica con el algoritmo base 64.

```

def contraseña_alarma(estado_internet):
    error = 0
    list1 = ['1','1','1','1']
    global cont_superados #contador de ingresos erroneos de contraseña de tres oportunidades
    global start_time_superados #bloqueo por 20 segundos
    start_time_contra = time.time() #tiempo en ingresar una contraseña
    if cont_superados < 1 :
        while error < 3: #3 oportunidades de ingresar una contraseña
            mylcd.lcd_display_string("Ingrese contraseña ", 1)
            contador = 0
            posicion = 5
            while contador < 4 : #ingreso de 4 digitos por el teclado
                for j in range (4):
                    GPIO.output(COL[j],0)
                    fin_time_contra = time.time() - start_time_contra #tiempo en ingresar una contraseña
                    if int(fin_time_contra) > 5: #tiempo de 5 segundos para ingresar
                        return 0
                for i in range (4):
                    if GPIO.input (ROW[i])== 0:
                        start_time_contra = time.time()#nuevo contador de tiempo
                        opc = MATRIX[i][j] #opc obtenida por el teclado
                        mylist[contador]= str(opc) #guarda en un arreglo
                        contador = contador + 1 #contador de espacio
                        mylcd.lcd_display_string(" ", 2,posicion)
                        posicion = posicion + 1
                        while (GPIO.input(ROW[i])== 0):
                            pass
                        GPIO.output(COL[j],1)
                        opc = ''
            c = 0
            if estado_internet == 1: #verificar conexion
                contraseña = refcontrasena.get() #obtener contraseña de la bd
                list1 = list(contrasena) #contrasena de la bd se guarda en un arreglo
                archivo = open("/home/pi/1.txt","w")
                archivo.write(contrasena) #escribe en un archivo la contraseña
                archivo.close()
    
```

Figura 42. Función contraseña\_alarma  
Fuente: Elaboración propia



El sistema de seguridad tiene el propósito de advertir al usuario el ingreso de una persona no grata al hogar. Cuando los sensores PIR detectan una presencia (función alarma\_activada), se prende el foco dependiendo de la hora (de 6:00 pm a 4:00 am), se toma una imagen y se envía una letra a Arduino, para que se envíe un mensaje de texto. Para finalizar se realiza el control de la puerta (cerrarla).

```
def alarma_activada(estado_internet):
    print("Intruso detectado toma foto")
    time_foco = time.strftime("%PM")
    #prende el foco en horas de la noche y madrugada
    if ((time_foco >= '1800' and time_foco <= '2459') or (time_foco >= '0000' and time_foco <= '0400')):
        mensaje_arduino('F') #mensaje para Arduino
        print("Se prendió el foco")
    tomar_foto()
    print("Intruso detectado ENVIA SMS")
    mensaje_arduino('H') #mensaje para Arduino
    enviar_foto(estado_internet)
    control_puerta()
    time.sleep(2)
    mylcd.lcd_clear()
```

Figura 43. Función activar\_alarma  
Fuente: Elaboración propia

El sistema de reconocimiento de voz se activa por la opción “#” del teclado. El sistema de reconocimiento de voz funciona con Wit.ai, al cual se envía el audio captado por el micrófono y como respuesta devuelve un archivo Json, con la palabra o frase que se dijo. La función escuchar llama a un subproceso que abre un archivo en bash. Al ejecutar todos los archivos, regresa el valor obtenido en texto, del cual solo se considera el texto antes de la siguiente línea. El sistema muestra en pantalla lo que entendió y realiza la opción ingresada por el usuario.

```
def escuchar():
    #Grabar audio del microfono en wav
    #y enviarlo a Google que devolviera el texto en una variable
    #creacion y gestion del nuevo proceso es manejado por la clase Popen, con cadenas unicode
    #el resultado se almacena en la variable envio (tuberia debe ser abierta)
    envio = subprocess.Popen(["./grabar.sh"], shell=True, universal_newlines=True, stdout=subprocess.PIPE)
    #[0] acceder al stdoutdata de la variable envio
    texto = envio.communicate()[0]
    #retorna una copia de cadena antes de una nueva línea (proxima línea)
    texto = texto.strip('\n')
    mylcd.lcd_display_string(texto, 2)
    time.sleep(1)
    return texto
```

Figura 44. Función escuchar (reconocimiento de voz)  
Fuente: Elaboración propia

Bash es un programa informático que consiste en interpretar órdenes en un lenguaje en consola. El archivo bash para reconocimiento de voz, graba un audio por medio del micrófono en formato wav. De ahí, se ejecuta un script Python para la conexión con Wit.ai.



constante revisión de los estados del sensor PIR y del sensor de gas. En caso alguno sea positivo se realizan las acciones descritas anteriormente.

```
try:
    while(True):
        mylcd.lcd_display_string("Ingresar una opcion: ", 1)#ingresar una opcion por teclado
        for j in range (4):
            GPIO.output(COL[j],0)
            for i in range (4):
                if GPIO.input(ROW[i])==0:
                    opcion = MATRIX[i][j]
                    print (MATRIX[i][j])
                    while (GPIO.input(ROW[i])!= 0):
                        pass
                    GPIO.output(COL[j],1)

            estado_pir = GPIO.input(pir_sensor)
            estado_pir2 = GPIO.input(pir2_sensor)
            if estado_alarma == '1' or estado_alarma_fuego == '1': #verificar alarma activada
                print ("Alarma estado activado")
            if (estado_pir == 1 or estado_pir2 == 1) and estado_alarma_fuego == '0' and estado_activado == 0: #verificar presencia del intruso
                print("INTRUSO")
                mylcd.lcd_clear()
                mylcd.lcd_display_string("Alarma Activada", 1)
                time.sleep(8.5)
                GPIO.output(buzzer,True)
                if estado_activado == 0:
                    alarma_activada(estado_internet)
                    estado_activado = 1
            elif ((estado_pir == 0 or estado_pir2 == 0) or (estado_pir == 1 or estado_pir2 == 1)) and estado_activado == 1:
                GPIO.output(buzzer,False)
                presencia_gas()#funcion de presencia de gas
                if start_time == 0:
                    start_time = time.time()#tiempo de inicio de deteccion
                    print("start.time(): %f " % (start_time))
                else:
                    fin_time = time.time() - start_time
                    print("fin_time: %f " % (fin_time))
                    print("fin_time: %f " % (int(fin_time)))
                    if int(fin_time) > 30 : #Tiempo entre mensajes si hay otra presencia
```

Figura 47. Función principal Main  
Fuente: Elaboración propia

Cuando se ingresan alguna de las siguientes opciones por medio del teclado A, B, C, D, \*, 0 y #, se realizan las acciones para las cuales fueron programadas. Ejemplo: La opción A, sirve para activar la alarma para que funcione esta opción el sistema debe de estar desactivado y la luz del foco apagada. Al ingresar a esta opción se visualizara por pantalla a que opción se ingresó, pedirá que se ingrese la contraseña, al ser correcta, se activa la alarma. También, actualiza en la base de datos el estado de la alarma para que se visualice en la aplicación móvil.

```
#Opciones para el ingreso del teclado
if opcion == 'A':
    mylcd.lcd_clear()
    if estado_alarma == '0' and estado_alarma_fuego == '0' and estado_luz == 0:
        mylcd.lcd_display_string("Opcion Activar ", 1)
        mylcd.lcd_display_string(" Alarma", 2)
        time.sleep(2)
        mylcd.lcd_clear()
        if contraseña_alarma(estado_internet):
            estado_alarma = activar_alarma()
            archivo_alarma(estado_alarma)
            if estado_internet == 1:
                refalarma.set(True)
            estadoapp = 1
```

Figura 48. Opción de ingreso por teclado  
Fuente: Elaboración propia

#### 2.4.1.2. Documentación del sistema en Arduino Uno R3

En Arduino se programó los actuadores que trabajan con 5v y el módulo GSM. En la parte de setup se configuro y se declaró las variables: led, foco y puerta. La variable puerta hace referencia al servo que cierra la puerta principal. El led y foco, se declararon como salidas (outputs). Para utilizar el servomotor se utilizó una clase de tipo Servo, se declaró con el número de pin que utiliza y el grado de rotación que tendrá al comienzo. Para el módulo GSM se utilizó una librería descargada de internet propia del módulo (SIM900.h y sms.h). Se utilizó la clase SMSGSM, para el envío de mensajes de texto. El módulo GSM y el puerto serial se inicializaron con una velocidad de 9600bps, la misma velocidad del puerto serial de la Raspberry Pi.

```
void setup()
{
  Serial.begin(9600); //Iniciando el puerto serie a 9600 baudios
  pinMode(led,OUTPUT); //Iniciando la variable led como salida
  pinMode(foco,OUTPUT); //Iniciando la variable foco como salida
  digitalWrite(foco, HIGH);

  if (gsm.begin(9600)) //comprobacion de iniciacion del modulo GSM
  {
    Serial.println("\nstatus=READY");
    started=true;
    digitalWrite(led,HIGH); //verificacion de estado GSM
    delay(1000);
    digitalWrite(led,LOW);
    delay(1000);
  }
  else
    Serial.println("\nstatus=IDLE");

  puerta.attach(10); //Iniciando el puertol0 como objeto servo
  puerta.write(180); //Se la da al servo la posicion de 180°
};
```

Figura 49. Método setup Arduino  
Fuente: Elaboración propia

La parte de loop se ejecuta cíclicamente. Entonces, el programa desarrollado en Arduino siempre está atento para recibir el comando que envía Raspberry Pi. Por medio de un comando switch case se ingresa la opción que envía Raspberry Pi por el puerto serial. Las opciones son: enviar un mensaje de texto cuando se activó la alarma, prender el foco, control del servo (puerta), prender y apagar el foco, envío de un mensaje de texto de advertencia por

presencia de gas y envió de un mensaje de texto de auxilio a algún familiar. El servo de la puerta se controla por medio de grados de rotación para cerrar o abrir.

```
void loop()
{
  if (Serial.available()) {
    Serial.print("Ingrese un valor");
    char c = Serial.read();
    switch (c){
      case ('H'):
        Serial.print("Estado Activo");
        iniciar(); //Funcion para enviar el sms |
        digitalWrite(led, HIGH);
        delay(1000);
        digitalWrite(led, LOW);
        break;

      case ('F'): //opcion para prender el foco unos segundos
        Serial.print("Opcion F");
        digitalWrite(foco, LOW);
        delay(9000);
        digitalWrite(foco, HIGH);
        break;

      case ('S'): //Control de servos cerrar puerta
        Serial.print("Opcion S");
        //puerta
        // Desplazamos a la posición 180°
        puerta.write(90);
        // Esperamos 1 segundo
        delay(1000);
        break;
    }
  }
}
```

Figura 50. Método loop Arduino  
Fuente: Elaboración propia

Para enviar un mensaje de texto de advertencia se desarrolló la función iniciar (). Que consiste en crear un objeto de la clase SMSGSM y llamar al método SendSMS de la clase sms. El método necesita como parámetros el número telefónico y el mensaje, para enviar un mensaje de texto.

```
void iniciar()
{
  if(started)
  {
    //el objeto sms llama a la funcion SendSMS parametros #telefonico y mensaje
    if (sms.SendSMS("+51959191932", "INTRUSO DETECTADO"))
    {
      Serial.println("\nSMS sent OK.");
    }
    else
    {
      Serial.println("\nError sending SMS.");
    }
  }
}
```

Figura 51. Método iniciar (enviar SMS)  
Fuente: Elaboración propia

### 2.4.2. Documentación aplicación móvil

Se utilizó Android Studio para realizar la aplicación móvil. Se trabajó con el sistema operativo Android por ser el que más se usa. Para programar en Android Studio se utilizó como lenguaje de programación Java y el meta lenguaje Xml.

Para conectar el sistema desarrollado en Raspberry Pi con la aplicación móvil se utilizó la base de datos en la nube Firebase. Para conectar la base de datos con Android Studio se agregaron al grandle de la aplicación las dependencias de googlefireabasecore, googlefirebasedatabase, googlefirebaseauth (autenticación) y googlefirebasemessaging.

```
dependencies {  
  
    implementation 'com.google.firebase:firebase-core:16.0.6'  
    implementation 'com.google.firebase:firebase-database:16.0.5'  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'com.android.support:appcompat-v7:27.1.1'  
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'  
    implementation 'com.google.firebase:firebase-auth:16.0.5'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.2'  
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
    implementation 'com.android.support:support-v4:27.1.1'  
    implementation 'com.android.support:design:27.1.1'  
    implementation 'com.android.support:cardview-v7:27.1.1'  
    //noinspection GradleCompatible  
    implementation 'com.google.firebase:firebase-messaging:17.3.4'  
    implementation 'com.google.android.gms:play-services-location:16.0.0'  
}  
apply plugin: 'com.google.gms.google-services'
```

Figura 52. Dependencias para conectar con Firebase  
Fuente: Elaboración propia

La aplicación móvil cuenta con un Login (inicio de sesión) que valida el ingreso de usuarios que están suscritos. Para crear una nueva cuenta se ingresa a la parte de registrar usuarios. El Login funciona con la base de datos en tiempo real (Firebase), en la cual se crean los usuarios (correo y contraseña). Firebase autentifica al usuario, le permite o deniega el ingreso. La autenticación funciona con una instancia de autenticación de usuario (FirebaseAuth), que verifica si antes hubo una sesión y permite el ingreso de otra.

```
//obtener instancia de autentificación de Firebase
auth = FirebaseAuth.getInstance();
getSupportActionBar().hide(); //ocultar Action Bar

if (auth.getCurrentUser() != null) { //verifica si hay un usuario actual
    startActivity(new Intent( packageContext: LoginActivity.this, LoginActivity.class));
    finish();
}
}
```

Figura 53. Obtener una instancia de FirebaseAuth  
Fuente: Elaboración propia

La autentificación se realiza por el correo y contraseña del usuario. Por medio de la función `signInWithEmailAndPasword` de la clase `auth`, propia de `FirebaseAuth`. Al ser un usuario correcto, le permite el ingreso al Menú principal. Si la información fuera errónea, le indica que el correo o la contraseña que se está ingresando no es el correcto, o que la contraseña tiene menos de 6 caracteres (indica un error). Al ingresar los datos de correo y contraseña, antes de enviarlos a la base de datos se verifica que los campos estén llenos, sino aparece un mensaje advirtiendo que faltan datos para el inicio de sesión.

```
//Authenticate user
auth.signInWithEmailAndPasword(email, password)
    .addOnCompleteListener { activity: LoginActivity.this, (task) => {
        //Si falla el inicio de sesión, muestra un mensaje al usuario. Si el inicio de
        //sesión se realiza correctamente, se notificará al agente
        //y se manejará la lógica para el inicio sesión.
        progressBar.setVisibility(View.GONE); //circular de carga desaparece
        if (task.isSuccessful()) {
            //ocurre un error en la validación del usuario
            if (password.length() < 6) {
                inputPassword.setError("Contraseña corta, ingrese minimo 6 caracteres"); //contraseña menor de 6 caracteres
            } else {
                //falla de autentificación correo o contraseña incorrecta
                Toast.makeText( context: LoginActivity.this, "Error de Autentificación, revise contraseña y correo elec...", Toast.LENGTH_LONG).show();
            }
        } else {
            //correcta validación
            Intent intent = new Intent( packageContext: LoginActivity.this, MainActivity.class); // Ingresar a MAIN
            startActivity(intent);
            finish();
        }
    }
}
```

Figura 54. Método de autentificación del usuario  
Fuente: Elaboración propia

El registro de un nuevo usuario se hace por `FirebaseAuth`. Se envía a `Firebase` correo y contraseña del nuevo usuario, si los datos son correctos se crea un nuevo usuario. En caso hubiera un usuario con el mismo correo no es posible crearlo nuevamente, se muestra un error de autentificación. Al crear un nuevo usuario se ingresa directamente a al menú principal.

```
//crear usuario
auth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener( activity: RegisterActivity.this, (task) -> {
        Toast.makeText( context: RegisterActivity.this, text: "createUserWithEmail:onComplete:" + task.isSuccessful()
        progressBar.setVisibility(View.GONE);
        //Si falla el inicio de sesión, muestre un mensaje al usuario.
        // Si el inicio de sesión se realiza correctamente, se notificará al oyente
        // y se manejará la lógica de inicio sesión en el escucha.
        if (!task.isSuccessful()) {
            Toast.makeText( context: RegisterActivity.this, text: "Error de Autenticación." + task.getException(),
            Toast.LENGTH_SHORT).show();
        } else {
            startActivity(new Intent( packageContext: RegisterActivity.this, MainActivity.class));
            finish();
        }
    });
```

Figura 55. Método para crear un nuevo usuario  
Fuente: Elaboración propia

En el menú principal se visualiza el estado del sistema de vigilancia y prevención. El sistema se puede activar o desactivar de forma remota. Para que funcione, se conecta con la base de datos por medio de FirebaseDatabase.

Para activar o desactivar el sistema de seguridad, se usa un Toggle Button, que tiene la función de un conmutador encendido/apagado. Cuando se presiona el botón, se da un cambio de estado, este dato se envía a la base de datos para que la variable este actualizada. La aplicación revisa constantemente el estado de la alarma en la base de datos, al darse un cambio, el estado del botón se modifica. Cuando el estado este en “true” el botón mostrara para que se apague el sistema, en caso este en “false” la opción que se muestra es que se encienda el sistema.

```
private void controlLED(final DatabaseReference refLed, final ToggleButton toggle_btn) {
    toggle_btn.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton compoundButton, boolean isChecked) {
            refalarma.setValue(isChecked); //cambia el valor en la bd de refalarma
        }
    });
    refalarma.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            Boolean estado_led = (Boolean) dataSnapshot.getValue(); //obtiene el valor de refalarma de la bd
            toggle_btn.setChecked(estado_led); //cambia el estado del marcador
            if (estado_led) { //segun el estado cambia lo que dice el boton
                toggle_btn.setTextOn("APAGAR");
            } else {
                toggle_btn.setTextOff("ENCENDER");
            }
        }
    });
}
```

Figura 56. Método para encender o apagar el sistema  
Fuente: Elaboración propia



Se cuenta con un TextView en el cual se visualiza el estado del sistema de vigilancia y detección, el valor se obtiene de la base de datos.

```
private void estadoPulsador(final DatabaseReference refPulsador_a, final TextView textEstadoPulsador) {
    refPulsador_a.addValueEventListener(new ValueEventListener() { //obtener estado de la base de datos
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            Boolean estado_pulsador = (Boolean) dataSnapshot.getValue(); //obtiene el valor de refalarma de la bd
            if (estado_pulsador==Boolean.TRUE){ //segun el valor del estado, muestra el Textview
                textEstadoPulsador.setText("Activado");
            }else {
                textEstadoPulsador.setText("Desactivado");
            }
        }
        @Override
        public void onCancelled(DatabaseError databaseError) { } //ocurre un error en la bd
    });
}
```

Figura 57. Método para visualizar el estado del sistema de vigilancia y detección  
Fuente: Elaboración propia

En el menú principal se implementó un botón de cerrar sesión, funciona con FirebaseAuth, el cual permite cerrar la sesión de forma segura. También, se aumentó la función del botón back del celular que al presionarlo dos veces se cierra la sesión de manera exitosa.

```
//metodo cerrar sesión
public void signOut() {
    auth.signOut();
    Intent i = new Intent(getApplicationContext(), LoginActivity.class);
    startActivity(i);
    finish();
}
```

Figura 58. Método para cerrar sesión  
Fuente: Elaboración propia

La aplicación cuenta con una función que permite cambiar la contraseña con la que se activa y desactiva el sistema de vigilancia y detección de fuga de gas por medio del panel de control. En esta opción se tiene dos edittext en los cuales se tiene que ingresar la contraseña, se verifica que las contraseñas sean iguales y que tengan los caracteres permitidos (0-9 y a, b, c, d), si cumplen con ambas condiciones, la contraseña se codifica por el algoritmo base 64, y codificada se envía a la base de datos en la nube.

```

bl.setOnClickListener((v) -> {
    String strPass1 = editText1.getText().toString();
    String strPass2 = editText2.getText().toString();
    if (strPass1.equals(strPass2)) {
        if (validarCampo(strPass1)) { //llamar al metodo para validar
            byte[] base64Data = strPass2.getBytes();
            String base64Str = "";
            try {
                base64Str = base64.encryptBASE64(base64Data);
            } catch (Exception e) {
                e.printStackTrace();
            }
            String strPass3 = base64Str.replaceAll( regex: "\\n", replacement: "");
            refContraseña.setValue(strPass3); //establecer el valor en la bd
            editText2.setText(base64Str);
            Toast.makeText(getApplicationContext(),
                text: "Cambio de contraseña exitoso...", Toast.LENGTH_SHORT).show();
            Intent i = new Intent(getApplicationContext(), MainActivity.class);
            startActivity(i);
            finish();
        }
    } else {
        Toast.makeText(getApplicationContext(), text: "Las contraseñas no son iguales!", Toast.LENGTH_SHORT).show();
    }
});

```

Figura 59. Función del botón “ok” para cambiar la contraseña  
Fuente: Elaboración propia

La aplicación permite ver en tiempo real la temperatura y humedad del hogar. Se obtiene los valores de la base de datos, por medio de una clase predicción, que tiene los métodos get para obtener los valores de la base de datos. Una vez que se obtienen los datos se muestran por medio de un TextView.

```

lblTemperatura = (TextView) findViewById(R.id.textView5);
lblHumedad = (TextView) findViewById(R.id.textView6);
eventListener = new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        Predicción pred = dataSnapshot.getValue(Predicción.class); //obtiene valor de la base de datos
        lblTemperatura.setText(pred.getTemperatura() + "°C"); //obtiene solo temperatura y la muestra en el textView
        lblHumedad.setText(pred.getHumedad() + "%"); //obtiene humedad y la muestra en el textView
        Log.e(TAGLOG, msg: "onDataChange:" + dataSnapshot.getValue().toString());
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
        Log.e(TAGLOG, msg: "Error!", databaseError.toException());
    }
};

```

Figura 60. Método para visualizar temperatura y humedad  
Fuente: Elaboración propia

El sistema de reconocimiento de voz para activar o desactivar el sistema de vigilancia y detección, le pide al usuario que ingrese una palabra clave. La palabra clave es generada en la aplicación móvil. La opción de generar una palabra clave funciona con un arreglo de strings para elegir de forma aleatoria la palabra. Las posibles palabras clave son: seguridad, innovador, prototipo, icono y algoritmo. También se declara una variable r de la clase Random.

```
refpalclave = myRef.child("palclave"); //obtener variable de la bd
final Random r = new Random(); // declara variable random
final String[] elementos = {"seguridad","innovador","prototipo","icono","algoritmo" };
//array de posibles palabras clave que se pueden seleccionar por random
```

Figura 62. Variables necesarias para elegir la palabra clave  
Fuente: Elaboración propia

Al presionar el botón generar, se elige de un arreglo de strings (elementos) la palabra clave. Se llama a la función `nextInt` de la clase `random`, la función devuelve un valor `int` entre 0 y el número de elementos en el arreglo (excluido) (`elementos.length`). Este valor es una posición en el arreglo, que viene a ser el que ocupa la palabra elegida. La palabra elegida se muestra en un `Textview`.

```
btnObtener.setOnClickListener((v) -> {
    String ranEle = elementos[r.nextInt(elementos.length)]; //seleccion palabra por random
    txtScram.setText(ranEle); //escribir palabra en textview
});
```

Figura 61. Función del botón generar  
Fuente: Elaboración propia

El botón “OK” es el que envía la nueva palabra clave a la base de datos, para que la guarde y se pueda usar en el sistema de reconocimiento de voz. Cuando actualiza la palabra clave en la base de datos, regresa la aplicación al menú principal (`main`) y muestra el mensaje palabra clave guardada.

```
boton.setOnClickListener((v) -> {
    refpalclave.setValue(txtScram.getText().toString()); //establecer palabra clave en bd
    txtScram.setText("");
    Toast.makeText(getApplicationContext(),
        text: "Palabra clave guardada", Toast.LENGTH_SHORT).show();
    Intent i = new Intent(getApplicationContext(), MainActivity.class);
    startActivity(i);
    finish();
});
```

Figura 63. Función del botón “ok” para guardar palabra clave en base de datos  
Fuente: Elaboración propia

El sistema de vigilancia y detección cuando detecta la presencia de un intruso toma una imagen, la cual es enviada a un correo electrónico. Por medio de la aplicación móvil se puede gestionar a que correo se envía. Antes de guardar un correo ingresado se valida si tiene el formato (patrón) de un correo electrónico. El patrón que debe tener un correo electrónico es: `prueba@prueba.com`. En el método `isEditTextContainEmail`, se verifica si el

editText tiene el patrón señalado y retorna un booleano de “true” o “false”, correspondiente si cumple con el formato o no.

```
public static boolean isEditTextContainEmail(EditText argEditText) {
    try {
        Pattern pattern = Pattern.compile("^[_A-Za-z0-9-]+(\\.[_A-Za-z0-9-]+)*@[A-Za-z0-9]+(\\.([A-Za-z0-9]+)*){1,3}$");
        //definir un patron de secuencia x@x.com
        Matcher matcher = pattern.matcher(argEditText.getText());
        //verifica si cumple el patron
        return matcher.matches(); //envia un estado de true o false si cumple con el patron
    } catch (Exception e) {
        e.printStackTrace(); //imprimir el registro del stack donde se ha iniciado la excepcion
        return false;
    }
}
```

Figura 64. Método para verificar formato de correo electrónico  
Fuente: Elaboración propia

Después de verificar el correo electrónico, que tiene el formato indicado, se guarda en la base de datos Firebase, y muestra un mensaje que se cambió el correo electrónico. Al cambiarlo regresa al menú principal.

```
bl.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if (isEditTextContainEmail(inputcorreo)) { //verificar que cumpla con el patron
            refcorreo.setValue(inputcorreo.getText().toString()); //establece correo en la bd
            inputcorreo.setText("");
            Toast.makeText(getApplicationContext(), text: "Cambio de correo exitoso...", Toast.LENGTH_SHORT).show();
            Intent i = new Intent(getApplicationContext(), MainActivity.class);
            startActivity(i);
            finish();
        } else {
            Toast.makeText(getApplicationContext(), text: "Vuelva a ingresar el correo", Toast.LENGTH_SHORT).show();
        }
    }
});
```

Figura 65. Función del botón ok para guardar correo electrónico  
Fuente: Elaboración propia

## 2.5. Pruebas de Ejecución

### 2.5.1. Pruebas de funcionalidad

Se llevaron a cabo tres pruebas de funcionalidad para el sistema de vigilancia y detección de fuga de gas, se consideraron los requerimientos adquiridos de las encuestas realizadas. Los requerimientos son:

- Requerimiento 1 (R1). Controlar el sistema de vigilancia y detección por medio de un panel de control.
- Requerimiento 2 (R2). Gestionar el sistema de vigilancia y detección por medio de una aplicación móvil.

- Requerimiento 3 (R3). Controlar el sistema de vigilancia y detección por medio de comandos de voz.
- Requerimiento 4 (R4). Enviar notificaciones por mensajes de texto (alerta de intruso y alerta de detección de fuga de gas)
- Requerimiento 5 (R5). Agregar al sistema de seguridad la opción de detectar gas.
- Requerimiento 6 (R6). Agregar un botón que alerte a los familiares de una emergencia.

### Prueba N°1 (T1)

En esta prueba se verificó el funcionamiento del panel de control, la aplicación móvil, el botón de emergencia y las acciones que realiza el sistema de seguridad al momento en que detecta la presencia de un intruso.

En este caso de prueba, el usuario activo la alarma por medio del panel de control presionando la tecla “A” del teclado matricial. El sistema le solicitó que ingrese una contraseña de cuatro dígitos, la cual fue correcta. Al activarse, el sistema emitió zumbido. El sistema brindó al usuario seis segundos para retirarse de la propiedad.



Figura 66. Ingreso de contraseña  
Fuente: Elaboración propia

Cuando ingresa un intruso al hogar, el sistema de seguridad fue activado, emitió un zumbido de alerta, envió un mensaje de texto, se prendió el foco, tomo una imagen que se guardó en Raspberry Pi y se envió la imagen al correo electrónico del propietario. El sistema emitió un mensaje de alerta que la puerta se iba a cerrar, después de tres segundos se cerró. Al pasar 30 segundos el sistema abrió el seguro de la puerta y podría volver a alertar la presencia de algún intruso. Este tiempo tiene la función de no estar enviando mensajes de texto y correos electrónicos de manera excesiva. Además, ese lapso de tiempo le permite al propietario desactivar la alarma si él fue el que ingreso.

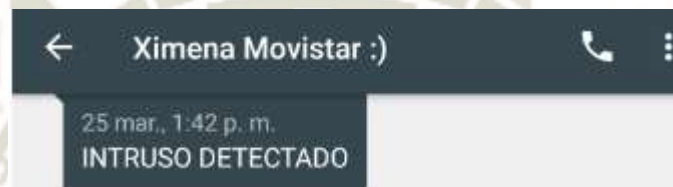


Figura 67. Mensaje de texto "Intruso detectado"

Fuente: Elaboración propia



Figura 68. Mensaje de correo electrónico con la imagen de un intruso

Fuente: Elaboración propia

Para desactivar el sistema de seguridad el usuario utilizó la aplicación móvil, inicio sesión con su cuenta y contraseña, ingreso al menú principal y presiono el botón de desactivar.



*Figura 69.* Alarma activada por la aplicación móvil  
Fuente: Elaboración propia

Para alertar a un familiar de una situación de peligro, el usuario presiono la tecla “0” del panel de control. Lo que envió un mensaje de texto de Auxilio, sin que el sistema de seguridad se encuentre activo.



*Figura 70.* Mensaje de Auxilio  
Fuente: Elaboración propia

## **Prueba 2 (T2).**

En esta prueba se verificó la operatividad del sistema de detección de fuga de gas que se puede controlar por medio del panel de control y por la aplicación móvil. Y las acciones que se llevan a cabo cuando el sistema detecta la presencia de gas.

El usuario activó el sistema de detección de fuga de gas por medio de la aplicación móvil, inicio sesión con su usuario y contraseña, e ingreso a la opción de “Presencia de gas” que se visualiza en el menú principal y presiono el botón “Encender”.



*Figura 71.* Alarma de presencia de gas activada por la aplicación móvil  
Fuente: Elaboración propia

En este caso se utilizó un encendedor para simular la presencia de gas. Se acercó el encendedor al sensor de gas MQ4, al momento en que detecto gas, se visualizó un mensaje de presencia de gas en la pantalla LCD y envió un mensaje de texto al celular del propietario. En caso la temperatura hubiera sido mayor de 35 y la humedad menor de 10, se mostraba un mensaje de posible incendio. Al pasar 15 segundos, y si aún se hubiera sentido presencia de gas, el sistema vuelve a enviar la alerta.



*Figura 72.* Prueba T2 con encendedor  
Fuente: Elaboración propia



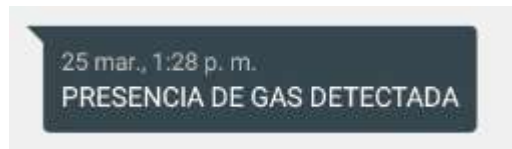


Figura 73. Mensaje de texto "Presencia de gas"  
Fuente: Elaboración propia

Para desactivar el sistema de detección de fuga de gas, el usuario utilizó el panel de control ingresando por medio del teclado la opción “D”, ingreso correctamente la contraseña de cuatro dígitos. En la pantalla LCD se visualiza el mensaje “Detección de fuga de gas desactivada”.

### **Prueba 3 (T3).**

En esta prueba se verificó el funcionamiento del sistema de reconocimiento de voz. Los comandos que se pueden decir son: Activar alarma, activar gas, dime temperatura, encender luz, apagar luz, desactivar alarma y desactivar gas.

El usuario para ingresar a la opción de comandos por voz, presionó la tecla “#”. Al ingresar a esta opción se visualizó en la pantalla LCD, el mensaje: “Lo escucho”. El usuario dijo “activar alarma”, se visualizó un mensaje “diga palabra clave”, el usuario dijo la palabra clave correctamente y la alarma se activó. Para desactivar la alarma por medio de comandos de voz, el usuario volvió a ingresar a esta opción y dijo “desactivar alarma”, se visualizó el mensaje “diga palabra clave”. El usuario tuvo tres oportunidades para decir la palabra clave correctamente.

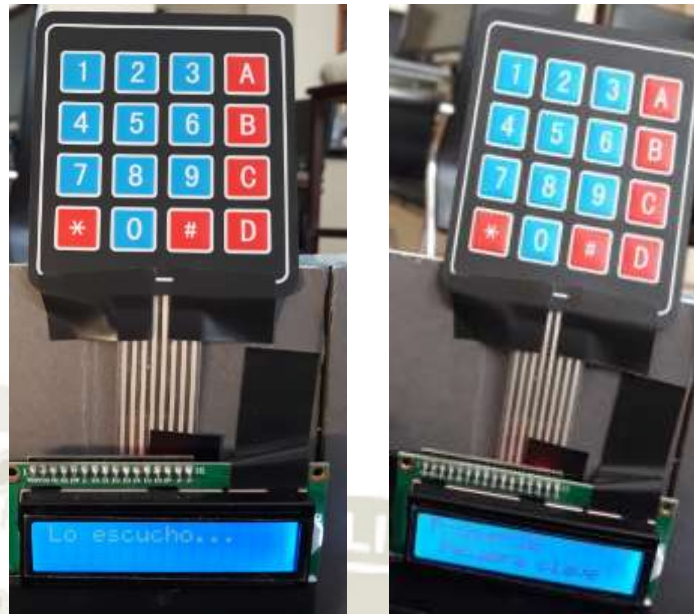


Figura 74. Opciones sistema de reconocimiento de voz  
Fuente: Elaboración propia

También, se probaron las opciones de “encender luz” y “apagar luz”. Para ingresar a esta opción el usuario pulso la tecla “#” y pronuncio “encender luz”, el foco que se encuentra en la sala se prendió. Para apagar el foco el usuario volvió a ingresar a esta opción y pronuncio “apagar luz” y el foco que se encuentra en la sala se apagó.



Figura 75. Comando de voz "encender luz"  
Fuente: Elaboración propia

Este es un buen sistema de reconocimiento de voz del idioma español, es gratuito y fácil de utilizar. Aunque cuenta con una latencia de 8 segundos, el tiempo en que se conecta con el servidor y hace el cambio de voz a texto.

Con los requerimientos y las pruebas realizadas se implementó una matriz de trazabilidad de pruebas (Tabla 21), en la cual se puede observar que dependiendo de la prueba realizada se cumplen algunos requerimientos. Al juntar todas las pruebas en una más compleja en este caso se llevarían a cabo todos los requerimientos descritos. El sistema cumple con todo lo que el cliente necesita, sin dejar de lado ningún requerimiento.

Tabla 21. Matriz de trazabilidad de pruebas  
*Matriz de trazabilidad de pruebas*

Matriz de trazabilidad de pruebas			
Requerimiento	Prueba 1 (T1)	Prueba 2 (T2)	Prueba 3 (T3)
R1	X	X	X
R2	X	X	
R3			X
R4	X	X	
R5		X	
R6	X		

Fuente: Elaboración propia



Figura 76. Prototipo del sistema de vigilancia y detección de fuga de gas

Fuente: Elaboración propia

Además, se llevaron a cabo 16 pruebas en diferentes escenarios. Las cuales se dividieron en pruebas al sistema de vigilancia, al sistema de detección de fuga de gas y al sistema de reconocimiento de voz.

Al sistema de vigilancia se llevaron a cabo cinco pruebas: Prueba1, se llevó a cabo a las 10:40 am. La alarma se activa por la aplicación móvil, ya que el propietario se encontraba fuera de la propiedad. La alarma detecta un intruso, suena la alarma, toma una imagen, envía un mensaje de texto, envía la imagen por correo electrónico y cierra la puerta. Al pasar casi un minuto el sistema está listo para detectar otra presencia y enviar una alerta. Prueba2, se llevó a cabo a las 11:20 am. El sistema detecta un intruso, suena la alarma, toma la imagen, enviar un mensaje de texto, envía la imagen por correo electrónico y cierra la puerta. Después de casi un minuto puede volver a detectar una presencia o puede ser desactivada. Estas dos pruebas se llevaron a cabo en la mañana por eso no se prende el foco.

Prueba3, se llevó a cabo a las 7:06 pm, se activó el sistema por medio de teclado (sale el propietario). A los 10 minutos, ingresa un intruso, se prende la luz, suena una alarma, toma una imagen, envía un mensaje de texto, envía la imagen por correo electrónico, cierra la puerta. Después, de casi un minuto puede volver a detectar una presencia y abre la cerradura. Se desactiva por medio de la aplicación móvil. Prueba 4, se llevó a cabo a las 7:27 pm, activar el sistema por medio de la aplicación móvil. Ingresa un intruso, prende la luz, suena la alarma, toma una imagen, envía un mensaje de texto, envía la imagen por correo electrónico y cierra la puerta. Regresa el propietario a la vivienda y desactiva la alarma por el teclado. Prueba 5, se llevó a cabo a las 8:06 pm. Se activó el sistema por medio del teclado y el propietario se va a quedar en su casa. Ingresa un intruso, prende la luz, suena la alarma, toma una imagen, envía un mensaje de texto, envía un correo electrónico y cierra la puerta.

Debe de pasar, casi un minuto (53 segundos) para que el sistema pueda volver a reportar si hay un ingreso de algún intruso.

Al realizar las pruebas se observó que el tiempo que demora en que la aplicación móvil envié el comando para activar o desactivar el sistema es de seis segundos. Las pruebas realizadas en la mañana varían con las pruebas realizadas en la noche en los tiempos de envío, ya que al prender el foco se demora unos segundos mas en enviar las alertas. Se puede visualizar los resultados y los tiempos en que se envían mensajes de texto y correo electrónico en la tabla 22.

Tabla 22. *Matriz de pruebas sistema de vigilancia*  
*Matriz de pruebas sistema de vigilancia*

Pruebas	Detecta presencia	Prende la luz	Tiempo en enviar un mensaje de texto	Tiempo en enviar un correo electrónico	Cierra la puerta	Casa llena	Casa vacía
Prueba1 10:40 am	SI	NO	9 segundos	20 segundos	14 segundos	x	
Prueba2 11:20 am	SI	NO	9 segundos	20 segundos	17 segundos		x
Prueba3 7:16 pm	SI	SI	17 segundos	19 segundos	21 segundos		x
Prueba4 7:27 pm	SI	SI	16 segundos	17 segundos	20 segundos		x
Prueba5 8:07 pm	SI	SI	16 segundos	19 segundos	20 Segundos	x	

Fuente: Elaboración propia

Al sistema de detección de fuga de gas se realizaron cuatro pruebas: Prueba6, se activa el sistema por medio del teclado ingresando a la opción C. A las 8:26pm, con un encendedor se generó un ambiente con presencia de gas. Al detectar gas, se muestra un mensaje por pantalla LCD y se envía un mensaje de texto, alertando de la presencia. Prueba7, se activa el sistema por medio de una aplicación móvil (opción de presencia de gas). A las 8:53 pm, por medio del encendedor se genera un ambiente con gas. Al detectar gas, se muestra un mensaje por la pantalla LCD y envía un mensaje de texto. Después, de 15 segundos el sistema puede volver a enviar un mensaje.

En las pruebas realizadas se observó que no importa si es de mañana o de noche los tiempos de detección son parecidos. Lo que, si varia es el tiempo en que el sensor este prendido, ya que estos sensores necesitan de un tiempo para calentarse. La primera detección se demora más en alertar que las siguientes, ya que el sensor ha calentado. Se puede visualizar los resultados y los tiempos en que se muestra el mensaje por la pantalla LCD y se envía el mensaje de texto en la tabla 23.

Tabla 23. *Matriz de pruebas sistema de detección de fuga de gas*

Pruebas	Detecta fuga de gas	Muestra mensaje por pantalla LCD	Tiempo en enviar un mensaje de texto	Casa llena	Casa vacía
Prueba6	SI	16 segundos	20 segundos		x
Prueba7	SI	4 segundos	9 segundos	x	
Prueba8	SI	5 segundos	10 segundos	x	
Prueba9	SI	4 segundos	9 segundos	x	

Fuente: Elaboración propia

Para el sistema de reconocimiento de voz se realizaron 7 pruebas según los comandos que se pueden decir como “activar alarma”, “desactivar alarma”, “dime temperatura”, “activar gas”, “desactivar gas”, “encender luz” y “apagar luz”; se probaron los comandos con una voz femenina y masculina, la voz masculina tuvo que decir el comando dos veces, ya que no sabia cuando decir el comando (palabra). En la tabla 24 se puede observar los resultados y el número de intentos que se llevaron a cabo. Cabe resaltar que enviar el comando (audio) al servidor para que lo convierta en texto demora entre 16 y 18 segundos.

Tabla 24. *Matriz de pruebas reconocimiento de voz*

Pruebas	Comando ingresado (voz)	Voz masculina	Voz femenina	Intento #	Palabra clave intento #
Prueba10	Activar alarma	x		2	2
Prueba11	Desactivar alarma		x	1	1
Prueba12	Dime temperatura		x	1	-
Prueba13	Activar gas	x		1	1
Prueba14	Desactivar gas	x		2	1
Prueba15	Encender luz		x	1	-
Prueba16	Apagar luz		x	1	-

Fuente: Elaboración propia

### 2.5.2. Pruebas de seguridad

Al realizar las pruebas de seguridad, se ingresó una contraseña errónea tres veces por el panel de control para activar el sistema de seguridad, bloqueando el sistema por 20 segundos. La contraseña es un conjunto de cuatro dígitos que solo puede contener números 0 - 9 y letras de a – d. Para adivinar esta contraseña se necesitarían 38,000 posibles combinaciones aproximadamente. La contraseña esta codificada por un algoritmo desarrollado en base 64, la cual al visualizarla en la base de datos no es la contraseña correcta.

Se realizaron pruebas de seguridad para activar el sistema de seguridad por medio de la opción de comandos de voz. El sistema le solicita al usuario que pronuncie la palabra clave, el usuario cuenta con tres oportunidades para decir la palabra clave sino el sistema regresa al menú principal. La palabra clave tiene que ser dicha de forma clara para que el sistema la entienda. Al utilizar una palabra clave, si una persona quiere adivinarla, tendría que ingresar millones de opciones. La palabra clave se gestiona por la aplicación móvil, en la cual hay la opción de “palabra clave”, que al presionar el botón generar, de forma aleatoria se elige una palabra entre las cinco palabras establecidas.

Para ingresar al sistema de vigilancia y detección desde la aplicación móvil, el usuario debe de iniciar sesión como método de autenticación. Si el usuario no cuenta con una cuenta entonces debe de generar una nueva, para ello debe de ingresar a “No tiene una cuenta aun, Cree una nueva”, en la cual le solicita al usuario: nombre, correo, una contraseña de seis dígitos y el código de familia. El código de familia es lo que diferencia a la persona como parte de la familia que utiliza el sistema, sin este código la persona no podrá crear una nueva cuenta. Para iniciar sesión el usuario necesita un correo y una contraseña de seis

dígitos que puede contener cualquier carácter. Por medio de Firebase se administra las cuentas creadas, se puede crear una, eliminar, inhabilitar o cambiar la contraseña.

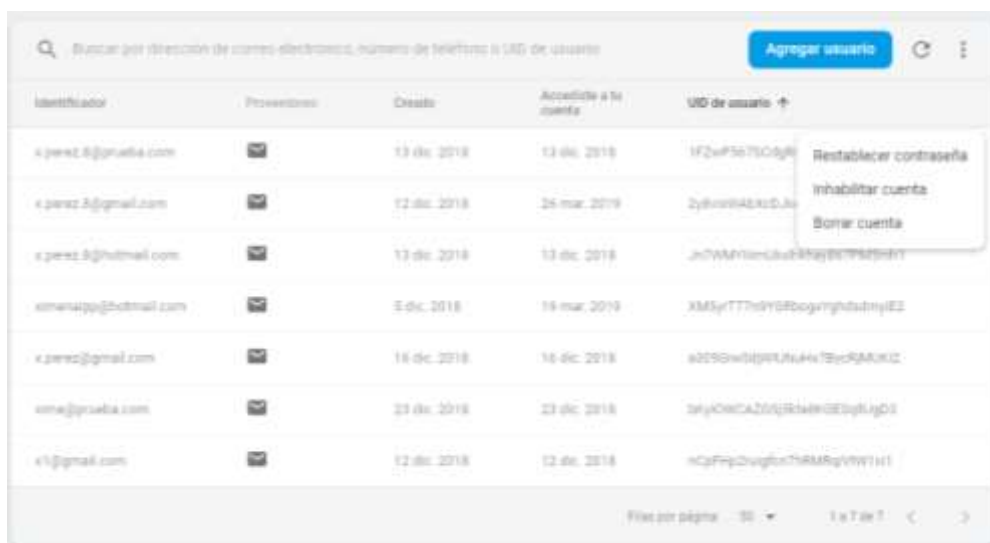


Figura 77. Panel de Firebase para administrar cuentas  
Fuente: Elaboración propia

En caso el usuario se olvide la contraseña, por medio del panel de administración de Firebase, se elige la cuenta que necesita cambiar la contraseña y se envía un correo electrónico con un link de cambio de contraseña.



Figura 78. Correo electrónico con el link para cambiar contraseña  
Fuente: Elaboración propia

### 2.5.3. Pruebas de usabilidad

Para realizar las pruebas de usabilidad del sistema de vigilancia y detección de fuga de gas, se realizaron encuestas por medio del método de conveniencia al elegir 50 personas por ser allegadas al investigador y que usaron u observaron el sistema, para ello se elaboró



un video donde se explica en que consiste y cómo funciona el sistema. Se realizaron preguntas sobre el uso del panel de control, la visualización de los mensajes del panel de control, el uso de la aplicación móvil, el uso del sistema de reconocimiento de voz y el funcionamiento del sistema en general. Las preguntas tienen relación a los requerimientos vistos en las pruebas de funcionalidad.

Los resultados fueron los siguientes:

- Al 36% le parece Muy Bueno controlar el sistema por medio del panel de control y al 52% le parece Bueno.
- En relación a los mensajes que muestra el panel de control son fácil de comprender, el 38% indicó que Muy Bueno y el 52% indicó Bueno.
- En relación a la aplicación móvil, el 38% la califico como Muy Bueno y el 54% la califico como Bueno.
- La aplicación móvil es amigable con el usuario el 100% indicó que Sí.
- El 88% indicó que le parece sencillo utilizar los comandos de voz y el 12% indico que No le parecían sencillos.
- En general calificaron el sistema de vigilancia y detección de fuga de gas el 20% indicó Excelente y el 50% indico Muy Bueno.

Estos resultados indican una buena aceptación del sistema de vigilancia y detección de fuga de gas de posibles clientes. Además, por medio de estas preguntas se verifica que se cumple con los requerimientos propuestos por posibles clientes en la primera encuesta desarrollada.

**Anexo C. ENCUESTA de Satisfacción del sistema de vigilancia y detección de fuga de gas.**

#### 2.5.4. Pruebas de velocidad y rendimiento

Para las pruebas de velocidad y rendimiento del sistema se consideró el uso del CPU, el uso de la memoria RAM y cómo funciona el sistema cuando no tiene una conexión a Internet.

En caso del uso del CPU, el microprocesador con el que se está trabajando es Raspberry Pi 3 cuenta con un procesador de cuatro núcleos ARM Cortex-A53, el uso de CPU se da en porcentaje, el 100% es un uso alto (generando lentitud), más de 80% es normal a alto, menor de 60% es normal y menor de 20% es bajo. Raspberry Pi 3 cuenta con 1 GB de RAM, el uso de 927 Mb es el 100% de uso, el 50% es de 464 Mb y menos del 25% es 232 Mb, el uso de RAM debe de ser menor al 50% para que se ejecute correctamente.

Para visualizar cuanto se utiliza del CPU (unidad central de procesamiento) y de RAM se ejecuta el programa que viene instalado en las versiones de Linux, el gestor de tareas. Se comparó el uso de CPU cuando no se ejecuta ningún programa, cuando se da el pico más alto de uso de CPU al ingresar la contraseña por teclado y cuando se ejecuta el programa.

Cuando no se ejecuta ningún programa solamente el gestor de tareas de forma gráfica (Ixtask), se observó el uso de 5% de CPU y 115 Megas de memoria RAM. Como se puede observar en la imagen el programa que está consumiendo CPU es el gestor de tareas y otros programas que están en segundo plano.

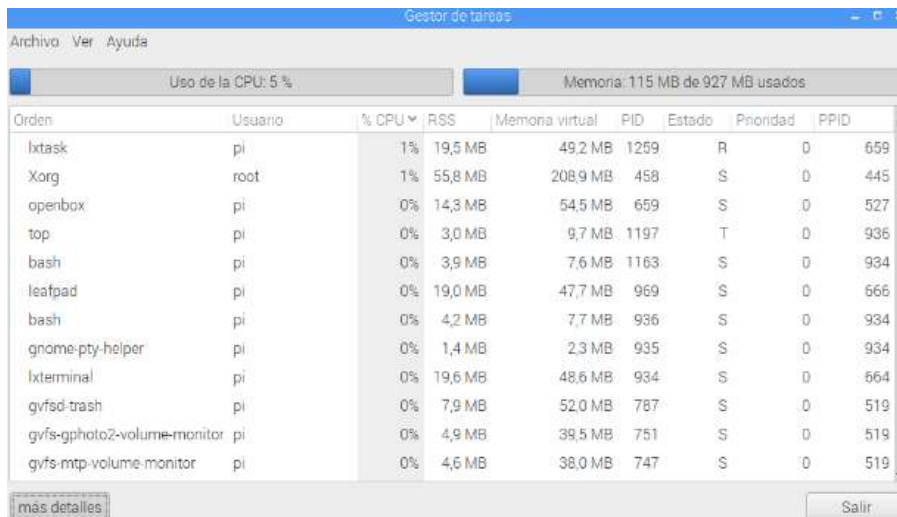


Figura 79. Uso del CPU sin ejecutar ningún programa  
Fuente: Elaboración propia

Al momento de ejecutar el programa, este tiene un pico de uso de CPU normal a alto. Se da cuando el usuario activa/desactiva cualquiera de las alarmas y el sistema le solicita al usuario ingresar una contraseña. En este tiempo en que el usuario ingresa una contraseña por el teclado matricial, el uso del CPU sube al 22% y la memoria RAM se mantiene en 147 Megas como cuando se ejecuta el programa sin ingresar datos por el teclado.

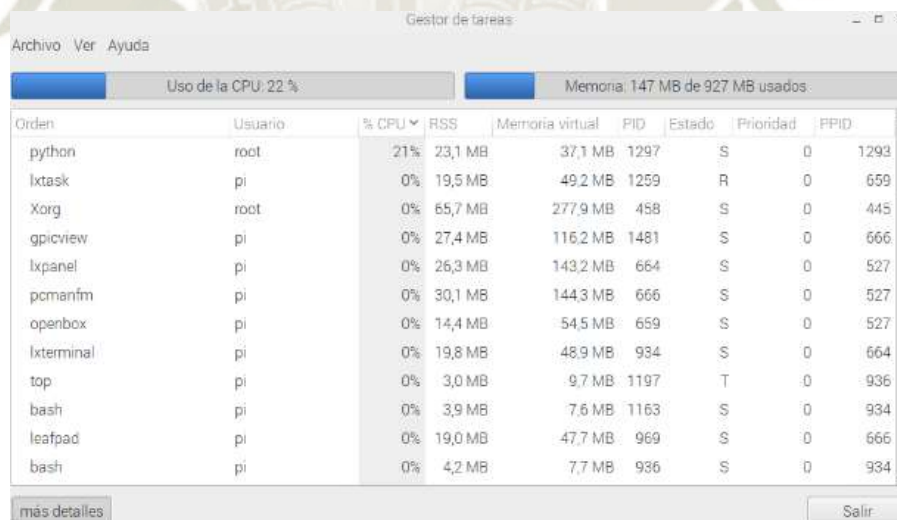
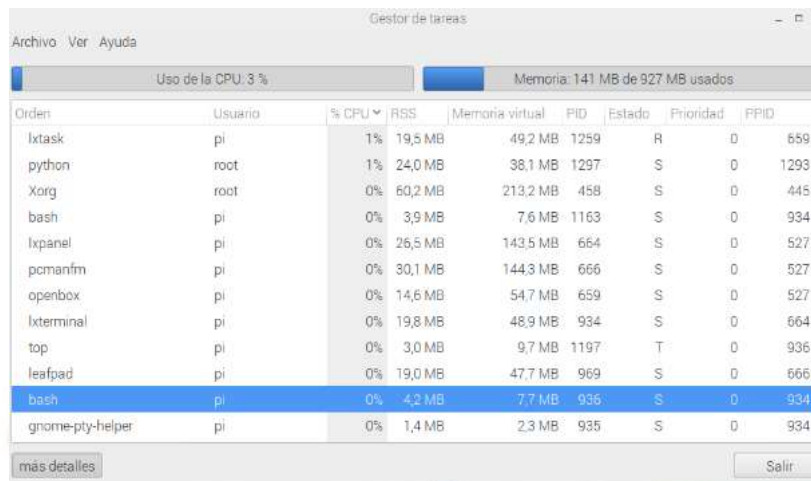


Figura 80. Uso de CPU en el pico más alto  
Fuente: Elaboración propia

Cuando se ejecuta el programa de forma continua y no se ingresan datos por el teclado, el uso del CPU es de 3% y la memoria baja a 141 Megas. Como se puede observar en la imagen el programa Python utiliza el CPU en un 1% y el siguiente porcentaje es propio del programa de gestor de tareas.



Orden	Usuario	% CPU	RSS	Memoria virtual	PID	Estado	Prioridad	PPID
lxtask	pi	1%	19,5 MB	49,2 MB	1259	R	0	659
python	root	1%	24,0 MB	38,1 MB	1297	S	0	1293
Xorg	root	0%	60,2 MB	213,2 MB	458	S	0	445
bash	pi	0%	3,9 MB	7,6 MB	1163	S	0	934
lxpanel	pi	0%	26,5 MB	143,5 MB	664	S	0	527
pcmanfm	pi	0%	30,1 MB	144,3 MB	666	S	0	527
openbox	pi	0%	14,6 MB	54,7 MB	659	S	0	527
lxterminal	pi	0%	19,8 MB	48,9 MB	934	S	0	664
top	pi	0%	3,0 MB	9,7 MB	1197	T	0	936
leafpad	pi	0%	19,0 MB	47,7 MB	969	S	0	666
bash	pi	0%	4,2 MB	7,7 MB	936	S	0	934
gnome-pty-helper	pi	0%	1,4 MB	2,3 MB	935	S	0	934

Figura 81. Uso de CPU ejecutando el programa  
Fuente: Elaboración propia

Como se puede observar el programa no hace uso demandante del CPU, se podría agregar otras características al sistema o se podrían ejecutar otros programas en la Raspberry Pi y este seguiría funcionando con normalidad. Se usó el programa gestor de tareas propio de Linux porque al ejecutarlo no utiliza mucho CPU, permitiendo visualizar el uso de CPU que hace el programa.

En cuanto a la memoria RAM el microprocesador Raspberry Pi 3 tiene 1 GB. Para observar la cantidad de memoria RAM que utiliza el sistema de vigilancia y detección se utilizó el comando `free` en el terminal de Raspbian. Al ejecutar el comando `free -m -h` se observa la memoria total, la memoria usada, la memoria libre, la memoria compartida, la memoria cache y la memoria disponible. Se realizó la prueba de uso de memoria RAM cuando no hay ningún programa ejecutándose, cuando al ejecutar el programa se consume mayor CPU (pico más alto), cuando al ejecutar el programa se utiliza mayor memoria RAM y cuando se ejecuta el sistema de seguridad.

Al no ejecutarse ningún programa en la Raspberry Pi, los procesos y programas que están en segundo plano consumen 98 Megas (campo `used`), quedando 595 Megas para usar (campo `free`), se usan para almacenar metadatos 233 Megas (campo `buff/cache`) y 728 Megas

para iniciar nuevos programas (campo available). Sin ejecutar ningún programa existe bastante memoria como para ejecutar otros programas.

```

pi@raspberrypi:~/Desktop $ free -m -h
              total        used          free      shared  buff/cache   available
Mem:          927M          98M          595M           20M          223M          750M
Swap:          99M           0B           99M
    
```

Figura 82. Uso de memoria sin ejecutar ningún programa  
Fuente: Elaboración propia

Se da el pico más alto de uso de CPU cuando al ejecutar el sistema, el usuario ingresa a las opciones del panel de control de activar/desactivar cualquiera de las alarmas y el sistema solicita el ingreso de una contraseña. En este caso el uso de la memoria RAM sube a 125 megas (campo used), disminuyen los megas libres a 554 (campo free), aumenta el uso de megas de almacenamiento de metadatos a 247 (campo buff/cache) y disminuyen a 730 megas las posibles para iniciar nuevos programas (campo available). Aun cuando se consume mayor CPU la memoria disponible es bastante como para ejecutar otros programas.

```

pi@raspberrypi:~/Desktop $ free -m -h
              total        used          free      shared  buff/cache   available
Mem:          927M          125M          554M           20M          247M          730M
Swap:          99M           0B           99M
    
```

Figura 83. Uso de memoria RAM en el pico más alto de uso de CPU  
Fuente: Elaboración propia

Cuando se utiliza mayor memoria RAM es cuando ingresa un intruso, se toma una imagen, y al enviarla se activa el servidor mutt. En este caso, la memoria utilizada sube a 134M (campo used), la memoria libre disminuye a 538M (campo free), el almacenamiento de metadatos aumenta a 254M (buff/ cache) y disminuye a 716M para iniciar nuevos programas (campo available). Aun cuando el sistema utiliza más memoria RAM, no es una cantidad representativa como para poner lento el sistema o que no permita ejecutar otro programa.

```

pi@raspberrypi:~/Desktop $ free -m -h
              total        used          free      shared  buff/cache   available
Mem:          927M          134M          538M           20M          254M          716M
Swap:          99M           0B           99M
    
```

Figura 84. Mayor uso de memoria RAM en el transcurso de uso del programa  
Fuente: Elaboración propia

Cuando solo se ejecuta el programa en la Raspberry Pi, utiliza 125M de memoria RAM (campo used), la memoria libre sube a 557M (campo free), la memoria que almacena metadatos disminuye a 244M (campo buff/cache) y la libre para ejecutar otros programas sube a 731M (campo available).



```
pi@raspberrypi:~/Desktop $ free -d -h
              total        used         free       shared  buff/cache   available
Mem:          927M         125M         557M           20M          244M         731M
Swap:          99M           0B           99M
```

Figura 85. Uso de memoria RAM al ejecutar el programa  
Fuente: Elaboración propia

Como se puede observar la memoria RAM que utiliza el sistema es poca. Se podría ejecutar otros programas en la Raspberry Pi o se podrían agregar otras características al sistema y seguiría funcionando a la normalidad. Se utilizó el terminar de Linux por que no consume muchos recursos y la lectura de uso de memoria considera varios puntos.

Una característica importante en el uso de Raspberry Pi es la temperatura. Un rango normal de temperatura es por debajo de los 60°C, una temperatura de 70°C a 80°C es normal a alta y una temperatura alta es mayor de 80°C que se da cuando se utiliza en un 100% la CPU, en este caso se debe de tener en cuenta adquirir disipadores o ventiladores. Se hizo la prueba de ejecutar el sistema de vigilancia y detección de fuga de gas por un lapso de dos horas en las horas más calurosas donde la temperatura ambiente oscila entre 23°C y 24°C. Para saber la temperatura de Raspberry Pi, se ejecutó un script, por medio del comando “/opt/vc/bin/vcgencmd measure\_temp”, que accede a la información específica de Raspberry Pi. Al ejecutar el comando se obtiene el resultado de 52.1°C siendo la temperatura que llega a subir la Raspberry Pi. Esta temperatura se encuentra en el rango de normal debajo de 60°C, es una temperatura normal que soporta Raspberry Pi y puede trabajar con normalidad.

```

pi@raspberrypi: ~/Desktop
Archivo Editar Pestañas Ayuda
root 9626  2  0 14:32 ?        00:00:00 [kworker/0:3]
root 9713  2  0 14:33 ?        00:00:00 [kworker/2:1]
root 9853  8868  8 14:35 pts/0    00:00:00 sudo ps -ef
root 9957  8953  0 14:35 pts/0    00:00:00 ps -ef
pi@raspberrypi:~/Desktop$ sudo bash temp.sh
lun mar 25 14:35:34 -05 2019 @ raspberrypi

temp.GPU => temp=52.1'C

-----
Otros datos del sistema
-----
CPU frequency(45)=600000000'Hz
CPU volt=1.2000V
Memoria repartida entre el sistema y la gpu:
Sistema arm=948M
GPU gpu=70M

-----
Memoria libre
-----
              total      used      free     shared  buff/cache
Mem:  927M    113M    878M     19M    243M    743M
Swap:   89M       0B     89M
    
```

Figura 86. Temperatura de Raspberry Pi 3 Modelo B  
Fuente: Elaboración propia

Después, de realizar las pruebas de funcionalidad, usabilidad, seguridad y velocidad y rendimiento, se realizaron las pruebas de validación en conjunto con expertos en el tema. Las pruebas se dieron en relación al código, la aplicación móvil y el sistema en general.

Al realizar la validación del código se consideró los siguientes ítems, las pruebas se realizaron en conjunto con el Ingeniero Fernando Paredes Marchena.

- En la validación de ingreso de opciones por medio del panel de control, se constató que las teclas que no permiten realizar ninguna acción se encuentran bloqueadas.
- Los mensajes que se visualizan en el panel de control (pantalla led) muestra mensajes claros según la opción ingresada y la acción a realizar.
- El código es entendible y está documentado por medio de comentarios que describen lo que realiza las principales líneas de desarrollo.
- En el código se utilizan funciones y expresiones lógicas (if-else, if-elif, while) que permiten reutilizar partes de código y minimizar el número de líneas.

Al realizar la validación de la aplicación móvil se consideró los siguientes ítems, las pruebas se realizaron en conjunto con el Doctor José Sulla Torres.

En cuanto al diseño:

- Los botones para ingresar a alguna opción, aceptar o cancelar podrían ser más grandes.
- Las imágenes e instrucciones son fáciles de comprender y ayudan al usuario a entender las opciones.
- Para ingresar contraseña, correo electrónico y otros datos del propietario se realiza una validación para que sean datos correctos (según el tipo de dato ingresado).

En cuanto a la seguridad:

- La aplicación móvil cuenta con un sistema de autenticación (inicio de sesión) que solo ingresan los usuarios que tienen una cuenta.
- Para crear una nueva cuenta el propietario tiene que validar que es parte de la familia, por medio de un código familiar.

En cuanto a la funcionalidad:

- La aplicación móvil cumple con el objetivo de gestionar y controlar el sistema de vigilancia y detección de fuga de gas de forma remota.

En relación al uso del usuario:

- Se podrían agregar mejores mensajes e imágenes para que sea un poco más intuitiva.
- La aplicación móvil es amigable con el usuario, ya que cuenta con una buena interfaz, es estética y rápida.

Al realizar la validación del sistema en general se consideró los siguientes ítems, las pruebas se realizaron en conjunto con el Ingeniero José Esquicha Tejada.



- El sistema permite mejorar la seguridad en el hogar, como método de prevención ante ingreso de un intruso o al detectar presencia de gas.
- El panel de control permite controlar el sistema de vigilancia y detección de fuga de gas, los mensajes podrían ser más legible y las opciones intuitivas.
- Por medio de comandos de voz se puede controlar el sistema, los comandos deben ser claros y decirlos cerca del micrófono para que el sistema comprenda el mensaje. Para que el ruido ambiental sea menor, se debería adquirir un micrófono con mejores características.
- Al agregar otros sensores de humo, de flama, etc., se reducirían los riesgos en el hogar.

**Anexo D: Fichas de Validación del sistema de vigilancia y detección de fuga de gas para mejorar la seguridad en el hogar.**

**2.6. Manuales de usuario**

**Anexo E. Manual de Usuario: sistema de vigilancia y detección de fuga de gas y Manual de usuario: aplicación móvil.**

## CONCLUSIONES

Después del desarrollo, implementación y pruebas del sistema de vigilancia y detección se puede concluir los siguientes puntos.

1. Se desarrolló un prototipo de sistema de vigilancia y detección de fuga de gas para mejorar la seguridad en maquetas de casas unifamiliares. El sistema alerta al propietario sobre el ingreso de un intruso o una fuga de gas por medio de mensajes de texto, utilizando Open Hardware y Open Source Software.
2. Se aplicó como herramienta el servicio gratuito Wit.ai, para reconocer la voz por medio de un micrófono convencional, ya que, detecta la voz de una persona con vocalización natural, no forzada, y funciona con ruido ambiental.
3. Se desarrolló una aplicación móvil en Android Studio utilizando la plataforma Firebase para gestionar y controlar el sistema de vigilancia y detección de fuga de gas en el espacio virtual ilimitado (Internet).
4. Se implementó y desarrolló un prototipo funcional en una maqueta de un hogar unifamiliar, en el cual se demostró que el sistema propuesto cumple con mejorar la seguridad. Además, se documentó y estructuró el sistema para la correcta conectividad entre el microprocesador Raspberry Pi, el microcontrolador Arduino y la aplicación móvil.
5. Se constató que, al utilizar una contraseña de cuatro dígitos da mayor seguridad, ya que se necesitarían de 38 000 posibles combinaciones para adivinarla aproximadamente. También, la plataforma Firebase brinda mayor seguridad al sistema, al realizar autenticación del usuario al momento en que ingresa al sistema y cuenta con SSL para la transferencia de datos. Además, se comprobó que la CPU que utiliza el sistema en el pico más alto es de 22%, la cual se encuentra debajo del límite de uso (menor de 80%); en cuanto al uso de memoria RAM, el mayor uso se

da al ejecutar el sistema 134 Mb que es menos del 25% de uso del total de 927 Mb.

Ambos, se encuentran debajo del límite que soporta Raspberry Pi en cuanto a la velocidad y rendimiento del sistema.

De esta manera se logran los objetivos formulados.



## RECOMENDACIONES

Primera: Automatización del sistema de detección de fuga de gas

Para el sistema de detección de fuga de gas propuesto, se le podría agregar un sistema automatizado que en el momento en que detecte una fuga de gas, por medio de válvulas se cierre el paso de gas. Así mismo se podría agregar otra notificación como emitir una luz o realizar un sonido de alerta.

Segunda: Enviar un mensaje de alerta al ingresar contraseñas incorrectas

Para activar o desactivar el sistema de vigilancia y detección de fuga de gas se debe de ingresar una contraseña por el teclado, si se ingresa una contraseña errónea tres veces se bloquea el sistema por 20 segundos. Una mejora que se podría dar al sistema es enviar un mensaje de alerta al propietario indicando que alguien esta intentando de ingresar una contraseña errónea más de tres veces.

Tercera: Implementar un sistema de reconocimiento facial

Al sistema de vigilancia se le podría agregar un sistema de reconocimiento fácil que reconozca a algunas personas (familiares) y se le permita el ingreso a la propiedad como persona de confianza. En este caso la autenticación del rostro seria antes de ingresar a la propiedad, se instalaría una cámara afuera de la casa, la cual tomaría una imagen de la persona, permitiendo o denegando el ingreso. La cámara a utilizar necesitaría ser de mejores características para tomar imágenes nítidamente y en ambientes de poca iluminación.

Cuarta: Implementar un asistente para que responda los comandos de voz

Al sistema de reconocimiento de voz se le podría agregar un asistente, el cual responda los comandos ingresados por el usuario. Entonces, cuando el usuario diga un comando de voz, el asistente le diría a que opción ingreso o que va a hacer el sistema en

relación al comando ingresado. Para ello, se tendría que agregar parlantes para oír al asistente. Se le podría agregar otras características que lo convertiría en un asistente virtual como Cortana, Siri, Alexa o Google Assistant. El cual no solo le respondería al usuario, podría configurar eventos en el calendario, configurar alertas o hasta reproducir música.

Quinta: Agregar mayores características al sistema de vigilancia y detección de fuga de gas

Así mismo podemos utilizar otros sensores, actuadores o periféricos que atiendan diversas necesidades del hogar, facilitando el manejo automatizado de puertas, ventanas, iluminación, ventilación, prevención de incendios (otros sensores de gas y detector de llama), ahorro de energía (paneles solares), utilizar agua de lluvia, etc. y lo convertiría en un sistema domótico y eco amigable.

## REFERENCIAS

- Abaya, S., Cabico, E., Domingo, J., Diaz, R., Kojima, H., y Rivera, R. (2016). An Embedded System of Dedicated and Real-time Fire Detector and Locator Technology as an Interactive Response Mechanism in Fire Occurrences. En *International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*, IEEE, India, pp. 407–411.
- Abellán, M.A. (2018) *Control de GPIO con Python en Raspberry Pi*. Recuperado de: <https://www.programoergosum.com/cursos-online/raspberry-pi/238-control-de-gpio-con-python-en-raspberry-pi/que-es-python>
- Adafruit (2018). *DHT11, DHT22 and AM2302 Sensors*. Recuperado de: <https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf>
- Arduino (2014). *Arduino UNO: Datasheet*. Recuperado de: <http://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf>
- Arduino (2015). *¿Qué es Arduino?* Recuperado de: <http://arduino.cl/que-es-arduino/>
- Ávila, K. (2016). *¿Qué es un reconocedor de voz?*. Recuperado de: <http://www.cavsi.com/preguntasrespuestas/que-es-un-reconocedor-de-voz/>
- Balbi, M. (2016). *Domótica, la casa del futuro que ahorra un 30 % de energía*. Recuperado de: <https://www.infobae.com/play-tv/2016/08/17/domotica-la-casa-del-futuro-que-ahorra-un-30-de-energia/>
- Bosu, K. y Choudhuri, R. (2017). *Learn Arduino Prototyping in 10 days*. Birmingham, Inglaterra: Packt Publishing
- Cobo, R. (2016). *GPRS*. Recuperado de: <http://www.aie.cl/files/file/comites/ca/articulos/octubre-06.pdf>

Corona, L., Abarca, G., y Mares, J. (2014). *Sensores y Actuadores: Aplicaciones con Arduino*. Azcapotzalco, México D.F.: Grupo Editorial Patria

Creaciones informáticas aplicadas (2012). *Mouse, Teclado y Micrófono*. Recuperado de: <https://creacionesinformaticasaplicadas.wordpress.com/2012/11/18/mouse-teclado-y-microfono/>

Crespo, J.E (2016). *Aprendiendo Arduino: Sensores y Actuadores*. Recuperado de: <https://aprendiendoarduino.wordpress.com/category/perifericos/>

Daramas, A., Pattarakitsophon, S., Eiumtrakul, K., Tantidham, T., y Tamkittikhun, N. (2016). HIVE: Home Automation System for Intrusion Detection. *Procedente de 2016 5th ICT International Student Project Conference IEEE, ICT-ISPC 2016*, Tailandia, 101–104. doi: 10.1109/ICT-ISPC.2016.7519246

del Valle Hernández, L. (2016). *Relé y Arduino MKR1000, controla la luz de tu casa*. Recuperado de: <https://programarfacil.com/blog/arduino-blog/rele-y-arduino-mkr1000/>

Dhruvajyoti, P., y Debashis, J. (2016). GSM Based Fire Sensor Alarm Using Arduino. *International Journal of Scientific and Engineering Research*, 7(4), 259–262.

D-Robotics Uk (2010). *DHT11 Humidity & Temperature Sensor*. Recuperado de: <http://www.micropik.com/PDF/dht11.pdf>

Espeso, P.(2014). *Wit.ai tiene una idea ambiciosa: reconocimietno de voz para todo tipo de proyectos*. Recuperado de: <https://www.xataka.com/otros/wit-ai-tiene-una-idea-ambiciosa-reconocimiento-de-voz-para-todo-tipo-de-proyectos>

García S. (2016). *¿Qué tipos de cables y conectores hay?* Recuperado de: <https://www.analfatecnicos.net/pregunta.php?id=33>

- Gyulyustan, H., y Enkov, S. (2017). Experimental speech recognition system based on Raspberry Pi 3. *IOSR Journal of Computer Engineering*, 19(3), 107–112. doi: 10.9790/0661-190302107112
- Hadwan, H. H., y Reddy, Y. P. (2016). Smart Home Control by using Raspberry Pi & Arduino UNO. *International Engineering Research Journal*, 5(4), 283–288.
- Hanwei Electronics. (2016). *MQ4 Semi conductor Sensor for Gas Natural*. Recuperado de: <https://www.pololu.com/file/0J311/MQ4.pdf>
- Huang, B. y Runberg, D. (2017). *The Arduino Inventor's Guide: Learn Electronics by Making 10 Awesome Projects*. California, Estados Unidos: SparkFun Electronics
- Huidobro, J. M. , Millán, R. J. (2010). *Manual de Domótica*. España: Creaciones Copyright SL
- Instituto Nacional de Defenza Civil (2017). *Cuales son las principales causas de incendio en el hogar*. Recuperado de: <https://rpp.pe/prevenir-es-simple/cuales-son-las-principales-causas-de-incendio-en-el-hogar-noticia-1088572>
- Instituto Nacional de Estadística e Informática (2019). *Principales indicadores seguridad ciudadana a nivel regional, semestre móvil Setiembre 2018 - Febrero 2019*. Recuperado de: [https://www.inei.gob.pe/media/MenuRecursivo/boletines/estadisticas\\_de\\_seguridad\\_ciuadana\\_a\\_nivel\\_regional\\_set.2018-feb.2019.pdf](https://www.inei.gob.pe/media/MenuRecursivo/boletines/estadisticas_de_seguridad_ciuadana_a_nivel_regional_set.2018-feb.2019.pdf)
- InVoxMedical (2018). *La Inteligencia Artificial en el reconocimiento de voz*. Recuperado de: <https://invoomedical.com/la-inteligencia-artificial-en-el-reconocimiento-de-voz/>
- Jasper (2016). *Configuring Jasper*. Recuperado de: <http://jasperproject.github.io/documentation/configuration/>



- Khong Fu, M., y Samson, S. (2017). PAL: Personal assistant system using low-cost computer. *Journal of Telecommunication, Electronic and Computer Engineering*, 9(3–11), 17–21.
- Kranz, M. (2017). *Internet of Things: Construye nuevos modelos de negocio*. Madrid, España: LID
- Kumar, K., Natraj, H., y Jacob, T. P. (2017). Motion activated security camera using Raspberry Pi. En *International Conference on Communication and Signal Processing (ICCSP 2017)*, IEEE, 17, pp. 1598–1601. doi: 10.1109/ICCSP.2017.8286658
- Kumar, P., y Pati, U. C. (2016). Arduino and Raspberry Pi based smart communication and control of home appliance system. En *International Conference on Green Engineering and Technologies (IC-GET 2016)*, IEEE. doi: 10.1109/GET.2016.7916808
- Kushagra (2012) LCD. Recuperado de: <https://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>
- Llamas, L. (2016). *Manejar cargas de mas de 220V con Arduino y salida por relé*. Recuperado de: <https://www.luisllamas.es/arduino-salida-rele/>
- Li, W. J., Yen, C., Lin, Y. S., Tung, S. C., y Huang, S. M. (2018). JustIoT Internet of Things based on the Firebase real-time database. En *2018 IEEE International Conference on Smart Manufacturing, Industrial and Logistics Engineering (SMILE), 2018*, Taiwan, 43–47. doi: 10.1109/SMILE.2018.8353979
- Logitech® (2011). *Webcam c170*. Recuperado de: <https://www.msyste.ms.gr/files/product/12615.pdf>
- Maestre, J.M. (2015). *Domótica para ingenieros*. Madrid, España: Ediciones Paraninfo

S.A

Marble (2004). *Mutt: Cliente de correo de consola*. Recuperado de:

<https://www.marblestation.com/?p=246>

Margolis, M. (2011). *Arduino Cookbook*. California, Estados Unidos: O'Reilly Media

Martín, J. (2011). *Sensores (Instalaciones domóticas)*. Madrid, España: Editex

Martínez, J.F (2017). *¿Qué es la automatización?*. Recuperado de:

<https://www.seas.es/blog/automatizacion/que-es-la-automatizacion/>

Mary, P. S., Harsha, S., y Reddy, C. (2017). Enhanced Home Automation System using Internet of Things. *International Conference on I-SMAC, IEEE*, 89–93.

Matus, D. (2018). *CDMA y GSM: Te explicamos que son y por que deben importarte*.

Recuperado de: <https://es.digitaltrends.com/celular/redes-cdma-y-gsm/>

McManus, S. y Cook, M. *Raspberry Pi for Dummies*. New Jersey, Estados Unidos: John Wiley & Sons, Inc..

Moro, M. (2011). *Instalaciones Domoticas*. Madrid, España: Ediciones Paraninfo S.A

Moya, P. (2016). *Todo sobre la nueva Raspberry Pi 3, más potente y conectada que nunca*.

Recuperado de: <https://omicron.elespanol.com/2016/02/raspberry-pi-3-model-b/>

Muñoz, A. (2017). *¿Qué es Internet de las Cosas?* Recuperado de:

<https://computerhoy.com/noticias/internet/que-es-internet-cosas-61528>

Pascual, A. (2013) *Dos millones de razones para saber qué es exactamente Raspberry Pi*.

Recuperado de: [https://www.elconfidencial.com/tecnologia/2013-11-22/dos-millones-de-razones-para-saber-que-es-exactamente-raspberry-pi\\_56003/](https://www.elconfidencial.com/tecnologia/2013-11-22/dos-millones-de-razones-para-saber-que-es-exactamente-raspberry-pi_56003/)

Parallax Inc (2011). *4x4 Matrix Membrane Keypad (#27899) Datasheet*. Recuperado de:

<https://www.parallax.com/sites/default/files/downloads/27899-4x4-Matrix-Membrane-Keypad-v1.2.pdf>

Peña, J. (2018). *¿Qué es Firebase y que nos aporta?*. Recuperado de:

<https://arpentechnologies.com/es/blog/aplicaciones-movil/que-es-firebase-y-que-nos-aporta/>

Punto Flotante S.A (2017). *HC-SR501 PIR Sensor infrarrojo de movimiento*. Recuperado de: <https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf>

Raghunath, M., y Anil Kumar, G. (2017). Integrated Smart Home Management and Security System based on Wireless Video Streaming using Internet of Things. *International Journal of Mechanical Engineering and Computer Applications*, 5(2), 29–34.

Raspberry Pi Foundation (2016). *Raspberry Pi 3 Model B*. Recuperado de:

[https://www.terraelectronica.ru/pdf/show?pdf\\_file=%252Fds%252Fpdf%252FT%252FTechicRP3.pdf](https://www.terraelectronica.ru/pdf/show?pdf_file=%252Fds%252Fpdf%252FT%252FTechicRP3.pdf)

Ray, R. K., y Uddin, M. A. (2016). GSM Based Bank Vault Security System. *International Journal of Computer Science and Information Security (IJCSIS)*, 14(2), 35–39.

Sanz, M. (2018). *Google señala el reconocimiento de voz como el proximo salto tecnologico*. Recuperado de: <https://computerhoy.com/noticias/industria/google-senala-reconocimiento-voz-como-proximo-salto-tecnologico-306003>

Saxena V. (2016). *Características que debes buscar en una camara web*. Recuperado de:

[https://techlandia.com/caracteristicas-debes-buscar-camara-web-info\\_303726/](https://techlandia.com/caracteristicas-debes-buscar-camara-web-info_303726/)

Soliman, M. S., Alahmadi, A. A., Maash, A. A., y Elhabib, M. O. (2017). Design and

Implementation of a Reliable Wireless Real-Time Home Automation System Based on Arduino Microcontroller Kit and LabVIEW Platform. *International Journal of Applied Engineering Research ISSN*, 12(18), 7259–7264.

Sunrom (2012). *DHT11 - Humidity and Temperature Sensor*. Recuperado de:  
<http://robocraft.ru/files/datasheet/DHT11.pdf>

Talavera, L.(2017). *Lecciones Aprendidas: Firebase Database*. Recuperado de:  
<https://www.belatrixsf.com/blog/lecciones-aprendidas-firebase-database/>

Torrente, O. (2013). *ARDUINO. Curso práctico de formación*. México D.F.: Alfaomega Grupo Editorial S.A.

Upton, E. y Halfacree, G. (2016). *Raspberry Pi, Guía de Usuario*. Inglaterra: Raspberry Pi Foundation

Vaduva, L. (2018). *MQ4 Gas sensor*. Recuperado de: <https://www.geekstips.com/mq4-sensor-natural-gas-methane-arduino/>

Vaidya, B., Patel, A., Panchal, A., Mehta, R., Mehta, K., y Vaghasiya, P. (2017). Smart home automation with a unique door monitoring system for old age people using Python, OpenCV, Android and Raspberry pi. En *International Conference on Intelligent Computing and Control Systems (ICICCS 2017)*, IEEE, 17, pp. 82–86. doi: 10.1109/ICCONS.2017.8250582

Valero, A. (2016). *Utiliza un relé con tu placa Arduino*. Recuperado de:  
<http://diwo.bq.com/utilizar-rele-arduino-zum-core/>

Vialfa, C. (2017). *Estándar GSM (Sistema global de comunicaciones móviles)*. Recuperado de: <https://es.ccm.net/contents/681-estandar-gsm-sistema-global-de-comunicaciones-moviles>

Vilca,G.(2017). *Infografía: aumenta la delincuencia en Arequipa*. Recuperado de:

<https://diariocorreo.pe/edicion/arequipa/infografia-preocupante-aumento-de-la-delincuencia-en-arequipa-734550/>

Wadhvani, S., Singh, U., Singh, P., y Dwivedi, S. (2018). Smart Home Automation and Security System using Arduino and IOT. *International Research Journal of Engineering and Technology (IRJET)*, 5(2), 1357–1359.

Younis, S. A., Ijaz, U., Randhawa, I. A., y Ijaz, A. (2017). Speech Recognition Based Home Automation System using Raspberry Pi and Zigbee. *NFC IEFR Journal of Engineering and Scientific Research*, 5(2), 40–45. doi: 10.24081/nijesr.2017.2.0008

Zamora, J.A.(2016). *¿Qué es Firebase? La mejorada plataforma de desarrollo de Google*. Recuperado de: <https://elandroidelibre.elespanol.com/2016/05/firebase-plataforma-desarrollo-android-ios-web.html>

## ANEXOS

## Anexo A. Glosario de Términos

- **API:** Interfaz de programación de aplicaciones.
- **Arduino:** Plataforma de hardware libre, se compone de una placa (microcontrolador) y un entorno de desarrollo.
- **ASCII:** Código Estándar Estadounidense para el Intercambio de Información.
- **Autenticación:** Proceso de confirmar que algo o alguien es quien dice ser.
- **CPU:** Unidad Central de Procesamiento, parte central de la computadora, realiza el procesamiento de todas las funciones.
- **Firestore:** Plataforma para el desarrollo de aplicaciones web y móviles, de Google.
- **GND:** Tensión negativa, en inglés se denomina ground, en español se denomina tierra.
- **GPIO:** Entrada/Salida de Propósito General, es un pin genérico, puede ser entrada o salida, se puede programar por el usuario en tiempo de ejecución.
- **GPRS:** General Packet Radio Service (Servicio General de paquetes vía radio), red para la transmisión móvil de voz y datos.
- **GSM:** Global System for Mobile communications (Sistema global para la comunicaciones móviles), sistema de teléfono móvil digital más utilizado.
- **HTTP:** Hypertext Transfer Protocol, protocolo de comunicación que permite la transferencia de datos en Internet.
- **HTTPS:** Hypertext Transfer Protocol Secure, protocolo de comunicación basado en HTTP, realiza la transferencia segura de datos de Hipertexto.
- **I2C:** Inter-Integrated Circuit, protocolo de comunicación serial diseñado por Philips Semiconductor para el envío de mensajes de baja velocidad entre dispositivos periféricos y microcontroladores.

- **ICSP:** Programación Serial en Circuito, reprogramación en microcontroladores sin que sea necesario removerlos del circuito.
- **IMAP:** Internet Message Access Protocol (Protocolo de acceso a mensajes de Internet), protocolo de aplicación para obtener mensajes, no descarga los correos electrónicos sino los lee del servidor.
- **IOT:** Internet of Things, siendo en español Internet de las cosas.
- **ISO 27001:** Norma internacional emitida por la Organización Internacional de Normalización (ISO), que permite el aseguramiento, confidencialidad e integridad de los datos y la información en una empresa.
- **JSON:** JavaScript Object Notation, formato de texto sencillo para el intercambio de datos.
- **LED:** Light-Emitting Diode, diodo que emite luz es un semiconductor, convierte en luz la corriente eléctrica de bajo voltaje que atraviesa su chip.
- **Maqueta:** Reproducción de una construcción (real o proyecto) hecha a escala en tamaño reducido.
- **Memoria RAM:** Random Access Memory (Memoria de Acceso Aleatorio), en esta memoria se cargan todas las instrucciones que ejecuta la unidad central de procesamiento y otras unidades del computador.
- **MQ:** Módulo de la familia de sensores de detector de humo y gases inflamables.
- **MuTT:** Cliente de correo electrónico libre basado en texto, para sistemas similares a Unix.
- **NoSql:** Amplia clase de sistemas de gestión de base de datos que son diferentes al modelo clásico, no usan SQL como lenguaje principal de consultas.

- **Open Hardware:** Hardware que su diseño se hace público y disponible para modificarlo, estudiarlo y distribuirlo. Basado en el diseño, se puede producir y vender el hardware.
- **Open Source Software:** Código abierto, termino para denominar cierto tipo de software que se distribuye mediante una licencia o forman parte del dominio público que le permite al usuario final utilizar el código fuente del programa para estudiarlo, modificarlo y realizar mejora en el mismo, incluso para redistribuirlo.
- **POP3:** Post Office Protocol (Protocolo de Oficina de Correo), protocolo que descarga los mensajes eliminándolos del servidor, permite revisar el correo electrónico incluso sin conexión a Internet.
- **Prototipo:** Primer ejemplar que se fabrica de una figura, un invento u otra cosa, y que sirve de modelo para fabricar otros iguales.
- **PWM:** Modulación por ancho de pulsos, es un tipo de señal de voltaje utilizado para enviar información o para modificar la cantidad de energía que se envía a una carga.
- **Python:** Lenguaje de programación interpretado, código legible.
- **Raspberry Pi:** Placa computadora (SBC) de bajo coste, ordenador de tamaño reducido, desarrollada por la fundación Raspberry Pi, con el objetivo de estimular la enseñanza de la informática en las escuelas.
- **Reconocimiento:** Acción de distinguir una cosa entre las demás como consecuencia de sus características y rasgos.
- **RNN:** Red Neuronal Recurrente, es una red neuronal que realiza la misma tarea para cada elemento de una secuencia y donde cada salida depende de los cálculos realizados previamente.



- **Seguridad:** Reducir los niveles de riesgo a aceptables, ya que el riesgo es inherente a cualquier actividad y no puede ser eliminado en su totalidad.
- **SMS:** Servicio de Mensajes Cortos, servicio de la telefonía celular (móvil) que permite enviar y recibir mensajes de texto de extensión reducida.
- **Smtip:** Simple Mail Transfer Protocol (Protocolo para Trasferencia Simple de Correo), protocolo de comunicación que permite el envío de correos electrónicos en Internet.
- **SOC:** Sistema de Organización de Servicios, realiza informes para asegurar que los controles que rodean los servicios que pueden afectar al cliente, se implementan correctamente.
- **SPI:** Serial Peripheral Interface, protocolo de comunicación serial. Es un bus de comunicación a nivel de circuitos integrados. La transmisión de datos se realiza en serie, un bit después de otro.
- **SSL:** Secure Locket Layer, protocolo de seguridad. Los datos viajan de manera íntegra y segura, entre un servidor y usuario, y en retroalimentación. Es cifrada o encriptada.
- **STT:** Speech to text (voz a texto), reconocimiento de voz por computadora, subcampo interdisciplinario de lingüística computacional que desarrolla metodologías y tecnologías que permite el reconocimiento y traducción del lenguaje hablado en texto por computadora.
- **Transductor:** Dispositivo que tiene la misión de recibir energía de una naturaleza eléctrica y suministrar otra energía de diferente naturaleza, pero de características dependientes de las que recibió.

- **UART:** Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal), protocolo serie asíncrono que controla los puertos y dispositivos serie.
- **VCC:** Voltaje en corriente directa, alimentación positiva del circuito.
- **VDC:** Voltaje of Continuos Current (Voltaje de Corriente Continua), es el voltaje que mantiene una polaridad y al momento de conectarlo se debe de respetar los signos + y -.
- **Vigilancia:** Monitoreo del comportamiento de personas, objetos o procesos que se encuentran insertos dentro de un determinado sistema con el fin de detectar aquellos que no cumplen con las normas.
- **Wav:** Formato de audio digital normalmente sin compresión de datos.
- **Wit.ai:** Plataforma con herramientas de inteligencia artificial que utiliza comandos escritos como hablados. Permite construir bots, aplicaciones móviles, aplicaciones de domótica, etc.
- **XML:** Es un subconjunto de SGML (Estandar Generalised Mark-up Language), simplificado y adaptado a Internet.

## Anexo B. ENCUESTA 1: Sistema de vigilancia y detección de fuga de gas

Esta encuesta se desarrolló para determinar los requisitos y requerimientos de software para el proyecto, el nivel de inseguridad en la muestra y si las personas encuestadas adquirirían un sistema como este y cuanto estarían dispuestas a pagar. Las personas encuestadas fueron elegidas por el método de conveniencia, por ello son allegadas al encuestador como familiares, amigos, amigos de familiares, etc. El tamaño de la muestra es de 52 personas.

La primera pregunta es para determinar: cuál es la edad de las personas encuestadas. La edad de las personas encuestadas se dividió en tres rangos, según el poder adquisitivo que tiene cada una. Considerando que las personas mayores de 30 años pueden adquirir un sistema de seguridad, ya que tienen una casa propia, en diferencia de personas menores de 30 que aún pueden vivir con sus padres. Como se describió anteriormente se consideró personas mayores de 30 años como personas interesadas en adquirir un sistema de seguridad. El 44.2% son personas de 30 a 50 años, el 32.7% son personas de 18 a 30 años y el menor grupo es de 23.1% personas de 50 a 70 años. Al enfocar la encuesta en personas mayores de 30 años ayuda a determinar si el sistema de vigilancia y detección de fuga de gas sería adquirido por posibles clientes.

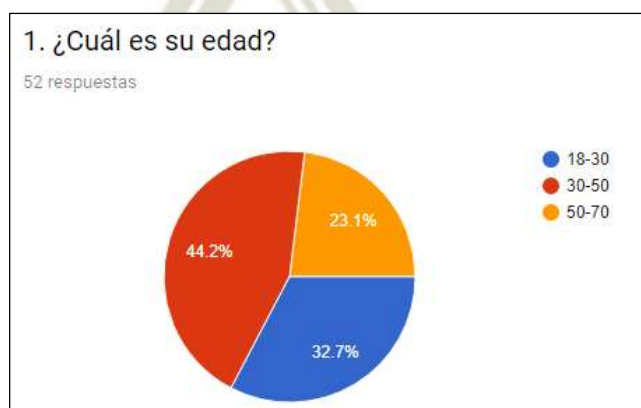


Figura B 1. Resultado de la primera pregunta Encuesta 1  
Fuente: Elaboración propia

El siguiente aspecto que se desea conocer de las personas encuestadas es su tipo de vivienda, según eso se realizará una maqueta para que sea un prototipo funcional. Los resultados indican que un 46.2% vive en casa de dos pisos, un 44.2% son personas que viven en departamentos y un 9.6% viven en casa de un piso. Con los resultados anteriores son determinantes para realizar una maqueta con las características de un departamento y una casa de dos pisos con áreas comunes como sala-comedor y cocina, con una puerta principal. Para que el usuario visualice que el sistema es posible tanto en casa como en un departamento.

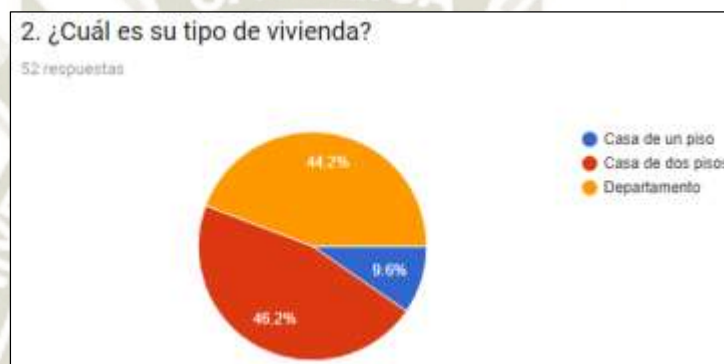


Figura B 2. Resultado de la segunda pregunta Encuesta 1  
Fuente: Elaboración propia

El siguiente aspecto a evaluar es el porcentaje de encuestados que cuentan con un sistema de alarma. En porcentaje un 55.8% “NO” cuenta con un sistema de alarma mientras que un 44.2% “Sí” cuenta con un sistema de alarma. Este resultado nos indica que es posible que las personas que no cuentan con un sistema de alarma les interesen adquirir un sistema como el de este proyecto. Por otro lado, las personas que si cuentan con un sistema de alarma, es posible que les interese una opción económica y con los mismos beneficios o mejores del sistema que tiene.



Figura B 3. Resultado de la tercera pregunta Encuesta 1  
Fuente: Elaboración propia

El siguiente aspecto a evaluar de las personas encuestadas es, si “alguna vez han ingresado a la propiedad sin su consentimiento”, los resultados fueron 51.9% “Sí” y un 48.1% “No”. Los resultados indican que un porcentaje considerable fue víctima de un ingreso de una persona ajena al hogar. Este indicador hace que las personas consideren en adquirir un sistema de vigilancia y detección, como medida de prevención.



Figura B 4. Resultado de la cuarta pregunta Encuesta 1  
Fuente: Elaboración propia

La siguiente pregunta se relaciona directamente con el problema que desea resolver este proyecto, que es el tema de inseguridad en el hogar. La pregunta se relaciona si la persona fue víctima de robo en el hogar, los resultados indican que el 63.5% “Si” fue víctima de robo y un 36.5% “No”. Los resultados indican que debido a la inseguridad a la cual fueron

víctimas los encuestados, puede que la persona desee adquirir un sistema de vigilancia y detección de fuga de gas para el hogar.

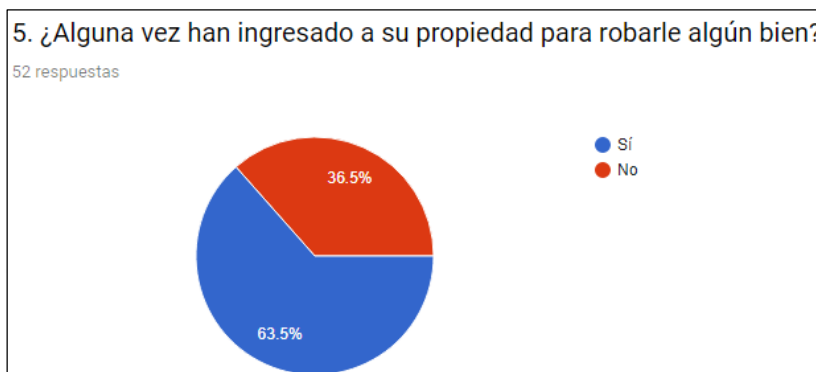


Figura B 5. Resultado de la quinta pregunta Encuesta 1  
Fuente: Elaboración propia

La siguiente pregunta tiene relación a como el usuario le gustaría controlar el sistema de vigilancia y detección de fuga de gas. Los resultados son: un 32.7% con un panel de control, 30.8% con una aplicación móvil, un 28.8% con todas las opciones anteriores (panel de control, app y reconocimiento de voz) y un 7.7% por medio de comandos de voz. Este es un requerimiento directo del cliente, que le indica al grupo de desarrollo realizar un panel de control novedoso, una aplicación móvil como control del sistema y comandos de reconocimiento de voz para facilitar el ingreso de opciones. Estas opciones brindan al sistema ser novedoso e interesante al cliente.

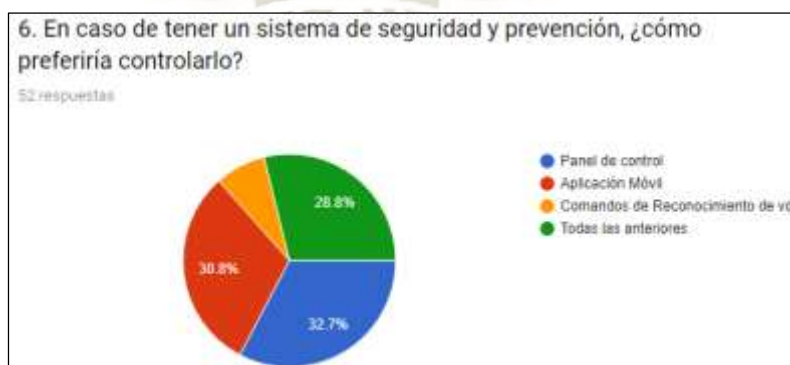


Figura B 6. Resultado de la sexta pregunta Encuesta 1  
Fuente: Elaboración propia

La siguiente pregunta tiene relación a como el usuario le gustaría ser notificado y recibir alertas del sistema. El 55.8% indico que le gustaría ser notificado por mensaje de texto y aplicación móvil. Con un porcentaje mayor de 25% por la aplicación móvil y un 19.2% por medio de mensaje de texto. El notificar al cliente por mensaje de texto es interesante, ya que es un método que no requiere de Internet.

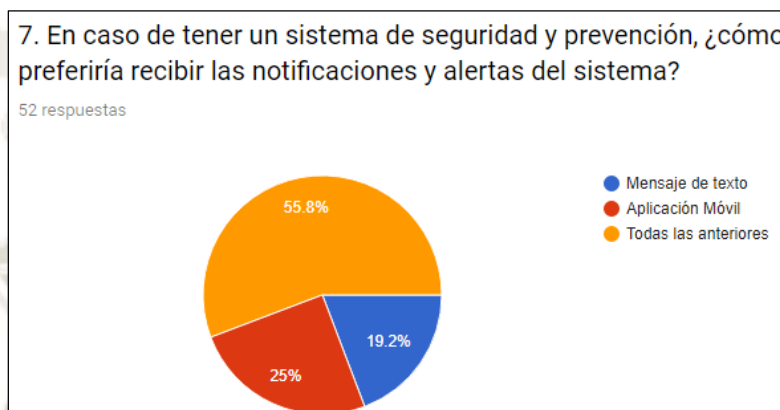


Figura B 7. Resultado de la séptima pregunta Encuesta 1  
Fuente: Elaboración propia

La siguiente pregunta ayuda a resolver un requerimiento funcional sobre el sistema de seguridad, además de detectar la presencia de un intruso, cual sería otra característica importante que se podría agregar al sistema. Los resultados presentan que un 69.2% indica que se debe agregar una alerta de presencia de gas e incendio, un 57.7% indica que agregar un botón de pánico que envía un alerta a un familiar, un 38.5% indica que la detección de puertas abiertas y un 19.2% indica que el control de iluminación. Considerando este punto se agrega al sistema un sensor de detección de fuga de gas y un sensor de temperatura que indica la presencia de un posible incendio. También, se agregar una función de botón de pánico que envía un SMS a un familiar.

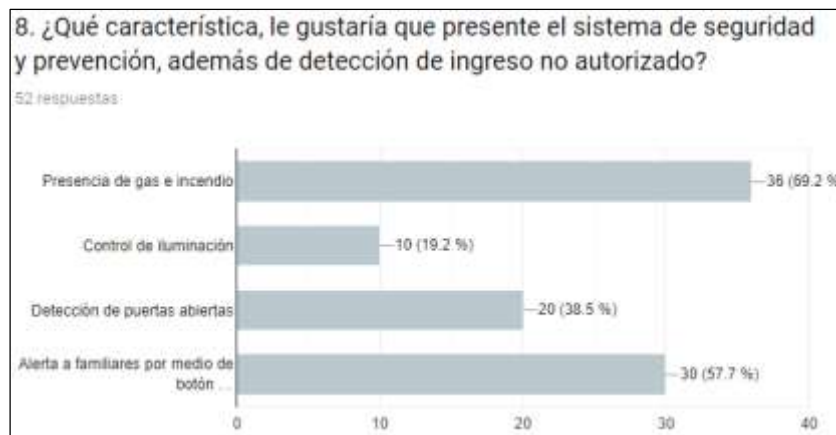


Figura B 8. Resultados de la octava pregunta Encuesta 1  
Fuente: Elaboración propia

La siguiente pregunta hace referencia, si los posibles usuarios han escuchado sobre los sistemas de alarma inteligente que se encuentran en el mercado. Del universo de encuestado el 61.5% indico que “sí” a escuchado, y un 38.5 “no” ha escuchado. Estas cifras son interesante porque indican que un porcentaje tiene una idea de cuáles son las características del sistema y los beneficios que aporta un sistema como este a la vida diaria en relación a la seguridad del hogar.

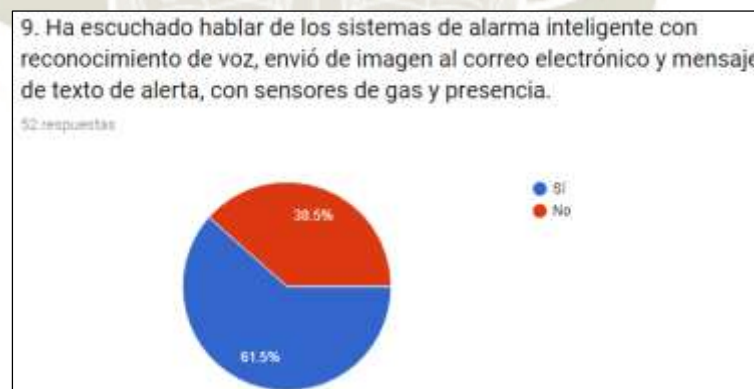


Figura B 9. Resultado de la novena pregunta Encuesta 1  
Fuente: Elaboración propia

La siguiente pregunta ayuda a determinar el precio del sistema de vigilancia y detección que sea posible de pagar al usuario. Se le pregunto al usuario cuanto está dispuesto a pagar por un sistema con las características anteriores, el cual indico: el 55.8% de 800 a 1000 soles, el 38.5% de 1000 a 3000 soles y un porcentaje menor del 5.8% de 3000 a 5000 soles. Con los resultados anteriores y el estudio de viabilidad el cual indica que el sistema se



puede vender entre 900 a 1000 soles, se encuentra en el rango en el cual el usuario puede pagar por un sistema de vigilancia y detección.

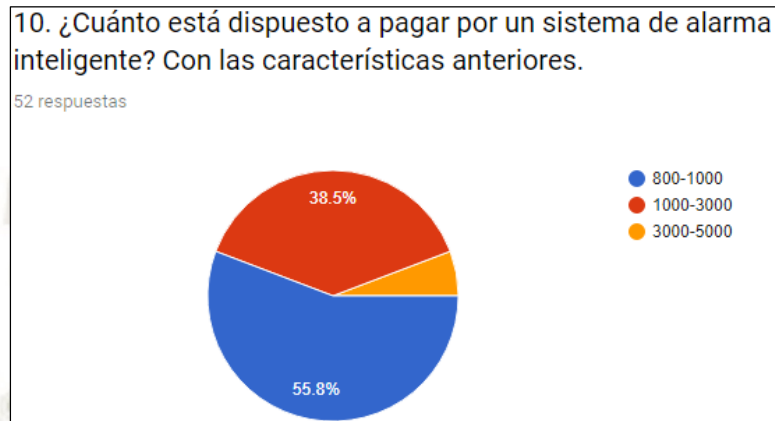


Figura B 10. Resultado de la décima pregunta Encuesta 1  
Fuente: Elaboración propia

La siguiente pregunta ayuda a saber si el usuario estaría interesado en adquirir un sistema de vigilancia y detección a un precio accesible. En porcentaje el 84.6% adquiriría un sistema y el 15.4% indica que no adquiriría. El porcentaje que si lo haría determina que hay posibles compradores y personas interesadas.

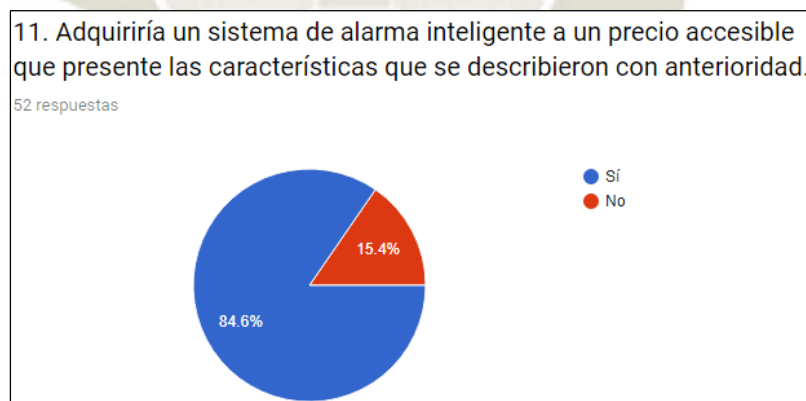


Figura B 11. Resultados de la décimo primera pregunta Encuesta 1  
Fuente: Elaboración propia

Por último, se le pregunto al usuario si prefiere pagar una cuota mensual o pagar una vez por un sistema de vigilancia y detección. El 75% prefiere pagar una sola vez y un 15% prefiere depender de una empresa que brinde el servicio. Con este ideal se indica que el sistema de vigilancia y detección sea de un solo pago (instalación), solo para el envío de mensajes de texto el dueño tiene que poner saldo de cinco soles a un chip movistar para el uso por mes.



Figura B 12. Resultados de la décimo segunda pregunta Encuesta 1  
Fuente: Elaboración propia

## Anexo C. ENCUESTA 2: Satisfacción del sistema de vigilancia y detección de fuga de gas

Para realizar las pruebas de usabilidad del sistema de vigilancia y detección de fuga de gas, se realizaron encuestas por medio del método de conveniencia al elegir 50 personas por ser allegadas al investigador y que usaron u observaron el sistema, para ello se elaboró un video donde se explica en que consiste y cómo funciona el sistema. Se realizaron preguntas sobre el uso del panel de control, la visualización de los mensajes del panel de control, el uso de la aplicación móvil, el uso del sistema de reconocimiento de voz y el funcionamiento del sistema en general. Las preguntas tienen relación a los requerimientos vistos en las pruebas de funcionalidad.

La primera pregunta es para verificar si al usuario le es fácil controlar el sistema por medio del panel de control. Las respuestas nos indican que un 36% le parece Muy Bueno, al 52% le parece Bueno y al 12% le parece Regular. Lo cual indica que a un porcentaje mayor al promedio al 88% le parece fácil ingresar a las opciones que se tienen en el panel de control, lo cual cumple con el requerimiento de ingresar opciones por el panel de control.

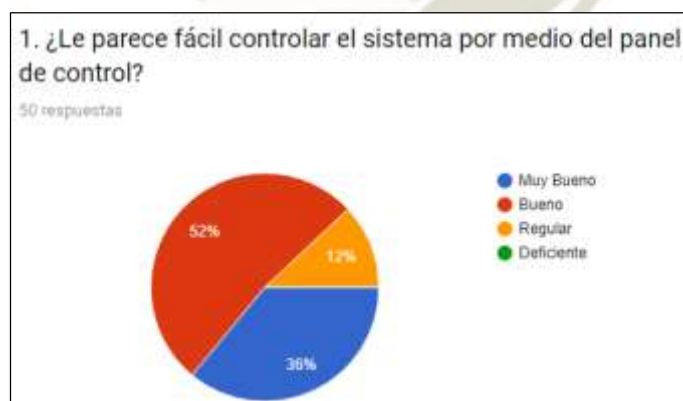


Figura C 1. Resultados de la primera pregunta Encuesta 2  
Fuente: Elaboración propia

La siguiente pregunta tiene relación con la pregunta anterior, hace referencia al requerimiento de controlar el sistema por medio de un panel de control. Se le pregunto a los usuarios, si los mensajes que muestra el panel de control son fáciles de comprender, un 38% le parece Muy Bueno, el 52% indica Bueno y un 10% indica Regular. Más del 50% indica que los mensajes son fáciles de comprender, para verificar el correcto funcionamiento del panel de control que presenta los mensajes con claridad.



Figura C 2. Resultados de la segunda pregunta Encuesta 2  
Fuente: Elaboración propia

La siguiente pregunta tiene relación con la aplicación móvil desarrollada para el sistema de vigilancia y detección de fuga de gas. Se le pide al usuario que califique la aplicación móvil, un 54% indicó que la aplicación móvil es Muy Buena, el 38% indicó que es Bueno y un 8% Regular. La aplicación móvil tiene más aceptación entre los usuarios, ya que es una aplicación intuitiva y fácil de utilizar.

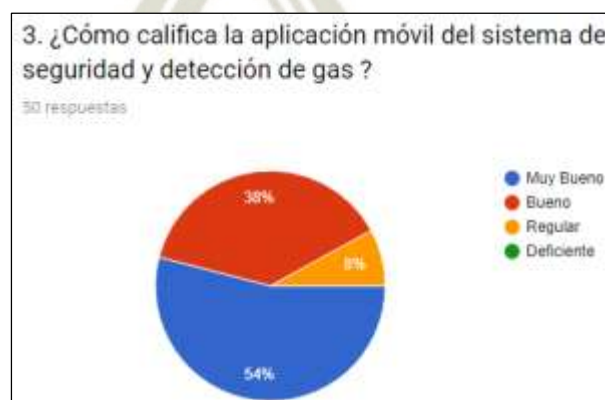


Figura C 3. Resultados de la tercera pregunta Encuesta 2  
Fuente: Elaboración propia

La siguiente pregunta hace referencia a la pregunta anterior y al requerimiento de gestionar el sistema por medio de una aplicación móvil. En esta pregunta el 100% de personas respondieron de forma afirmativa que la aplicación es amigable con el usuario. Lo cual rectifica que es una aplicación intuitiva, fácil de utilizar y amigable.

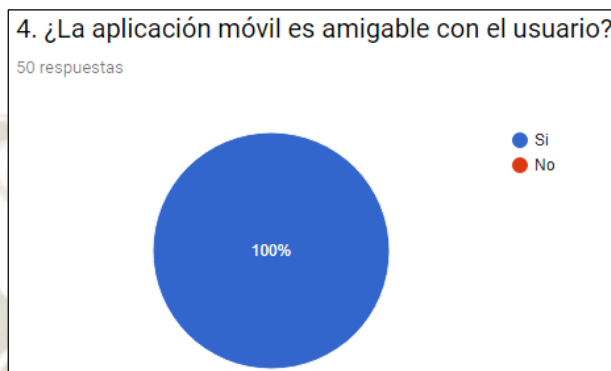


Figura C 4. Resultados de la cuarta pregunta Encuesta 2  
Fuente: Elaboración propia

La siguiente pregunta es para verificar el funcionamiento del sistema de reconocimiento de voz y el requerimiento de manejar el sistema por medio de comandos de voz. Se le pregunto al usuario si le parece sencillo utilizar los comandos de voz, un 88% indicó que Sí y un 12% indicó que no. Lo cual indica que el sistema de reconocimiento de voz para la mayoría de usuarios les pareció sencillo, con este ideal es que se implementó un sistema que se pueda manejar por comandos de voz y que simplifique el uso del sistema.

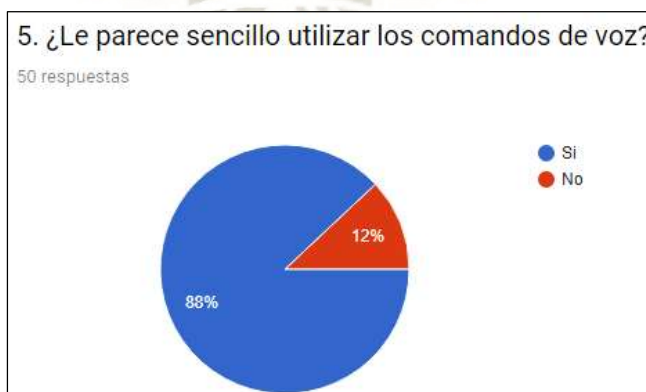


Figura C 5. Resultados de la quinta pregunta Encuesta 2  
Fuente: Elaboración propia

La última pregunta de la encuesta permite verificar como califica el usuario al sistema de vigilancia y detección de fuga de gas en general. Un 20% indica Excelente, un 50% Muy Bueno y un 30% Bueno. Lo cual representa que el sistema tiene una buena acogida por los usuarios.



Figura C 6. Resultados de la sexta pregunta Encuesta 2  
Fuente: Elaboración propia

Algunos encuestados dieron algunas recomendaciones sobre el sistema de vigilancia y detección de fuga de gas. Una de ellas es que se debería de dar mayor tiempo en que se activa la alarma para que el propietario pueda salir del hogar, ya que en el momento en que el propietario activa la alarma solo cuenta con 6 segundos para salir de la propiedad considerando que este tiempo es muy corto, que debería ser de promedio entre 1 a 2 minutos.

Otra recomendación por los usuarios es alargar el tiempo entre alertas cuando ingresa un intruso o se detecta la presencia de gas, el sistema está configurado para cuando ingrese un intruso pasen 30 segundos y si se vuelve a detectar la presencia de un intruso se envíe una alerta, en el mismo caso cuando se detecta gas pero deben de pasar 15 segundos. Estos tiempos se deben de constatar con el propietario y el tiempo que transcurre para que sea alertado, según lo que se desee y si desea aumentar el tiempo hasta en 1 minuto.

**Anexo D. Fichas de Validación del sistema de vigilancia y detección de fuga de gas para mejorar la seguridad en el hogar.**

I. Ficha de Validación de la Aplicación Móvil: *Cautela Smart Alarm*

I. DATOS GENERALES:

NOMBRE Y APELLIDOS: José Alfredo Sulla Torres

GRADO ACADEMICO: Doctor

FECHA DE REVISIÓN: 31/07/2019

II. ASPECTOS DE VALIDACIÓN

Indicador	Criterio	Excelente	Muy Buena	Buena	Regular	Deficiente
Diseño	La aplicación móvil tiene botones grandes		x			
	La aplicación móvil presenta imágenes precisas	x				
	La aplicación móvil presenta instrucciones claras	x				
	Correcto Ingreso de datos (Validación de contraseña, correo electrónico, etc.)	x				
Seguridad	La aplicación móvil cuenta con un Inicio de sesión	x				
	La aplicación móvil solo permite el ingreso de propietarios	x				
Funcionalidad	Por medio de la aplicación móvil se puede gestionar y controlar el sistema de seguridad remotamente.	x				
Relación con el usuario	La aplicación móvil es intuitiva		x			
	La aplicación móvil es amigable con el usuario	x				

Comentarios: Buen trabajo

2. Ficha de Validación del sistema: Sistema de vigilancia y detección de fuga de gas para mejorar la seguridad en el hogar

I. DATOS GENERALES:

NOMBRE Y APELLIDOS: José David Esquicha Tejada

GRADO ACADEMICO: Magister

FECHA DE REVISIÓN: 12/08/2019

II. ASPECTOS DE VALIDACIÓN DEL SISTEMA GENERAL

Criterio	Excelente	Muy Buena	Buena	Regular	Deficiente
El sistema permite mejorar la seguridad del hogar.		X			
El sistema tiene un panel de control fácil de utilizar.		X			
¿Por medio del panel de control se puede controlar el sistema de seguridad?	X				
Los comandos de voz son fáciles de utilizar.		X			
Los comandos de voz permiten controlar el sistema de seguridad.		X			
¿Qué le parece utilizar mensajes de texto como alerta?		X			
¿Le parece eficiente el envío de imágenes por correo electrónico?		X			
¿Cree que al agregar un sensor de gas, se reducen los riesgos en el hogar?	X				
¿Le parece eficiente el uso de un botón de alerta, que notifique a un familiar sobre una emergencia?	X				
Por medio de la aplicación móvil se puede gestionar y controlar el sistema de seguridad remotamente.	X				

Comentarios: Utilizar el computador de bajo consumo Raspberry Pi con el microcontrolador Arduino permite realizar proyectos innovadores como este, es importante resaltar que los proyectos son flexibles en brindar nuevas funciones como por ejemplo un sistema de grabación en caso de que suceda un incidente dentro de la casa.



**Anexo E. Manual de Usuario: sistema de vigilancia y detección de fuga de gas  
y Manual de usuario: aplicación móvil.**

**Manual de Usuario:  
Sistema de vigilancia y detección de fuga de gas**

**Introducción**

El sistema de vigilancia y detección de fuga de gas es desarrollado para mejorar la seguridad en el hogar de una familia unifamiliar, con el fin de generar un ambiente de confianza que no solo alerte sobre el ingreso de un intruso sino que alerte sobre un posible incendio, por medio de un sensor que detecta la presencia de gas. El sistema cuenta con un panel de control y un teclado, por donde es posible ingresar las siguientes opciones: A: Activar alarma, B: Desactivar alarma, C: Activar alarma incendio, D: Desactivar alarma incendio, \*: Visualizar temperatura y humedad, 0: botón de emergencia y #: opciones de reconocimiento de voz. Al ingresar a cualquiera de estas opciones se visualiza un mensaje por medio de la pantalla LCD.

**Requisitos e instalación**

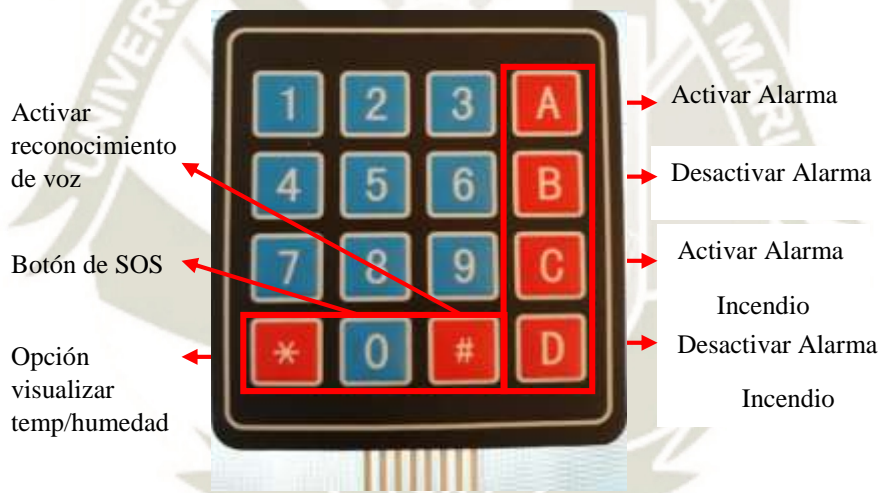
Debido a que es sistema envía un correo electrónico al propietario con la imagen del intruso y mensajes de texto según la alerta activada, se requiere como mínimo dos requisitos.

- Conexión a Internet
- Un chip de Movistar con saldo mínimo de 5 soles.

El sistema no requiere instalación, ya viene instalado en el Raspberry Pi y Arduino Uno que se utilizan en el sistema.

## 1. Ingreso de opciones por el panel de control (teclado)

El sistema de vigilancia y detección de fuga de gas cuenta con una pantalla LCD y un teclado matricial como panel de control para que el usuario interactúe directamente con el sistema sin necesidad de la aplicación móvil. En el panel de control, las opciones que se pueden ingresar por el teclado son: A: “Opción Activar Alarma”, B: “Opción Desactivar Alarma”, C: “Opción Activar Alarma Incendio”, D: “Opción Desactivar Alarma Incendio”, \*: “Opción visualizar temperatura y humedad”, #: “Opción reconocimiento de voz” y 0: “Opción Emergencia: botón de emergencia”. Cabe resaltar que en caso la alarma no está desactivada no se puede usar la opción de desactivar y si la alarma esta activada no se puede ingresar a la opción de activar.



*Figura E 1.* Opciones de Ingreso por teclado  
Fuente: Elaboración propia

### 1.1. Opción A: Activar sistema de vigilancia y detección de fuga de gas

En el panel de control como menú principal se puede observar el mensaje “ingrese una opción”, cuando el usuario presiona la tecla A ingresa a la opción “Activar Alarma”, función que permite activar el sistema de vigilancia y detección de fuga de gas en general.



*Figura E 2. Pantalla "Ingreso de opción"*

Fuente: Elaboración propia



*Figura E 3. Pantalla "Opción: Activar Alarma"*

Fuente: Elaboración propia

Al ingresar a esta opción se le solicita al usuario ingresar una contraseña, la cual debe ser ingresada por el teclado. El usuario cuenta con tres oportunidades para ingresar una contraseña correcta, en caso no la ingrese correctamente, el sistema se bloquea por veinte segundos.



*Figura E 4. Pantalla "Ingrese contraseña"*

Fuente: Elaboración propia



*Figura E 5. Pantalla "Ingreso correcto"*

Fuente: Elaboración propia

Al ingresar una contraseña correcta, se muestra un mensaje de “Alarma activa en: 5”, comenzando una cuenta regresiva de 5 a 1 (6 segundos), tiempo para que el usuario pueda salir de la propiedad antes de que este active el sistema. Cuando la cuenta llega a 1, el buzzer emite un pitido y se activa el sistema. Después, se muestra un mensaje de que

la alarma esta activada. En caso el usuario vuelva a presionar el botón A y la alarma esta activada se muestra un mensaje “Alarma en estado activado”.



*Figura E 6.* Pantalla "Alarma Activada en:"  
Fuente: Elaboración propia



*Figura E 7.* Pantalla "Alarma Activada"  
Fuente: Elaboración propia

## 1.2. Opción B: Desactivar sistema de vigilancia y detección de fuga de gas

En el panel de control como menú principal se visualiza el mensaje “Ingrese una opción”. Cuando la alarma esta activada y el usuario quieren desactivarla, debe de presionar la tecla B del teclado. Al presionarla, en la pantalla se muestra “Opción Desactivar Alarma”, después, le solicita al usuario que ingrese una contraseña por el teclado, al ser correcta, se desactiva el sistema y muestra un mensaje de “Alarma desactivada”. En caso la contraseña no sea correcta, el usuario tiene tres oportunidades para ingresar la contraseña correcta sino el sistema queda bloqueado por veinte segundos.



*Figura E 8.* Pantalla "Opción:  
Desactivar Alarma"  
Fuente: Elaboración propia



Figura E 9. Pantalla "Correcto Alarma desactivada"

Fuente: Elaboración propia

### 1.3. Opción C: Activar sistema de detección de fuga de gas

El sistema permite activar el sistema de detección de fuga de gas a parte del sistema general, como un subsistema, por si el usuario está en casa y solo quiere detectar si hay una fuga de gas, como medida de prevención ante un posible incendio.

En el panel de control como menú principal se visualiza "Ingrese una opción", para activar el sistema de detección de fuga de gas se presiona la tecla C del teclado. Al ingresar a esta opción, se muestra en pantalla "Opción Activar Alarma Incendio", después, se le solicita al usuario ingresar una contraseña, si la contraseña es correcta se activa y se muestra en pantalla el mensaje "Alarma incendio activada". En caso la contraseña sea incorrecta el usuario cuenta con tres oportunidades para ingresar la contraseña correcta. En caso el sistema de seguridad en general o el sistema de detección de fuga de gas estuvieran activados no se puede ingresar a esta opción.

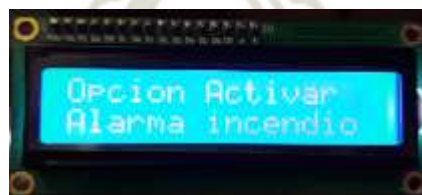


Figura E 10. Pantalla "Opción Activar alarma incendio"

Fuente: Elaboración propia



Figura E 11. Pantalla "Alarma incendio activada"

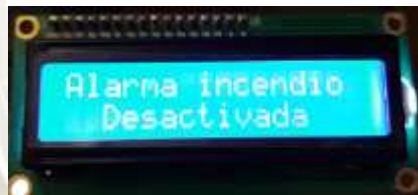
Fuente: Elaboración propia

#### 1.4. Opción D: Desactivar sistema de detección de fuga de gas

Para desactivar el sistema de detección de fuga de gas, primeramente debe estar activado. En el panel de control se visualiza en el menú principal el mensaje “ingrese una opción”, se presiona la opción D del teclado. En la pantalla se visualiza “Opción Desactivar Alarma Incendio”. Después, se le solicita al usuario que ingrese la contraseña, si es correcta se desactiva y se muestra el mensaje de “Alarma incendio desactivada”. En caso la contraseña no sea correcta, el usuario cuenta con tres oportunidades para ingresar la contraseña correcta.



*Figura E 12.* Pantalla "Opción: Desactivar alarma incendio"  
Fuente: Elaboración propia



*Figura E 13.* Pantalla: "Alarma incendio desactivada"  
Fuente: Elaboración propia

#### 1.5. Opción \*: Visualizar temperatura y humedad

El sistema cuenta con la opción de visualizar la temperatura y humedad de la zona de la cocina. Para ingresar a esta opción se presiona el botón “\*” del teclado, cuando se visualiza en el panel de control el menú principal “Ingrese una opción”. Al presionar el botón, se muestra en pantalla a que opción ingresó. Después, muestra la temperatura y humedad actual. Si alguno de los sistemas estuviera activado, el sistema no permite ingresar a esta opción.



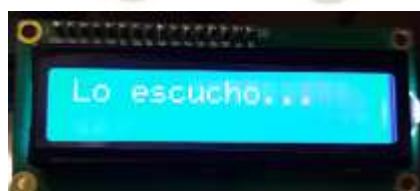
*Figura E 14.* Pantalla "Opción: tem/hume"  
Fuente: Elaboración propia



*Figura E 15.* Pantalla "temperatura y humedad"  
Fuente: Elaboración propia

### 1.6. Opción #: Reconocimiento de voz

El sistema cuenta con la opción de decir comandos de voz por medio de un eficiente sistema de reconocimiento de voz. En el panel de control cuando se visualiza el menú principal “Ingrese una opción”, se presiona el botón # de teclado para ingresar a esta opción. Al presionar el botón, el sistema muestra por medio de la pantalla el mensaje “Lo escucho...”, es ahí donde el usuario debe de decir un comando acercándose al micrófono. Los comandos que se pueden decir son: “activar alarma”, “desactivar alarma”, “activar gas”, “desactivar gas”, “dime temperatura”, “dime humedad”, “apagar luz” y “encender luz”. Si el usuario dice cualquier otra cosa el sistema muestra “No es una opción, vuelva a intentar”.



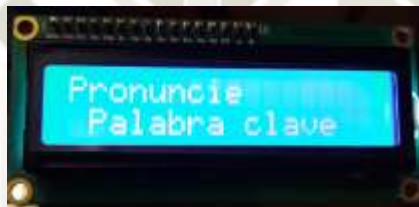
*Figura E 16.* Pantalla "Opción: Lo escucho"  
Fuente: Elaboración propia

Cuando se dice el comando activar o desactivar cualquiera de los sistemas, el sistema solicita al usuario que pronuncie la palabra clave como método de seguridad. El usuario cuenta con tres intentos para ingresar la palabra clave, en caso no diga la palabra clave, se muestra un mensaje de Intento # y el sistema regresa al menú principal. Al ingresar la palabra clave correcta, se realiza la opción que se ingresó.



*Figura E 17.* Pantalla "Dijo: activar alarma"

Fuente: Elaboración propia



*Figura E 18.* Pantalla "Pronuncie palabra clave"

Fuente: Elaboración propia



*Figura E 19.* Pantalla "Intento 1"

Fuente: Elaboración propia

Los otros comandos solo se pueden realizar si los sistemas están desactivados (visualizar temperatura/humedad y control de luz). Si el sistema de vigilancia y detección está activado solo se puede desactivar y en caso este desactivado solo se puede activar.

En el caso el sistema no encuentre el comando que dijo el sistema mostrara el siguiente mensaje “No es una opción: Vuelva intentar” y regresa al menú principal.



### 1.7. Opción 0: Botón de emergencia

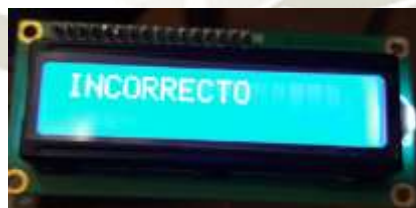
Para avisar a un familiar que se encuentra en una emergencia, puede presionar el botón 0 del teclado, cuando se visualiza en el menú principal el mensaje “ingrese una opción”. Al ingresar a esta opción, aun cuando el sistema este activado o desactivado, se envía un mensaje de texto de alerta. Al momento de enviar el mensaje no se muestra por el panel de control ninguna alerta, para que el intruso no sospeche, y el sistema se ejecuta con normalidad.



*Figura E 20.* Mensaje de Auxilio  
Fuente: Elaboración propia

### 1.8. Ingreso de contraseña por teclado

Para los sistemas de vigilancia y detección de fuga de gas que se activan o desactivan, el usuario tiene que ingresar una contraseña. Cuando el usuario ingresa una contraseña incorrecta sale el mensaje “Incorrecto”.



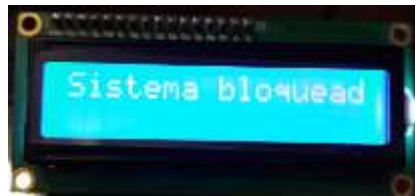
*Figura E 21.* Pantalla "Incorrecto"  
Fuente: Elaboración propia

El usuario cuenta con tres intentos para ingresar una contraseña valida. Al pasar estos tres intentos el sistema se bloquea por 20 segundos. Por pantalla se muestra el mensaje “Sistema bloqueado” y el tiempo que ha transcurrido.



*Figura E 22.* Pantalla "Intentos superados"

Fuente: Elaboración propia



*Figura E 23.* Pantalla "Sistema bloqueado"

Fuente: Elaboración propia



*Figura E 24.* Pantalla "Han Trascurrido"

Fuente: Elaboración propia

### **1.9. Alarma activada por la presencia de un intruso**

Cuando el sistema de vigilancia y detección está activo, y los sensores de infrarrojo (PIR) detectan un movimiento, el sistema muestra un mensaje “Alarma Activada”, suena el buzzer, envía un mensaje de texto como notificación de la presencia, prende un foco dependiendo de la hora (de 6 pm a 4 am), toma una imagen, la guarda en una carpeta del sistema operativo y la envía al correo electrónico del propietario. Para finalizar se muestra “la puerta cerrara en:” haciendo un conteo en retroceso de 3 al 1 y el servomotor cierra la puerta. Al pasar 30 segundos el sistema puede volver a alertar la presencia de un intruso, enviar mensajes de texto y correos electrónicos y abrir la cerradura de la puerta.



Figura E 25. Pantalla "Alarma Activada"

Fuente: Elaboración propia

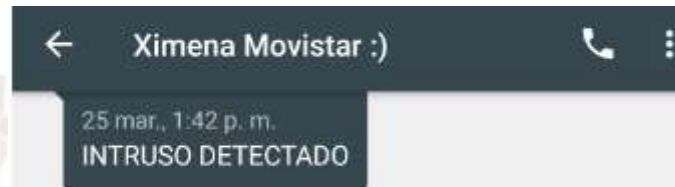


Figura E 26. Mensaje de texto "Intruso detectado"

Fuente: Elaboración propia



Figura E 27. Correo electrónico con la imagen del intruso

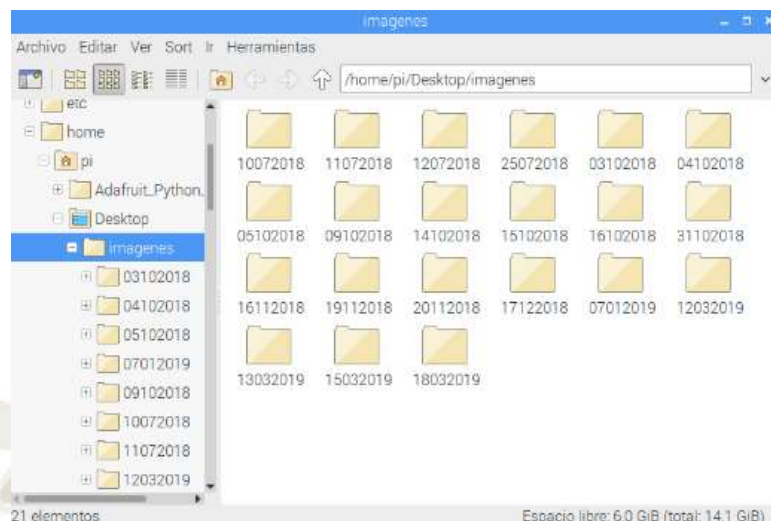
Fuente: Elaboración propia



Figura E 28. Pantalla "Puerta se cerrara en: "

Fuente: Elaboración propia

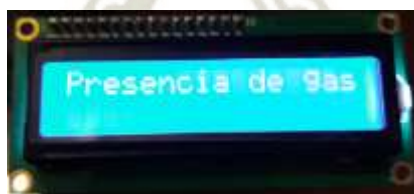
La imagen que capta el sistema es guardada en una carpeta en el sistema operativo de Raspberry Pi, la carpeta se encuentra en el escritorio con el nombre de imágenes. En caso el sistema no cuenta con una conexión a Internet y no pueda enviar un correo electrónico con la imagen, se puede visualizar en la carpeta indicada.



*Figura E 29.* Carpeta para guardar las imágenes  
Fuente: Elaboración propia

### 1.10. Alarma activada por presencia de gas

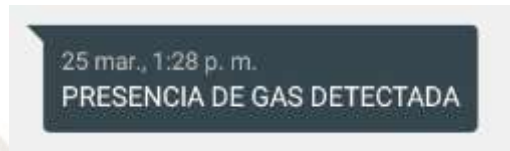
En el caso del sistema de detección de fuga de gas, se activa cuando el sensor MQ4 siente la presencia de gas. Se visualiza en la pantalla “Alarma incendio Activada”, y de ahí se muestra un mensaje de “presencia de gas” y se envía un mensaje de texto de advertencia. Por medio del sensor de temperatura se verifica, en este caso si la temperatura es mayor de 23 °C y si la humedad es mayor de 18%, muestra en pantalla “Posible presencia de fuego y humo” y muestra la temperatura y humedad. Cabe resaltar que la temperatura de un incendio es mayor de 35 °C y la humedad tiende a bajar (dependiendo del ambiente).



*Figura E 30.* Pantalla "Presencia de gas"  
Fuente: Elaboración propia



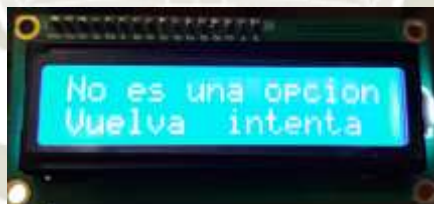
*Figura E 31.* Pantalla "Posible presencia de fuego y humo"  
Fuente: Elaboración propia



*Figura E 32.* Mensaje de texto Presencia de gas detectada  
Fuente: Elaboración propia

### **1.11. Opciones de 1 a 9: No disponibles**

Las opciones numéricas 1 a 9 del teclado, no realizan ninguna función. Entonces al presionarlas se muestra el mensaje "No es una opción válida" y regresa al menú principal.



*Figura E 33.* Pantalla "No es una opción"  
Fuente: Elaboración propia

## Manual de Usuario: Aplicación Móvil: Cautela Smart Alarm

### Introducción

El sistema de vigilancia y detección de fuga de gas cuenta con una aplicación móvil “Cautela Smart Alarm”, la aplicación móvil permite al usuario activar o desactivar la alarma, activar o desactivar el sistema de detección de fuga de gas y visualizar temperatura y humedad. También, se puede administrar la contraseña que se ingresa por el panel de control, el correo electrónico al que se le envía la imagen que toma el sistema y la palabra clave del sistema de reconocimiento de voz. Por medio de la aplicación móvil se puede controlar y gestionar el sistema de vigilancia y detección de fuga de gas de forma remota.

### Requisitos e instalación

La aplicación móvil se desarrolló para usuarios que cuenten con un teléfono móvil con un sistema operativo de Android 7.0 en adelante. Por el momento la aplicación móvil solo es para el sistema operativo Android, aún no se cuenta con una versión disponible para celulares con sistema operativo IOS. Para usar la aplicación móvil el usuario necesita una señal de Internet, sin ella no puede ingresar a la aplicación móvil, ya que la aplicación constantemente se conecta con el servicio de Firebase en la nube. Por ello, se requiere como mínimo dos requisitos.

- Conexión a Internet
- Teléfono móvil con un sistema operativo de Android 7.0 en adelante.

La aplicación móvil se instala en los dispositivos móviles por medio de su apk, después podría ser descargada de Google Play.

## Descripción del funcionamiento de la aplicación móvil

Al iniciar la aplicación como modo de presentación se muestran dos pantallas, antes de la página para iniciar sesión. La siguiente vista de la aplicación corresponde al inicio de sesión, en la cual el usuario debe ingresar su correo y contraseña. Después de ingresar los datos, el usuario debe de presionar el botón “Iniciar sesión” para que los datos sean autenticados por el sistema y acceda a la aplicación. Si el usuario ingresa datos erróneos se muestra en la parte de abajo de la aplicación una alerta que indica “Error de autenticación, revise contraseña y correo electrónico”. Lo que debe de hacer el usuario es ingresar un correo y contraseña correcta y volver a “Iniciar sesión”.



*Figura E 34.* Aplicación móvil  
"Inicio sesión"

Fuente: Elaboración propia

En caso el usuario no cuente con una cuenta para ingresar, puede crear una nueva cuenta al realizar clic en “Aun no tiene una cuenta? Registre una”. Se abre la vista para que el usuario registre un nuevo usuario. El usuario debe ingresar todos los datos que se piden correctamente: nombre, un correo electrónico válido, contraseña, confirmar contraseña y código familia (verificar que es miembro de la familia). El código familia es único por familia para que otros usuarios no creen una cuenta sino son parte de la familia; y presionar

el botón “Register”. Al registrar un nuevo usuario, la aplicación permite al usuario el acceso directamente, no regresa a la pantalla para que el usuario inicie sesión.

Antes de registrar un nuevo usuario la aplicación valida los datos ingresados, al ingresar cualquiera de los cinco datos incorrectos se muestra en la parte de abajo un mensaje de cuál es el dato incorrecto, para que el usuario lo vuelva a ingresar.



*Figura E 35.* Ingreso de nuevo usuario  
Fuente: Elaboración propia

Después de autenticar al usuario, se accede al sistema donde se controla y visualiza el estado del sistema de vigilancia y detección de fuga de gas. Además, se visualiza un menú para ingresar a las opciones de: cambiar contraseña, visualizar temperatura y humedad, cambiar palabra clave, controlar y visualizar el estado del sistema de detección de fuga de gas, cambiar correo electrónico y visualizar información de la aplicación.

Por medio del botón que se encuentra abajo del candado se activa o desactiva el sistema de vigilancia y detección. En la parte superior derecha se encuentra el botón para



cerrar sesión. También, se puede cerrar sesión haciendo doble clic en el botón de retroceso del celular.



Figura E 36. Aplicación móvil:  
Menú principal  
Fuente Elaboración propia

Al hacer clic en el botón cambiar contraseña se ingresa a la siguiente pantalla, en esta pantalla el usuario puede cambiar la contraseña con la cual se activa o desactiva el sistema por medio del teclado matricial. El usuario puede ingresar una contraseña con una longitud de cuatro dígitos y solo con los siguientes caracteres 0-9 y de a-d. Para cambiar la contraseña se confirma haciendo clic en el botón “OK”, si los datos son correctos se hace un cambio de contraseña y regresa al menú principal. Si no se quiere hacer ningún cambio se hace clic en el botón “Regresar” o en el botón de atrás del celular para ir al menú principal. Si al cambiar la contraseña el usuario ingresar cualquier otro carácter se mostrara un mensaje indicando “caracteres no permitidos, vuelva a ingresar”, para que el usuario vuelva a ingresar una contraseña correcta. Además, como método de confirmación se le pide al usuario ingresar dos veces la contraseña. En caso, las contraseñas ingresadas no sean igual se muestra un

mensaje “las contraseñas no son iguales!” para que el usuario las vuelva a ingresar correctamente.

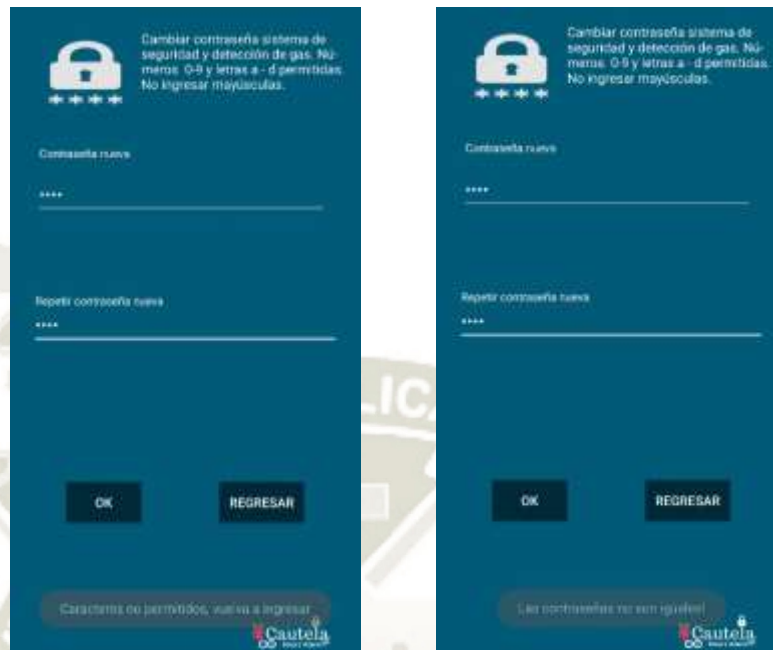


Figura E 37. Aplicación Móvil: Ingreso nueva contraseña  
Fuente: Elaboración propia

En el menú principal hay otro botón que se llama “temperatura”, al hacer clic lleva a la siguiente pantalla. En esta pantalla se puede observar la temperatura y humedad captada por el sensor que se encuentra en el área de la cocina. Para obtener los datos se presiona el botón “Encender”, siempre se debe de apagar, para hacerlo se presiona el botón “Apagar”. Para volver al menú principal se presiona el botón “Regresar” o hacer clic en el botón de atrás del celular.



*Figura E 38.* Aplicación móvil:  
Visualizar temperatura y  
humedad  
Fuente: Elaboración propia

En el menú principal se encuentra un botón “palabra clave voz”, que lleva a la siguiente pantalla. La palabra clave voz es para cuando se desea activar o desactivar las alarmas por medio del sistema de reconocimiento de voz, se utiliza como una contraseña. La palabra clave se elige aleatoriamente de cinco palabras determinadas al presionar el botón “generar palabra”, para cambiar la palabra clave se debe de presionar el botón “OK”, al cambiarla el sistema regresa automáticamente al menú principal. Si no se quiere hacer ningún cambio se presiona el botón “regresar” o el botón de atrás del celular para ir al menú principal.



*Figura E 39.* Aplicación Móvil:  
cambiar palabra clave  
Fuente: Elaboración propia

En el menú principal se visualiza un botón “presencia de gas” que lleva a la siguiente pantalla. En esta pantalla se visualiza el estado y se controla el activar o desactivar del sistema de detección de fuga de gas. Para activarlo se presiona el botón de encender, el estado cambiara a “Activo” y para desactivarlo se presione el botón “Apagar”, ahora el estado se mostrara “Desactivado”. Al presionar el botón regresar o el botón de atrás del celular se vuelve al menú principal.



Figura E 40. Pantalla  
Presencia de Gas  
Fuente: Elaboración propia

El menú principal cuenta con un botón “correo electrónico”, que lleva a la siguiente pantalla. En esta pantalla se visualiza el actual correo electrónico al cual se envía la imagen que capta el sistema al detectar la presencia de un intruso en la propiedad. Si se desea cambiar el correo electrónico por otro, se ingresa el nuevo correo y se presiona el botón “OK”. Se valida el correo, si es correcto, se cambia y regresa al menú principal, sino es correcto se muestra un mensaje “vuelva a ingresar el correo” para que el usuario ingrese otro correo electrónico. Si no se desea cambiar el correo electrónico se presiona el botón regresar o el botón de atrás del celular para ir al menú principal.