



## **Diseño de un sistema de seguridad inalámbrico para protección de viviendas y locales**

Mañay Portilla, Marlon Stalin

Departamento de Eléctrica y Electrónica

Carrera de Tecnología en Electrónica Mención Instrumentación y Aviónica

Monografía, Previo a la Obtención del Título de Tecnólogo en: Electrónica Mención  
Instrumentación y Aviónica

Ing. Alpúsig Cuichán, Silvia Emperatriz

Latacunga, 17 de marzo del 2021



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**  
**CARRERA DE TECNOLOGÍA EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y**  
**AVIÓNICA**  
**CERTIFICACIÓN**

Certifico que la monografía, **“Diseño de un sistema de seguridad inalámbrico para protección de viviendas y locales”** fue realizado por el señor **Mañay Portilla, Marlon Stalin** la cual ha sido revisada y analizada en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 17 de marzo del 2021

.....  
**Alpúsig Cuichán, Silvia Emperatriz**

**Directora**

C.C.: 0502779697



### Document Information

Analyzed document	Monografia_Mañay_Portilla_Marlon.pdf (D98653978)
Submitted	3/17/2021 3:13:00 PM
Submitted by	
Submitter email	msmanay@espe.edu.ec
Similarity	2%
Analysis address	sealpusig.espe@analysis.orkund.com

### Sources included in the report

<b>SA</b>	<b>TESIS-Astudillo-Rodriguez_87.docx</b> Document TESIS-Astudillo-Rodriguez_87.docx (D72202466)	 1
<b>SA</b>	<b>Tesis_Aguilar Freddy.docx</b> Document Tesis_Aguilar Freddy.docx (D29385722)	 6
<b>SA</b>	<b>Gualo Lligalo Alex Israel.pdf</b> Document Gualo Lligalo Alex Israel.pdf (D50722071)	 1
<b>W</b>	URL: <a href="https://repositorio.umsa.bo/bitstream/handle/123456789/21569/PG-2202.pdf?sequence=...">https://repositorio.umsa.bo/bitstream/handle/123456789/21569/PG-2202.pdf?sequence= ...</a> Fetched: 7/24/2020 6:05:23 PM	 1
<b>W</b>	URL: <a href="https://docplayer.es/33620635-Universidad-politecnica-salesiana-sede-quito.html">https://docplayer.es/33620635-Universidad-politecnica-salesiana-sede-quito.html</a> Fetched: 5/11/2020 5:42:32 AM	 1



Alpúsig Curchán, Silvia Emperatriz

Directora

C.C.: 0502779697



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**  
**CARRERA DE TECNOLOGÍA EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y**  
**AVIÓNICA**  
**RESPONSABILIDAD DE AUTORÍA**

Yo **Mañay Portilla, Marlon Stalin**, con cédula de ciudadanía n°1721610671, declaro que el contenido, ideas y criterios de la monografía: **Diseño de un sistema de seguridad inalámbrico para protección de viviendas y locales** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 17 de marzo del 2021

**Mañay Portilla, Marlon Stalin**

C.C.: 1721610671



## DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

### CARRERA DE TECNOLOGÍA EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y

### AVIÓNICA

#### AUTORIZACIÓN DE PUBLICACIÓN

Yo **Mañay Portilla, Marlon Stalin** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar la monografía: **Título: “Diseño de un sistema de seguridad inalámbrico para protección de viviendas y locales”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 17 de marzo del 2021

A handwritten signature in blue ink, which appears to read 'Mañay Portilla, Marlon Stalin', is positioned above the printed name.

**Mañay Portilla, Marlon Stalin**

C.C.: 1721610671

**Dedicatoria**

Dedico este momento a mis padres quien han sido mi apoyo más importante durante mi vida ya que siempre me ayudan y me guían a pesar de cualquier situación, ayudándome en lo que necesito y guiándome adecuadamente para conseguir mis metas, también se lo dedico a mis hermanos que a pesar de la distancia siempre están pendientes y buscan la manera de ayudarme si lo llevo a necesitar dándome ánimos y nunca dejándome solo, enseñándome que no debo rendirme a pesar de las adversidades y que debo luchar a pesar de cualquier circunstancia.

## **Agradecimientos**

Agradezco a Dios por sobre todas las cosas por cuidar a mí y a mi familia y mis seres queridos que me han acompañado durante este maravilloso proceso, en especial a mis padres que bajo toda circunstancia supieron apoyarme y entenderme con toda la paciencia que tienen junto con sus palabras de aliento diciéndome que todo sería recomenzado, gracias al apoyo de mi tutora la Ingeniera Alpúsig Silvia, que supo guiarme y ayudo en la dirección durante el proceso del proyecto de tesis facilitando enormemente el cumplimiento de una de mis metas más importantes que es graduarme de la universidad, a mis amigos que a pesar de la distancia siempre mantuvimos contacto y a través de la comunicación se transmitió información y consejos muy valiosos para el desarrollo de un buen proyecto. A todos mis maestros y compañeros que compartimos aula en la duración de la carrera, los cuales me ayudaron a crecer como ser humano.

**Tabla de contenidos**

<b>Carátula.....</b>	<b>1</b>
<b>Certificación.....</b>	<b>2</b>
<b>Urkund.....</b>	<b>2</b>
<b>Responsabilidad de autoría.....</b>	<b>4</b>
<b>Autorización de publicación.....</b>	<b>5</b>
<b>Dedicatoria.....</b>	<b>6</b>
<b>Agradecimientos .....</b>	<b>7</b>
<b>Resumen.....</b>	<b>16</b>
<b>Abstract.....</b>	<b>17</b>
<b>Introducción.....</b>	<b>18</b>
<b>Antecedentes .....</b>	<b>18</b>
<b>Planteamiento del problema.....</b>	<b>21</b>
<b>Justificación .....</b>	<b>22</b>
<b>Objetivos .....</b>	<b>23</b>
<i><b>Objetivo general .....</b></i>	<b>23</b>
<i><b>Objetivos específicos .....</b></i>	<b>23</b>
<b>Alcance .....</b>	<b>23</b>
<b>Fundamentación teórica .....</b>	<b>25</b>
<b>Sistemas de Protección.....</b>	<b>25</b>
<i><b>Protección para el hogar .....</b></i>	<b>25</b>
<i><b>Protección Comercial .....</b></i>	<b>25</b>
<b>Seguridad .....</b>	<b>26</b>
<b>Arduino .....</b>	<b>26</b>

<i>Cómo funciona Arduino</i> .....	28
<i>Arduino Mega</i> .....	29
ESP8266 módulo WiFi para Arduino.....	31
Comunicación WIFI ESP8266 .....	32
Teclado matricial 4x4.....	33
Pantalla LCD 20x4 - Display LCD 20x4 .....	35
Cámara IP .....	37
<i>Funcionamiento una cámara IP</i> .....	38
Sensor Magnético Reed-switch .....	39
<i>Funcionamiento</i> .....	39
Sirena.....	41
Android Studio .....	42
Firestore .....	45
UPS.....	47
Relés .....	48
<i>Función de un relé</i> .....	49
Reloj de tiempo real RTC.....	49
Comunicación I2C.....	51
Desarrollo del tema .....	53
Programa de Arduino Mega 2560.....	54
Programa en ESP8266 módulo WiFi para Arduino .....	60
Base de datos.....	64

Aplicación móvil.....	67
Armado del sistema en físico y prueba del sistema en marcha .....	80
Implementación en un domicilio local .....	85
Conclusiones y Recomendaciones.....	92
Conclusiones .....	92
Recomendaciones .....	93
Bibliografía.....	94
Anexos.....	99

**Índice de tablas**

<b>Tabla 1</b> <i>Especificaciones del arduino mega</i> .....	30
<b>Tabla 2</b> <i>Especificaciones de voltaje del arduino mega</i> .....	30
<b>Tabla 4</b> <i>Especificaciones técnicas</i> .....	36
<b>Tabla 5</b> <i>Especificaciones técnicas</i> .....	42
<b>Tabla 6</b> <i>Principales características de firebase</i> .....	46

## Índice de figuras

<b>Figura 1</b> <i>Placa electrónica de Arduino UNO</i> .....	27
<b>Figura 2</b> <i>Conexión de un módulo a la placa electrónica de Arduino UNO</i> .....	28
<b>Figura 3</b> <i>Diferente tipos de placas de Arduino.</i> .....	29
<b>Figura 4</b> <i>Partes de un Arduino Mega.</i> .....	29
<b>Figura 5</b> <i>Módulo ESP8266</i> .....	31
<b>Figura 6</b> <i>Pines del módulo ESP8266.</i> .....	32
<b>Figura 7</b> <i>Comunicación serial entre Arduino Mega y módulo ESP8266.</i> .....	33
<b>Figura 8</b> <i>Diagrama de un pequeño teclado numérico de 16 teclas</i> .....	34
<b>Figura 9</b> <i>Diagrama de conexión de un teclado matricial con arduino</i> .....	34
<b>Figura 10</b> <i>Diagrama de conexión de un teclado matricial con Arduino.</i> .....	35
<b>Figura 11</b> <i>Ejemplo de una cámara IP.</i> .....	37
<b>Figura 12</b> <i>Funcionamiento de un sensor magnético Reed-switch.</i> .....	39
<b>Figura 13</b> <i>Esquema interno de un sensor magnético Reed-switch</i> .....	40
<b>Figura 14</b> <i>Partes de un sensor magnético Reed-switch.</i> .....	40
<b>Figura 15</b> <i>Sirena para interiores.</i> .....	41
<b>Figura 16</b> <i>Alarma visible</i> .....	42
<b>Figura 17</b> <i>Los archivos de proyecto en la vista de Android</i> .....	43
<b>Figura 18</b> <i>Ventana principal de Android Studio</i> .....	44
<b>Figura 19</b> <i>Funcionamiento interno de un UPC</i> .....	48
<b>Figura 20</b> <i>Partes de un Relé</i> .....	49
<b>Figura 21</b> <i>DS1307</i> .....	50
<b>Figura 22</b> <i>La conexión de un módulo con DS1307</i> .....	51
<b>Figura 23</b> <i>Conexión de tres dispositivos a un bus de comunicación I2C</i> .....	52
<b>Figura 24</b> <i>Diagrama de bloques del sistema de seguridad</i> .....	53
<b>Figura 25</b> <i>Librería utilizada en arduino mega2560</i> .....	54

<b>Figura 26</b> <i>Declaración de la contraseña y variables del teclado matricial.</i> .....	54
<b>Figura 27</b> <i>Pines de entrada utilizados en la programación.</i> .....	55
<b>Figura 28</b> <i>Pines de entrada utilizados en la programación.</i> .....	55
<b>Figura 29</b> <i>Condiciones de activación de la alarma</i> .....	56
<b>Figura 30</b> <i>Funciones de la contraseña</i> .....	57
<b>Figura 31</b> <i>Mensaje de alerta en el display</i> .....	57
<b>Figura 32</b> <i>Activación de luces led</i> .....	58
<b>Figura 33</b> <i>Comprobación de la contraseña y mensajes de indicación.</i> .....	58
<b>Figura 34</b> <i>Activación de la alarma por el uso de la contraseña</i> .....	59
<b>Figura 35</b> <i>Desactivación de la alarma</i> .....	59
<b>Figura 36</b> <i>Cuadro de texto de bienvenida</i> .....	60
<b>Figura 37</b> <i>Configuración del módulo RTC para el tiempo</i> .....	60
<b>Figura 38</b> <i>Librería utilizada en la programación del módulo ESP8266</i> .....	61
<b>Figura 39</b> <i>Dato utilizado para la conexión con internet</i> .....	61
<b>Figura 40</b> <i>Declaración de entradas utilizadas</i> .....	61
<b>Figura 41</b> <i>Programación para conectar el módulo con una base dedatos</i> .....	62
<b>Figura 42</b> <i>Conexión del módulo con el microcontrolador mega 2560</i> .....	62
<b>Figura 43</b> <i>Configuración para la lectura de los datos</i> .....	63
<b>Figura 44</b> <i>Condiciones de cada sensor</i> .....	63
<b>Figura 45</b> <i>Configuración para muestra de errores</i> .....	64
<b>Figura 46</b> <i>Tipo de datos utilizados en la programación</i> .....	64
<b>Figura 47</b> <i>Creación de un nuevo proyecto en la base de datos</i> .....	65
<b>Figura 48</b> <i>Nombramiento del proyecto</i> .....	65
<b>Figura 49</b> <i>Registro de la base de datos</i> .....	66
<b>Figura 50</b> <i>Realtime Database</i> .....	66
<b>Figura 51</b> <i>Información subida en base de datos</i> .....	67

<b>Figura 52</b> Creación de un proyecto nuevo .....	67
<b>Figura 53</b> Nombramiento de los datos de la aplicación .....	68
<b>Figura 54</b> Diseño de un Splash Screen .....	68
<b>Figura 55</b> Programación del Splash Screen .....	69
<b>Figura 56</b> Diseño del inicio de sesión .....	69
<b>Figura 57</b> Configuración del inicio de sesión .....	70
<b>Figura 58</b> Configuración de los botones del inicio de sesión .....	70
<b>Figura 59</b> Sincronización de la base de datos con la aplicación .....	71
<b>Figura 60</b> Declaración de librerías para funciones con la base de datos .....	71
<b>Figura 61</b> Condiciones al ingresar la contraseña .....	72
<b>Figura 62</b> Diseño de un registro .....	72
<b>Figura 63</b> Configuración para la contraseña .....	73
<b>Figura 64</b> Verificación de datos .....	73
<b>Figura 65</b> Diseño de una actividad para recuperar la contraseña .....	74
<b>Figura 66</b> Configuración del botón de recuperación de contraseña .....	74
<b>Figura 67</b> Diseño de una actividad de menú .....	75
<b>Figura 68</b> Configuración de los botones de la actividad de menú .....	75
<b>Figura 69</b> Diseño para una visualización en vivo de los sensores .....	76
<b>Figura 70</b> Configuración para los datos q se mostraran en la lista .....	77
<b>Figura 71</b> Referencias para toma de datos de la base de datos .....	77
<b>Figura 72</b> Diseño de un historial .....	78
<b>Figura 73</b> Base de datos interna .....	78
<b>Figura 74</b> Configuración de valores para el historial .....	79
<b>Figura 75</b> Información almacenada en la base de datos del teléfono .....	79
<b>Figura 76</b> Comunicación serial entre Arduino MEGA y PC .....	80
<b>Figura 77</b> Puesta en marcha del LCD .....	80

<b>Figura 78</b> <i>Prueba de contraseña con un código incorrecto.....</i>	81
<b>Figura 79</b> <i>Funcionamiento del módulo RTC .....</i>	81
<b>Figura 80</b> <i>Conexión de los sensores magnéticos a la placa principal.....</i>	82
<b>Figura 81</b> <i>Conexión de luces led.....</i>	82
<b>Figura 82</b> <i>Conexión de relés a la placa principal.....</i>	83
<b>Figura 83</b> <i>Conexión de módulo WiFi ESP8866.....</i>	83
<b>Figura 84</b> <i>Base de datos “FIREBASE”.....</i>	84
<b>Figura 85</b> <i>Aplicación android.....</i>	84
<b>Figura 86</b> <i>Puesta en marcha del sistema de seguridad en forma de maqueta.....</i>	85
<b>Figura 87</b> <i>Caja de protección para dispositivos táctiles.....</i>	85
<b>Figura 88</b> <i>Caja principal del sistema de seguridad.....</i>	86
<b>Figura 89</b> <i>Instalación de sensores magnéticos.....</i>	86
<b>Figura 90</b> <i>Conexión de los dispositivos hacia la caja principal.....</i>	87
<b>Figura 91</b> <i>Conexión a 110v.....</i>	87
<b>Figura 92</b> <i>Conexión del módulo de relés.....</i>	88
<b>Figura 93</b> <i>Conexión de cada módulo al microcontrolador Arduino Mega.....</i>	88
<b>Figura 94</b> <i>Conexión de cables del teclado a Arduino Mega.....</i>	89
<b>Figura 95</b> <i>Conexión de sensores a la caja principal.....</i>	89
<b>Figura 96</b> <i>Fijación de la caja de dispositivos táctiles.....</i>	90
<b>Figura 97</b> <i>Fijación de la caja principal junto con las sirenas de alarma.....</i>	90
<b>Figura 98</b> <i>Instalación de cámara IP.....</i>	91

## **Resumen**

El presente proyecto técnico se basa en un sistema de seguridad basado en la tecnología de Arduino para la protección de viviendas o locales. Para el diseño del programa se realizó un análisis de cómo es el funcionamiento de un sistema de alarma, por consiguiente se utilizó sensores magnéticos que entregan una señal de entrada digital al controlador y una sirena que funciona como actuador, estos datos se envían al arduino el cual dependiendo de cada caso activara o no el paso de corriente a la sirena lo cual provoca una alerta, esto permite una facilidad para que el propietario pueda tomar acciones pertinentes, de igual manera este sistema conto con una contraseña de activación y desactivación de la alarma esto a través de un teclado matricial junto con una visualización mediante un display LCD en donde muestra información de los sensores o cuando el sistema este activado o desactivado, todos estos datos obtenidos son enviados a una base de datos mediante un módulo WiFi, para que el usuario pueda observar en tiempo real por medio de internet lo que ocurre en el domicilio, esto mediante una aplicación móvil llamada "Sistema de seguridad", la aplicación únicamente permite un monitoreo , de igual manera se implementó cámaras IP para una visualización de video en tiempo real que facilito aún más la vigilancia.

Palabras clave:

- **ALARMA DOMICILIARIA**
- **MONITOREO DE SENSORES**
- **SEGURIDAD DOMICILIARIA**
- **SENSORES MAGNÉTICOS**
- **ARDUINO MEGA2560**

**Abstract**

This technical project is based on a security system based on Arduino technology for the protection of homes or premises. For the design of the program an analysis of how is the operation of an alarm system was performed, therefore magnetic sensors were used to deliver a digital input signal to the controller and a siren that works as an actuator, these data are sent to the arduino which depending on each case will activate or not the passage of current to the siren which causes an alert, this allows an ease for the owner to take appropriate action, Similarly, this system has a password activation and deactivation of the alarm through a matrix keyboard along with a display using an LCD display which shows information from the sensors or when the system is on or off, all these data obtained are sent to a database via a WiFi module, so that the user can observe in real time via internet what happens in the home, this through a mobile application called "Security System", the application only allows monitoring, in the same way IP cameras were implemented for a video display in real time that further facilitated the surveillance.

Key words:

- **HOME ALARM**
- **SENSOR MONITORING**
- **HOME SECURITY**
- **MAGNETIC SENSORS**
- **ARDUINO MEGA2560**

## CAPÍTULO I

### 1 Introducción

#### 1.1 Antecedentes

Desde siempre el ser humano ha tenido la necesidad de cuidar y salvaguardar su integridad física y sus bienes materiales como los hogares o locales, esto ha impulsado al desarrollo de sistemas de seguridad y por lo tanto a empresas que se dediquen a su fabricación e implementación, debido a ello existen muchas empresas que diseñan varios tipos de sistemas de seguridad, en la actualidad esto generó una gran competencia en el mercado y como resultado este tipo de productos y servicios cuentan con un costo exorbitante y gracias a esto no todos los hogares cuentan con la condición económica para solicitar un servicio de seguridad.

Tomando en cuenta los diferentes hechos mencionados el objetivo de un sistema de seguridad es el de brindar respaldo y protección para de esta manera mejorar el estilo de vida de las personas previniendo que pongan en riesgo su seguridad o sus vidas al tratar de proteger sus bienes, se ha indagado y analizado varios temas, proyectos y trabajos referentes a sistemas de seguridad tomando en cuenta su desarrollo y su posterior puesta en marcha, para de esta manera comprender como han contribuido a la comunidad, y por ende se tomó a consideración para el diseño y creación de un nuevo proyecto de seguridad, y han sido un gran sustento como base bibliográfica debido a sus teorías.

A continuación, se describe a cada autor con su respectivo tema y su propuesta:

Con referencia al trabajo (Conejos, 2014) planteó un “Sistema de seguridad perimetral programable inteligente” el cual lo realizó como tesis para la “Escuela Técnica Superior de Ingeniería Informática universidad Politécnica de Valencia” en septiembre de 2014. Dicho proyecto funciona de tal manera que la alarma se activa y el programa realiza acciones de manera automática, en este caso al acercarse un

individuo a una área protegida el sistema enviará un mensaje de alerta, el mensaje es enviado a un número específico el cual puede ser el propietario del lugar, esto demostró que la telefonía va tomando una parte fundamental del entorno de vida, debido a la digitalización pero dicha comunicación también es utilizada para el control de actuadores siendo así un sistema de control.

El trabajo de (Gutiérrez, 2016) planteó un “Sistema De Seguridad Domiciliaria Basada En Tecnología Arduino Y Aplicación Móvil” el cual lo realizó como tesis para la “Universidad Mayor De San Andrés Facultad De Ciencias Puras Y Naturales Carrera De Informática” en LA Paz – Bolivia 2016. El trabajo se basa en una investigación realizada a un estudio acerca de la domótica y que busca brindar a los usuarios un sistema de seguridad utilizando el microcontrolador de Arduino, para ello implementó un prototipo que simula el funcionamiento domótico, todo esto utilizando herramientas de software y hardware libre junto con componentes que existen en el mercado para ello utiliza la tecnología Bluetooth lo cual es una manera interesante de automatizar los procesos de domótica en el hogar para el control utilizo servomotores sensores y medios inalámbricos, utilizando Android y la aplicación App Inventor 2 esto para que el dispositivo Android y el microcontrolador arduino pueda comunicarse adecuadamente sin problemas.

La propuesta de (Monje, 2014 ) es el “Diseño e implementación de un sistema de seguridad para el hogar con interfaz en Android” el cual lo realizo como tesis para la “Universidad Del País Vasco Facultad De Informática” en junio del 2014. Este proyecto se ha creado para el diseño de un sistema de seguridad domótico, que se puede instalar en hogares, el objetivo principal de este programa es la de controlar la seguridad y cuenta con un sistema para diferenciar falsas alarmas, todo esto siendo monitoreado mediante el uso de una interfaz para conexión móvil, en este caso utiliza un sistema Android con el que monitorea la seguridad, este sistema cuenta con un

detector de intrusos o movimientos, una vez detectado algún riesgo activa una alarma disuasoria la cual sirve para alertar al propietario y toma acciones pertinentes según sea el caso. Para probar el sistema se ha implantado en un entorno real, siendo el prototipo implementado completamente funcional.

El documento de (Montoya Rivera, 2015) está creado como “Diseño de un sistema de alarma inalámbrica para pequeñas empresas y hogares en Pereira y Dosquebradas” el cual lo realizó como tesis para la “Universidad Tecnológica de Pereira” en Pereira 2015. En este documento se ha diseñado un sistema de seguridad con monitoreo inalámbrico, el cual funcionará de manera remota y en tiempo real utilizando una red IP este podrá ser implementado tanto en casas pequeñas como en empresas, su proyecto se basa en una alarma de video vigilancia inalámbrica esto mediante una IP estos datos son enviados al microcontrolador arduino, el sistema ha sido pensado con la intención de reducir al máximo el cableado y de esta manera facilitar la instalación, por lo que la comunicación entre los distintos dispositivos se hace de manera inalámbrica

El tema desarrollado por (Morán, 2018) tiene con fin “Diseñar Un Sistema De Alarma Inalámbrico De Bajo Costo Para La Protección De Viviendas Tipo, En Sectores De Bajo Recursos Económicos De La Ciudad De Guayaquil” el cual lo realizó como tesis para la “Universidad De Guayaquil Facultad De Ciencias Matemáticas Y Físicas Carrera De Ingeniería En Networking Y Telecomunicaciones” en Guayaquil, septiembre del 2018. El tema habla sobre un sistema operado con un controlador Arduino que trabaja con un módulo SIM900 el cual permite una comunicación mediante envío de llamadas o mensajes a números específicos este sistema es un prototipo por lo que está implementado como maqueta sus sistema está conformado por una placa principal llamada arduino conectada con varios sensores complementado con un SIM900 que se intercomunica con la placa para enviar información por medio de mensajes de texto y en

el caso de que se active la alarma cuenta con cámaras IP para complementar el monitoreo.

## **1.2 Planteamiento del problema**

Los sistemas de seguridad han evolucionado de manera muy eficiente permitiendo el control remoto de los mismos por ende se han generado grandes empresas que brindan servicios de protección y cuidado ya sea de viviendas o personas, pero a pesar de que en varios sectores son algo primordial por su alto costo la mayor parte de hogares no cuenta con uno, pero a medida que la seguridad ha crecido la delincuencia también ha aumentado y en la actualidad la delincuencia se ha vuelto parte de todo tipo de sectores tanto para casas como locales, siendo así un gran problema que afecta el diario vivir en la ciudadanía en general, este problema de inseguridad ha afectado enormemente al desarrollo económico principalmente de sectores donde la policía no frecuenta o sectores donde la concurrencia de personas no es abundante, esto ha contribuido a que la delincuencia aumente en barrios lo que afecta a microempresas y domicilios por ello se ha visto la necesidad de la incorporación de servicios de seguridad, pero no todos los sectores cuentan con los recursos económicos necesarios para implementar un sistema de seguridad en sus domicilios, tomando en cuenta estos aspectos económicos y buscando la seguridad en los hogares y locales se planteó el desarrollo de un sistema de seguridad de bajo costo.

En la actualidad existen varios sistemas de bajo costo para monitorear las viviendas, este tipo de sistemas utilizan redes de sensores inalámbricos los cuales pueden medir variables de movimiento, temperatura, humedad, humo entre otros, cumpliendo la función de sistemas de alarma. Un sistema de alarma permite el control y monitoreo de dichos sensores permitiendo así el control de acceso de personas que no están autorizadas o intrusos, estos datos pueden ser almacenados por video mediante cámaras IP que en conjunto con un sensor de movimiento permite emitir señales de

alerta, notificando que existen intrusos en dicho sitio, esta alerta será enviada por medio de mensaje al residente y de esta manera pueda tomar las decisiones y medidas de control necesarias para proteger su hogar.

De manera general, la mayor parte de viviendas no cuenta con un servicio de seguridad, esto debido a que solicitar un servicio de seguridad implica un alto costo tanto por el mantenimiento como por su implementación y por ende no han podido ser adoptados por la ciudadanía.

### **1.3 Justificación**

La aparición de casas que cuenten con sistemas automatizados es decir que tengan incorporados sensores, actuadores y controladores remotos ofrecen una domotización y eventualmente se convertirá en algo que incorporen todos los domicilios, pero estos instrumentos no sirven únicamente para la domotización de casas tienen un sinnúmero de usos y en especial depende de la imaginación de cada persona, un uso que se les puede dar a este tipo de instrumentos es el de la seguridad ya sea para el control de un domicilio o para la monitorización del mismo, de esta manera se ofrece una solución ante robos, incendios y ayudan a proteger los domicilios esto mediante sensores de humo, movimiento e incluso de temperatura, en la actualidad el desarrollo de la tecnología facilita la implementación de este tipo de sistemas en hogares teniendo en cuenta que estos sistemas no llegaran a costos exorbitantes.

Por lo tanto se diseñara un sistema de seguridad inalámbrico para protección de viviendas y locales, este tipo de sistemas permitirá la facilidad de que todo tipo de hogares pueda contar con un sistema de seguridad y sin la necesidad de precios altos, de esta manera se proporcionará a los moradores un mayor confort y tranquilidad en lo referente a su seguridad, este sistema se ha diseñara como una alternativa de un sistema de seguridad profesional que se distribuyen de manera local, para ello se

analizó los elementos técnicos y de esta manera tener la información necesaria para diseñar el sistema de alarma inalámbrica, este tipo de sistemas es disuasivo.

## **1.4 Objetivos**

### **1.4.1 Objetivo general**

- Diseñar un sistema de seguridad inalámbrico para protección de viviendas y locales.

### **1.4.2 Objetivos específicos**

- Establecer información acerca de un sistema de seguridad inalámbrico.
- Analizar los procesos que componen un sistema de seguridad inalámbrico.
- Implementar un sistema de seguridad domiciliaria basada en tecnología Arduino y aplicación móvil para brindar mayor seguridad en el hogar de las personas.

## **1.5 Alcance**

Este proyecto tiene como finalidad un prototipo que permita el control de seguridad de una vivienda mediante el uso implementado del microcontrolador Arduino, este sistema tiene planteado el ayudar a la seguridad de domicilios por lo tanto este sistema podrá mantenerse funcional las 24 hora los siete días a la semana y podrá controlarse de manera remota.

El sistema detectará la presencia de intrusos mediante el uso de sensores, los cuales son capaces de detectar el movimiento en el día y de noche, esto con la finalidad de proteger las viviendas domicilios o locales. La información tomada por los sensores será enviada mediante una aplicación al celular para que de esta manera el dueño pueda tomar medidas de protección o para que se pueda comunicar directamente con una unidad de policía comunitaria (UPC), una vez que se ha verificado el suceso. Este sistema está diseñado para ser implementado en domicilios o locales pequeños que no cuenten con un presupuesto para implementar un sistema de seguridad profesional,

este sistema puede activarse y desactivarse mediante el uso de contraseñas y también mediante el uso del celular por medio de comando, el sistema al detectar algún intruso dentro de una zona restringida o a una hora inadecuada se activará. El sistema no enviará señales a la policía nacional, solo al propietario el cual deberá tomar medidas pertinentes según sea el caso, la razón por la que no se envía datos directo a la policía es debido a que en varios casos han activado por errores del usuario o por niños los cuales al no saber del sistema de seguridad y de su comunicación con la policía pueden generar una falsa señal de alarma lo cual es penado por la ley, debido a eso el sistema envía los datos directo al propietario el cual sabrá las medidas a tomar.

Los límites del presente proyecto de un sistema de seguridad para un hogar no contempla un sistema completo de seguridad ya que no funciona como alerta y no puede actuar para casos como:

- Casos de incendio para lo cual se necesitaría de otro tipo de equipos.
- Para problemas de instalaciones eléctricas en el hogar.
- Este sistema no sustituye a un sistema completo y profesional de alto costo, pero brindara funciones un sistema de alerta para el cliente.

Para el desarrollo de un sistema de seguridad inalámbrico para protección de viviendas y locales se realizará mediante el control de sensores, los cuales al cambiar de estado lanzaran una señal a la placa principal, para ello se empleará conocimientos de programación para el microcontrolador de Arduino, el cual se utilizará como plataforma de control. Dentro del presente capítulo se verá el concepto de Arduino, junto con sus características, hardware, software y la descripción de los dispositivos utilizados como los sensores, sistemas de producción, y las diferentes tecnologías utilizadas para implementar el presente sistema y los diferentes tipos de seguridad que puede integrar este sistema.

## CAPÍTULO II

### 2 Fundamentación teórica

#### 2.1 Sistemas de Protección

La implementación del sistema de seguridad para domicilios o locales basada en tecnología Arduino mejorara en gran proporción la seguridad en los hogares de las personas, esto mediante un monitoreo a distancia que funcionara con la ayuda de una aplicación móvil, por lo cual dependiendo de la necesidad de cada hogar se manejan en diferentes niveles. A continuación, se describen los principales niveles de protección.

##### ***2.1.1 Protección para el hogar***

La evolución en la tecnología ha permitido desarrollar una nueva ciencia en sistemas distribuidos comúnmente llamada computación móvil, la tecnología utilizada en agentes móviles tiene una especial importancia en todo lo referente a entornos inalámbricos, en los cuales los recursos de comunicación son necesarios. Una vez definido este concepto se toma en cuenta que esta tecnología es de gran importancia actualmente y forma gran parte del diario vivir de las personas, esto debido a diferentes aspectos como los son la comunicación, seguridad y esto ha llevado a grandes investigaciones para mejorar estos servicios e integrando diferentes tecnologías y en este caso se tomara en cuenta todo lo referente a la seguridad de las viviendas o locales. Para el desarrollo de este sistema en domicilios se utilizara un sistema de seguridad basado en sensores magnéticos en puertas y ventanas, junto con una sirena que pueda servir de alerta a los vecinos y moradores, de esta manera el propietario del domicilio podrá tomar las medidas necesarias ante esta situación.

##### ***2.1.2 Protección Comercial***

En el caso de sitios comerciales donde el propietario o dueño del negocio no se encuentre en todo momento en el local, se ha implementado un sistema de vigilancia

inalámbrico el cual puede ser monitoreado mediante el celular, esta implementación consta de cámaras IP y de una aplicación móvil mediante la cual se puede observar el estado en el que se encuentra el negocio, esto permitirá saber si existe algún intruso en el periodo de tiempo que no se encuentra el propietario en dicho sitio. En el presente proyecto se puede observar el desarrollo de un sistema que integra los lenguajes de programación de arduino que funcionara como envío de datos y un software de aplicación móvil, al unir estos dos lenguajes se permite una comunicación inalámbrica. Esto se maneja a través del internet. Para que de esta manera el propietario pueda tomar acciones pertinentes como llamar a la policía si se da la necesidad.

## **2.2 Seguridad**

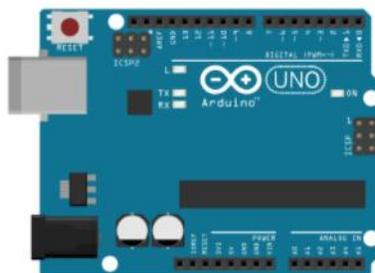
La seguridad se la puede definir como un conjunto de sistemas los cuales constan de varios medios organizativos, humanos mayormente utilizadas para la reducción o control de los riesgos, amenazas que puedan perjudicar a una persona, entidad, instalación, objeto. Por ello, la definición de seguridad es el requerimiento de proporcionar las condiciones necesarias para afrontar el peligro y de esta manera permitir la minimización del riesgo (Mora, 2008).

## **2.3 Arduino**

Arduino es una plataforma la cual cuenta con una placa electrónica de código abierto, esta placa está basada en hardware y software libre. La placa electrónica incorpora un microcontrolador re-programable y una serie de pines hembra mediante los cuales se realizan conexiones entre el microcontrolador y los diferentes sensores y actuadores de manera muy sencilla como se muestra en la figura 1. (Arduino, 2020).

## Figura 1

*Placa electrónica de Arduino UNO*



*Nota:* El gráfico representa la vista frontal del microcontrolador. Tomado de Arduino UNO (Arduino, 2020).

Para una comprensión más específica de este concepto, se analizará los conceptos de hardware libre y el software libre. El concepto de hardware libre se lo puede entender como dispositivos cuyas especificaciones y diagramas son de acceso público. El concepto de software libre se lo puede entender como los programas informáticos cuyo código es accesible por cualquiera permitiendo su uso y modificación. En esta parte Arduino ofrece la plataforma Arduino IDE (Entorno de Desarrollo Integrado), el cual es un entorno de programación que permite crear aplicaciones para las placas Arduino, de manera que se les puede dar todo tipo de utilidades (Fernandez, 2020).

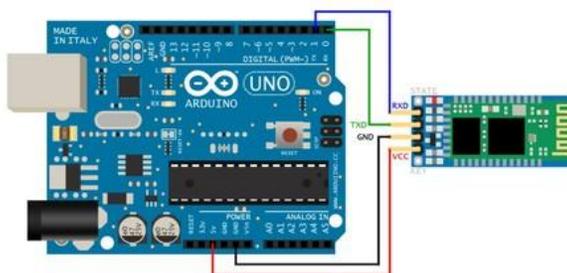
Su entorno de programación es multiplataforma por ende se puede instalar en varios sistemas operativos como lo son Windows, Mac OS y Linux. Además cuenta con uno de los lenguajes de programación más utilizados el cual es el lenguaje de programación basado en C++, su placa electrónica es re-utilizable esto gracias a que cuenta con un chip reprogramable lo que permite que al terminar el proyecto se puede empezar con uno nuevo en el mismo dispositivo (Arduino, 2020).

### 2.3.1 *Cómo funciona Arduino*

El Arduino está basado en un microcontrolador ATMEL y gracias a ello pueden grabar diferentes tipos de instrucciones para después ir generando programas, al tener pines de entrada la placa puede recibir diferente tipo de señales e información y se encargara de procesar dicha información, esto variara dependiendo del uso que se le vaya a dar a la placa. Estos microcontroladores son circuitos integrados en los que se pueden grabar instrucciones las cuales se convierten en programas. Los periféricos que se pueden utilizar para enviar datos al microcontrolador dependen del uso como se muestra en la figura 2. Sus periféricos son capaces de recibir todo tipo de información como cámaras, teclados o sensores. De igual manera es capaz de enviar información por sus pines transmitiendo información a través de LCDS o pantallas e incluso altavoces (Fernandez, 2020)

#### **Figura 2**

*Conexión de un módulo a los periféricos a la placa electrónica de Arduino UNO*

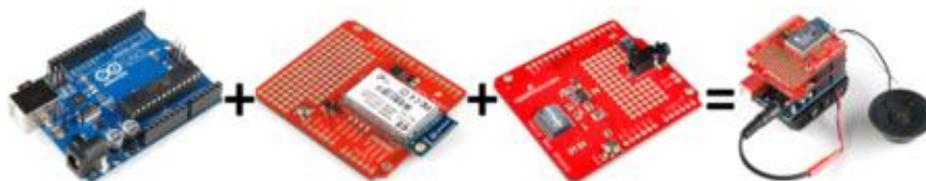


*Nota:* El gráfico representa las Conexiones de Arduino con el módulo WIFI. Tomado de (Fernandez, 2020).

Arduino cuenta con diferentes tipos de placas como se muestra en la figura 3. Estas placas varían dependiendo de la función que vayan a cumplir por ello su tipo y forma van variando e incluso su tamaño, todo para satisfacer las necesidades del proyecto que se vaya a trabajar, y dependiendo de sus funciones específicas van variando sus precios (Fernandez, 2020).

**Figura 3**

*Diferentes tipos de placas de Arduino.*



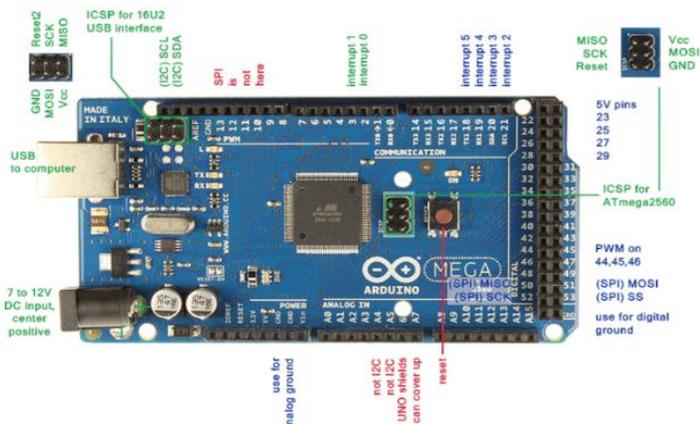
*Nota:* El gráfico representa los tipos de placas de Arduino. Tomado de (Fernandez, 2020).

### 2.3.2 Arduino Mega

El Arduino Mega es uno de los microcontroladores que cuenta con más capacidad de la familia Arduino. Este microprocesador posee 54 pines digitales los cuales funcionan como entrada/salida de las cuales 16 entradas son analógica como se muestra en la figura 4 y cuenta con una conexión USB además de una entrada para energizar la placa en la tabla 1 podemos observar algunas de sus características más *Notables*. Para la comunicación entre la computadora y Arduino se produce a través del Puerto Serie. Posee un convertidor USB-Serie, por lo que sólo se necesita conectar el dispositivo a la computadora utilizando un cable USB (González, 2013).

**Figura 4**

*Partes de un Arduino Mega.*



*Nota:* El gráfico representa las partes de la placa Arduino MEGA. Tomado de (González, 2013).

**Tabla 1***Especificaciones Del Arduino Mega*

Propiedades	Características
Microcontrolador	mega2560
Voltaje Operativo	5V
Voltaje de Entrada	7-12V
Voltaje de Entrada(límites)	6-20V
Pines digitales de Entrada/Salida	54 (de los cuales 15 proveen salida PWM)
Pines análogos de entrada	16
Corriente DC por cada Pin Entrada/Salida	40 mA
Corriente DC entregada en el Pin 3.3V	50 mA
Memoria Flash	256 KB (8KB usados por el bootloader)
SRAM	8KB
EEPROM	4KB
Clock Speed	4KB

*Nota:* Características principales del Arduino Mega. Tomado de (González, 2013).

El Arduino Mega puede ser alimentado de dos maneras mediante el puerto USB o con el uso de una fuente externa. El voltaje debe estar en el rango de los 7V hasta los 12V (González, 2013).

Arduino Mega posee algunos pines para la alimentación del circuito aparte del adaptador para la alimentación como se muestra en la tabla 2:

**Tabla 2***Especificaciones De Voltaje Del Arduino Mega*

Pines	Características
VIN	A través de este pin es posible proporcionar alimentación a la placa.
5V	Podemos obtener un voltaje de 5V y una corriente de 40mA desde este pin.

Pines	Características
3.3V	Podemos obtener un voltaje de 3.3V y una corriente de 50mA desde este pin.
GND	El Ground (0V) de la placa.

*Nota:* Características de voltaje de pines del Arduino Mega. Tomado de (González, 2013).

#### 2.4 ESP8266 módulo WiFi para Arduino

El módulo ESP8266, está diseñado para trabajar con el microcontrolador Arduino de bajo costo este módulo permite la conexión Wi-Fi como se muestra en la figura 5. Este módulo permite el control de proyectos para así trabajar a distancia sin la necesidad de una conexión física, de esta forma se pueden llegar a controlar o automatizar varios como es un control de una puerta de garaje, hasta una alarma casera (López, 2019 ).

#### Figura 5

*Módulo ESP8266*



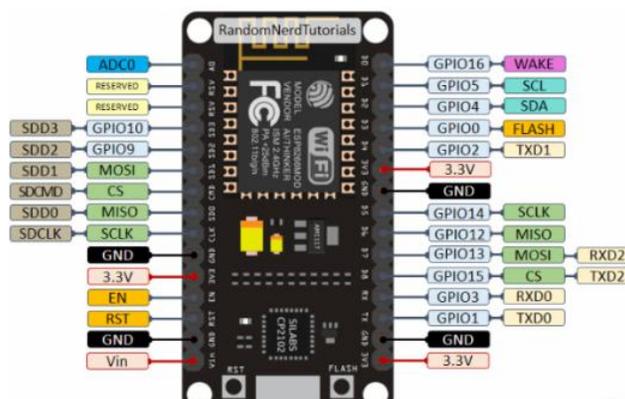
*Nota:* El gráfico representa al módulo ESP8266. Tomado de (López, 2019 ).

Este módulo cuenta con un microcontrolador el cual tiene una CPU RISC de 32 bits fabricada por Tensilica, esto quiere decir que el modulo trabaja con una frecuencia de 80 MHz, por lo que es capaz de soportar overclock, tanto en su CPU como en su memoria flash, pero además se puede extender o impulsar su rendimiento hasta los 160

MHz en su procesador y entre los 40 y 80 MHz en su memoria flash, puede soportar para IEEE 802.11 b/g/n , TR switch, soporte para WEP y WPA/WPA2 como se muestra en la figura 6 (López, 2019 ).

**Figura 6**

*Pines del Módulo ESP8266.*



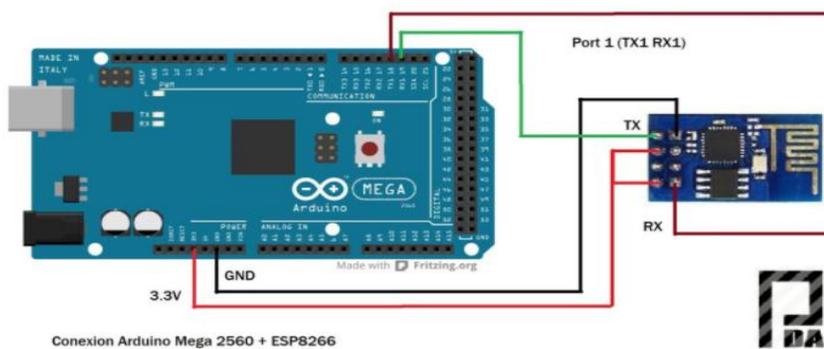
*Nota:* El gráfico muestra los pines del módulo ESP8266. Tomado de (López, 2019 ).

## 2.5 Comunicación WIFI ESP8266

Para la comunicación WiFi, del ESPXX cuenta con una comunicación integrada 802.11 b/g/n, en donde incluye modos Wi-Fi Direct (P2P) y soft-Ap, además de una pila de TCP/IP completa, esto sirve para liberar la mayor parte del trabajo al momento de la comunicación con el procesador. El módulo ESPXX sirve para dar una conectividad WiFi a diferentes proyectos de arduino. Pero se debe tener en cuenta que los módulos de Ethernet y su comunicación con internet puede generar una carga excesiva para Arduino. Para establecer una comunicación con el ESPXX se realiza mediante comandos AT, que son comandos de texto enviados de manera serial. Estos comandos se pueden enviar por un conversor USB-TTL o Arduino como muestra en la figura 7 (Llamas L. , 2017).

## Figura 7

Comunicación serial entre Arduino Mega y módulo ESP8266.



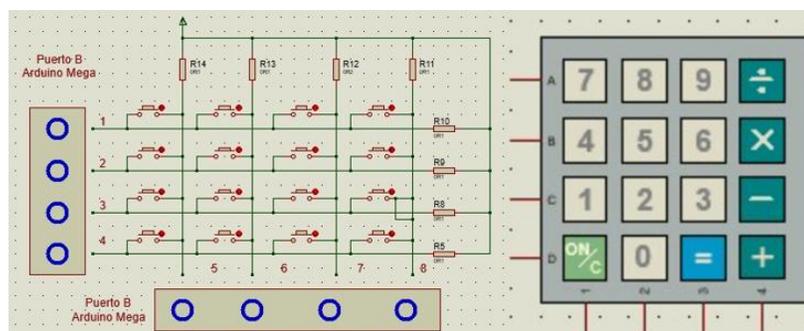
*Nota:* En esta figura se muestra como energizar y conectar el módulo ESP8866 al microcontrolador Arduino. Tomado de (PDAControl, 2016).

### 2.6 Teclado matricial 4x4

El teclado matricial es un conjunto de pulsadores, y se le asigna a cada pulsador una determinada función o símbolo, este tipo de teclados es mayormente utilizado para trabajar con el microcontrolador de arduino, para que el Arduino sea capaz de identificar que teclas está pulsando se debe agregar tensión en las filas de manera secuencial de esta manera al leer columnas se podrá identificar cual es la que permite el paso de la tensión. Los teclados matriciales usan una combinación de filas y columnas para conocer el estado de los botones como se muestra en la figura 8. Cada tecla es un pulsador conectado a una fila y a una columna. Cuando se pulsa una de las teclas, se cierra una conexión única entre una fila y una columna (Prometec, 2018).

**Figura 8**

*Diagrama de un pequeño teclado numérico de 16 teclas.*

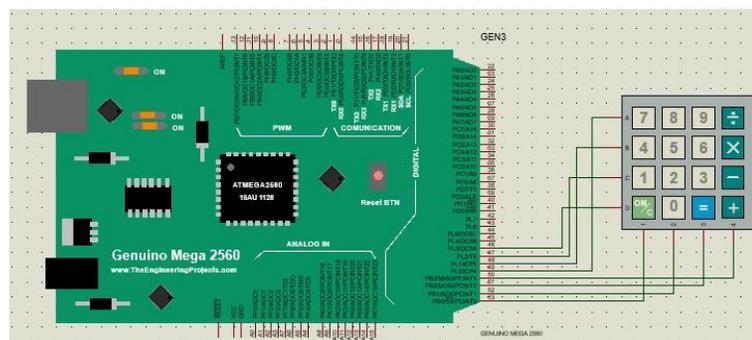


*Nota:* Esquemático interno de un teclado matricial. Elaborado por el Autor.

Una manera de configurarlo es, colocando la tensión en la primera fila y después leemos seguidamente los hilos que corresponden a las columnas. Por ende si ninguno de los puertos recibe tensión significa que no se ha pulsado ninguna tecla de la primera fila. Por otro lado en la segunda fila si se coloca la tensión, y al momento de leer los hilos 4, 3, 2,1 se encuentra con un hilo 1 con tensión, es que se ha pulsado la tecla correspondiente a la “B”. Simplificando de esta manera para tomar los datos de un teclado de 4x4 únicamente se necesita de 8 en lugar de 16 como se muestra en la figura 9. (Prometec, 2018).

**Figura 9**

*Diagrama de conexión de un teclado matricial con Arduino.*



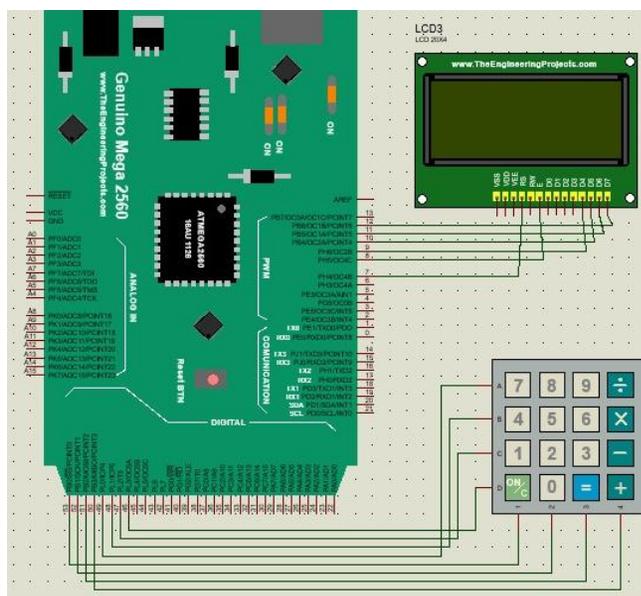
*Nota:* El gráfico representa la conexión entre un Arduino Mega y un teclado matricial. Elaborado por el Autor.

## 2.7 Pantalla LCD 20x4 - Display LCD 20x4

La pantalla LCD es una pantalla de 20x4 este tipo de display son alfanuméricos, monocromática de cristal líquido, es una pantalla visual electrónica que funciona mediante las propiedades de modulación de la luz de los cristales líquidos, en si estos cristales líquidos no emiten luz directamente el cual es compatible con arduino como se muestra en la figura 10. Una de sus mayores ventajas es que permite la presentación de caracteres alfanuméricos y otros símbolos en un formato de 20 caracteres por línea, en 4 líneas. Viéndolo desde este punto el LCD 20x4 puede mostrar 20 caracteres en cada una de las 4 filas del display LCD, dando un total de 80 caracteres. Es muy utilizado en este display es muy útil para proyectos con microcontroladores PIC, AVR y Arduino ya proyectos con microcontroladores ya que permite la visualización de datos como letras o estados de sensores (TECmikro, 2020).

**Figura 10**

*Diagrama de conexión de un teclado matricial con Arduino.*



*Nota:* El gráfico representa la conexión entre un Arduino Mega y un Display LCD 20x4.

Elaborado por el Autor.

Esta pantalla se muestra en una matriz de 5x7 píxeles y funciona con un comando para control y muestra la información que el usuario desea visualizar. En cuanto a sus pines no es diferente a uno normal de 16x2 o 20x4 (TECmikro, 2020).

**Tabla 3**

*Especificaciones Técnicas*

Especificaciones	Características
Formato de presentación	20 caracteres X 4 líneas
Interfaz de entrada	4-Bits / 8-Bits
Retroiluminación	LED blanco
Controlador	SPLC780D (compatible con Hitachi HD44780)
Modo de presentación	Fondo azul / Caracteres blancos
Caracteres	5X8 puntos
Tamaño de cada punto	0.55X0.55 mm
Separación entre puntos	0.60X0.60 mm
Medidas del módulo	98.0(ancho) x 60.0(alto) x 12(espesor) mm
Medidas del display (Área de presentación)	76.0(ancho) x 27.0(alto) mm
Temperatura de funcionamiento	-10 a +60 grados Celsius
Temperatura de almacenamiento	-20 a +70 grados Celsius
Voltaje de funcionamiento	5V
Consumo de corriente del LCD	aproximadamente 2 mA
Consumo de la luz de fondo (retroiluminación)	40 mA
Descripción de los pines (pinout)	Vss: Tierra GND (de la lógica y del LCD) VDD: Alimentación del circuito lógico (+5V)

Especificaciones	Características
	VO: Alimentación del panel LCD (variable, se emplea para el contraste)
	RS: Selección de Dato o Instrucción
	R/W: Lectura o Escritura de Datos
	E: Habilitación
	DB0-DB7: Bus de Datos
	(A: Alimentación de la Retroiluminación (+5V)
	K: Tierra GND (de la Retroiluminación)

---

*Nota:* Características de un display LCD. Tomado de (TECmikro, 2020)

## 2.8 Cámara IP

Las cámaras IP son videocámaras cuya diferencia con las cámaras normales es que están diseñadas con la función de enviar señales de video y audio a Internet, esto mediante el uso de una red local y su información de video e audio se la puede visualizar con un dispositivo que cuente con conectividad a internet y desde ahí acceder a las funciones que ofrezca la cámara un ejemplo de una cámara IP se muestra en la figura 11 (Nivian, 2017).

### Figura 11

*Ejemplo de una Cámara IP.*



*Nota:* El gráfico representa un ejemplo de una cámara IP. Tomado de (Radic, 2016).

### **2.8.1 Funcionamiento una cámara IP**

Una necesidad muy importante para un buen funcionamiento de una cámara IP es que debe contar con una conexión a Internet, el primer paso para poner a funcionar una cámara IP es conectarla a una fuente de alimentación y conectarla al Router o conexión a internet, cada cámara IP contara con una aplicación móvil la con la cual se puede controlar las diferentes funciones de la misma, para ello se escanea el código QR de la parte trasera de las cámaras desde la aplicación móvil y de esta manera la aplicación podrá conectarse a la cámara desde cualquier lugar. Integran potentes sistemas de seguridad, y gracias a ello es posible obtener grabaciones de lo que sucede dentro de su campo de visión. Las grabaciones obtenidas por la cámara pueden ser almacenadas a través de un PC que se encuentre conectado a la misma red, un disco duro externo, o incluso una tarjeta SD que se encuentre alojada en la cámara (Nivian, 2017).

Dependiendo de cada tipo de cámara pueden contar con LEDs infrarrojos, los cuales se encargan de proporcionar una visión nocturna y de esta manera obtendrá una imagen más nítida. El uso más común para este tipo de cámaras es como un vigila

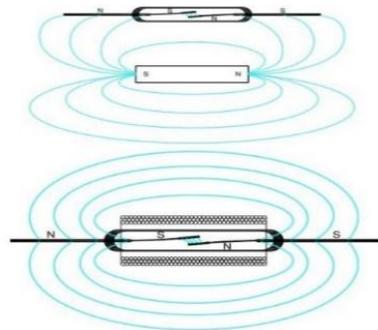
bebes, también es muy utilizada para proteger un negocio o una casa ya que sirve como sistema de intimidación frente a robos. (Nivian, 2017).

## 2.9 Sensor Magnético Reed-switch

Un sensor magnético Reed-switch es un sensor que se activa por la presencia de un campo magnético. Cuenta con dos tipos de estado, uno normalmente abierto en este tipo de estado los contactos internos del sensor se ante la presencia del campo magnético, el otro tipo es normalmente cerrado y sus contactos internos se abren en presencia de un campo magnético como se muestra en la figura 12 (Shoptronica, 2020).

### Figura 12

*Funcionamiento de un sensor magnético Reed-switch.*



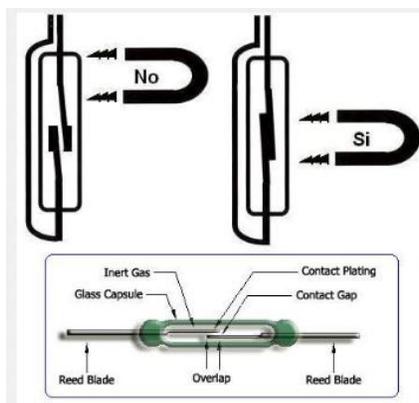
*Nota:* El gráfico representa el funcionamiento de un sensor magnético reed-switch (Shoptronica, 2020).

### 2.9.1 Funcionamiento

El funcionamiento de un sensor magnético reed switch debemos entender su composición, este sensor está compuesto de dos laminas ferromagnéticas las cuales están compuestas de Ni y Fe, se encuentran herméticamente selladas dentro de una cápsula de vidrio como se muestra en la figura 13.

### Figura 13

*Esquema interno de un sensor magnético Reed-switch.*



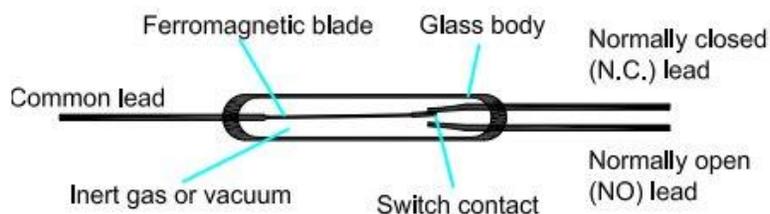
*Nota:* El gráfico representa el funcionamiento interno de un sensor magnético reed-switch.

Tomado de (Shoptronica, 2020).

Las láminas ferromagnéticas internas no llegan a conectarse dentro de la cápsula de vidrio dejando un pequeño espacio entre ellas, y se ponen en contacto ante la presencia de un campo magnético adecuado. El área que entra en contacto está recubierto por metales duros los cuales ayudan a elevar la vida útil como se muestra en la figura 14. (Shoptronica, 2020).

### Figura 14

*Partes de un sensor magnético Reed-switch.*



*Nota:* El gráfico representa las partes de un sensor magnético reed-switch. Tomado de (Shoptronica, 2020).

## 2.10 Sirena

Un concepto básico sobre una alarma se lo puede asociar como una señal sonora o luminosa la cual sirve para informar sobre la presencia real e inminente de alguna amenaza. Esto ha sido de esta manera desde épocas ancestrales, el ser humano ha empleado todo tipo de señales para comunicar ese peligro cuando se aproxima. En la actualidad, los sistemas de alarma se han modernizado esto debido al avance tecnológico. Y gracias a esto, quien desee dar aviso a sus vecinos de que hay una situación de peligro dentro del vecindario, puede activar una alarma de manera inmediata utilizando únicamente un botón de su control remoto. (Montes, 2018)

Estos tipos de sirena pueden ser tanto sonoras como visibles, esto significa que una sirena puede ser una alarma audible que alerte de un suceso específico como se muestra en la figura 15.

### Figura 15

*Sirena para interiores.*



*Nota:* La figura muestra una sirena auditiva para interiores. Elaborado por el Autor.

Mientras que una alarma visible pueden ser luces que notifiquen el estado de un proceso, estas luces regularmente son intermitentes como se muestra en la figura 16.

**Figura 16***Alarma visible.*

*Nota:* La figura muestra una alarma visual utilizada para interiores. Elaborado por el Autor.

**2.11 Android Studio**

La plataforma de Android Studio es un entorno de desarrollo integrado (IDE) principalmente para desarrollar de aplicaciones Android, basado en IntelliJ IDEA, cuenta con un editor de códigos y herramientas para desarrolladores de IntelliJ, ofrece funciones que aumentan tu productividad al momento de desarrollar aplicaciones para Android como se muestra en la tabla 5. (AndroidStudio, 2021).

**Tabla 4***Especificaciones Técnicas*

Especificaciones	Características
Sistema de compilación	Flexible basado en Gradle
Emulador	Rápido con varias funciones
Entorno unificado	Para todos los dispositivos Android
Aplicación de cambios	Sirve para insertar cambios de códigos sin reiniciar la aplicación
Integración con GitHub y plantillas	Código para ayudarte a compilar e importar código de muestra

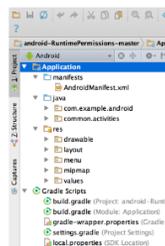
Especificaciones	Características
Variedad	Marcos de trabajo y herramientas de prueba
Herramientas de Lint	Identifica problemas de rendimiento, uso y compatibilidad de la versión.
Compatibilidad con Google Cloud Platform	Integración de Google Cloud Messaging y App Engine
Incluye uno o más módulos	Archivos de código fuente y archivos de recursos
Tipos de módulos	Módulos de aplicaciones para Android
	Módulos de biblioteca
	Módulos de Google App Engine
Compatibilidad	Con C++ y NDK

*Nota:* Características de Android Studio. Tomado de (AndroidStudio, 2021).

Android Studio muestra los archivos de tu proyecto, como se muestra en la Figura 17. Está organizada en módulos para acceder rápidamente a los archivos fuente de tu proyecto. (AndroidStudio, 2021).

### Figura 17

*Los archivos de proyecto en la vista de Android.*



*Nota:* El gráfico representa los archivos de proyecto en la vista de Android Studio.

Tomado de (AndroidStudio, 2021).

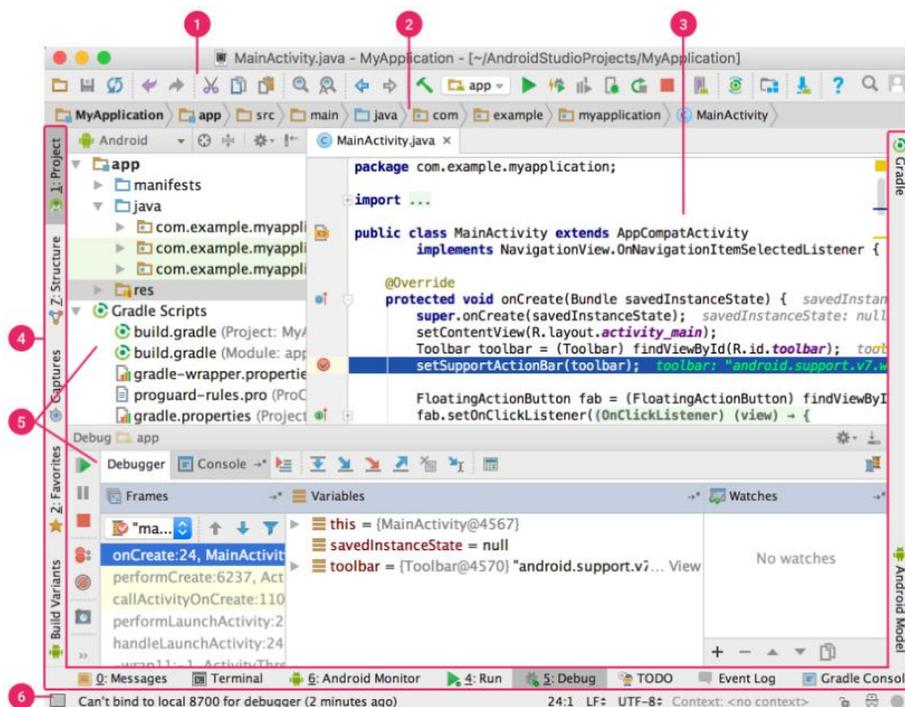
Archivos de compilación en la parte superior, cada módulo de app contiene las siguientes carpetas:

- Manifests: Archivo de AndroidManifest.xml.
- Java: Código fuente Java y código de prueba de JUnit.
- Res: Contiene los recursos sin código, como diseños XML, strings de IU e imágenes de mapa de bits.

La interfaz gráfica de usuario en Android Studio la podemos visualizar como una ventana principal consta de varias áreas lógicas identificadas en la figura 18.

**Figura 18**

*Ventana principal de Android Studio.*



*Nota:* El gráfico muestra la pantalla principal de Android Studio. Tomado de (AndroidStudio, 2021).

Según (AndroidStudio, 2021) las partes de la ventana principal de Android Studio son:

1. La barra de herramientas permite ejecutar e iniciar las herramientas de Android.
2. La barra de navegación ayuda a explorar, abrir archivos para editar.
3. La ventana del editor para crear y modificar código.
4. La barra de la ventana de herramientas contiene los botones que te permiten expandir o contraer ventanas de herramientas individuales.
5. Las ventanas de herramientas acceso a tareas específicas.
6. En la barra de estado, el estado del proyecto además de advertencias o mensajes.

La plataforma permite organizar la ventana principal ocultando o desplazando las barras y ventanas de herramientas, además de acceder a sus funciones mediante una combinación de teclas (AndroidStudio, 2021).

## **2.12 Firebase**

La Plataforma de Firebase está vinculada con Google, y tiene una función principal que es facilitar el desarrollo para la creación de aplicaciones y de esta manera cuentan con una elevada calidad. Es una plataforma en la nube y es compatible con diferentes plataformas como iOS, Android y web. Dentro de Firebase encontraremos diversas funciones que ayudaran a que los desarrolladores combinen y adapten la plataforma de acuerdo a sus necesidades como se muestra en la tabla 6 (Cardona, 2016).

**Tabla 5***Principales características de Firebase*

Especificaciones	Características
Desarrollo	Permite el desarrollo de mejores aplicaciones, ya que minimiza el tiempo de optimización y desarrollo, gracias a la detección de errores y testeo, a la vez que almacena todo en la nube, de manera remota.
Analítica	Cuenta con un control de rendimiento de la aplicación por el uso de métricas analíticas, desde un panel y de forma gratuita.
Poder de crecimiento	Ayuda a la gestión de todos los usuarios de las aplicaciones, pueden identificar nuevos usuarios, con el uso de invitaciones o notificaciones.
Monetización	Cuenta con la función de AdMob, la cual permite que puedas ganar dinero.
Rapidez	Su API que es muy intuitiva y esta sostenida en un solo SDK.

Especificaciones	Características
Agilidad	Cuenta con apps multiplataforma e integra una Apis a SDK individuales para iOS, Android y Javascript, y puede gestionar diferentes apps sin salir de la plataforma.

*Nota:* Características principales de Firebase. Tomado de (Cardona, 2016)

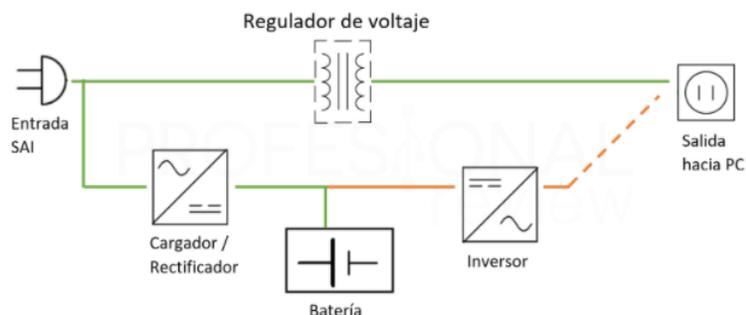
Dentro de la plataforma se encuentra documentación diversa diseñada para el aprendizaje esta documentación cuenta con ejemplos, tutoriales y documentación complementaria sobre todos sus usos. (Cardona, 2016)

### 2.13 UPS

Un UPS viene de (Uninterruptible Power Supply), o en español llamado SAI (Sistema de alimentación ininterrumpida) es dispositivo que gracias a sus baterías y a sus diferentes elementos se encarga de almacenar energía, por lo que puede suministrar energía eléctrica durante un apagón eléctrico por un tiempo limitado. Esta energía se distribuirá a todos los dispositivos que se encuentren conectados, una de sus ventajas es que filtra bajadas y subidas de tensión, por lo que elimina el armónico de la red. Este dispositivo da energía eléctrica a los equipos conocidos como cargas críticas, como por ejemplo pueden ser los aparatos industriales, médicos, o informáticos, que siempre necesitan tener alimentación en la figura 19 se puede observar el funcionamiento interno de un UPS (Salvatore, 2019).

**Figura 19**

*Funcionamiento interno de un UPS*



*Nota:* Partes que conforman el UPS. Tomado de (Castillo, 2019).

### 2.14 Relés

Los relés son dispositivos electromecánicos y gracias a sus componentes nos permiten controlar cargas que tienen un nivel tensión o intensidad muy superior mediante microcontroladores. El uso del relé es muy común dentro de la automatización de procesos, y mayormente utilizado para accionar cargas como motores, bombas, o cualquier otro tipo de maquinaria. De manera física un relé actúa como un interruptor pero que en lugar de accionarse manualmente, es activado de forma electrónica. Los relés son aptos para accionar cargas tanto de corriente alterna como continua (Llamas L. , 2016 ).

Un relé dispone de dos circuitos:

- El circuito primario: Es el encargado de conectar la electrónica de baja tensión, de la cual recibe la señal de encendido y apagado.
- El circuito secundario: Es el que funciona como interruptor encendiendo o apagando la carga.

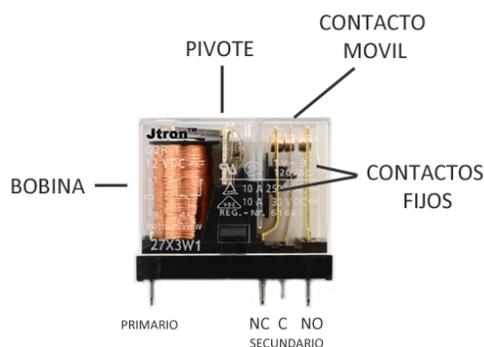
Su vida está determinada por el número de conmutaciones en el rango de 100.000 a 1.000.000 de conmutaciones. (Llamas L. , 2016 ).

### 2.14.1 Función de un relé

Su circuito primario recibe la señal de baja tensión, y está formado por una bobina arrollada a un núcleo metálico, lo cual crea un electroimán. Mientras que el secundario, está formado por unos contactos eléctricos en láminas de metal flexible como se muestra en la figura 20. Estos elementos están fijados a una base aislante la cual impiden que exista el contacto eléctrico con el exterior (Llamas L. , 2016 ).

#### Figura 20

*Partes de un Relé*



*Nota:* El gráfico muestra las partes de un relé. Tomado de (Llamas L. , 2016 )

Los relés disponen de tres contactos en el secundario C (común), NO (normalmente abierto) y NC (normalmente cerrado). Para controlar la carga con un relé de manera que funcione como un interruptor se conectara uno de los polos al terminal C, y el otro polo se conectara a uno de los terminales NO o NC, en función de si queremos que al entrar en funcionamiento el relé el circuito se cierre o se abra (Llamas L. , 2016 ).

### 2.15 Reloj de tiempo real RTC

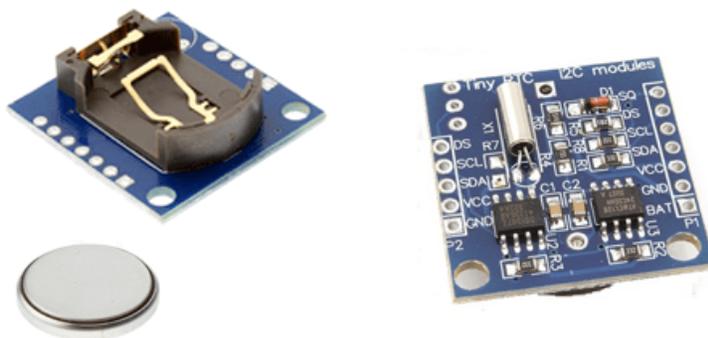
Un módulo de reloj de tiempo real (RTC) es un dispositivo el cual permite obtener datos de tiempo, estos dispositivos son parecidos a los relojes y calendarios esto debido a que funcionan con segundos, minutos, horas. Este reloj de tiempo real está formado por un resonador de cristal para contabilizar el paso del tiempo de manera

adecuada. Una de sus ventajas es la de reducir el consumo de energía a la vez que cuenta con una gran precisión al utilizarlo con Arduino, e incorporan baterías que permiten mantener el tiempo establecido a pesar de una pérdida de alimentación (Llamas, 2016).

Para Arduino existen dos RTC DS1307 y el DS3231, fabricados por Maxim, el DS3231 es la versión mejorada del DS1307, El modelo DS1307 sufre errores debido a la temperatura lo que afecta a la medición del tiempo provocando un desfase temporal que puede ser de uno a dos minutos al día. Mientras que el DS3231 cuenta con medición y compensación de temperatura esto sirve para garantizar precisión. Su comunicación es a través del bus I2C, además que su alimentación es de 4.5 a 5.5 para el DS1307, y 2.3 a 5.5V para el DS3231, incorporan una batería CR2032 esto para que el dispositivo pueda ser alimentado por varios años y no tenga un desfase de tiempo (Llamas, 2016). En la figura 21 se muestra al DS1307.

### Figura 21

DS1307

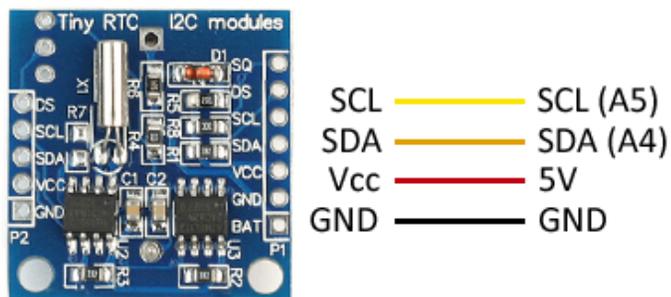


*Nota:* Representación gráfica del módulo DS1307. Tomado de (Llamas, 2016).

La conexión del dispositivo RTC con Arduino es simple para lo cual se debe alimentar al módulo desde Arduino mediante 5V y Gnd, y se conecta pines del bus I2C a los correspondientes de Arduino como se muestra en la figura 22.

## Figura 22

Conexión de un módulo con DS1307



*Nota:* El módulo cuenta con dos partes lo que permite conectarlo a dos Arduinos.

Tomado de (Llamas, 2016).

### 2.16 Comunicación I2C

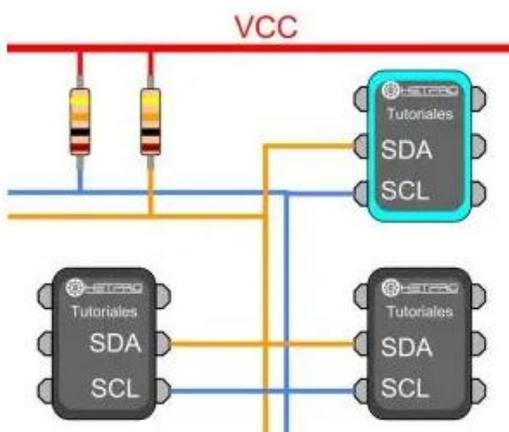
La comunicación I2C es una comunicación serial y viene del término Inter Integrated Circuits el cual es un puerto y protocolo de comunicación serial, este funciona definiendo la trama de datos y las transfiere una velocidad de 100 kbps y 3,4 MHz bits entre 2 dispositivos digitales. El puerto incluye dos cables de comunicación, SDA y SCL mediante estos cables de comunicación se permiten conectar hasta 127 dispositivos esclavos (Hetpro, 2018).

Los modos de comunicación en I2C varían dependiendo de las tramas que pueden formarse en el bus como se muestra en la figura 23. Las tramas van variando dependiendo de si se quiere leer al sensor esclavo, o si lo queremos configurar. Existen principalmente dos modos de comunicación de acuerdo con (Hetpro, 2018):

- Maestro-Transmisor y Esclavo-Receptor. Se usa cuando se desea configurar algún parámetro del esclavo I2C.
- Maestro-Receptor Y Esclavo-Transmisor. Se usa cuando se quiere leer información del sensor I2C.

**Figura 23**

Conexión de tres dispositivos a un bus de comunicación I2C.



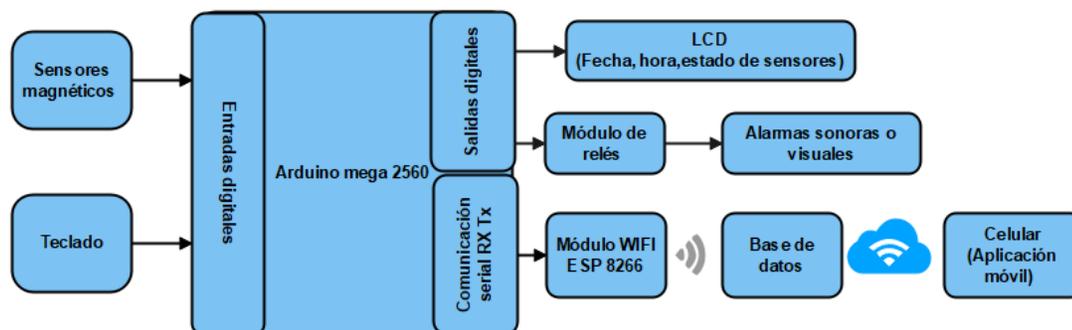
*Nota:* En la figura se muestra una conexión entre tres dispositivos mediante la comunicación I2C. Tomado de (Hetpro, 2018).

## CAPÍTULO III

### 3 Desarrollo del tema

**Figura 24**

*Diagrama de bloques del sistema de seguridad*



*Nota:* La figura muestra como es el funcionamiento del sistema de seguridad.

En la figura 24 se muestra al sistema de seguridad inalámbrico el cual brinda protección de viviendas y locales, utiliza tecnología basada en el microcontrolador arduino mega 2560, este microcontrolador recibe señales digitales emitidas por varios dispositivos como los son sensores magnéticos y un teclado matricial, para después enviar esta información a dispositivos que permiten una mejor visibilidad de la información en este caso es un LCD 20x4 y a un módulo WiFi ESP8266, dependiendo de las señales recibidas y del estado del sistema accionara las alarmas, (en caso de que el sistema este activado y reciba una señal de sensores activa, accionara las alarmas que funcionan como alerta), por otro lado el módulo ESP8266 recibirá los datos obtenidos por los sensores y el teclado para de esta manera almacenarlos en una base de datos, esta información podrá ser visualizada mediante dispositivos móviles como celulares siempre y cuando cuenten con una aplicación de Android llamada "Sistema de Seguridad".

### 3.1 Programa de Arduino Mega 2560

Para el diseño del sistema de seguridad se utilizó un microcontrolador mega 2560, el cual funciona como módulo principal y recibe toda la información del sistema, en primera instancia se incluye las librerías a utilizar como lo muestra la figura 25.

#### Figura 25

*Librerías Utilizadas en la programación del microcontrolador mega2560*

```
//Librerias utilizadas en el programa
#include <LiquidCrystal.h> //Libreria del LCD
#include <Wire.h>          //Libreria para modulo wifi
#include <Password.h>     //Libreria de contraseña
#include <Keypad.h>       //Libreria del Teclado
```

*Nota:* En la figura se muestra las librerías necesarias utilizadas en el programa.

Una vez añadidas las librerías se declara la contraseña, la cual es insertada a través de un teclado matricial, el mismo que cuenta con 4 filas y 4 columnas como se muestra en la figura 26, para ello se define los pines que se van al arduino mega.

#### Figura 26

*Declaración de la contraseña y variables del teclado matricial.*

```
//Declaracion de la contraseña
Password password = Password( "1234" );
//Numero de filas y columnas del teclado
const byte ROWS = 4; // Cuatro filas
const byte COLS = 4; // Cuatro columnas

char keys[ROWS][COLS] = { // Definicion del Keypad
  {'7','8','9','D'},
  {'4','5','6','C'},
  {'1','2','3','B'},
  {'*','0','#','A'}};

byte rowPins[ROWS] = {
  46, 47, 48, 49}; //Coneccion de filas de pines del keypad
byte colPins[COLS] = {
  50, 51, 52, 53}; //Coneccion de columnas de pines del keypad

// Creacion del Keypad
Keypad keypad = Keypad( makeKeypad(keys), rowPins, colPins, ROWS, COLS );
```

*Nota:* La figura muestra qué valor tendrá cada botón del teclado matricial

Una vez establecido el teclado se asigna los pines del Display LCD, junto con la declaración de los pines de entrada que se van a utilizar como se muestra en la figura 27.

### Figura 27

*Pines de entrada utilizados en la programación.*

```
// Inicializacion de la libreria con numeros de pines
LiquidCrystal lcd(7, 8, 9, 10, 11, 12); // Asignacion de los pines al lcd

//Declaracion de luces led
int ledDelay = 50; // delay by 50ms
int redPin = 29;
int bluePin = 31;
// Declaracion de Constantes
//int blueLED = 36;
int greenLED = 37;
int redLED = 38;
int pirPin1 = 23;
int pirPin2 = 24;
int pirPin3 = 25;
int pirPin4 = 26;
int pirPin5 = 27;
int pirPin6 = 28;
```

*Nota:* La figura muestra cada pin que se utilizó en la programación.

Se habilita los pines de entrada y de salida como se muestra en la figura 28.

### Figura 28

*Pines de entrada utilizados en la programación.*

```
//Habilitacion de pines de salida
pinMode(redPin, OUTPUT);
pinMode(bluePin, OUTPUT);

pinMode(redLED, OUTPUT);
pinMode(greenLED, OUTPUT);
pinMode(speakerPin, OUTPUT);

//pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
pinMode(relay3, OUTPUT);
pinMode(relay4, OUTPUT);

//Habilitacion de pines de entrada
pinMode(pirPin1, INPUT);
pinMode(pirPin2, INPUT);
pinMode(pirPin3, INPUT);
pinMode(pirPin4, INPUT);
pinMode(pirPin5, INPUT);
pinMode(pirPin6, INPUT);
pinMode(reedPin1, INPUT);
pinMode(reedPin2, INPUT);
pinMode(buzzer, OUTPUT);
```

*Nota:* La figura muestra cada pin según su función si es por entrada o salida.

Una vez ya configurados los pines se establece como se activa la alarma, para ello se utilizó el comando “if” dentro de este comando se establece de acuerdo al número de sensores a utilizar ya que una vez se active cualquier sensor nos llama al comando de alarma como se muestra en la figura 29.

### Figura 29

*Condiciones de activación de la alarma*

```

//Condicon para la activacion de la alarma
if (alarmActive == 1){
  if (digitalRead(pirPin1) == HIGH)  {
    zone = 0;
    alarmTriggered();
  }
  if (digitalRead(reedPin1) == LOW)  {
    zone = 1;
    alarmTriggered();
  }
  if (digitalRead(reedPin2) == LOW)  {
    zone = 2;
    alarmTriggered();
  }
}

```

*Nota:* La figura muestra activación de la alarma.

Posteriormente se agrega las funciones y condiciones para la contraseña, el teclado junto con un buzzer, lo que produce un sonido cada vez que se presione una tecla, para validar la contraseña se presiona la tecla “\* “, pero en caso de ingresar un valor erróneo se lo puede sobrescribir presionando la tecla “# “, al ser una contraseña los valores escritos son remplazados con el símbolo de “\*” como se muestra en la figura 30.

## Figura 30

### Funciones de la contraseña

```

//Funciones de la contraseña
void keypadEvent(KeypadEvent eKey){
  if (keypadEvent!= NO_KEY){suenaBuzzer(100);
  switch (keypad.getState()){
  case PRESSED:
    if (passwd_pos - 15 >= 15) {
      return ;
    }
    lcd.setCursor((passwd_pos++),0);
    switch (eKey){
    case '*': // * boton para validar password
      passwd_pos = 15;
      checkPassword();
      break;
    case '#': //# boton para volver a escribir password
      password.reset();
      passwd_pos = 15;
      displayCodeEntryScreen();
    // Limpieza de la contraseña
      break;
    default:
      password.append(eKey);
      lcd.print("*"); } }
  }
}

```

*Nota:* La figura muestra Estos comando sirven para evitar que al presionar varios pulsadores el sistema entre en confusión.

Para el control de la alarma como son sirenas y una luz de emergencia se utiliza por la activación de relés y muestra mensajes de alerta en el display dependiendo del sensor que ha sido activado como se muestra en la figura 31.

## Figura 31

### Mensajes de alerta en el display

```

void alarmTriggered(){
  int expected_pos;
  int incr;
  digitalWrite(speakerPin, HIGH);
  digitalWrite(relay2, LOW);
  digitalWrite(relay3, LOW);
  digitalWrite(redPin, HIGH);
  password.reset();
  // Indicaciones mostradas en el display
  alarmStatus = 1;
  // alarmActive = 0;
  lcd.clear();
  lcd.setCursor(0,2);
  lcd.print(" INTRUSO DETECTADO ");
  lcd.setCursor(0,4);
  Serial1.println(" INTRUSO DETECTADO ");
  if (zone == 1) {
    lcd.print(" ++++++ ");
    expected_pos = 65;
    delay(1000); }
  if (zone == 0){
    expected_pos = 40;
    lcd.print(" VENTANA ABIERTA ");
    Serial1.println(" VENTANA ABIERTA ");
    delay(1000); }
}

```

*Nota:* La figura muestra Los mensajes sirven para especificar que tipo de sensor se activo.

Junto al teclado se coloca luces led como indicación cuando se coloque correctamente o incorrectamente la contraseña, además de mostrar una luz parpadeante cuando la alarma este activada como se muestra en la figura 32.

### Figura 32

*Programación para la activación de luces led*

```
//Activacion de luces led
void StrokeLight(){
  digitalWrite(redPin, HIGH); // turn the red light on
  delay(ledDelay); // wait 50 ms
  digitalWrite(redPin, LOW); // turn the red light off
  delay(ledDelay); // wait 50 ms
  digitalWrite(redPin, HIGH); // turn the red light on
  delay(ledDelay); // wait 50 ms
  digitalWrite(redPin, LOW); // turn the red light off
  delay(ledDelay); // wait 50 ms
}
```

*Nota:* La figura muestra la activación de luces led.

El sistema se activa y se desactiva una vez se coloque la contraseña correcta, en caso de que la contraseña sea incorrecta se muestra un mensaje en el display.

Como se ve en la figura 33.

### Figura 33

*Comprobación de la contraseña y mensajes de indicación.*

```
// Comprobacion si la contraseña es correcta
void checkPassword(){
  if (password.evaluate()) {
    if(alarmActive == 0 && alarmStatus == 0) {
      activate(); }
    else if( alarmActive == 1 || alarmStatus == 1) {
      deactivate(); } }
  else {
    invalidCode(); } }
// Mensajes si la contraseña es incorrecta
void invalidCode()
{
  password.reset();
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("¡ CODIGO ERRONEO !");
  Serial1.println("CODIGO ERRONEO !");
  lcd.setCursor(1,2);
}
```

*Nota:* La figura muestra la configuración de la contraseña.

Una vez activada la alarma muestra un mensaje en el display y enciende una luz verde indicando que el sistema se encuentra encendido como se muestra en la figura

34.

## Figura 34

### Activación de la alarma por el uso de la contraseña

```
// Activacion de la alarma mediante la contraseña
void activate()
{
  if((digitalRead(reedPin1) == HIGH) && (digitalRead(reedPin2) == HIGH)){
    digitalWrite(redLED, HIGH);
    digitalWrite(greenLED, LOW);
    digitalWrite(2, HIGH);
    lcd.setCursor(0,0);
    lcd.print("!!SISTEMA ACTIVADO!!");
    Serial1.println("!!SISTEMA ACTIVADO!!");
    alarmActive = 1;
    suenaBuzzer(50);
    delay(50);
    suenaBuzzer(50);
    delay(50);
    password.reset();
    delay(2000);
  }
  else{
    deactivate();
  }
}
```

*Nota:* La figura muestra una notificación de la activación de la alarma se muestra un mensaje en el display

Quando el sistema se desactive muestra un mensaje en el display notificando que el sistema se encuentra desactivado y encenderá un led rojo para una visualización de que el sistema se encuentra desactivado. Además que si alguna de las alarmas están activadas las desactivara como se muestra en la figura 35.

## Figura 35

### Desactivación de la alarma

```
//Desactivacion de la alarma
void deactivate()
{
  //digitalWrite(camera, LOW);
  alarmStatus = 0;
  digitalWrite(redLED, LOW);
  digitalWrite(greenLED, HIGH);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("SISTEMA DESACTIVADO!");
  Serial1.println("SISTEMA DESACTIVADO!");
  digitalWrite(speakerPin, LOW);
  alarmActive = 0;
  password.reset();
  delay(5000);
  digitalWrite(relay2, HIGH);
  digitalWrite(relay3, HIGH);
  displayCodeEntryScreen();
}
```

*Nota:* La figura muestra programación acerca de la desactivación de la alarma.

Quando el dispositivo se conecta muestra unos mensajes de bienvenida como se ve en la figura 36.

## Figura 36

### *Cuadros de texto de bienvenida*

```
// Mensajes de inicio al encender el Dispositivo
void displayCodeEntryScreen()
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" PASSWORD :");
  lcd.setCursor(0,2);
  lcd.print("HOME SECURITY SYSTEM");
  lcd.setCursor(0,3);
  lcd.print(" By MS MANAY, ESPE");
}
void suenaBuzzer(int tiempo)
{
  digitalWrite(buzzer, HIGH);
  delay(tiempo);
  digitalWrite(buzzer, LOW);
}
```

*Nota:* La figura muestra el nombre del sistema de seguridad.

Programación del módulo RTC DS1307, en la figura 37 se muestra las líneas de programación seguidas para la configuración de la hora del RTC.

## Figura 37

### *Configuración del módulo RTC para el tiempo*

```
//Configuracion del tiempo
Wire.begin();
RTC.begin();
RTC.adjust(DateTime(__DATE__, __TIME__));
```

*Nota:* La figura muestra el tiempo que se establecerá a la hora del ordenador.

### **3.2 Programa en ESP8266 módulo WiFi para Arduino**

Para que el sistema cuente con un monitoreo inalámbrico se utiliza el módulo ESP8266 el cual permite una conexión mediante WiFi, y así poder almacenar estos datos en la nube, se utilizó las siguientes librerías como se muestra en la figura 38.

## Figura 38

*Librerías utilizadas en la programación del módulo ESP8266*

```
//Librerías utilizadas en el programa
#include "FirebaseESP8266.h"//Librerías para un enlace con la base de datos
#include <ESP8266WiFi.h> //Modulo utilizado
#include <Ticker.h>
```

*Nota:* La figura muestra las librerías necesarias para poder conectar con la base de datos.

Para establecer la conexión con la base de datos se coloca el enlace de la base de datos y la autenticación, además se debe establecer el nombre de la red WiFi a la que se conecta junto con su contraseña como se muestra en la figura 39.

## Figura 39

*Datos utilizados para la conexión con internet*

```
//Datos para la conexión al internet
#define FIREBASE_HOST "sistema-de-seguridad-ms-default-rtdb.firebaseio.com" //Sin http:// o https://
#define FIREBASE_AUTH "4wOyUJJ9E6bNvDGutlew3TKMfJLeI9Y7rQFp8hmv"
#define WIFI_SSID "GLORIA"
#define WIFI_PASSWORD "anaid627"
```

*Nota:* La figura muestra la conexión con una red WIFI.

Se coloca un nombre en donde se almacena los datos obtenidos del microcontrolador junto con las entradas que se va a utilizar en el programa como se muestra en la figura 40.

## Figura 40

*Declaración de entradas utilizadas*

```
//Define un objeto de Firebase
String path = "/Equipo6";
FirebaseData firebaseData;

//Define entradas del arduino mega al modulo
int SistemaActivado = 0;
int SistemaActivadoAn = 0;
int SistemaDesactivado = 0;
int SistemaDesactivadoAn = 0;
intCodigoErroneo = 0;
intCodigoErroneoAn = 0;
intIntrusoDetectado = 0;
intIntrusoDetectadoAn = 0;
```

*Nota:* La figura muestra las entradas de los sensores.

Se establece la velocidad a la que trabaja el módulo junto con los procedimientos para conectarse a la base de datos, como se muestra en la figura 41.

### Figura 41

*Programación para conectar el módulo con una base de datos*

```
//Conexión del modulo a internet
void setup() {
  Serial.begin(115200);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Conectando a ....");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Conectado con la IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);

  //Establezca el tiempo de espera de lectura de la base de datos en 1 minuto (máximo 15 minutos)
  Firebase.setReadTimeout(firebaseData, 1000 * 60);

  //Tamaño y tiempo de espera de escritura, tiny (1s), small (10s), medium (30s) and large (60s).
  //tiny, small, medium, large and unlimited.
  Firebase.setwriteSizeLimit(firebaseData, "tiny");
}
```

*Nota:* La figura muestra la conexión con la base de datos.

Para la conexión con del ESP8266 y el arduino mega deben tener la misma velocidad de comunicación serial en los pines Rx y Tx, los datos obtenidos se almacenan en una variable. En el caso de que el sistema este activado empezara a tomar los datos de los sensores en tiempo real figura 42.

### Figura 42

*Conexión del módulo con el microcontrolador mega 2560*

```
//Conexion con Aeduino MEGA
void loop()
{
  char mega=(char)Serial.read();
  //Condicion del sistema activado
  if (mega && SistemaActivadoAn == 'A'){
    SistemaActivado = 1 - SistemaActivado;
    delay (111); }
  SistemaActivadoAn = mega;
  if (SistemaActivado == 1)
  {Serial.print("SistemaActivado: ");Serial.println(1);
  if (Firebase.setInt(firebaseData, path + "/Estado del Sistema/ACTIVO", 1){InforSetLuzSensor();}else{CausaError();}
  delay(500);}
  else{Serial.print("SistemaActivado: ");Serial.println(0);
  if (Firebase.setInt(firebaseData, path + "/Estado del Sistema/ACTIVO", 0){InforSetLuzSensor();}else{CausaError();}
  delay(500);}
```

*Nota:* La figura muestra la comunicación utilizada entre el Arduino y el módulo ESP8866.

En el caso de que el sistema este desactivado se bloquea la información del estado de los sensores en la figura 43.

### Figura 43

*Configuración para la lectura de los datos recibidos del microcontrolador*

```
//Condicion del sistema desactivado
if (mega && SistemaDesactivadoAn == 'B'){
  SistemaDesactivado = 1 - SistemaDesactivado;
  delay (111); }
SistemaDesactivadoAn = mega;
  if (SistemaDesactivado == 1)
    {Serial.print("SistemaDesactivado: ");Serial.println(1);
    if (Firebase.setInt(firebaseData, path + "/Estado del Sistema/INACTIVO", 1)){InforSetLuzSensor();}else{CausaError();}
    delay(500);
    Serial.print("SistemaActivado: ");Serial.println(0);
    if (Firebase.setInt(firebaseData, path + "/Estado del Sistema/ACTIVO", 0)){InforSetLuzSensor();}else{CausaError();}
    delay(500);
    Serial.print("IntrusoDetectado: ");Serial.println(0);
    if (Firebase.setInt(firebaseData, path + "/Estado del Sistema/INTRUSO DETECTADO", 0)){InforSetLuzSensor();}else{CausaError();}
    delay(500);
    Serial.print("Sensor1: ");Serial.println(0);
    if (Firebase.setInt(firebaseData, path + "/sensores/Sensor1", 0)){InforSetLuzSensor();}else{CausaError();}
    delay(500);
```

*Nota:* La figura muestra el sistema desactivado.

Para evitar errores en la lectura de los sensores se configura como debe tomar la información en caso de que este desactivado o activado como se muestra en la figura 44.

### Figura 44

*Condiciones de cada sensor en caso de estar activado o desactivado*

```
//Condiciones para cada sensor en caso que este activado o desactivado
if (mega && Sensor1An == 'E'){
  Sensor1 = 1 - Sensor1;
  delay (111); }
Sensor1An = mega;
  if (Sensor1 == 1)
    {Serial.print("Sensor1: ");Serial.println(1);
    if (Firebase.setInt(firebaseData, path + "/sensores/Sensor1", 1)){InforSetLuzSensor();}else{CausaError();}
    delay(500);}
  else{Serial.print("Sensor1: ");Serial.println(0);
  if (Firebase.setInt(firebaseData, path + "/sensores/Sensor1", 0)){InforSetLuzSensor();}else{CausaError();}
  delay(500);}
```

*Nota:* La figura muestra la programación para cada sensor.

Si la base de datos sufre algún error notificara las razones por las que no se pudo obtener dicha información como se muestra en la figura 45.

## Figura 45

### Configuración para muestra de errores

```

}
//Muestra de informacion en caso de error
void CausaError(void)
{
  Serial.println("ERROR");
  Serial.println("RAZON: " + firebaseData.errorReason());
  Serial.println("-----");
  Serial.println();
}

```

*Nota:* La figura muestra En caso de un error al subir los datos el modulo pide un mensaje para poder corregirlo.

La información que se pueden enviar a la base de datos se la declara de acuerdo a su tipo como se muestra en la figura 46.

## Figura 46

### Tipos de datos utilizados en la programación

```

//Muestra de tipo de datos enviados a la base de datos
void printResult(FirebaseData &data)
{
  if (data.dataType() == "int")
    Serial.println(data.intData());
  else if (data.dataType() == "float")
    Serial.println(data.floatData(), 5);
  else if (data.dataType() == "double")
    printf("%.9lf\n", data.doubleData());
  else if (data.dataType() == "boolean")
    Serial.println(data.boolData() == 1 ? "true" : "false");
  else if (data.dataType() == "string")
    Serial.println(data.stringData());
}

```

*Nota:* La figura muestra la declaración para subir todo tipo de información.

### 3.3 Base de datos

Para la base de datos se utilizó Firebase en donde únicamente se necesita una cuenta de google y es gratuita, una vez registrado en la página nos pide crear un proyecto como se muestra en la figura 47.

## Figura 47

*Creación de un nuevo proyecto en la base de datos*



*Nota:* La figura muestra la base de datos.

Se coloca el nombre del proyecto como se muestra en la figura 48.

## Figura 48

*Nombramiento del proyecto*

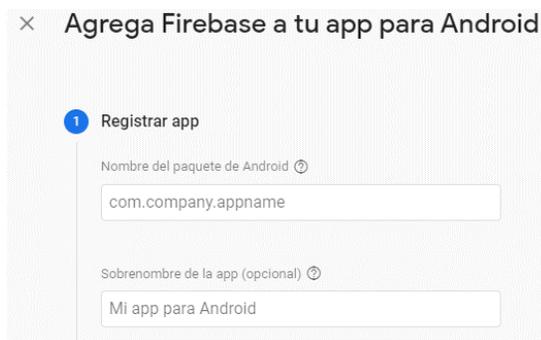
The image shows a form for naming a Firebase project. The heading is 'Comencemos con el nombre de tu proyecto®'. Below it, the instruction 'Ingresa el nombre de tu proyecto' is followed by a text input field containing 'my-awesome-project-id'. A red error message below the field reads 'El nombre del proyecto es obligatorio.' At the bottom of the form is a 'Continuar' button.

*Nota:* La figura muestra el nombre del proyecto.

Una vez creado el proyecto permite crear varias aplicaciones que pueden funcionar dentro de una misma base de datos y nos permite trabajar tanto con Android y Apple, en este caso se trabajó para la plataforma de Android y se debe registrar los datos de la aplicación como se muestra en la figura 49.

## Figura 49

### Registro de la base de datos

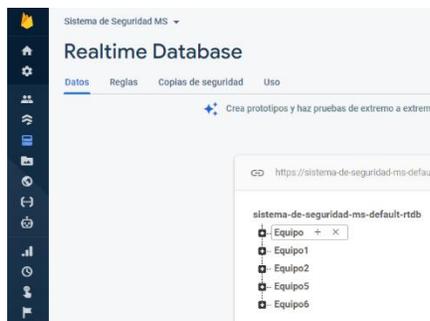


*Nota:* La figura muestra el registro para poder enlazar la base de datos con otras aplicaciones.

Una vez registrada la aplicación la plataforma nos ofrece varios tipos de utilidades en este caso se utilizó Real Time Data Base la cual es una base de datos en tiempo real como se muestra en la figura 50.

## Figura 50

### Realtime Database

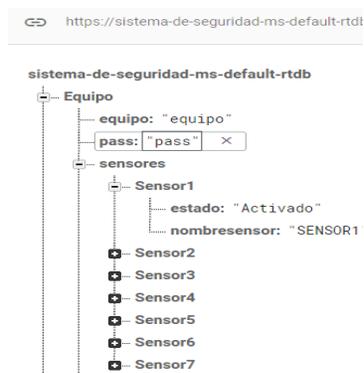


*Nota:* La figura muestra Realtime Database permite recibir la información de los sensores y permite su lectura a través de una aplicación móvil.

Esta base de datos permite la comunicación con el módulo utilizado por lo que los datos del módulo se suben automáticamente en tiempo real a la base de datos y se actualizan en tiempo real permitiendo una visualización inmediata de los cambios en el sistema como se muestra en la figura 51.

## Figura 51

Información subida en base de datos



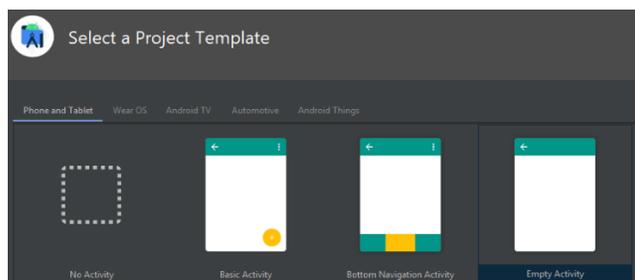
*Nota:* La figura muestra la información es obtenida a través del módulo ESP8866.

### 3.4 Aplicación móvil

Para la creación de una aplicación Android se realizó mediante “Android Studio” que es una interfaz para desarrollar aplicaciones, se comenzó con la creación de un proyecto nuevo donde se establece el tipo de proyecto como se muestra en la figura 52.

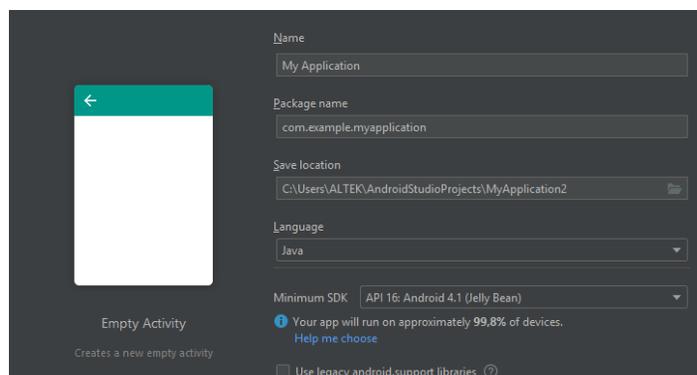
## Figura 52

Creación de un proyecto nuevo



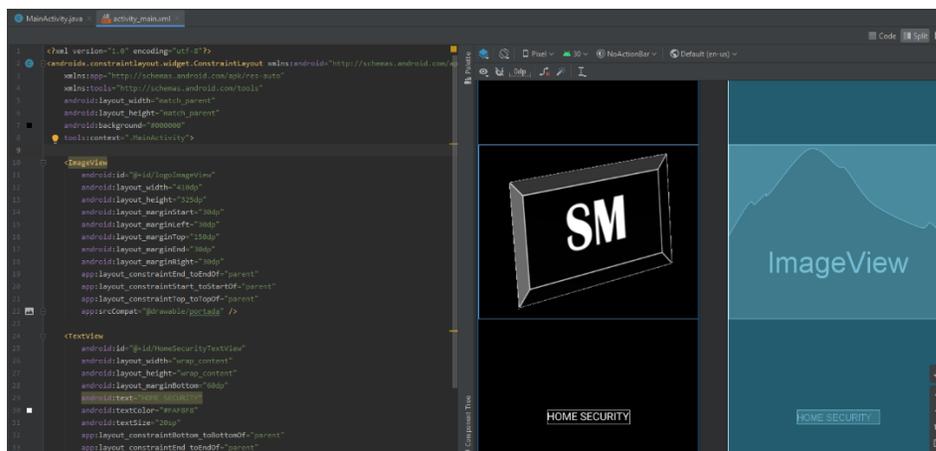
*Nota:* La figura muestra varios tipos de diseños para cada proyecto.

Una vez escogido el tipo de proyecto se coloca un nombre, se selecciona su ubicación y el tipo de lenguaje de programación como se muestra en la figura 53.

**Figura 53***Datos de la aplicación*

*Nota:* La figura muestra el lenguaje de programación.

Una vez creado el proyecto brinda una ventana de diseño y otra de programación en la ventana de diseño es en la que se mostraran todos los datos visuales, y en la ventana de programación se especifica como funcionaran los objetos colocados en la ventana de diseño. En la primera ventana se muestra un inicio de aplicación y el logo por unos segundos, para lo cual se da un diseño como se ve en la figura 54.

**Figura 54***Diseño de un Splash Screen*

*Nota:* La figura muestra la primera vista de la aplicación móvil.

Una vez creado el diseño se programa el tiempo que se va a mostrar esta pantalla y hacia donde nos redirige, para ello se crea una nueva actividad a donde nos llevara una vez mostrada esta pantalla de inicio como se muestra en la figura 55.

**Figura 55**

*Programación del Splash Screen*

```

activity_main.xml
class MainActivity extends AppCompatActivity {
    override
    onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //ANIMACIONES
        Animation animacion1 = AnimationUtils.loadAnimation(this, R.anim.desplazamiento_arriba);
        Animation animacion2 = AnimationUtils.loadAnimation(this, R.anim.desplazamiento_abajo);

        TextView HomeSecurityTextView = findViewById(R.id.HomeSecurityTextView);
        ImageView LogoImageView = findViewById(R.id.LogoImageView);

        HomeSecurityTextView.setAnimation(animacion2);
        LogoImageView.setAnimation(animacion1);
    }
}

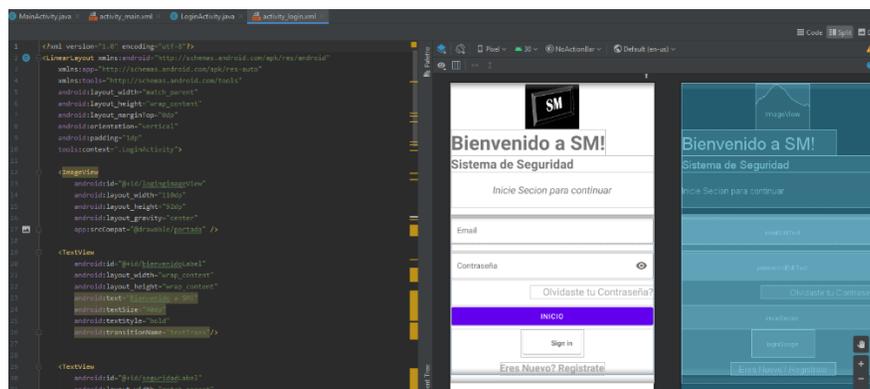
```

*Nota:* La figura muestra el tiempo para cambiar de actividad.

En esta nueva actividad se muestra un inicio de sesión para tener constancia de las personas que ingresan a la aplicación, para ello se creó un diseño para ingresar el correo electrónico y la contraseña, además de un botón para crear una cuenta como se muestra en la figura 56.

**Figura 56**

*Diseño del inicio de sesión*



*Nota:* La figura muestra el inicio de sesión.

Una vez ya establecido el diseño se declara cada una de sus funciones, primero se parte declarando por su tipo como lo son texto, editor de texto y botones, cada uno de estos se los debe enlazar con la actividad de diseño para que funcione de manera sincronizada como se muestra en la figura 57.

### Figura 57

*Configuración del inicio de sesión*

```

activity.java  activity_main.xml  LoginActivity.java  activity_login.xml

public class LoginActivity extends AppCompatActivity {

    TextView bienvenidaLabel,continuarLabel,nuevoUsuario,olvidastecontrasena;
    ImageView loggingimageView;
    TextInputLayout usuarioTextField, contrasenaTextField;
    MaterialButton inicioSesion;
    TextInputEditText emailEditText,passwordEditText;
    private FirebaseAuth mAuth;

    SignInButton signInButton;
    GoogleSignInClient mGoogleSignInClient;
    public static final int RC_SIGN_IN =0;

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

        loggingimageView = findViewById(R.id.loggingimageView);
        bienvenidaLabel = findViewById(R.id.bienvenidaLabel);
        continuarLabel = findViewById(R.id.continuarLabel);
        usuarioTextField = findViewById(R.id.usuarioTextField);
    }
}

```

*Nota:* La figura muestra los campos para que la aplicación funcione adecuadamente

Para cada botón se crea su actividad y se programa su redirección, como lo son los botones de registro, de olvidar contraseña y para un registro por google como se muestra en la figura 58.

### Figura 58

*Configuración de los botones del inicio de sesión.*

```

nuevoUsuario.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent= new Intent( packageContext, LoginActivity, this, Registro.class);
        startActivity(intent);
    }
});

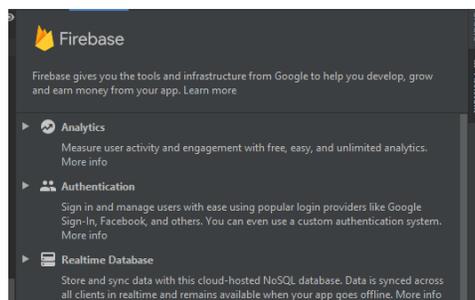
```

*Nota:* La figura muestra la función que cumplen os botones una vez se presionan.

Para el registro y el inicio de sesión se utilizó una de las ventajas de la base de datos de “Firebase” que cuenta con una sección de inicio y registró, para ello se enlaza la aplicación diseñada con Firebase, se inicia sesión con la misma cuenta que se inició la base de datos y se descarga todas las librerías y complementos como se muestra en la figura 59.

### Figura 59

*Sincronización de la base de datos con la aplicación*



*Nota:* La figura muestra las librerías que se va a utilizar

Ya una vez sincronizados nos da la facilidad de un registro e inicio de sesión seguro para los usuarios, y permite el uso de sus librerías de programación dependiendo de cada necesidad, ya con las librerías se declara cada campo y como actuara con la base de datos como se muestra en la figura 60.

### Figura 60

*Declaración de librerías para funciones con la base de datos*

```

        mGoogleSignInClient = GoogleSignIn.getClient( activity, this, gso);
    }

    private void signInwithGoogle () {
        Intent signInIntent = mGoogleSignInClient.getSignInIntent();
        startActivityForResult(signInIntent, RC_SIGN_IN);
    }

    @Override
    public void onActivityResult(int requestCode,int resultCode,Intent data){
        super.onActivityResult(requestCode,resultCode,data);
        if (requestCode == RC_SIGN_IN){
            Task<GoogleSignInAccount>task=GoogleSignIn.getSignedInAccountFromIntent(data);
            try {
                GoogleSignInAccount account = task.getResult(ApiException.class);
                firebaseAuthWithGoogle (account.getIdToken());
            }catch (ApiException e){
                Toast.makeText( context, LoginActivity.this, text: "Fallo al iniciar Google",Toast.LENGTH_SHORT).show();
            }
        }
    }

```

*Nota:* La figura muestra el enlace con la base de datos.

También se establece unas condiciones en caso de que la contraseña sea incorrecta o de que el correo utilizado no este sincronizado con la aplicación como se muestra en la figura 61.

**Figura 61**

*Condiciones al ingresar la contraseña*

```
public void validate(){
    String email = emailEditText.getText().toString().trim();
    String password = passwordEditText.getText().toString().trim();

    if (email.isEmpty() || !Patterns.EMAIL_ADDRESS.matcher(email).matches()){
        emailEditText.setError("Correo invalido");
        return;
    } else {
        emailEditText.setError(null);
    }
}

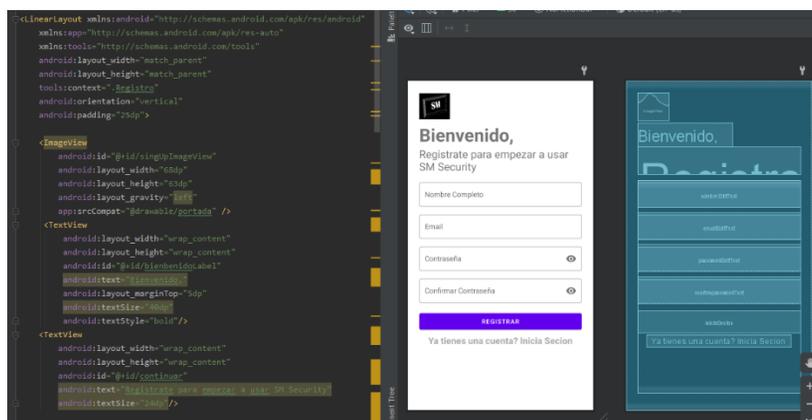
if (password.isEmpty() || password.length() < 8){
    passwordEditText.setError("Se necesitan mas de 8 caracteres");
    return;
} else if (!Pattern.compile("[0-9]").matcher(password).find()){
    passwordEditText.setError("Al menos un numero");
    return;
} else {
    passwordEditText.setError(null);
}
```

*Nota:* La figura muestra cómo se verifica la información con la base de datos.

Se debe establecer cómo será el registro de cada usuario para ello se crea una nueva actividad en donde se mostrara un formulario para que el usuario llene y pueda registrarse como se muestra en la figura 62.

**Figura 62**

*Diseño de un registro*



*Nota:* La figura muestra el registro se pide una verificación de la contraseña.

En el caso del nombre y la contraseña se establecen ciertos parámetros para que sea más seguro como se muestra en la figura 63.

## Figura 63

### Configuración para la contraseña

```

public void validate(){
    String email = emailEditText.getText().toString().trim();
    String password = passwordEditText.getText().toString().trim();
    String confirmPassword = confirmPasswordEditText.getText().toString().trim();
    String nombre=nombreEditText.getText().toString().trim();
    if (nombre.isEmpty() || nombre.length() < 4){
        nombreEditText.setError("Campo Obligatorio");
        return;
    }else {
        nombreEditText.setError(null);
    }

    if (email.isEmpty() || !Patterns.EMAIL_ADDRESS.matcher(email).matches()){
        emailEditText.setError("Correo invalido");
        return;
    }else {
        emailEditText.setError(null);
    }

    if (password.isEmpty() || password.length() < 8){
        passwordEditText.setError("Se necesitan mas de 8 caracteres");
        return;
    }else if (!Pattern.compile("[0-9]").matcher(password).find()){
        passwordEditText.setError("Al menos un numero");
        return;
    }
}

```

*Nota:* La figura muestra cada uno de los campos de contraseña.

De igual manera se establece que pasa en caso de que los datos sean incorrectos impidiendo el registro como se muestra en la figura 64.

## Figura 64

### Verificación de datos

```

@Override
public void onComplete(@NonNull Task<AuthResult> task) {
    if (task.isSuccessful()){
        String nombre =nombreEditText.getText().toString().trim();
        Map<String, Object>usuario = new HashMap<>();
        usuario.put("Nombre", nombre);
        usuario.put("Correo", email);
        referenceUsuarios.push().setValue(usuario);

        Intent intent =new Intent( packageContext, Registro.this, UserActivity.class);
        startActivity(intent);
        finish();
    }else{
        Toast.makeText( context, Registro.this, text: "Fallo en registrarse", Toast.LENGTH_LONG).show();
    }
}

```

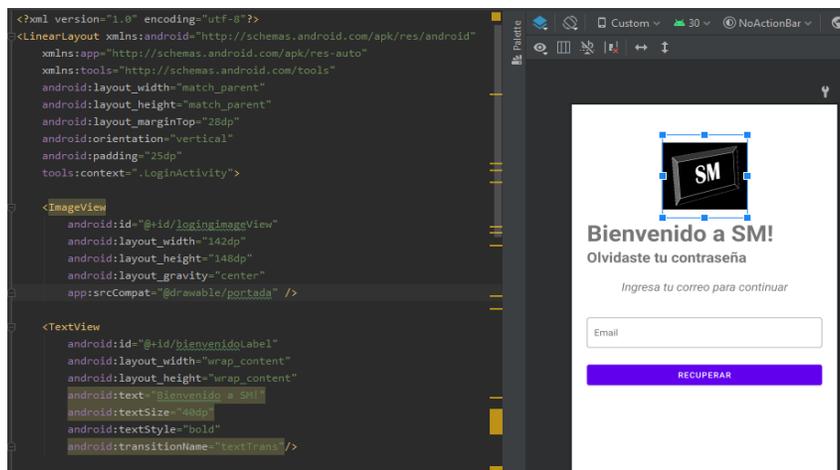
*Nota:* La figura muestra si los datos son incorrectos.

Ya una vez registrado lo regresa nuevamente a la actividad de inicio de sesión donde le permitirá ingresar sus datos.

En caso de olvidarse la contraseña se envía un mensaje al correo para la validación y cambio de contraseña, para ello se crea una actividad en donde ingresara el correo como se muestra en la figura 65.

**Figura 65**

*Diseño de una actividad para recuperar la contraseña*



*Nota:* La figura muestra un campo donde se ingresa el correo electrónico del usuario

Al estar conectado con la base de datos de Firebase esta permite que la recuperación de la contraseña sea automática, únicamente se debe declarar los campos utilizados y la pedida de la contraseña mediante la pulsación del botón como se muestra en la figura 66.

**Figura 66**

*Configuración del botón de Recuperación de contraseña*

```
@Override
public void onBackPressed() {
    super.onBackPressed();

    Intent intent = new Intent( packageContext, forgotpassword.this.LoginActivity.class);
    startActivity(intent);
    finish();
}

public void validate(){
    String email = editText.getText().toString().trim();
    if (email.isEmpty() || !Patterns.EMAIL_ADDRESS.matcher(email).matches()){
        editText.setError("CORREO INVALIDO");
        return;
    }
}
```

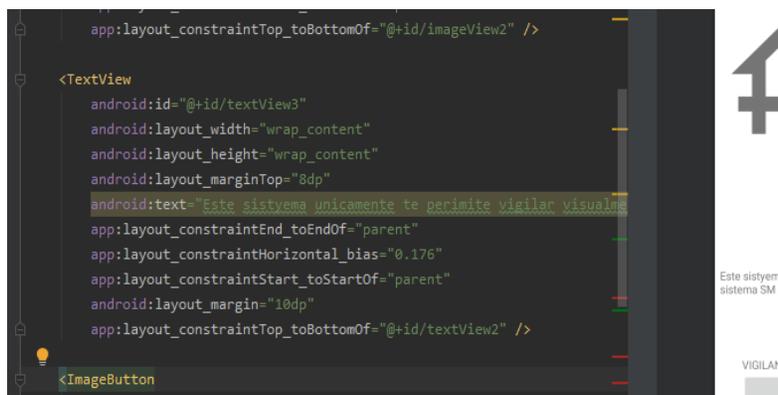
*Nota:* La figura muestra conexión con la base de datos

En esta nueva actividad nos muestra un botón para seleccionar otro dispositivo, un botón para una vigilancia en vivo y otro para visualizar el historial, para ello se

agregó imágenes las cuales le dan un diseño más estético como se muestra en la figura 67.

### Figura 67

*Diseño de una actividad de menú*



*Nota:* La figura muestra el menú que permite moverse entre las ventanas de la aplicación

Esta actividad funcionaria como menú por lo que se deben declarar los botones y sincronizarlos, y se establece a donde este enlazado cada uno de los botones como se muestra en la figura 68.

### Figura 68

*Configuración de los botones de la actividad de menú*

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu);
    botonhistorial = findViewById(R.id.botonhistorial);
    botonvivo = findViewById(R.id.botonvivo);
    botonbuscador = findViewById(R.id.botonbuscador);

    botonhistorial.setOnClickListener(new OnClickListener() {
        Intent intent = new Intent( Menu.this, HistorialSM.class);
        startActivity(intent);
        finish();
    });
    botonvivo.setOnClickListener(new OnClickListener() {
        Intent intent = new Intent( Menu.this, VivoSM.class);
        startActivity(intent);
        finish();
    });
    botonbuscador.setOnClickListener(new OnClickListener() {
        Intent intent = new Intent( Menu.this, Buscador.class);
        startActivity(intent);
        finish();
    });
}

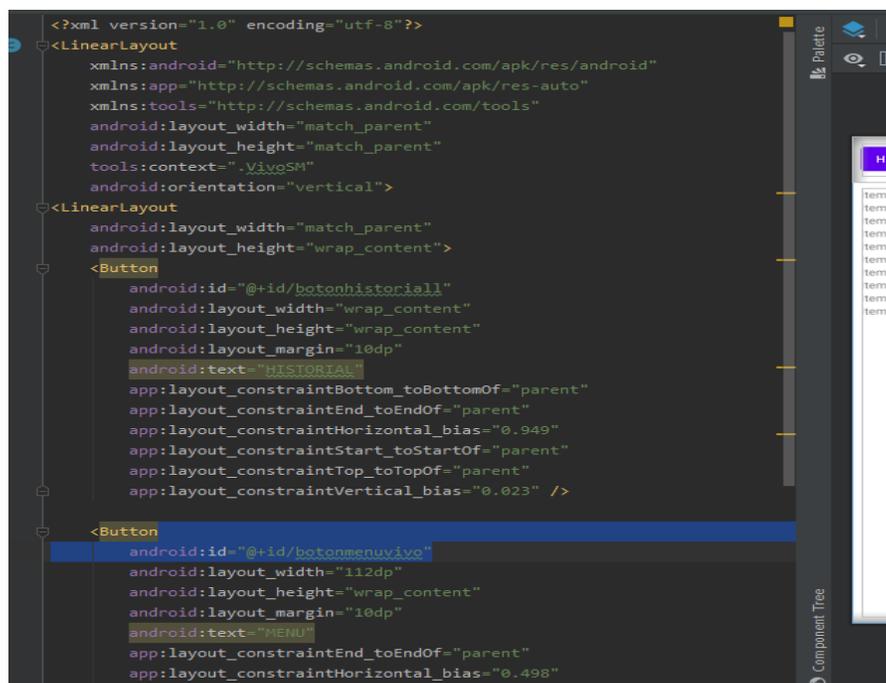
```

*Nota:* La figura muestra el enlace de cada botón.

El botón de vigilancia en vivo nos llevara a una actividad sincronizada con los sensores en tiempo real y nos permitirá ver que sensores están activados y cuales no como se muestra en la figura 69.

**Figura 69**

*Diseño para una visualización en vivo de los sensores*



*Nota:* La figura muestra la vista en vivo en forma de lista.

Se establece una conexión con la base de datos y se instancia que los datos se mostraran dentro de esa lista para ello se configura cada dato como se muestra en la figura 70.

## Figura 70

*Configuración para los datos q se mostraran en la lista*

```

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

    botonhistoriall = findViewById(R.id.botonhistoriall);
    botonmenuvivo = findViewById(R.id.botonmenuvivo);

    ref= FirebaseDatabase.getInstance().getReference("Equipo").child("sensores");

    rvs=findViewById(R.id.rvs);
    lm=new LinearLayoutManager(this);
    rvs.setLayoutManager(lm);
    list=new ArrayList<>();
    adapterSensores= new AdapterSensores(list);
    rvs.setAdapter(adapterSensores);

    ref.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (dataSnapshot.exists()){
                for (DataSnapshot snapshot: dataSnapshot.getChildren()){
                    Sensores se = snapshot.getValue(Sensores.class);
                    list.add(se);
                }
                adapterSensores.notifyDataSetChanged();
            }
        }
    });
}

```

*Nota:* La figura muestra los datos son obtenidos de la base de datos.

Para establecer una conexión con la base de datos se la llaman mediante referencias y se crea un adaptador y un listado que sirven para mostrar los datos, primero se establecen los nombres de las referencias para poder llamar la información de la base de datos a la aplicación como se muestra en la figura 71.

## Figura 71

*Referencias para toma de datos de la base de datos*

```

public class Sensores {
    private String estado;
    private String nombresensor;

    public Sensores() {
    }

    public Sensores(String estado, String nombresensor) {
        this.estado = estado;
        this.nombresensor = nombresensor;
    }

    public String getEstado() { return estado; }

    public void setEstado(String estado) { this.estado = estado; }

    public String getNombresensor() { return nombresensor; }

    public void setNombresensor(String nombresensor) { this.nombresensor = nombresensor; }
}

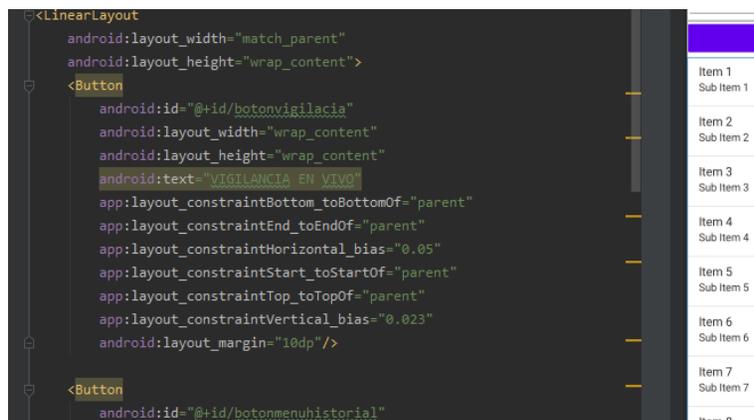
```

*Nota:* La figura muestra Las referencias y los nombres en la base de datos debe ser la misma

De igual manera para el historial se diseñó una vista en donde se encontrara una lista y dos botones como se muestra en la figura 72.

**Figura 72**

*Diseño de un historial*



*Nota:* La figura muestra la vista del historial.

El historial almacena datos por lo que se crea una base de datos en el celular, una nueva actividad funciona como base de datos en esta se instancia los datos que se van a almacenar en este caso de tipo texto como se muestra en la figura 73.

**Figura 73**

*Base de datos interna*

```
public class BaseHelper extends SQLiteOpenHelper {
    String tabla="CREATE TABLE Personas( Id INTEGER PRIMARY KEY AUTOINCREMENT, Nombre Text,Apellido Text, Edad Text,S1 Text,S2 Text, S3 Text,S4 Text,S5 Text, S6 Text)";
    public BaseHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version){
        super(context,name,factory,version);
    }
    @Override
    public void onCreate (SQLiteDatabase db){
        db.execSQL(tabla);
    }
    @Override
    public void onUpgrade(SQLiteDatabase DB, int oldVersion, int newVersion){
```

*Nota:* La figura muestra la base de datos interna que almacena información en el celular.

Una vez se establece la base de datos se crea una lista en donde tomara la fecha y hora y la enviara la base de datos del celular junto con el estado de los

sensores, para ello de declarar cada uno de los sensores a utilizar como se muestra en la figura 74.

### Figura 74

*Configuración de valores para el historial*

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_historiallist);
    mDatabase= FirebaseDatabase.getInstance().getReference( path: "Equipo").child("sensores");
    S1=(EditText) findViewById(R.id.lnombre);
    S2 =(EditText) findViewById(R.id.lapellido);
    S3 =(EditText) findViewById(R.id.ledad);

    botonregreso=findViewById(R.id.botonregreso);
    botonvista=findViewById(R.id.botonhvista);

    sensor1=findViewById(R.id.sensor1);

    fecha=findViewById(R.id.fechacompleta);
    S4=findViewById(R.id.mes);
    S5=findViewById(R.id.dia);
    S6=findViewById(R.id.ayo);
    hora=findViewById(R.id.hora);
    Date date = new Date();

    SimpleDateFormat fechac = new SimpleDateFormat( pattern: "d,MMM 'del',yyy");
    String sfecha= fechac.format(date);
    fecha.setText(sfecha);

    SimpleDateFormat h=new SimpleDateFormat( pattern: "h:mm a");
    String ahora = h.format(date);
    hora.setText(ahora);
}
```

*Nota:* La figura muestra el historial que almacena los cambios de la base de datos.

Estos datos se envían a la lista del historial para que pueda ser visibles como se muestra en la figura 75.

### Figura 75

*Configuración para mostrar la información almacenada en la base de datos del teléfono*

```
public void cargar(){
    BaseHelper baseHelper = new BaseHelper( context: this, name: "DEMO08", factory: null, version: 1);
    SQLiteDatabase db = baseHelper.getReadableDatabase();
    if (db!=null){
        Cursor c = db.rawQuery( sql: "select * from Personas ", selectionArgs: null);
        int cantidad =c.getCount();
        int i =0;
        String[] arreglo = new String[cantidad];
        if (c.moveToFirst()){
            do {
                String linea = c.getInt( columnIndex: 0)+" "+c.getString( columnIndex: 1)+" "+ c.getString( columnIndex: 2)+
                arreglo[i] = linea;
                i++;
            }while (c.moveToNext());
        }
        ArrayAdapter<String>adapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1,arreglo);
        ListView lista = (ListView) findViewById(R.id.listahiss);
        lista.setAdapter(adapter);
    }
}
```

*Nota:* La figura muestra los datos en forma de una lista.

### 3.5 Armado del sistema en físico y prueba del sistema en marcha

Al momento de armar el sistema se comenzó con la implantación de la programación en el Arduino mega, esto mediante un cable USB como se muestra en la figura 76.

#### Figura 76

*Comunicación serial entre Arduino MEGA y PC.*



*Nota:* La figura muestra como la programación es implantada a través de un cable USB.

Para saber que el programa estaba funcionando correctamente se conectó por medio de cables el LCD al microcontrolador y se comprueba que el display muestre un texto legible como se muestra en la figura 77.

#### Figura 77

*Puesta en marcha del LCD*



*Nota:* La figura muestra al LCD es uno de 20x4 funcionando.

Seguidamente se conectó el teclado numérico de 4x4 para verificar que el microcontrolador reciba la contraseña y pueda activar el sistema, y se realizó una prueba ingresando un código incorrecto como se muestra en la figura 78.

### Figura 78

*Prueba de contraseña con un código incorrecto.*



*Nota:* La figura muestra una prueba de funcionamiento del programa.

Se conectó el módulo RTC que indica la hora del día como se muestra en la figura 79.

### Figura 79

*Funcionamiento del módulo RTC*



*Nota:* La figura muestra al módulo RTC enviando información al display.

Se conectó los sensores magnéticos a la placa principal partiendo desde un GND a cada pin de lectura respectivo, como se muestra en la figura 80.

### Figura 80

*Conexión de los sensores magnéticos a la placa principal.*



*Nota.* La figura muestra la conexión de los sensores conectados a la placa principal.

Se añadió los LEDs y un buzzer que funcionan como indicadores para saber si el sistema está activado y cuando esta desactivado como se muestra en la figura 81.

### Figura 81

*Conexión de luces led*

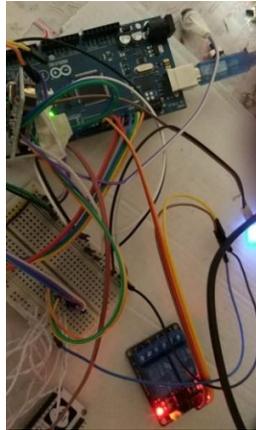


*Nota:* La figura muestra una luz led una roja que indica que el sistema está pagado.

Se conecta los relés que activan una sirena en caso de detectar algún intruso como se muestra en la figura 82.

**Figura 82**

*Conexión de relés a la placa principal.*

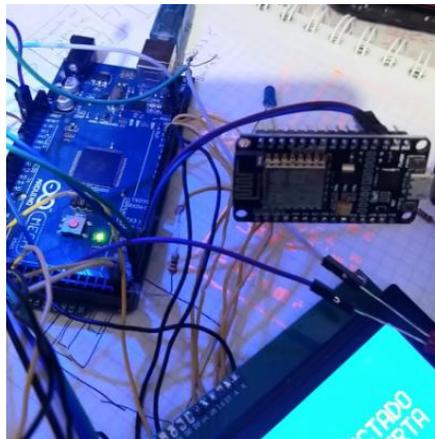


*Nota.* La figura muestra al módulo relé conectado la placa principal.

El módulo WiFi ESP8866 se lo conecto a los pines Rx1 y Tx1 para subir los datos que del arduino mega a la nube como se muestra en la figura 83.

**Figura 83**

*Conexión de módulo WiFi ESP8866*

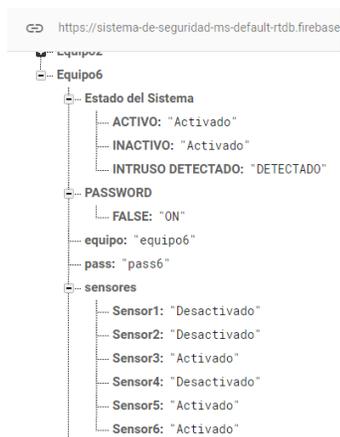


*Nota:* La figura muestra el módulo ESP8266.

Se comprobó que los datos enviados del arduino sean subidos a la base de datos en tiempo real como se muestra en la figura 84.

## Figura 84

### Base de datos "FIREBASE"



*Nota:* La figura muestra los datos que se almacenan y cambian en tiempo real.

Se comprobó que la aplicación era capaz de leerlos sin ningún problema como se muestra en la figura 85.

## Figura 85

### Aplicación Android

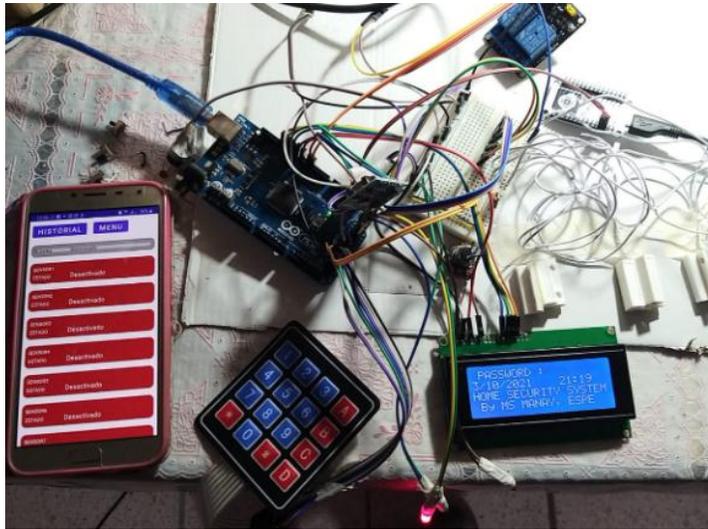


*Nota:* La figura muestra a la aplicación enlazada con la base de datos.

Finalmente se comprobó todo el sistema de seguridad funcionando en forma de maqueta, para pruebas y corrección de fallas como se muestra en la figura 86.

**Figura 86**

*Puesta en marcha del sistema de seguridad.*



*Nota:* La figura muestra que el sistema no cuenta con ninguna falla.

### **3.6 Implementación en un domicilio local**

Para la implementación del sistema en un domicilio local se colocó cada uno de los elementos en una caja protectora para evitar posibles golpes, separando una caja en donde se colocó el teclado y LCD junto con indicadores luminosos, los cuales son los dispositivos se encuentran al alcance más público como se muestra en la figura 87.

**Figura 87**

*Caja de protección para dispositivos táctiles.*



*Nota:* La figura muestra En esta caja se contralara el contraste del LCD.

Para la segunda caja se acoplo el microcontrolador de Arduino el cual funciona como modulo principal junto con el módulo WiFi, RTC y el módulo de relés como se muestra en la figura 88.

### **Figura 88**

Caja principal del sistema de seguridad.



*Nota:* La figura muestra cómo se colocó a los dispositivos en una caja protectora.

Se realizó el cableado necesario para los sensores magnéticos, estos sensores se conectaron tanto en puertas y ventanas como se muestra en la figura 89.

### **Figura 89**

*Instalación de sensores magnéticos*



*Nota:* La figura muestra la conexión de cada uno de los sensores en el domicilio.

Una vez instalado los dispositivos se realizó la conexión de cada uno empezando de la caja donde se encuentra el teclado y el Display a la caja principal como se muestra en la figura 106.

### Figura 90

*Conexión de los dispositivos hacia la caja principal.*



*Nota:* La figura muestra la conexión del LCD, teclado y luces indicadoras hacia el arduino.

En la caja principal se introdujo un conector 110v para la toma de corriente como se muestra en la figura 91.

### Figura 91

*Conexión de 110v.*



*Nota:* La figura muestra la conexión de la toma corriente

Se realizó la conexión del módulo de relés junto con borneras para una facilidad al momento de conectar los dispositivos de 110v como se muestra en la figura 92.

### **Figura 92**

*Conexión del módulo de relés*

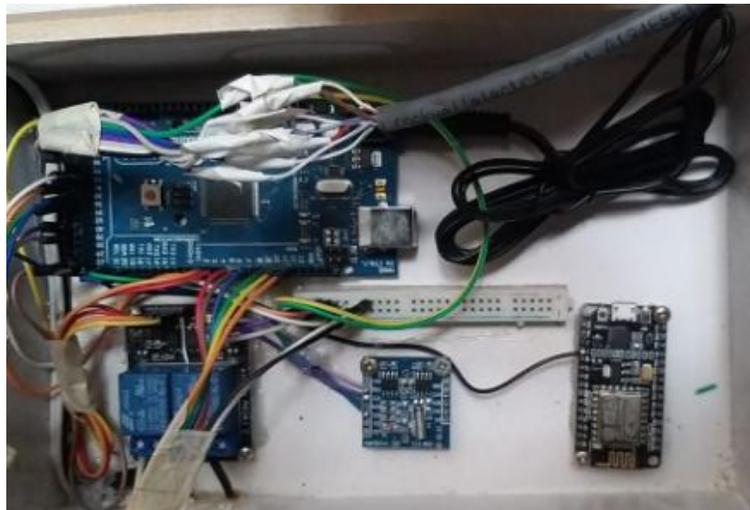


*Nota:* La figura muestra la conexión de las salidas del módulo de relés a borneras.

Se realizó la conexión de cada uno de los módulos dentro de la caja principal hacia el arduino como se muestra en la figura 93.

### **Figura 93**

*Conexión de cada módulo al microcontrolador ARDUINO MEGA.*



*Nota:* La figura muestra el enlace de cada módulo por medio de cables.

Se conectó los cables provenientes la caja donde se encuentra el teclado a la caja principal donde se están los módulos como se muestra en la figura 94.

### **Figura 94**

Conexión de cables del teclado a Arduino Mega

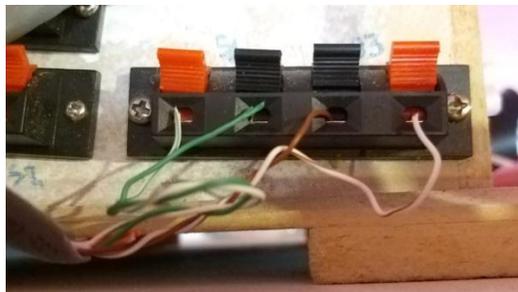


*Nota:* La figura muestra la conexión de cables entre dispositivos.

Se conectan los sensores a la caja principal y se verifica el correcto funcionamiento del sistema como se muestra en la figura 95.

### **Figura 95**

*Conexión de sensores a la caja principal.*



*Nota:* La figura muestra a las borneras que facilitan la conexión de los sensores.

Se instaló la caja donde se encuentran los dispositivos táctiles en el domicilio como se muestra en la figura 96.

**Figura 96**

*Fijación de la caja de dispositivos táctiles.*



*Nota:* La figura muestra a la caja del teclado ya fijada en una pared.

Se colocó la caja principal del sistema en un lugar estratégico el cual no esté cerca de intrusos y junto a ella se conectaron las sirenas como se muestra en la figura 97.

**Figura 97**

*Fijación de la caja principal junto con las sirenas de alarma.*



*Nota:* La figura muestra el uso de borneras para facilitar la conexión de las sirenas.

Finalmente se realizaron las conexiones respectivas y se agregó una cámara IP en una zona cercana al techo para que tenga un mejor rango de grabación como se muestra en la figura 98.

**Figura 98**

*Instalación de cámara IP.*



*Nota:* La figura muestra La cámara debe estar cerca de una conexión a internet para que pueda funcionar adecuadamente.

## CAPÍTULO IV

### 4 Conclusiones y Recomendaciones

#### 4.1 Conclusiones

- Mediante un microcontrolador Arduino mega 2560 se realizó el diseño de un sistema de seguridad inalámbrico para protección de viviendas y locales comerciales, que necesiten de protección continua y precisa.
- En base a datos recolectados se establece que un sistema de seguridad inalámbrico posee grandes beneficios y presenta una mayor fiabilidad cuando de protección domiciliaria se trata.
- Dentro de un sistema de seguridad inalámbrico se puede encontrar una gran variedad de elementos receptores y actuadores que permiten establecer un rango determinado de seguridad.
- Con un microcontrolador Arduino y con una variedad de módulos disponibles para éste, se desarrolló y sistema capaz de informar al usuario cuando un perímetro ha sido violado.

## 4.2 Recomendaciones

- Para realizar un trabajo escrito como la monografía o tesis es importante partir del trabajo técnico ya realizado, esto debido a que una vez ya realizado el trabajo se puede especificar más claramente tanto los componentes como el procedimiento que se siguió para conseguirlo.
- Para una comunicación serial se debe asegurar que la velocidad en baudios sea la misma en ambos dispositivos que se enlazarán, caso contrario su comunicación no será satisfactoria o no se llegaran a comunicar.
- Para una buena transmisión de datos a la nube dependerá de la calidad de internet que cuente, esto debido a la subida y bajada de datos.
- La conexión de cables del microcontrolador a cada dispositivo debe ser segura, ya que una mala conexión de cables puede desencadenar una mala recepción de datos y por ende el sistema no funcionara adecuadamente.
- La placa principal debe estar asegurada en una caja y en un lugar donde evite golpes y humedad, ya que los elementos son sensibles y una caída puede afectarlos proporcionalmente.

## Bibliografía

- AndroidStudio. (22 de Mayo de 2021). *developers*. Recuperado el 14 de Diciembre de 2020 Obtenido de <https://developer.android.com/studio/intro?hl=es-419>
- ARDUINO. (12 de DICIEMBRE de 2020). *ARDUINO*. Recuperado el 18 de Diciembre de 2020 Obtenido de <http://arduino.cl/que-es-arduino/>.
- Arduino. (s.f.). *Arduino*. Recuperado el 08 de Marzo de 2020, de <http://arduino.cl/que-es-arduino/>
- CARDONA, M. P. (14 de OCTUBRE de 2016). Recuperado el 6 de Diciembre de 2020 *IEBS*. Obtenido de <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>
- Castillo, J. (23 de Febrero de 2019). *Profesionalreview*. Recuperado el 6 de Enero de 2021 Obtenido de <https://www.profesionalreview.com/2019/02/23/que-es-sai/>
- Conejos, J. B. (Septiembre de 2014). *Sistema de seguridad perimetral programable inteligente*. Recuperado el 6 de Diciembre de 2020 Obtenido de Escola Tècnica Superior d'Enginyeria Informàtica Universitat Politècnica de València: <https://riunet.upv.es/bitstream/handle/10251/39849/Memoria.pdf?sequence=1>
- ELECTROPRO. (2017). *www.electropro.pe*. Recuperado el 4 de Enero de 2021 Obtenido de <http://electropro.pe/image/data/imgProductos/140.%20M%C3%B3dulo%20GSM%20SIM800/SIM800L.pdf>
- FERNÁNDEZ, Y. (3 de Agosto de 2020). Recuperado el 16 de Febrero de 2021 *xataka*. Obtenido de <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

González, A. G. (23 de Enero de 2013). *PANAMAHITEK*. Recuperado el 8 de Enero de 2021 Obtenido de <http://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>

Gutiérrez, O. E. (BOLIVIA de LA PAZ de 2016). *SISTEMA DE SEGURIDAD DOMICILIARIA BASADA EN TECNOLOGÍA ARDUINO Y APLICACIÓN MÓVIL*. Obtenido de UNIVERSIDAD MAYOR DE SAN ANDRÉS FACULTAD DE CIENCIAS PURAS Y NATURALES CARRERA DE INFORMÁTICA: Recuperado el 2 de Febrero de 2021 Obtenido de [:https://repositorio.umsa.bo/bitstream/handle/123456789/7661/T.3103.pdf?sequence=1&isAllowed=y](https://repositorio.umsa.bo/bitstream/handle/123456789/7661/T.3103.pdf?sequence=1&isAllowed=y)

Hetpro. (22 de Febrero de 2018). *I2C – Puerto, Introducción, trama y protocolo*. Recuperado el 22 de Febrero de 2021. Obtenido de <https://hetpro-store.com/TUTORIALES/i2c/#:~:text=En%20el%20caso%20de%20I2C,Datos%20Seriales%20y%20Reloj%20Serial>.

Jaramillo, E. M. (2011). *repositorio.ute*. Recuperado el 27 de Diciembre de 2020 Obtenido de [http://repositorio.ute.edu.ec/bitstream/123456789/4898/1/44172\\_1.pdf](http://repositorio.ute.edu.ec/bitstream/123456789/4898/1/44172_1.pdf)

Llamas, L. (23 de JULIO 23 de 2016 ). *MANEJAR CARGAS DE MÁS DE 220V CON ARDUINO Y SALIDA POR RELÉ* . Recuperado el 9 de Febrero de 2021. Obtenido de <https://www.luisllamas.es/arduino-salida-rele/#:~:text=Un%20rel%C3%A9%20es%20un%20dispositivo,que%20su%20electr%C3%B3nica%20puede%20soportar.&text=El%20circuito%20primario%20se%20conecta,se%20B1al%20de%20encendido%20y%20apagado>.

Llamas, L. (27 de Mayo de 2017). Recuperado el 14 de Diciembre de 2020 Obtenido de

CONECTAR ARDUINO POR WIFI CON EL MÓDULO ESP8266 ESP01:

<https://www.luisllamas.es/arduino-wifi-esp8266-esp01/>

Llamas, L. (27 de Mayo de 2017). *CONECTAR ARDUINO POR WIFI CON EL*

*MÓDULO ESP8266 ESP01*. Recuperado el 24 de Diciembre de 2020 Obtenido

de <https://www.luisllamas.es/arduino-wifi-esp8266-esp01/>

López, J. (21 de Octubre de 2019 ). *HZ HARDZONE*. Recuperado el 25 de Febrero de

2021 Obtenido de <https://hardzone.es/reportajes/tema/esp8266-2n2222-arduino/>

Monje, D. G. (Junio de 2014 ). *Diseño e implementación de un sistema de seguridad*

*para el hogar con interfaz en Android*. Obtenido de Universidad del país vasco

Facultad de Informática Recuperado el 21 de Diciembre de 2020. Obtenido de:

<https://addi.ehu.es/bitstream/handle/10810/13331/PFC.pdf?sequence=2&isAllowed=y>

Montes, L. (08 de Mayo de 2018). *SEGUCOMP*. Recuperado el 22 de Diciembre de

2020. Obtenido de <https://seguridadcompartida.mx/importancia-de-las-sirenas-en-las-alarmas/>

Montoya Rivera, D. A. (2015). *Diseño de un sistema de alarma inalámbrica para*

*pequeñas empresas y hogares en Pereira y Dosquebradas*. Recuperado el 12

de Enero de 2020. Obtenido de Pereira : Universidad Tecnológica de Pereira

Recuperado el 14 de Diciembre de 2020. Obtenido de:

<http://repositorio.utp.edu.co/dspace/handle/11059/6731>

MORA, H. (2008). *Manual del vigilante de seguridad*. Editorial Club Universitario. Madrid.

Recuperado el 4 de Febrero de 2021. Obtenido de

[http://epn.gov.co/elearning/distinguidos/SEGURIDAD/1\\_conceptos\\_de\\_seguridad.html](http://epn.gov.co/elearning/distinguidos/SEGURIDAD/1_conceptos_de_seguridad.html)

- Morán, K. D.-F. (septiembre de 2018). *Diseñar un Sistema de Alarma Inalámbrico de Bajo costo para la protección de viviendas tipo, en sectores de bajo recursos económicos de la ciudad de guayaquil*. Recuperado el 2 de Enero de 2021. Obtenido de Universidad de Guayaquil Facultad de Ciencias Matemáticas y Físicas Carrera de Ingeniería en networking y telecomunicaciones: <http://repositorio.ug.edu.ec/bitstream/redug/32554/1/B-CINT-PTG-N.294%20%20Gaibor%20Carrillo%20Karina%20Dolores%20.%20Loor%20Mor%C3%A1n%20Fernando%20Antonio.pdf>
- NIVIAN. (18 de Abril de 2017). *nivianhome.com*. Recuperado el 16 de Diciembre de 2020 Obtenido de <https://www.nivianhome.com/es/que-es-una-camara-ip/>
- PDAControl. (6 de Abril de 2016). *Prueba Arduino Mega 2560 y ESP8266 (Comandos AT)*. Recuperado el 14 de Febrero de 2020. Obtenido de <http://pdacontroles.com/prueba-arduino-mega-2560-y-esp8266/>
- PROMETEC. (2018). *www.prometec.net*. Recuperado el 13 de Febrero de 2021. Obtenido de <https://www.prometec.net/teclados-matriciales/#:~:text=Los%20teclados%20matriciales%20usan%20una,una%20fila%20y%20a%20una%20columna.&text=De%20este%20modo%2C%20para%20leer,a%20la%20hora%20de%20programar.>
- Quispe, I. S. (Noviembre de 2015). *Draapurimac*. Recuperado el 22 de Enero de 2021 Obtenido de <http://draapurimac.gob.pe/sites/default/files/revistas/MANUAL%20CUY%20PDF.pdf>
- Radic. (2016). *Radic Comunicaciones*. Recuperado el 12 de Enero de 2021. Obtenido de <http://www.radiocomunicaciones.net/camaras-ip/125-camara-ip-wptz.html>

Salvatore. (13th de Agosto de 2019). *Tier4*. Recuperado el 28 de Diciembre de 2020.

Obtenido de <https://www.tier4.com.mx/que-es-y-para-que-sirve-un-ups/>

SHOPTRONICA. (2020). *www.shoptronica.com*. Recuperado el 22 de Diciembre de

2020 Obtenido de [https://www.shoptronica.com/curiosidades-tutoriales-y-gadgets/3981-que-son-los-interruptor-magnetico-reed-switch-](https://www.shoptronica.com/curiosidades-tutoriales-y-gadgets/3981-que-son-los-interruptor-magnetico-reed-switch-0689593949974.html)

[0689593949974.html](https://www.shoptronica.com/curiosidades-tutoriales-y-gadgets/3981-que-son-los-interruptor-magnetico-reed-switch-0689593949974.html)

TECMIKRO. (2020). *tecmikro.com*. Recuperado el 25 de Diciembre de 2020 Obtenido de

<https://tecmikro.com/led-lcd/182-pantalla-lcd-20x4.html>

TWENERGY. (5 de DICIEMBRE de 2019). *twenergy.com/*. Recuperado el 16 de

Diciembre de 2020 Obtenido de [https://twenergy.com/eficiencia-](https://twenergy.com/eficiencia-energetica/consejos-sobre-ahorro-de-energia/que-son-pilas-recargables/)

[energetica/consejos-sobre-ahorro-de-energia/que-son-pilas-recargables/](https://twenergy.com/eficiencia-energetica/consejos-sobre-ahorro-de-energia/que-son-pilas-recargables/)

**ANEXOS**