



**Sistema de seguridad para el control y monitoreo de la unidad educativa Lago San Pablo cantón Pujilí  
a través de una plataforma web**

Chiluisa Guato, Jaime Tarquino

Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

Centro de Posgrados

Maestría en Electrónica y Automatización mención Redes Industriales

Trabajo de titulación, previo a la obtención del título de master en Electrónica y Automatización  
mención Redes Industriales

Msc. Naranjo Hidalgo, César Alfredo

12 de diciembre del 2022

Latacunga



## Tesis escrito Jaime\_Chiluisa.pdf

Scanned on: 13:42 November 7, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	288
Words with Minor Changes	103
Paraphrased Words	598



CESAR ALFREDO  
NARANJO HIDALGO

Ing. César Alfredo Naranjo Hidalgo, MSc.



## Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

### Centro de Posgrados

#### Certificación

Certifico que el trabajo de titulación, **“Sistema de seguridad para el control y monitoreo de la unidad educativa Lago San Pablo Cantón Pujilí a través de una Plataforma Web”** fue realizado por el señor **Chiluisa Guato, Jaime Tarquino** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o similitud de contenidos ; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 12 de diciembre de 2022

Firma:



Firmado electrónicamente por:  
**CESAR ALFREDO NARANJO  
HIDALGO**

.....  
Ing. Naranjo Hidalgo, César Alfredo MSc.

Director

C.C.:0501498505



Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

Centro de Posgrados

**Responsabilidad de autoría**

Yo **Chiluisa Guato, Jaime Tarquino**, con cédula de ciudadanía n° 0503318321, declaro que el contenido, ideas y criterios del trabajo de titulación: **"Sistema de seguridad para el control y monitoreo de la unidad educativa Lago San Pablo Cantón Pujilí a través de una Plataforma Web"** 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, 12 de diciembre de 2022

Firma

Chiluisa Guato, Jaime Tarquino

C.C.: 0503318321



Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

Centro de Posgrados

Autorización de publicación

Yo **Chiluisa Guato, Jaime Tarquino**, con cedula de ciudadanía n° 0503318321 autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: "**Sistema de seguridad para el control y monitoreo de la Unidad Educativa Lago San Pablo Cantón Pujilí a través de una Plataforma Web**" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Latacunga, 12 de diciembre de 2022

Firma

Chiluisa Guato, Jaime Tarquino

C.C.: 0503318321



### **Dedicatoria**

Este trabajo lo dedico a mis padres Leonardo y Luz María, a mi esposa Carla y a mi hija Doménica Sophia que son los pilares más importantes en mi vida. Su amor incondicional, el regalo más grande que Dios me dio, han sido mi inspiración para seguir adelante en todos mis proyectos.

A todas las personas que ayudaron directa e indirectamente para la culminación de este proyecto.



## Agradecimiento

Agradezco a Dios y a Divino Niño de Isinche por darme salud, sabiduría y la fuerza para sobrellevar las dificultades del camino, y lograr todas las metas que me he propuesto y me propondré durante mi vida profesional, agradezco a mi familia de manera especial a mis padres Leonardo y Luz María, a mi esposa Carla, A mi hija Doménica Sophia, a mis Abuelo Andrés y Aurora y a mis hermanos Martha, Andrés, Gloria, Carlos, quienes han estado y estarán presentes en cada peldaño de la vida.

Enfatizo mis más sinceros agradecimientos a la Universidad de las Fuerzas Armadas ESPE, en especial al Centro de Posgrados por su programa de maestría en Electrónica y automatización mención Redes Industriales.

A todas las personas que de una u otra manera aportaron para que el presente proyecto haya llegado a su feliz término

A todos ustedes mi eterna gratitud.

**ÍNDICE DE CONTENIDOS**

<b>Carátula .....</b>	<b>1</b>
<b>Reporte de verificación de contenido .....</b>	<b>2</b>
<b>Certificación .....</b>	<b>3</b>
<b>Responsabilidad de autoría .....</b>	<b>4</b>
<b>Dedicatoria .....</b>	<b>6</b>
<b>Agradecimiento .....</b>	<b>7</b>
<b>Índice de contenidos.....</b>	<b>8</b>
<b>Índice de figuras .....</b>	<b>12</b>
<b>Índice de Tablas .....</b>	<b>14</b>
<b>Tabla de acrónimos.....</b>	<b>15</b>
<b>Resumen .....</b>	<b>16</b>
<b>Abstract .....</b>	<b>17</b>
<b>Capítulo I: Introducción .....</b>	<b>18</b>
<b>Justificación .....</b>	<b>19</b>
<b>Hipótesis de investigación .....</b>	<b>20</b>
<b>Antecedentes.....</b>	<b>20</b>
<b>Objetivo general del proyecto .....</b>	<b>21</b>
<b>Objetivos específicos del proyecto.....</b>	<b>21</b>
<b>Capítulo II: Marco teórico .....</b>	<b>23</b>
<b>Introducción .....</b>	<b>23</b>
<b>Marco conceptual .....</b>	<b>24</b>
<b><i>Domótica e Inmótica</i> .....</b>	<b>24</b>
<b><i>Sistema de seguridad</i> .....</b>	<b>25</b>

<i>Sistemas de control BMS</i> .....	25
<i>Arquitectura de sistemas de control BMS</i> .....	26
<i>Protocolo de comunicación</i> .....	27
<i>Plataforma de gestión</i> .....	28
<i>Sistema de gestión API</i> .....	28
<i>Protocolo de comunicación API</i> .....	29
SNMP .....	29
MIB y OID.....	30
Mensaje SMNP.....	30
Macrodatos.....	31
<i>Raspberry PI</i> .....	32
Puertos USB. ....	33
Pines GPIO. ....	34
Conector DSI. ....	35
Conector CSI.....	35
Puerto Ethernet. ....	36
Alimentación.....	36
Salidas de audio. ....	36
Conector HDMI. ....	36
Almacenamiento.....	36
Software. ....	36
<i>Tipos de Raspberry PI</i> .....	37
<i>Lenguaje de programación JavaScript</i> .....	38
Posibilidades y limitaciones. ....	38

	10
Sintaxis.....	39
<b>Frameworks de programación .....</b>	<b>40</b>
<b>Frameworks backend.....</b>	<b>40</b>
<i>Express.....</i>	<i>40</i>
<i>Node.Js.....</i>	<i>41</i>
<b>Frameworks frontend. ....</b>	<b>43</b>
<i>React Js.....</i>	<i>43</i>
<i>Arquitectura de una aplicación React.js.....</i>	<i>44</i>
<i>Estructura y componentes.....</i>	<i>45</i>
<b>Capítulo III: Diseño de sistema de Seguridad .....</b>	<b>46</b>
<b>Selección de equipo .....</b>	<b>48</b>
<b>Diseño del backend.....</b>	<b>51</b>
<i>Usuarios implementados .....</i>	<i>53</i>
<i>Tabla consumo.....</i>	<i>54</i>
<i>Tabla de credenciales.....</i>	<i>55</i>
<i>Tabla alarmas .....</i>	<i>56</i>
<b>Diseño del frontend .....</b>	<b>57</b>
<b>Consumo de API'S.....</b>	<b>58</b>
<i>API eWeLink.....</i>	<i>58</i>
<b>Conformación de sistema de seguridad .....</b>	<b>59</b>
<b>Diseño del tablero de control para transferencia automática de energía .....</b>	<b>61</b>
<b>Diagrama de fuerza y control de sistema de transferencia automática .....</b>	<b>63</b>
<b>Capítulo IV: Pruebas y resultados .....</b>	<b>65</b>
<b>Creación de nuevo usuario desde el sistema de seguridad. ....</b>	<b>66</b>

<i>Ingreso a la página web de sistema de seguridad de la Unidad Educativa</i> .....	68
<i>Pruebas de envío de peticiones desde el sistema de seguridad hacia los equipos</i> .....	72
Pruebas de comunicación módulo sonoff con backend. ....	72
Pruebas de comunicación con los dispositivos en formato JSON .....	73
Prueba de recolección de Datos utilizando MongoDB.....	73
Pruebas de rendimiento del sistema al almacenar Datos. ....	74
<b>Capítulo V: Conclusiones y Recomendaciones</b> .....	77
<b>Conclusiones</b> .....	77
<b>Recomendaciones</b> .....	79
<b>Bibliografía</b> .....	80
<b>Anexos</b> .....	82

## ÍNDICE DE FIGURAS

<b>Figura 1</b> <i>Sistema de seguridad</i> .....	25
<b>Figura 2</b> <i>Gestiones de sistemas BMS</i> .....	26
<b>Figura 3</b> <i>Arquitectura de sistemas BMS</i> .....	27
<b>Figura 4</b> <i>Estructura de funcionamiento API</i> .....	29
<b>Figura 5</b> <i>Tipos de datos que maneja Big Data</i> .....	31
<b>Figura 6</b> <i>Placa Raspberry PI</i> .....	32
<b>Figura 7</b> <i>Diagrama de Puerto USB</i> .....	34
<b>Figura 8</b> <i>Distribución de pines GPIO raspberry PI</i> .....	35
<b>Figura 9</b> <i>Funcionamiento Node.js</i> .....	42
<b>Figura 10</b> <i>Estructura Básica FRONT- END</i> .....	43
<b>Figura 11</b> <i>División de carpetas del proyecto en React.js</i> .....	44
<b>Figura 12</b> <i>Diagrama de flujo frontEnd de sistema de seguridad</i> .....	47
<b>Figura 13</b> <i>Diagrama de flujo backEnd de sistema de seguridad</i> .....	47
<b>Figura 14</b> <i>Esquema general del sistema implementado</i> .....	49
<b>Figura 15</b> <i>Terminal de comandos integrada en visual studio code.</i> .....	51
<b>Figura 16</b> <i>Archivo de configuración package.json</i> .....	52
<b>Figura 17</b> <i>Instalación del framework express</i> .....	52
<b>Figura 18</b> <i>Diagrama de bloques del backend</i> .....	53
<b>Figura 19</b> <i>Estructura de la tabla de usuarios</i> .....	54
<b>Figura 20</b> <i>Estructura de la tabla de alarmas</i> .....	57
<b>Figura 21</b> <i>Diseño de Frontend.</i> .....	58
<b>Figura 22</b> <i>Instalación del API eWeLink</i> .....	59
<b>Figura 23</b> <i>Conformación de sistema de seguridad</i> .....	60

<b>Figura 24</b> <i>Diagrama de conexiones sistema de seguridad</i> .....	61
<b>Figura 25</b> <i>Sistema de transferencia de energía</i> .....	63
<b>Figura 26</b> <i>Diagrama de potencia de sistema de transferencia</i> . ....	63
<b>Figura 27</b> <i>Diagrama de control de sistema de transferencia</i> . ....	64
<b>Figura 28</b> <i>Diagrama de control de sistema de transferencia</i> .....	66
<b>Figura 29</b> <i>Registro en el sistema de seguridad</i> .....	67
<b>Figura 30</b> <i>Plataforma principal, ingreso como administrador</i> . ....	67
<b>Figura 31</b> <i>Plataforma principal, ingreso como administrador</i> . ....	68
<b>Figura 32</b> <i>Plataforma principal de ingreso</i> .....	69
<b>Figura 33</b> <i>Datos incorrectos al ingresar al sistema de seguridad</i> .....	69
<b>Figura 34</b> <i>Plataforma principal de sistema de seguridad</i> . ....	70
<b>Figura 35</b> <i>Pantalla de mandos de sistema de seguridad</i> . ....	70
<b>Figura 36</b> <i>Pantalla de eventos de sistema de seguridad</i> .....	71
<b>Figura 37</b> <i>Pantalla de videos de sistema de seguridad</i> . ....	71
<b>Figura 38</b> <i>Pantalla de control de acceso de seguridad</i> . ....	72
<b>Figura 39</b> <i>Respuesta de testing módulo sonoff con backend</i> .....	72
<b>Figura 40</b> <i>Respuesta testing comunicación frontend y backend</i> . ....	73
<b>Figura 41</b> <i>Prueba de lectura de base de datos con MongoDB</i> . ....	74
<b>Figura 42</b> <i>Tendencia de rendimiento del sistema al almacenar datos</i> .....	75

**ÍNDICE DE TABLAS**

<b>Tabla 1</b> <i>Partes de raspberry pi</i> .....	33
<b>Tabla 2</b> <i>Tipos de raspberry PI y sus características</i> .....	37
<b>Tabla 3</b> <i>Equipos para sistema de seguridad</i> .....	48
<b>Tabla 4</b> <i>Usuarios implementados en sistema de seguridad</i> .....	53
<b>Tabla 5</b> <i>Tabla de consumo de sistema de seguridad</i> . ....	55
<b>Tabla 6</b> <i>Tabla de credenciales de sistema de seguridad</i> . ....	55
<b>Tabla 7</b> <i>Tabla de alarmas de sistema de seguridad</i> . ....	56
<b>Tabla 8</b> <i>Tabla de consumo eléctrico de sistema de seguridad</i> . ....	61

**Tabla de acrónimos**

<b>BMS:</b>	Building management system, Sistemas de gestión de edificios.
<b>WSN:</b>	Wireless network sensor.
<b>IOT:</b>	Internet of Things, Internet de las cosas
<b>IIOT:</b>	(industrial internet of things, Internet Industrial de las Cosas
<b>BACnet</b>	Building Automation and Control Networks, Redes de control y automatización de edificios
<b>RTU</b>	Remote Terminal Unit, Unidad de terminal remota
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol, Protocolo de control de transmisión/Protocolo de Internet
<b>PLC</b>	Controlador lógico programable
<b>DALI</b>	Digital Addressable Lighting Interface, Interfaz digital de iluminación direccionable
<b>DSI</b>	Display Serial Interface, Interfaz serie de la pantalla
<b>GPIO</b>	General Purpose Input/Output, Entrada/salida de propósito general
<b>CSI</b>	Camera Serial Interface, Interfaz serie de la cámara
<b>XBMC</b>	Xbox Media Center, Centro multimedia de Xbox
<b>CRUD</b>	Create, Read, Update y Delete, crear, leer, actualizar, eliminar.
<b>DOM</b>	Document Object Model, Modelo de objetos del documento
<b>HTML</b>	HyperText Markup Language, Lenguaje de Marcas de Hipertexto
<b>CSS</b>	Cascading style sheets, hojas de estilo en cascada.
<b>SPA</b>	Single Page Application, Aplicación de una sola página

## Resumen

En este proyecto de titulación se propone implementar un sistema de seguridad mediante la integración de equipos de diferentes tecnologías para el control, monitoreo y generación de macrodatos de la Unidad Educativa Lago San Pablo Cantón Pujilí, a través de una plataforma Web. Para lo cual, primeramente, se considera investigar temas relacionado a sistemas de seguridad. Para implementar el sistema de seguridad se desarrollará un tablero máster que permita integrar los diferentes sensores y actuadores con un controlador y su respectiva conexión al switch ethernet. También se desarrollará un tablero de transferencia automática para conexión a un generador de 110V en caso de detectar alguna falla en el suministro de energía, para interactuar con el sistema de seguridad se desarrollará una interfaz de usuario HMI a través de un software libre, en el cual los habitantes del Barrio puedan tener acceso para el control, monitoreo y revisión de datos almacenados en la nube, la plataforma constará de seguridad, notificaciones, privilegios, históricos. Finalmente realizar pruebas experimentales para evaluar el funcionamiento del sistema de seguridad implementado en la Unidad Educativa Lago San Pablo. Para la evaluación se considerará a los habitantes del barrio Cañar Bajo del Cantón Pujilí para comprobar el correcto funcionamiento del sistema de seguridad implementado.

*Palabras Clave:* Sistemas domóticos, automatización de edificios, sistemas BMS.

### **Abstract**

In this degree project we propose to implement a security system through the integration of equipment of different technologies for the control, monitoring and generation of macro data of the Educational Unit Lake San Pablo Canton Pujilí, through a Web platform. For which, first, it is considered to investigate issues related to security systems. To implement the security system, a master board will be developed to integrate the different sensors and actuators with a controller and its respective connection to the ethernet switch. An automatic transfer board will also be developed for connection to a 110V generator in case of detecting any failure in the power supply, to interact with the security system an HMI user interface will be developed through a free software, in which the inhabitants of the neighborhood can have access to control, monitoring and review of data stored in the cloud, the platform will consist of security, notifications, privileges, historical. Finally, experimental tests will be carried out to evaluate the performance of the security system implemented in the San Pablo Lake Educational Unit. For the evaluation will be considered the inhabitants of the neighborhood Cañar Bajo del Canton Pujilí to verify the proper functioning of the implemented security system.

*Keywords:* Home automation systems, Building automation, BMS systems.

## Capítulo I

### Introducción

La tendencia de la Domótica y el sistema de gestión de edificios (BMS), ha desencadenado un constante desarrollo que permite combinar la tecnología con la experiencia de la vida cotidiana, asegurar la red de una edificación inteligente requiere el aprovechamiento de varios protocolos, tecnologías, máquinas, herramientas y procedimientos para proteger los datos y disminuir las amenazas (Raghad Al-Syouf, 2021).

Desde que se propuso el concepto de domótica en los años 1980, los hogares inteligentes se han desarrollado rápidamente, a mediados del siglo 80 se produjo una tendencia de casas inteligentes en todo el mundo, aunque el término de casa inteligente ha sido alto, no se ha concretado por una serie de razones. La primera razón es los altos precios para su desarrollo, otra de las razones ha sido la distribución estructural del hogar que hace imposible modificar según sus propias necesidades lo que implica que las casas inteligentes no sean tan inteligentes como las personas lo imaginan (Chentao Li, 2019).

Los hogares en la actualidad serán cada vez más inteligentes y automatizados debido a las ventajas que presenta la automatización moderna. El avance de la tecnología proporciona ahora la opción de la seguridad y la protección de las personas junto con sus pertenencias y patrimonios. Una de las razones para desarrollar edificaciones inteligentes es prevenir el creciente riesgo de robos, hurtos y accidentes pues, debido al ajetreado estilo de vida la gente quiere controlar el estado de sus hogares a distancia (Pranav Kumar Madupu, 2018). En Latino América las edificaciones ya están adoptando poco a poco estas tecnologías, se está migrando a tener edificios controlados y supervisados, de forma que se garantiza su funcionamiento a niveles máximos de eficiencia y ahorro.

El sistema que se pretende diseñar permitirá la integración, monitoreo y el control de los diferentes dispositivos, lo que permitirá tener seguridad, comunicación y eficiencia energética en una edificación.

Gracias a la utilización de un hardware de bajo coste y software libre se podrá realizar diversas tareas y aplicaciones sin ninguna restricción, dependiendo lo que se requiera supervisar y controlar mientras que al diseñar el sistema de seguridad con un software propietario conllevaría a pagar una alta suma de dinero por las licencias y con limitadas tareas y funciones.

### **Justificación**

Los sistemas de automatización del hogar han evolucionado continuamente en los últimos años, algunos de los sistemas están centrando su atención en proporcionar acceso remoto, control del sistema y almacenamientos de datos, mientras que otros están más enfocados en la interconectividad de los dispositivos y su integración en sistemas más grandes (Stangaciu V., 2017)

Con el fin de precautelar la salud y bienestar de los habitantes del barrio Cañar Bajo el Ministerio de Educación Distrito Pujilí, ante la solicitud realizada por la directiva se llegó a un acuerdo para el préstamo de las instalaciones de la Unidad Educativa al Barrio para realizar asambleas de la junta de agua potable, actividades culturales, actividades sociales, capacitaciones del Barrio, ya que la institución se encuentra actualmente inhabilitada temporalmente por falta de estudiantes.

En la Acta de Préstamo de uso N°. 002-2021 especifica que los habitantes del barrio deben dar la seguridad necesaria para el cuidado de las instalaciones. Actualmente, una de las problemáticas que afecta a la Unidad Educativa Lago San Pablo del Cantón Pujilí, es la falta de seguridad en sus instalaciones, por lo que existen robos. Para contrarrestar esta problemática la directiva del barrio en la asamblea general realizado el día sábado 18 de abril del 2021, solicita la implementación de un sistema de seguridad a través de una plataforma Web que permita el control y monitoreo de las instalaciones

desde la comodidad de sus hogares y la generación de datos históricos para ser analizado por la directiva.

El desarrollo de proyectos de sistemas de seguridad aplicado en zonas de bajo desarrollo tecnológico brindará beneficios al permitir la innovación, mejora en la calidad de vida y optimización de recursos, además de contribuir con la seguridad y eficiencia energética.

### **Hipótesis de investigación**

La integración de diferentes tecnologías permitirá controlar, monitorear y generar macrodatos de sistema de seguridad a través de la web de las instalaciones de la Unidad Educativa Lago San Pablo Cantón Pujilí.

### **Antecedentes**

Existen diferentes tipos de sistemas que permite realizar el control y monitoreo de infraestructuras, no obstante, muchos de ellos son solo aplicaciones locales.

En 2016 en la Universidad de Malang Indonesia realizan un prototipo de control del hogar basado en la web, la alta movilidad de las personas para realizar sus actividades necesita un dispositivo que pueda controlar los aparatos eléctricos a distancia. Conectar una computadora a Internet podría generar un sistema de control a distancia para monitorear y controlar esos dispositivos (Sendari S., 2016).

En el 2017 se implementa una arquitectura novedosa de un sistema de medición y control especializado que se pretende integrar en un complejo sistema de automatización para edificios. El diseño propuesto se realiza alrededor del microcontrolador Atmega 328 integrado en la placa de desarrollo Arduino Uno. Para la monitorización ambiental interior se utilizan diferentes sensores asociados a diversas magnitudes físicas (Lita I., 2016).

En 2017 se desarrolla un sistema automático que permite el seguimiento y control de las condiciones ambientales en los espacios interiores a través de Internet. El sistema le permite recopilar

datos de humedad, temperatura y luminosidad que se almacenan en una base de datos estructurada. El acceso a los datos de monitoreo se logra a través de tres aplicaciones: un portal web, una aplicación de Android y una aplicación de escritorio (Marques, 2017)

### **Objetivo general del proyecto**

Desarrollar un sistema de seguridad a través de una plataforma web, que permita integrar equipos de diferentes tecnologías, para controlar, monitorear y generar macrodatos de sensores y actuadores implementados en la Unidad Educativa Lago San Pablo del Barrio Cañar Bajo, Cantón Pujilí.

### **Objetivos específicos del proyecto**

- Investigar en las diferentes bases de datos científicos sobre integración de equipos de diferentes tecnologías para el diseño e implementación de sistemas de seguridad basados en tecnología web.
- Desarrollar un sistema BMS que permita integrar equipos de diferentes tecnologías para monitorear, controlar y generar macrodatos de sistema de seguridad.
- Desarrollar una interfaz de usuario basado en plataforma web, con la finalidad de centralizar la información para el control y monitoreo del sistema de seguridad de la Unidad Educativa.
- Instalar la electrónica necesaria para el acondicionamiento de señales de sensores y actuadores implementados en el sistema de seguridad para la Unidad Educativa Lago San Pablo.
- Implementar un tablero de control que centralice la información de los diferentes sensores y actuadores implementados en el sistema de seguridad, con el propósito de generar una red de comunicación inalámbrica con una aplicación web.
- Implementar un tablero de transferencia automática, a través de contactores de potencia y control eléctrico para conexión a un generador de 110V en caso de corte de energía eléctrica.
- Realizar pruebas experimentales para evaluar el funcionamiento del sistema de seguridad implementado en la Unidad Educativa Lago San Pablo. Para la evaluación se considerará a los

habitantes del barrio Cañar Bajo para comprobar el correcto funcionamiento del sistema de seguridad implementado.

## Capítulo II

### Marco teórico

#### Introducción

Actualmente, la inseguridad se da por varios factores como son el desempleo, la inflación, la corrupción entre otros, generando así la delincuencia en diferentes modalidades, como homicidios, robo a instituciones públicas y privadas, robos a personas, vandalismo, entre otros. Estos delitos en la actualidad son controlados por varios dispositivos como elementos de prevención o disuasión; dentro de estos se hallan los sistemas de video-vigilancia, domótica e inmótica.

La seguridad pública es muy importante en el país, pero a nivel nacional la ciudadanía no ha encontrado una respuesta positiva debido a bajos ingresos económicos por la cual presenta bajos índices de nivel de seguridad y bienestar ciudadana, provocando pérdidas económicas y humanas. Es así como, en la provincia de Cotopaxi entre enero y julio del 2021 la provincia registró los siguientes números de denuncias; 32% de robo a personas, 22% robo a domicilios, 15% robo de accesorios y autopartes de vehículos, 10% robo de carros, 9% robo a motos, 9% robo a unidades económicas. De acuerdo con estos datos existe preocupaciones tanto para organizaciones como para los gobiernos, siendo un obstáculo social y económico.

Los lugares que no cuentan con un sistema de seguridad para evitar o minimizar actos delictivos son las unidades Educativas rurales, en donde las instituciones se ven afectados por varios acontecimientos que suceden día a día en el lugar, específicamente generando pérdidas económicas y al no contar con un sistema de seguridad no se tiene evidencias de anomalías presentadas generalmente en la noche para poder dar aviso a la directiva del Barrio o al personal policial encargado del sector.

Nikolav Valova y Irena Malova en 2020 en su publicación “Home Automation System with Raspberry Pi” habla de automatización del hogar mediante la aplicación de dispositivo raspberry PI, con el objetivo de acceder a distancia y controlar los principales electrodomésticos: lámparas, calefacción,

refrigeración, cierre y control de acceso, etc., en cualquier lugar donde el propietario tenga acceso a Internet. Este proyecto fue un prototipo que se probó con luces Led de diferentes colores que indican los eventos de salida. Para realizar este trabajo utiliza la Raspberry Pi como unidad de procesamiento, la cual recopila información de sensores, gestiona diferentes dispositivos, poner en marcha aplicaciones web y gestiona bases de datos (Nicolay Valov, 2020).

## **Marco conceptual**

### **Domótica e Inmótica**

La domótica y la inmótica hacen referencia a sistemas que permiten automatizar una vivienda o edificio, aportando servicios de gestión energética, seguridad, bienestar y comunicación, puede ser cableadas o inalámbricas, y cuyo control se puede realizar desde dentro y fuera del hogar (Domóticas Integradas, SL, 2021).

**Domótica.** - Es el conjunto de técnicas o sistemas para automatizar las diferentes instalaciones de una vivienda. Esta tecnología se puede aplicar a cualquier tipo de vivienda, ya sea casa, apartamento, etc. para dotarlas de un sistema inteligente de automatización, control y gestión. La domótica es una de las opciones más favorables y económicas que existen hoy en día para disfrutar de una vivienda segura, cómoda y favorable con el cuidado del medio ambiente y con el mayor ahorro energético y calidad de confort. (Domóticas Integradas, SL, 2021)

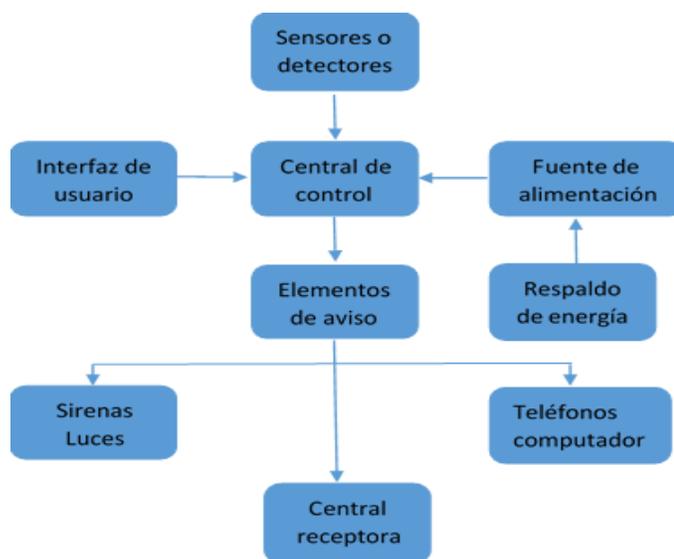
**Inmótica.** - La inmótica se define como la automatización integral de inmuebles con alta tecnología, es decir, edificios de grandes dimensiones tales como edificios de oficinas, hoteles, centros comerciales, plantas industriales, bancos, universidades, parques tecnológicos, etc., gracias a este sistema se puede gestionar servicios comunes como pueden ser la climatización, la calefacción, los ascensores, la iluminación, el sistema de riego, con el objetivo de reducir el consumo de energía, aumentar el confort y seguridad de los mismos. (Domóticas Integradas, SL, 2021)

### ***Sistema de seguridad***

Los sistemas de seguridad permiten proteger a personas, instalaciones o bienes materiales dentro de un hogar o empresa, mediante la aplicación de elementos tecnológicos interconectados entre sí con capacidad de gestionar diferentes eventos como sirenas, cámaras de video, luces, envío de mensajes a bomberos, policía, etc. En la figura 1, se presenta el esquema general de un sistema de seguridad electrónica.

**Figura 1**

*Sistema de seguridad*



*Nota.* Tomado de (Navarro, 2022)

### ***Sistemas de control BMS***

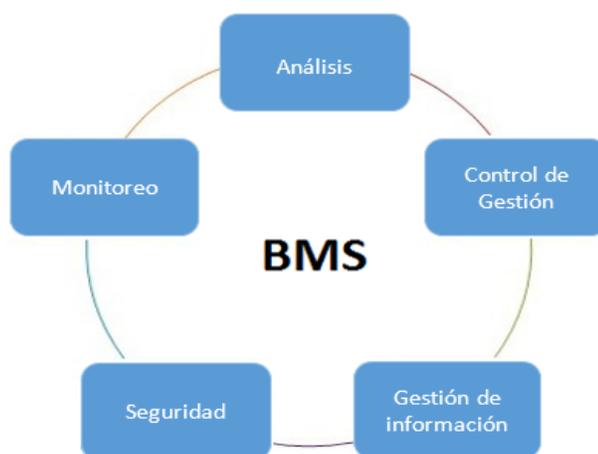
BMS (Building Management System), es un sistema de gestión de edificaciones que supervisa y controla servicios tales como iluminación, control de acceso, sistemas de seguridad, calefacción, sistema contra incendio entre otros (GSL, 2022).

Actualmente los sistemas de seguridad y de confort, han evolucionado grandemente debido al avance tecnológico e integración de nuevos conceptos como domótica, IoT (Internet de las cosas) e IIoT (Internet Industrial de las cosas) (GSL, 2022).

Los sistemas de seguridad electrónica fueron desarrollados por BMS utilizando tecnología industrial de sistemas basados en sensores y actuadores y aplicando IIoT para tener una red de seguridad capaz de proteger vidas humanas y patrimonios mediante el uso de sistemas de alarma, detectores de incendios, acceso a cámaras IP, entre otros, estos parámetros son gestionados por BMS para control, monitoreo y supervisión (CASTILLO, 2019).

**Figura 2**

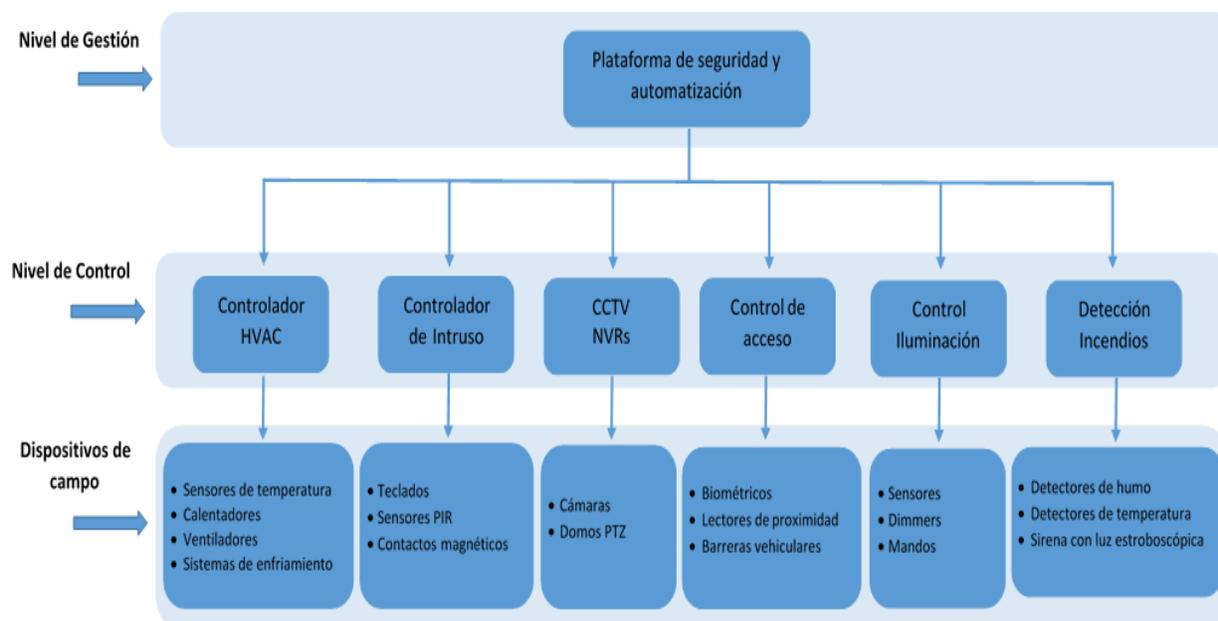
*Gestiones de sistemas BMS*



*Nota.* Tomado de (Omorogiuva, 2019)

### ***Arquitectura de sistemas de control BMS***

La siguiente estructura de sistema BMS muestra los niveles para control y monitoreo. Empezando de la parte inferior hacia la superior, el nivel inferior corresponde a instrumentación industrial (sensores) digital y analógica, seguido de nivel medio para procesos de control como son los PLCs, HMIs, switches, drivers, etc., finalmente la etapa de gestión conformada por sistemas de monitoreo, apps, servidores.

**Figura 3****Arquitectura de sistemas BMS**

Nota. Tomado de (CASTILLO, 2019)

**Protocolo de comunicación**

Se define como mecanismo de transmisión de datos o información entre los diferentes dispositivos que componen una red industrial, dentro de la pirámide de automatización, siendo necesario para llevar a cabo tareas de control y gestión de un proceso, es decir es el lenguaje usado para la comunicación entre dispositivos entre los cuales tenemos:

**BACnet**, Protocolo de comunicación diseñado para la comunicación de diferentes aparatos electrónicos presentes en los edificios actuales.

**Modbus**, Protocolo de comunicaciones, basado en la arquitectura maestro/esclavo RTU o cliente/servidor TCP/IP, diseñado para la conexión de dispositivos electrónicos industriales.

**Lonwork**, Conjunto de dispositivos inteligentes, que se conectan mediante uno o más medios físicos y que se comunican utilizando un protocolo común, Cada nodo es autónomo y proactivo, de forma que puede ser programado para enviar mensajes.

**DALI**, Es un estándar bidireccional en el que la información corre desde un controlador hacia los equipos de iluminación que ejecutan las órdenes y solicitudes de información.

**KNX**, Protocolo de comunicaciones de red, basado en OSI, para edificios inteligentes. Es el único estándar abierto para el Control de Casas y Edificios.

**EnOcean**, Protocolo domótico basado en tecnologías Wireless, usada principalmente en sistemas automáticos en edificios.

### ***Plataforma de gestión***

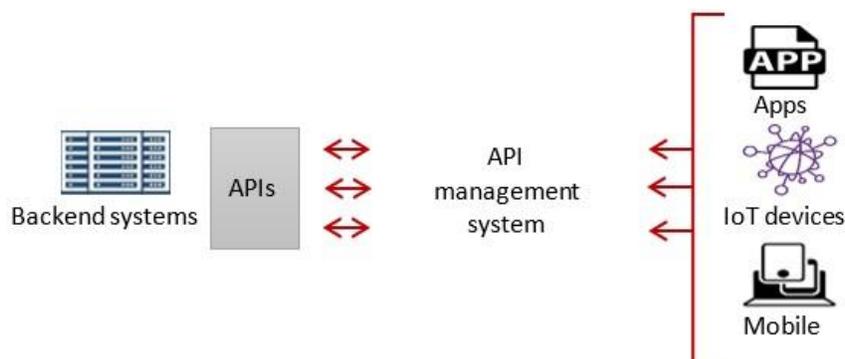
La plataforma de gestión sistema capaz de prevenir eventos críticos y reaccionar de manera eficiente ante situaciones inesperadas por parte de los sistemas de seguridad mediante su automatización (CASTILLO, 2019).

La plataforma de seguridad y automatización debe ser:

- Flexible
- Escalable
- Robusta
- Redundante
- Multitarea, orientada a objetos.

### ***Sistema de gestión API***

Application programming interface API define como interactuar con los diferentes componentes del software para lograr que elementos con distintos lenguajes de programación puedan interactuar entre ellos, de manera que las capacidades existentes sean intercambiables, para obtener un sistema más robusto y seguro estimulando así, la innovación del sistema.

**Figura 4***Estructura de funcionamiento API*

*Nota.* Tomado de (Portilla, 2021)

Los principales estándares de seguridad API son:

- **Autenticación:** Validación de quien tiene acceso a los distintos recursos expuesto.
- **Autorización:** La validación de que servicios puede consumir un cliente autenticado.
- **Validación de peticiones:** Cuando un servicio desee consumir una API se realiza siguiendo los estándares de datos definidos, esta validación de datos abarca desde las cabeceras de seguridad o información hasta el cuerpo de la petición en caso de ser necesario.

**Protocolo de comunicación API****SNMP.**

Protocolo Simple de Administración de Red (SNMP: Simple Network Management Protocol) opera en la Capa de Aplicación del Modelo de Referencia OSI, para administrar y controlar dispositivos de red. SNMP es uno de los protocolos del conjunto TCP/IP utilizado para gestionar y monitorizar elementos de red.

El Protocolo SNMP facilita la gestión de dispositivos que estén conectados en cualquier red accesible desde Internet, y no necesariamente aquellos localizados en la propia red local. Una red con aplicaciones SNMP posee al menos una estación de administración, cuya función es supervisar y

administrar dispositivos de red por medio de un software denominado NMS (Network Management System), el cual se encarga de la gestión de la red.

El dispositivo administrado es un nodo de red que implementa la interfaz SNMP, y ejecuta permanentemente un componente de software denominado agente, el cual intercambia datos con la estación de administración, y permite el acceso del NMS a la información específica del nodo.

### **MIB y OID.**

Colección de datos organizados jerárquicamente en forma de árbol, con una raíz anónima y varios niveles, asignados estos últimos por diferentes organizaciones. Los nodos ubicados en la parte superior del árbol definen estándares, mientras los identificadores de la parte inferior son colocados por las entidades asociadas.

Las Bases de Información de Administración (MIB) contienen Identificadores de Objetos (OID) organizados de forma jerárquica, donde cada nodo está identificado indistintamente con un número y una cadena de texto.

Para que se comuniquen con éxito la entidad administradora y un dispositivo administrado en la red, ambos necesitan conocer qué OID están disponibles. Esta es la razón por la cual existen las MIB, y por qué las necesitan los administradores del sistema.

Para supervisar las características de un dispositivo dado, se deben conocer las MIB de dicho dispositivo, por lo cual los administradores deben asegurarse de que todas las MIB necesarias estén almacenadas tanto en los dispositivos agentes SNMP como en el sistema de la entidad administradora.

### **Mensaje SMNP.**

Los agentes SNMP reciben y envían solicitudes UDP en el puerto 161 y 162. Un administrador SNMP puede enviar sus solicitudes desde cualquier puerto, aunque por lo general, también utiliza los mismos puertos que el agente.

Para realizar la administración, el protocolo SNMP es encapsulado dentro del protocolo UDP, para enviar los mensajes entre el dispositivo administrador y sus terminales monitoreados. La utilización de encapsulamiento UDP asegura que las tareas de administración no afecten el rendimiento de la red, ya que se evita la utilización de mecanismos de control y recuperación.

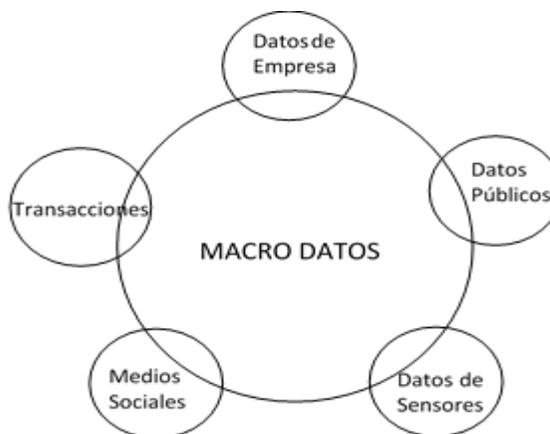
### **Macrodatos.**

Los macrodatos consisten en reunir, almacenar y procesar volúmenes muy grandes de datos e información, los macrodatos a menudo se definen como las tres V: volumen, velocidad variedad (Shane Hamilton, 2019).

La importancia estratégica de los macrodatos no radica en la cantidad sino en las aplicaciones potenciales que ofrecen para abordar cuestiones científicas, económicas y sociales, los datos pueden ayudar a gestionar ciudades o recursos naturales, a estudiar el cambio climático o a promover el desarrollo de regiones, pero también plantea la posibilidad de ciertos riesgos, entre ellos, el uso de enfoques puramente basados en datos que no tienen en cuenta la comprensión del fenómeno, que ignoran aspectos críticos de la recolección de datos, y procesan los datos de forma inadecuada y que confunden el ruido con la señal (Rossell, 2014).

### **Figura 5**

*Tipos de datos que maneja Big Data*



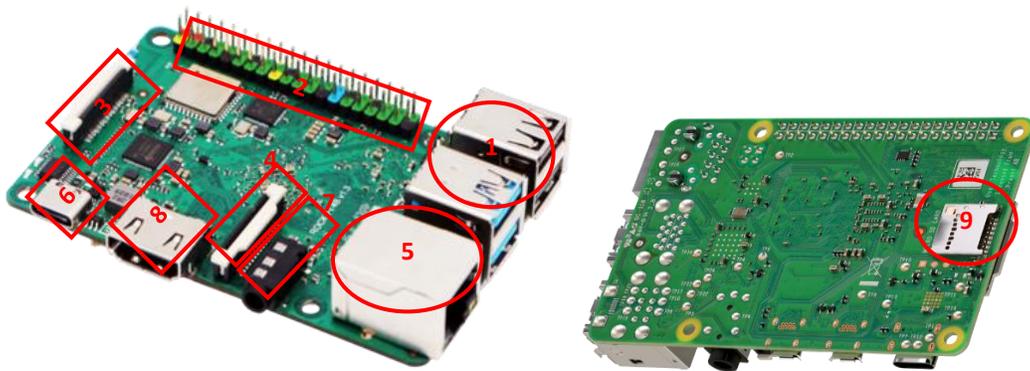
*Nota.* Tomado de (Rossell, 2014)

## **Raspberry Pi**

La tarjeta Raspberry-Pi es un ordenador de placa y consumo eléctrico reducida de bajo costo, el cual opera bajo sistema operativo Linux. Este equipo tiene una arquitectura de procesador ARM Broadcom BCM2835, posee diversas interfaces de hardware para la comunicación y conexión con otros dispositivos. El diseño no incluye un disco duro ni unidad de estado sólido, sin embargo, permite el almacenamiento permanente a través de una tarjeta SD. De igual forma la Raspberry nos proporciona diversos componentes de comunicación como lo son puertos UART/USB, Audio, salida de video HDMI y RCA, puerto Ethernet y un arreglo pines de entrada/salida de propósito general, mediante los cuales es posible activar estados lógicos o establecer comunicación con microcontroladores u otros dispositivos por protocolos como I2C, SPI, UART (Tzuc, 2015).

### **Figura 6**

*Placa Raspberry Pi*



*Nota.* Tomado de (Tzuc, 2015)

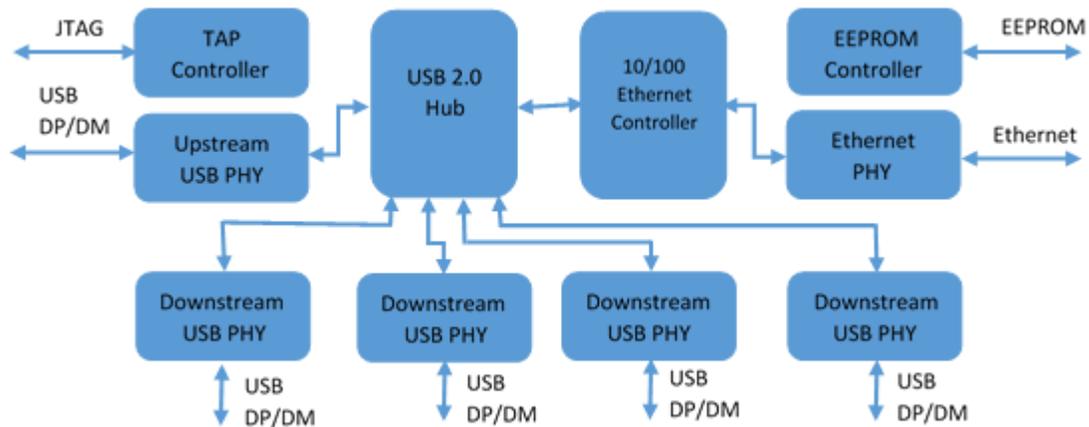
**Tabla 1***Partes de raspberry pi*

<b>Nº</b>	<b>IDENTIFICADOR</b>
1	Salida puertos USB 2.0
2	Pines GPIO
3	Conector DSI Display Module
4	Conector CSI Camera Module
5	Puerto Ethernet (RJ45)
6	Alimentación micro USB
7	Salida de audio
8	Salida HDMI
9	Ranura para la tarjeta microSD

*Nota.* Tomado de (Tzuc, 2015)

### **Puertos USB.**

Raspberry PI modelo B+ dispone de cuatro puertos USB del estándar 2.0 gestionados por microchip LAN9514. Este chip está diseñado para integrar puertos en pequeñas placas. La cantidad de puertos que contiene permite conectar a la vez mouse, teclado y conexión WIFI USB. (Domínguez, 2015).

**Figura 7***Diagrama de Puerto USB*

*Nota.* Tomado de (Domínguez, 2015)

### **Pines GPIO.**

GPIO (General Purpose Input/Output) es un sistema de Entrada/Salida de propósito general, dispone dos filas de 20 pines cada una, estos pines sirven de interfaz entre la Raspberry Pi y el exterior, con estos pines se pueden controlar multitud de dispositivos como luces leds, motores o relés, etc. Dispone de 12 pines de alimentación 5V, 3.3V y tierra, pines GPIO normales, y pines GPIO especiales para usar como puerto UART, I2C o SPI. (Domínguez, 2015).

**Figura 8***Distribución de pines GPIO raspberry PI*

3,3V DC POWER	1	2	5V DC POWER
GPIO 2(SDA)	3	4	5V DC POWER
GPIO3(SDL)	5	6	GROUND
GPIO4(GPCLK0)	7	8	GPIO 14(TXD)
GROUND	9	10	GPIO 15(RXD)
GPIO 17	11	12	GPIO 18(PWM0)
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3,3V DC POWER	17	18	GPIO 24
GPIO10(MOSI)	19	20	GROUND
GPIO10(MISO)	21	22	GPIO 25
GPIO11(SCLK)	23	24	GPIO08(CE0)
GROUND	25	26	GPIO07(CE1)
GPIO0(SDA0)	27	28	GPIO07(SCL0)
GPIO05	29	30	GROUND
GPIO06	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	GROUND
GPIO 19	35	36	GPIO 16
GPIO26	37	38	GPIO 20
GROUND	39	40	GPIO 21

*Nota.* Tomado de (Domínguez, 2015)

#### **Conector DSI.**

Display Serial Interface (DSI), permite la conexión de pequeñas pantallas LCD directamente a la GPU del dispositivo.

#### **Conector CSI.**

Camera Serial Interface (CSI) Conector tipo bus de 15 pines utilizado para conectar la cámara de la Raspberry Pi.

**Puerto Ethernet.**

Posee un conector RJ-45 conectado al integrado, que proporciona conectividad de 10/100 Mbps, mediante el puerto ethernet es posible conectar la raspberry directamente a un PC sin pasar por un router conectando ambos equipos de manera directa con un cable RJ45, también es posible añadir soporte Wi-Fi a la Raspberry utilizando un adaptador USB para red inalámbrica.

**Alimentación.**

Dispone de alimentación de 5V DC a través de un conector micro USB tipo B que alimenta la placa con un consumo de corriente de 600mA, no dispone de botones de encendido y apagado.

**Salidas de audio.**

Posee un conector de audio Jack de 3,5mm, Cuando se usa el puerto HDMI de la Raspberry Pi se debe configurar el puerto para que este transporte señales de audio y video mediante la conexión de un único cable a la pantalla. Si el Display no tiene entrada HDMI se tendría que utilizar la salida de audio Jack.

**Conector HDMI.**

Permite la conexión de un dispositivo compatible con la interfaz HDMI 1.3 y 1.4 para la extracción de vídeo y audio.

**Almacenamiento.**

Raspberry Pi no dispone de un disco duro, por lo cual la placa trae una ranura para insertar memoria microSD en donde se albergará el sistema operativo y todos los archivos. El arranque del sistema se hará desde la tarjeta microSD.

**Software.**

Raspberry Pi cuenta con sistema operativo GNU/Linux de código abierto. Las versiones de Linux que soportan la Raspberry Pi son:

- **Raspbian OS.-** Sistema operativo oficial de Raspberry Pi se basa en Debian Wheezy (Debian 7.0), con un entorno de escritorio LXDE y Midori como navegador web predeterminado. Además, incluye herramientas de desarrollo muy interesantes, como IDLE para Python, Scratch para programar videojuegos.
- **RISC OS.-** Sistema operativo británico desarrollado por Acorn Computers que distribuye bajo licencia Open-Source creado en torno a la plataforma ARM desde cero.
- **Arch Linux** Se caracteriza por su simplicidad, elegancia, coherencia del código, recomendable solo para gente con conocimientos más elevados. Arch Linux soporta ARM.
- **Pidora** Distribución de Linux Fedora, optimizada para funcionar en ARM.
- **OpenELEC** Se trata de distribución Linux especialmente pensada para crear un centro multimedia barato, incluye paquetes de drivers y codecs de audio y vídeo, se basa en el famoso Kodi anteriormente conocido como XBMC (Xbox Media Center), con el que se puede disponer de todo el contenido multimedia y acceso a Internet.

### ***Tipos de Raspberry PI***

**Tabla 2**

#### *Tipos de raspberry PI y sus características*

MODELO	CPU	CONECTIVIDAD INALAMBRICA	GPU	MEMORIA	PUERTOS	ALIMENTACION	ETHERNET	WIFI	CAMARA	CORE
<b>Raspberry PI 4B</b>	1.5-GHz, 4-core Broadcom BCM2711 (Cortex-A72)	802.11ac / Bluetooth 5.0	Video Core VI	1/2/4GB LPDDR4 RAM	1/2/4GB LPDDR4 RAM	5V/3A vía USB-C, 5V vía cabezal GPIO	Si	Si	Si	Cortex-A72.
<b>Raspberry PI zero W</b>	1-GHz, 1-core Broadcom BCM2835 (ARM1176JZF-S)	802.11n / Bluetooth 4.1	250MHz video Core	512MB	1x micro USB, 1x mini HDMI	Conector 40-pin GPIO.	No	Si	Si	Cortex-A53
MODELO	CPU	CONECTIVIDAD INALAMBRICA	GPU	MEMORIA	PUERTOS	ALIMENTACION	ETHERNET	WIFI	CAMARA	CORE

<b>Raspberry Pi 3 B+</b>	1.4-GHz, 4-core Broadcom BCM2837 B0 (Cortex-A53)	802.11ac, Bluetooth 4.2, Ethernet	400mM Hz	1GB	4 x USB 2.0, HDMI, 3.5mm audio	micro USB de 5,1 V	Si	No	15PIN	Cortex-A53
<b>Raspberry Pi 1B+</b>	700 MHz	10/100 Ethernet	Video Core IV	Micro SD	BCM2835	micro USB de 5,1 V	Si	No	SI	ARM1176 JZF-S
<b>Raspberry Pi 2B</b>	900 MHz	Puertos E/S.	250MHz video core	1 GB	USB2,0	USB a microUSB	Si	No	Si	Cortex-A7
<b>Raspberry Pi 3B</b>	1.2 GHz	2.4GHz 802.11n	400MHz	1 GB	USB 2.0	micro USB de 5,1 V	Si	Si	Si	Cortex-A53 64-bit
<b>Raspberry Pi zero</b>	1 GHz	2,4 GHz	Video Core IV	512 MB	1x micro USB, 1x mini HDMI	micro USB (5V)	No	No	Si	ARM1176 JZF-S
<b>Raspberry Pi moodB</b>	ARM11-6 700mHZ	MAX485	BROADCOM VIDCOR E IV	HDMI AUDIO 3,5MM	Mongo DB	26 pines GPIO	Si	Eth 10/100	CSI	.2GHz

*Nota.* Tomado de (Domínguez, 2015)

### **Lenguaje de programación JavaScript**

JavaScript es un lenguaje de programación que se utiliza para crear páginas web dinámicas capaces de interactuar con el usuario. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos, es decir, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Pérez, 2019).

### **Posibilidades y limitaciones.**

JavaScript siempre fue utilizado de forma masiva por la mayoría de los sitios de Internet, entre las ventajas principales tenemos;

- Es un lenguaje sencillo de utilizar.

- Es rápido, ejecuta las funciones de forma inmediata.
- Tiene varias opciones de efectos visuales.
- Compatible con los dispositivos más modernos, incluyendo iPhone, móviles y PS3.
- Versátil, útil para desarrollar páginas dinámicas y aplicaciones web.
- Es multiplataforma, puede ser ejecutado de manera híbrida en cualquier sistema operativo móvil.

En cuanto a las desventajas, JavaScript fue diseñado de forma que se ejecutara en un entorno muy limitado, las principales limitaciones son;

- Sus códigos son visibles, pueden ser leídos por cualquier usuario.
- Tiende a introducir gran cantidad de fragmentos de código en los sitios web.
- Opciones 3D son limitadas.
- Los usuarios pueden desactivar JavaScript desde su navegador.
- Los scripts son limitados por razones de seguridad y no es posible realizar todo con JavaScript, por lo tanto, es necesario complementarlo con otros lenguajes evolucionados y más seguros.

### **Sintaxis.**

Sintaxis es el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación.

La sintaxis de JavaScript es similar a lenguajes de programación como Java y C. Las normas básicas que definen la sintaxis de JavaScript son las siguientes:

- No se tienen en cuenta los espacios en blanco y las nuevas líneas.
- Se distinguen las mayúsculas y minúsculas.
- No se define el tipo de las variables.
- No es necesario terminar cada sentencia con el carácter de punto y coma.
- Se pueden incluir comentarios.

### ***Frameworks de programación***

Un framework es una aplicación reutilizable, semi-completa que puede ser especializada para producir aplicaciones concretas y específicas. Describe los objetos que componen el sistema y cómo éstos interactúan, sus interfaces y el flujo de control entre ellos, y como las responsabilidades de los sistemas mapean en objetos. (Delía, 2009).

La finalidad de emplear un framework a la hora de desarrollar aplicaciones web es la de disminuir el tiempo de desarrollo, reutilizar código existente, y promover buenas prácticas de desarrollo como el uso de patrones o estándares. El uso de frameworks define una filosofía de trabajo, donde el objetivo de un proyecto puede ser sintetizado en uno solo y es en su alcance, donde el desarrollador se centrará, dejando los detalles de operación a las estructuras del framework. Con la ayuda del framework, es posible implementar una serie de código repetitivo de una manera eficiente y sin tener que partir desde cero. Es posible establecer diferentes parámetros para cada estructura que se desea implementar, pero el desarrollo parte ya desde una base pre-programada, para esto existen diferentes librerías, funciones y APIs que están disponibles a la hora de desarrollar aplicaciones (OJEDA, 2008).

### **Frameworks backend.**

Un framework de desarrollo backend es la parte o rama del desarrollo web encargada de que toda la lógica de una página funcione. Este término es utilizado para referirse al área lógica de toda página web. Nos referimos a la arquitectura interna del sitio que asegura que todos elementos desarrollen la función correcta (Nestrategia, 2022).

### ***Express.***

Express es el framework web más utilizado de Node, y es la librería que permite el funcionamiento de otros frameworks ya que proporciona utilidades base para el desarrollo web, podría decirse que actúa como un programa que acepta las peticiones web y las maneja (Navarrete, 2017)

Express puede agregar diferentes funcionalidades de acuerdo con la necesidad, o cambiar lo que no cumpla las expectativas del desarrollador, proporciona un marco mínimo y flexible para las aplicaciones web. Funciona sobre los módulos de nodo central sin ocultar ninguna de las funciones de Node.js. Además, proporciona funciones robustas y limpias para agregar a los módulos de nodos, por lo que el desarrollo de la aplicación Node.js usando express es mucho más fácil que usar los módulos de nodo nativos (Express official website). El uso de Express framework en la parte superior de Node.js ayuda a mantener la claridad del código. También hace que la integración del módulo sea fácil de manejar y proporciona una estructura de solución para las aplicaciones. Express se instala utilizando el administrador de paquetes npm que emite el comando "npm install express" (Express official website).

Un servidor Express se compone de tres bloques de construcción: enrutador, rutas y middleware. La funcionalidad principal de un servidor web depende de sus excelentes métodos de enrutamiento. En una comunicación de cliente servidor, un cliente solicita algunos recursos del servidor, el servidor localiza los recursos y responde enviando los recursos al cliente. Esta es la funcionalidad principal de un servidor web y requiere excelentes métodos de enrutamiento para atender la solicitud. Express hace que este tedioso trabajo sea realmente fácil al permitir a los desarrolladores crear rutas en una estructura simple. Una ruta en Express es una combinación de un verbo HTTP y una ruta. El verbo HTTP generalmente es uno de los cuatro métodos HTTP: GET, POST, PUT y DELETE, y la ruta es la ubicación del recurso (URI).

### ***Node.js.***

Node.js se encarga de ejecutar código de JavaScript en el lado de servidor. Node.js por sí solo es un servidor, pero la gestión de peticiones HTTP por el mismo es muy complicada, por ello hay que dotarlo de Express.js, que facilite enormemente la gestión de peticiones.

Node trabaja con un único hilo de ejecución, una de las preocupaciones más comunes de este tipo de ejecución es el bloqueo de hilo por una tarea que conlleve mucho tiempo, para solventar este

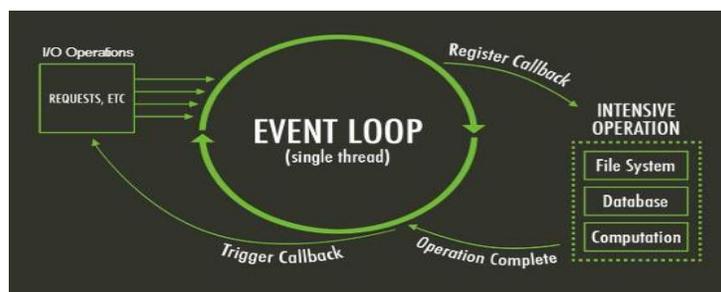
inconveniente lo que hace Node es gestionar todas sus tareas de forma asincrona, lo cual es un concepto revolucionario e innovador, ya que la gestion de las ejecuciones se realiza de manera mucho mas eficiente debido a que todo el codigo se ejecuta en paralelo, lo cual significa que en vez de bloquear el flujo de ejecución por la espera de la respuesta de una instrucción, Node sigue ejecutando otras instrucciones hasta que el proceso anterior acabe y devuelva una respuesta.

El procedimiento asincrono se realiza de la siguiente manera, el servidor web recibe una petición, entonces se registra su callback y se ejecuta el proceso pedido, mientras esta ejecución se lleva a cabo Node sigue ejecutando el resto de instrucciones en paralelo, y si se realiza otra petición se sigue el mismo procedimiento que para la petición anterior.

Cuando alguno de los procesos termina se avisa la bucle de evento, el cual es el encargado de enviar el callback a nuestra aplicación para que continúe con la ejecución anterior.

### **Figura 9**

#### *Funcionamiento Node.js*



*Nota.* Tomado de (Carrillo, 2016)

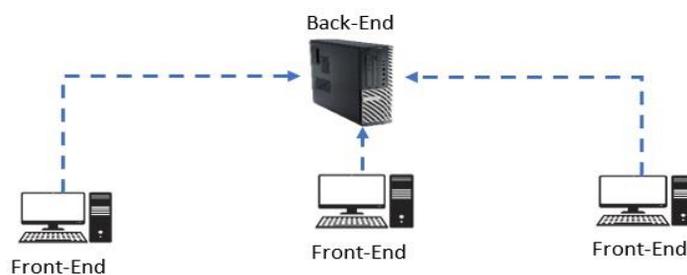
Node.js permite conectarse a una base de datos o utilizar la API de una aplicación. Suelen instalarse a través de un gestor de paquetes llamado Node Package Manager el cual es una colección pública de paquetes de código abierto para node.js, aplicaciones web front-end, aplicaciones móviles, robots, routers y otras necesidades de javascript

## Frameworks frontend.

Front-end es la parte que interactúa con los usuarios. Es decir, front-end es todos esos formularios que gestionan el ingreso por medio de los navegadores a los sitios y aplicaciones web, el objetivo es desarrollar la interfaz gráfica de usuario (GUI). De esa manera el back-end es la parte que procesa la entrada de datos realizado por los usuarios desde el front-end es decir, son los procesos que efectúan los respectivos sistemas para resolver las peticiones de los usuarios. De este modo en manera conjunta el front-end y el back-end interactúan en un sistema web o software para resolver las necesidades de los usuarios. La idea de esta abstracción es mantener separadas las diferentes partes de un sistema web o software con el fin de tener un mejor control. En pocas palabras el objetivo es que el front-end recolecte los datos y el back-end los procese (Melendez, 2014).

### Figura 10

#### Estructura Básica FRONT- END



*Nota.* Tomado de (Melendez, 2014)

### React Js.

React es una de las principales librerías de JavaScript de código abierto, para crear interfaces de usuario. Su principal ventaja son los componentes, que son pequeñas piezas de la interfaz de usuario que se pueden reutilizar en diferentes sitios de la aplicación. Sus características incluyen el ser declarativo, es decir, ayuda a crear interfaces de usuario interactivas de forma sencilla. Esto se debe a que se encarga de actualizar y renderizar (generar una imagen) de manera eficiente los componentes correctos cuando los datos de una aplicación cambian. Las vistas declarativas hacen que el código sea

más predecible, por lo tanto, fácil de depurar. También está basado en componentes; se pueden crear componentes encapsulados que manejen su propio estado y convertirlos en interfaces de usuario complejas. Debido a que la lógica de los componentes está escrita en JavaScript y no en plantillas, se pueden pasar datos de forma sencilla a través de una aplicación y mantener el estado fuera del DOM. Además, React permite desarrollar nuevas características sin necesidad de volver a escribir el código existente, y también puede renderizar desde el servidor usando Node, así como potencializar aplicaciones móviles usando React Native (React, 2021).

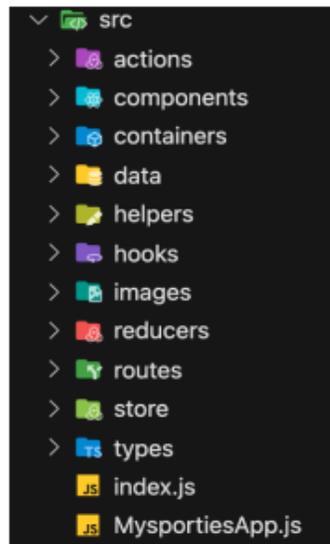
### ***Arquitectura de una aplicación React.js.***

Para desarrollar una aplicación utilizando React.js, es importante tener conocimiento en JavaScript, HTML y CSS, previo a desarrollar una aplicación web utilizando la librería de React.js. Puesto que, durante todo el desarrollo, se deben poseer ciertos conceptos para poder entender el funcionamiento a bajo nivel y luego, sobre esa base, comenzar a desarrollar. React.js es una librería y un framework de NodeJS. Las aplicaciones web en React.js se basan en el concepto de SPA (Aplicación de página única). Por esto, al generar un proyecto se crea la carpeta “src”. En ella, se encuentra todo el código JavaScript y estilos, mientras que la carpeta “public” contiene el único archivo con extensión .html de todo el proyecto. En dicha carpeta “src” se hará la división de carpetas dentro del proyecto.

Para crear una aplicación en React.js, existen varios paquetes de desarrollo recomendados que entrega al usuario varias soluciones. Éstos son repositorios vacíos para implementar, que contienen la división en carpetas, librerías y componentes que se utilizan frecuentemente a la hora de crear una aplicación en un determinado framework.

### ***Figura 11***

*División de carpetas del proyecto en React.js*



*Nota.* Tomado de (Tonarelli, 2021)

En la parte inferior de la carpeta, se encuentran los archivos base de la aplicación, los cuales serán ejecutados cuando se inicialice el proyecto. Luego, es importante destacar la carpeta “route”, que será donde se registre todas las navegaciones de la aplicación. La carpeta “store”, permitirá a los componentes de la aplicación enviar eventos y leer datos, que serán almacenados en el contexto de ésta. Por último, tenemos la carpeta “actions”, que refleja las acciones que deben tomar, dichas acciones serán llamadas a la hora de querer modificar el estado del contexto de la aplicación. La carpeta “routes” contiene las funciones que definirán como se manejarán las rutas dentro de la aplicación. La carpeta “components” busca contener las funciones que denotan diseño. El funcionamiento se los dará mediante la utilización de las acciones del contexto y las funciones que se encuentran dentro de la carpeta “containers”.

### ***Estructura y componentes.***

React.js permite crear interfaces de usuario interactivas de forma sencilla, encargándose de actualizar y renderizar de manera eficiente los componentes correctos cuando éstos cambian. (React, 2021) Los componentes son lógica escrita en JavaScript, los cuales manejan su propio estado, convirtiéndose en interfaces complejas. Además, permiten pasar datos de forma sencilla entre ellos,

manteniendo el estado fuera del DOM. Un componente está compuesto por código HTML y JavaScript, en su mayoría. Luego, en él se puede hacer referencia a una hoja de estilos CSS, pero también se le puede dar estilos dinámicamente a través de JavaScript. Además, React.js permite devolver componentes que, a su vez, pueden contener otros componentes dentro, los cuales se muestran según la necesidad y el comportamiento establecidos.

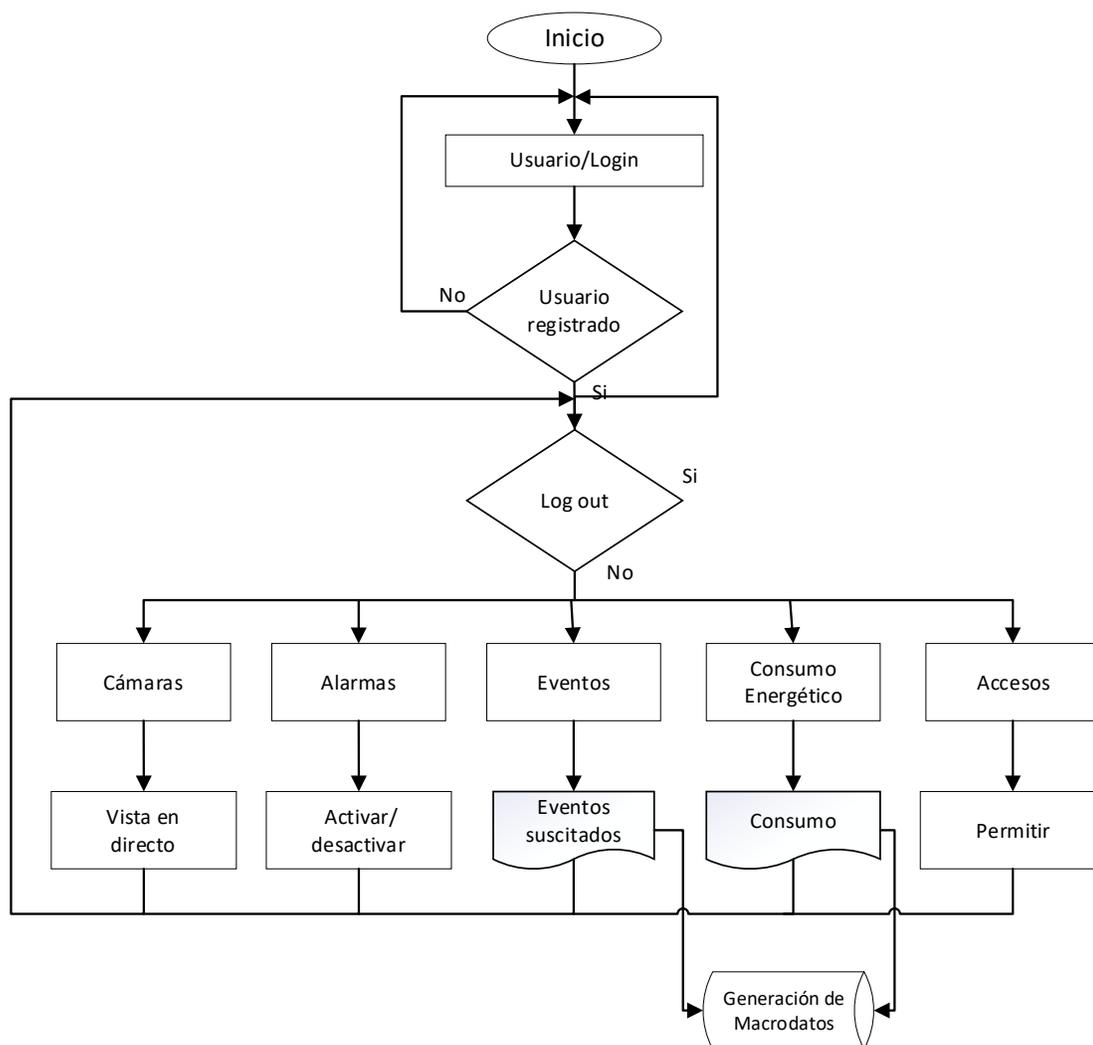
### **Capítulo III**

#### **Diseño de sistema de Seguridad**

Para entender de una mejor manera el diseño de sistema de seguridad en la figura 12 y figura 13, se observa el diagrama de flujo del funcionamiento del sistema.

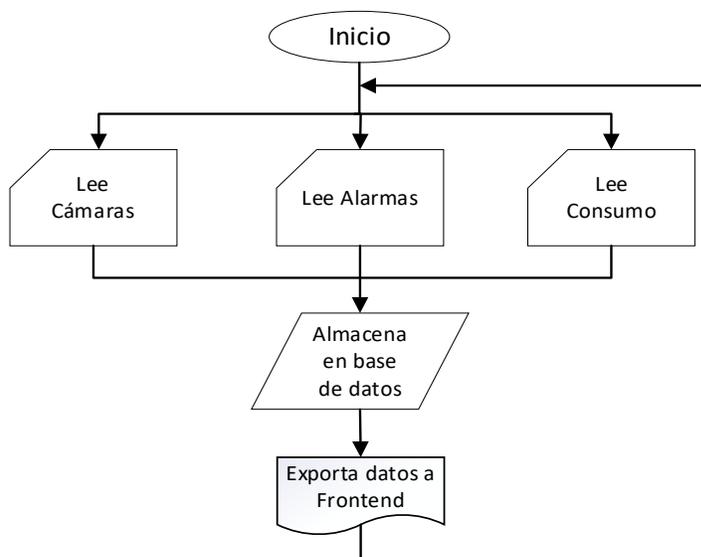
**Figura 12**

Diagrama de flujo frontEnd de sistema de seguridad



**Figura 13**

Diagrama de flujo backEnd de sistema de seguridad



### Selección de equipo

El sistema BMS planteado en este proyecto consta de dos partes fundamentales, por un lado, está el hardware que son los dispositivos de seguridad electrónica (cámaras, alarmas, controles de acceso) con dispositivos utilizados para la automatización residencial (controles de iluminación, medidores de energía). Y por otro lado está el software, que es el encargado de solicitar los datos a cada uno de los dispositivos, guardarlos en una base de datos y mostrarlos al usuario del sistema de tal forma que este los pueda interpretar.

Entre los dispositivos utilizados para la implementación de este sistema tenemos los detallados en la tabla 3.

**Tabla 3**

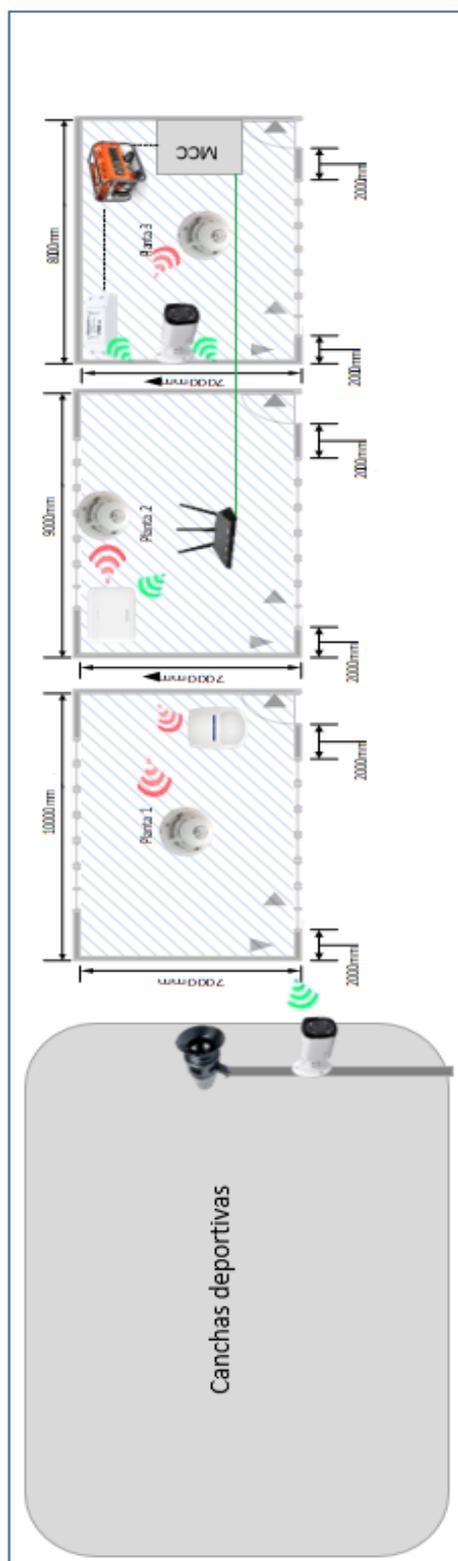
*Equipos para sistema de seguridad*

<b>MEDIO DE COMUNICACIÓN</b>				
<b>EQUIPO</b>	<b>WIFI</b>	<b>ETHERNET</b>	<b>RS485</b>	<b>RF</b>
<b>Sonoff Pow</b>	SI	NO	NO	SI
<b>Sonoff Basic</b>	SI	NO	NO	SI
<b>Central de Alarma</b>				
<b>Hikvision Axpro</b>				
	SI	SI	SI	SI
<b>Cámara Onvif</b>	SI	SI	NO	NO

Tras analizar los medios de comunicación de cada uno de los equipos, se observó que todos tienen la capacidad de comunicarse mediante WIFI. En función de esto, se procedió a diseñar el sistema utilizando una red wifi en la cual se concentren cada uno de los equipos para su posterior adquisición y almacenamiento de datos.

**Figura 14**

*Esquema general del sistema implementado*



La plataforma para adquirir los datos de los equipos antes listados se desarrolló utilizando el lenguaje de programación Javascript, puesto que es un lenguaje de programación open source y facilita la implementación y el consumo de API'S.

El sistema BMS implementado, al ser una plataforma web consta de dos partes fundamentales que son:

- El backend. - Este se ejecuta del lado del servidor y es el encargado de recoger los datos de cada uno de los dispositivos y guardarlos en una base de datos para luego enviarlos al frontend.
- El frontend. - Este se ejecuta del lado del cliente y es el encargado de solicitar los datos al backend mediante una API, para luego mostrarlos en pantalla mediante una interfaz gráfica.

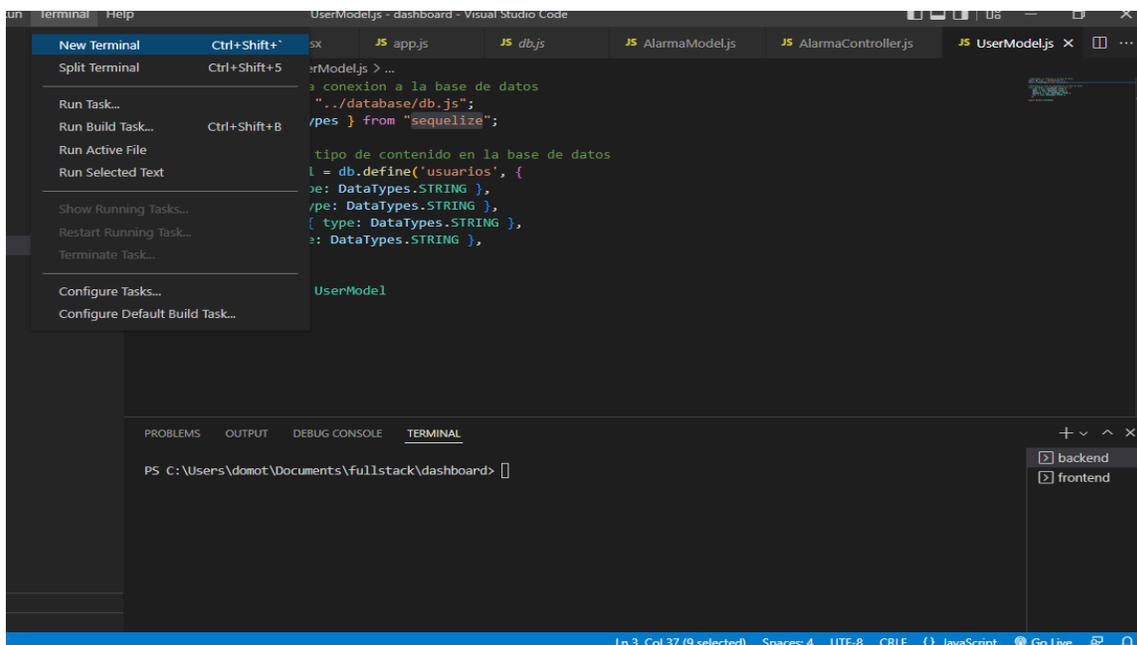
### Diseño del backend

El IDE de programación usado para el desarrollo de este proyecto fue Visual Studio Code, en el cual se instaló el framework express, siguiendo los pasos detallados a continuación.

1. En visual studio code abrimos una nueva terminal de comandos.

### Figura 15

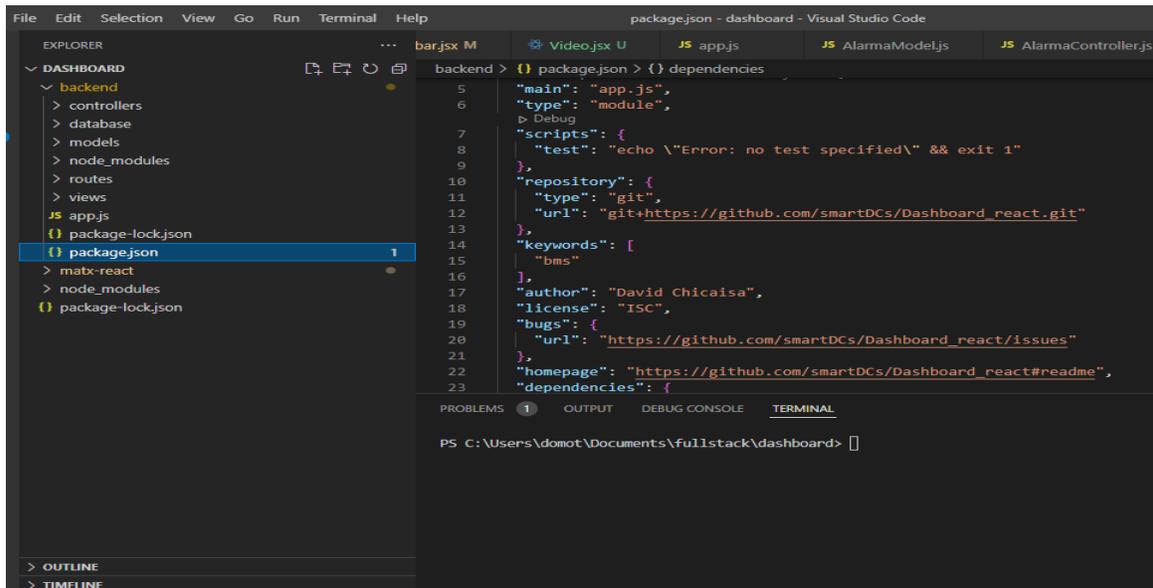
*Terminal de comandos integrada en visual studio code.*



2. Ingresamos el comando `npm init`, para crear el archivo `package.json` que contendrá la configuración de nuestro proyecto, junto con todas las dependencias necesarias.

**Figura 16**

Archivo de configuración `package.json`



The screenshot shows the Visual Studio Code interface. The Explorer view on the left displays a project structure with a 'backend' folder containing subfolders like 'controllers', 'database', 'models', 'node\_modules', 'routes', and 'views'. The 'package.json' file is selected and open in the Editor. The content of the file is as follows:

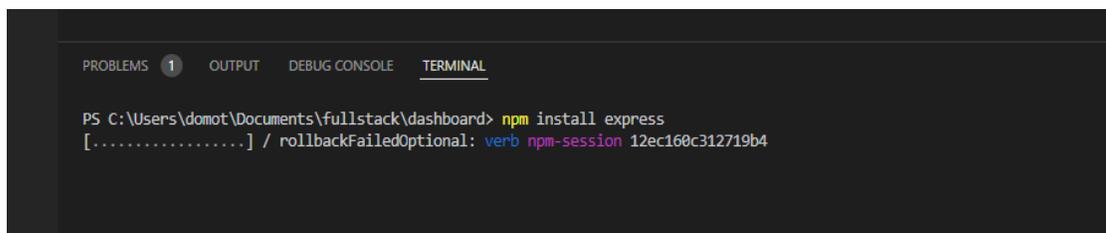
```
5  "main": "app.js",
6  "type": "module",
7  "scripts": {
8    "test": "echo \\\"Error: no test specified\\\" && exit 1"
9  },
10 "repository": {
11   "type": "git",
12   "url": "git+https://github.com/smartDCs/Dashboard_react.git"
13 },
14 "keywords": [
15   "bms"
16 ],
17 "author": "David Chicaiza",
18 "license": "ISC",
19 "bugs": {
20   "url": "https://github.com/smartDCs/Dashboard_react/issues"
21 },
22 "homepage": "https://github.com/smartDCs/Dashboard_react#readme",
23 "dependencies": {
```

The Terminal view at the bottom shows the command prompt: `PS C:\Users\domot\Documents\fullstack\dashboard>`

3. Una vez creado el archivo `package.json` ejecutamos el comando `npm install express`, en la cual nos abre una pantalla donde podemos empezar a desarrollar nuestro backend.

**Figura 17**

Instalación del framework `express`



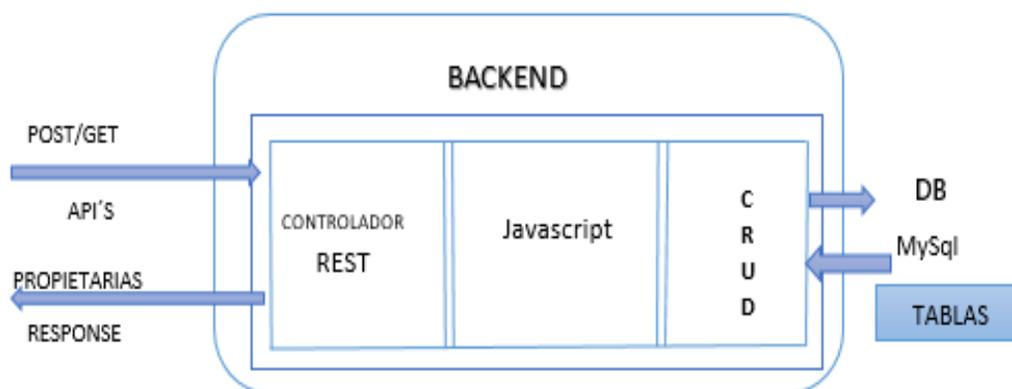
The screenshot shows the terminal window with the following output:

```
PS C:\Users\domot\Documents\fullstack\dashboard> npm install express
[.....] / rollbackFailedOptional: verb npm-session 12ec160c312719b4
```

En la Figura 18 se muestra un diagrama de bloques que corresponde al backend, visto de forma general.

**Figura 18**

Diagrama de bloques del backend



El software correspondiente al backend está desarrollado usando node js y el framework express, con estas dos herramientas se implementó un CRUD que gestiona la base de datos, esta base de datos incluye las tablas correspondientes a:

- Los usuarios del sistema,
- Parámetros de consumo de energía,
- Credenciales de acceso de las diferentes API'S,
- Históricos de los eventos suscitados.

La base de datos se implementó utilizando el motor de mySql.

### ***Usuarios implementados***

El número de usuarios implementados está conformado por los siguientes campos.

**Tabla 4**

*Usuarios implementados en sistema de seguridad.*

Ítem	Nombre	Tipo
1	Id	Llave primaria de tipo entero auto incremental, 11 dígitos máximo
2	Name	String máximo 200 caracteres. (corresponde al nombre de usuario)

Ítem	Nombre	Tipo
3	Email	String máximo 200 caracteres. (dirección de correo electrónico correspondiente al usuario a registrar)
4	Password	String máximo 200 caracteres. (contraseña del usuario a registrar)
5	Rol	String máximo 20 caracteres. (rol desempeñado por el usuario registrado)
6	createdAt	Date. (fecha en la que se crea el usuario)
7	updatedAt	Date. (fecha en la que se actualiza los datos del usuario).

**Figura 19**

*Estructura de la tabla de usuarios.*

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2 name	varchar(200)	utf8mb4_general_ci		Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	3 email	varchar(200)	utf8mb4_general_ci		Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	4 password	varchar(200)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	5 rol	varchar(20)	utf8mb4_general_ci		No	guest			Cambiar Eliminar Más
<input type="checkbox"/>	6 createdAt	date			Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	7 updatedAt	date			Sí	NULL			Cambiar Eliminar Más

En esta tabla se encuentra configurado un súper usuario, el que tendrá activado todos los privilegios, capaz de crear, modificar y eliminar usuarios.

### **Tabla consumo**

La tabla consumo almacena los parámetros de consumo de energía que se obtienen desde el módulo SONOFF POW, y está estructurado de la siguiente forma.

**Tabla 5**

*Tabla de consumo de sistema de seguridad.*

<b>Ítem</b>	<b>Nombre</b>	<b>Tipo</b>
<b>1</b>	Id	Llave primaria de tipo entero auto incremental, 11 dígitos máximo.
<b>2</b>	Device Id	String máximo 200 caracteres. (se refiere al nombre del dispositivo SONOFF).
<b>3</b>	Voltaje	Decimal 10 dígitos enteros y 2 decimales. (se refiere al voltaje medido entre línea y neutro).
<b>4</b>	Potencia	Decimal 10 dígitos enteros y 2 decimales. (corresponde a la potencia consumida en Kwh).

**Tabla de credenciales**

Esta tabla almacena las credenciales requeridas para el consumo de cada una de las API'S y está estructurada de la siguiente forma.

**Tabla 6**

*Tabla de credenciales de sistema de seguridad.*

<b>Ítem</b>	<b>Nombre</b>	<b>Tipo</b>
<b>1</b>	Id	Llave primaria de tipo entero auto incremental, 11 dígitos máximo.
<b>2</b>	Tecnología	String máximo 200 caracteres. (se refiere a la marca del dispositivo, es decir, hikvision, SONOFF u ONVIF).
<b>3</b>	Name	String máximo 200 caracteres. (corresponde al nombre de usuario que administra el dispositivo en la plataforma original).

Ítem	Nombre	Tipo
4	Password	String máximo 200 caracteres. (contraseña del usuario que administra el dispositivo en la plataforma original).

### **Tabla alarmas**

Esta tabla almacena todos los eventos de alarma producidos, y está estructurado de la siguiente forma.

#### **Tabla 7**

*Tabla de alarmas de sistema de seguridad.*

Ítem	Nombre	Tipo
1	Id	Llave primaria de tipo entero auto incremental, 11 dígitos máximo.
2	Tipo	String máximo 20 caracteres. (se refiere al tipo de alarma, que puede ser: de movimiento, incendio o fallo de suministro de energía).
3	Detalle	String máximo 200 caracteres. (en este campo se indica los detalles de la ubicación del sensor que produjo el evento).
4	Fecha	Datetime. (corresponde a la fecha y la hora en la que se produjo el evento).
5	Estado	Boleano. (muestra el estado de la alarma, si fue aceptada o no por el administrador del sistema).
6	createdAt	Date. (se refiere a la fecha en la que se produjo el evento)
7	updatedAt	Date. (se refiere a la fecha en la que se actualizó los detalles del evento).

**Figura 20**

*Estructura de la tabla de alarmas.*

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2 tipo	varchar(20)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3 detalle	varchar(200)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4 fecha	datetime			Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	5 estado	tinyint(1)			No	0			Cambiar Eliminar Más
<input type="checkbox"/>	6 createdAt	date			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	7 updatedAt	date			No	Ninguna			Cambiar Eliminar Más

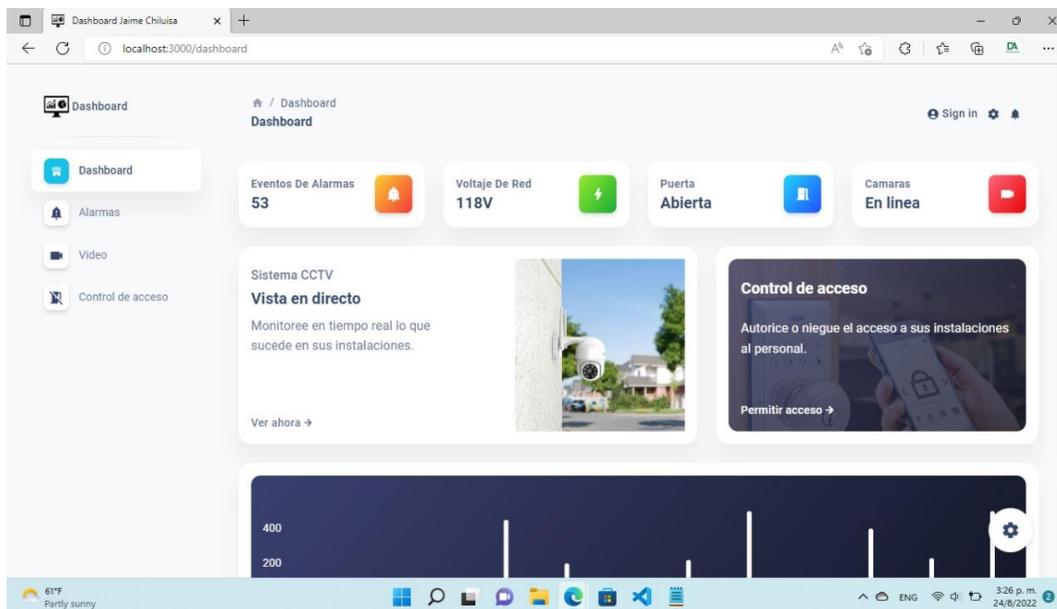
## Diseño del frontend

La implementación del frontend requiere procedimientos previos para el diseño, primero se realiza el proceso de diseño dashboard o panel de control, posteriormente se lleva a cabo el análisis de componentes, lógica y estilos.

Aplicando este procedimiento, se logra una definición de elementos que cuenten con una similitud en su forma y comportamiento, Entre los elementos se pueden encontrar Botones, barras, contenedores, visualizadores de información, entre otros.

En el caso de los botones permiten, habilitar accesos, encender dispositivos. En cuanto a las barras tiene un componente llamado Dashboard que se encuentra presente todo momento donde muestra el logotipo, las diferentes subpantallas que puede acceder. Por último, el perfil que permite accesos a las credenciales y cierre de sesión.

Los visualizadores de información permiten desplegar información del sistema de seguridad en nuestro caso los históricos del sistema de seguridad de la Unidad Educativa.

**Figura 21***Diseño de Frontend.***Consumo de API'S**

Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones.

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones; y ofrecen oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales). (Hat, 2017)

**API eWeLink**

eWeLink es la plataforma de aplicaciones que admite múltiples marcas de dispositivos inteligentes entre los cuales se incluye Sonoff.

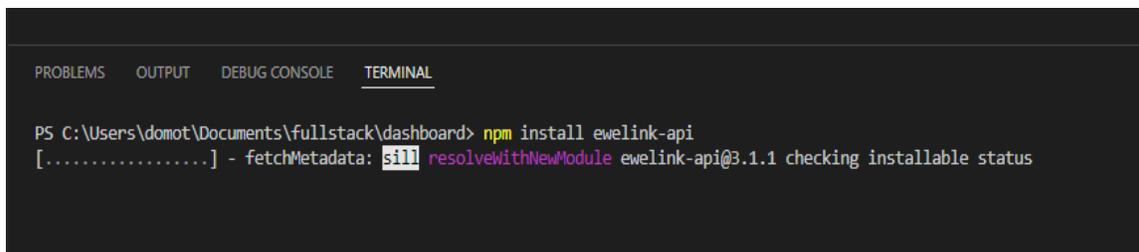
El API de eWeLink para Javascript es un módulo que permite interactuar directamente con los dispositivos inteligentes que se tenga configurado en el aplicativo eWeLink, mediante peticiones HTTP.

Para su instalación simplemente abrimos un terminal de comandos en nuestro proyecto y ejecutamos el siguiente código:

```
npm install ewelink-api
```

### **Figura 22**

#### *Instalación del API eWeLink*



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\domot\Documents\fullstack\dashboard> npm install ewelink-api
[.....] - fetchMetadata: sill resolveWithNewModule ewelink-api@3.1.1 checking installable status
```

Una vez terminada la instalación ya podemos hacer uso de la API y empezar a recolectar los datos de nuestros dispositivos inteligentes, para eso podemos ayudarnos de la documentación proporcionada por el desarrollador, en el siguiente enlace: [ewelink-api/docs at master · skydiver/ewelink-api · GitHub](https://github.com/skydiver/ewelink-api/docs).

### **Conformación de sistema de seguridad**

En la figura 23, se observa cómo está conformado la comunicación del sistema de seguridad con los respectivos componentes.

**Figura 23**

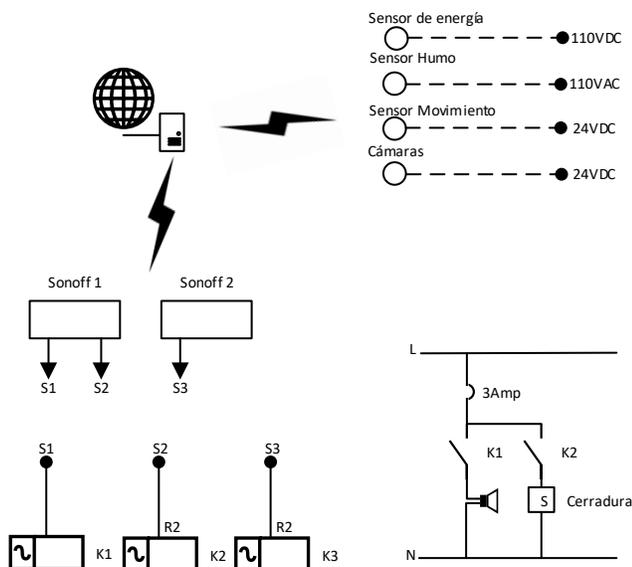
*Conformación de sistema de seguridad*



En el diagrama se observa que el sistema de control de seguridad diseñado se ejecuta en una raspberry, la cual se comunica a un Switch por el protocolo de comunicación UDP el mismo que está conectado a los diferentes dispositivos de control mediante señal wifi; desde el módulo sonoff pow R2 podemos controlar dispositivos auditivos y de seguridad, se utiliza módulo sonoff pow medidor de consumo eléctrico para monitorear señales eléctricas y en caso de ser necesario tener la opción de encender equipos de respaldo eléctrico, en la figura 24 se detalla diagrama de conexiones.

**Figura 24**

Diagrama de conexiones sistema de seguridad.



### Diseño del tablero de control para transferencia automática de energía

Para diseñar el tablero de transferencia automática se toma en cuenta la carga total que debe suministrar el equipo de respaldo.

**Tabla 8**

Tabla de consumo eléctrico de sistema de seguridad.

Equipos	Voltaje	Consumo(A)	Potencia(W)
Raspberry PI 4b	5V DC	3A	15W
Kit de alarma AXPRO de hikvision	12V AC	1.3A	15.6W
Cámaras Onvif	12V AC	500mA	6W
Sensor de humo fotoeléctrico	12V AC	130mA	1.56W
Chapa magnética 1200Lb	12V AC	500mA	6W
Reflector led 200W	127V AC	1.75 A	200W

<b>Equipos</b>	<b>Voltaje</b>	<b>Consumo(A)</b>	<b>Potencia(W)</b>
<b>Alarma comunitaria</b>	12V DC	7A	84W
<b>Reserva</b>	127V	5A	635W
<b>Total</b>		20.35A	968.16W

De acuerdo con los datos de consumo de energía del proyecto que se ilustra en la Tabla 8, el sistema de seguridad necesita una fuente de respaldo de energía con las siguientes características; Tomando en cuenta que los equipos traen su propio cable de alimentación según su necesidad, se va a implementar un equipo que suministre 127V AC con corriente superior o igual a 20.35A con una capacidad de potencia de 1Kw.

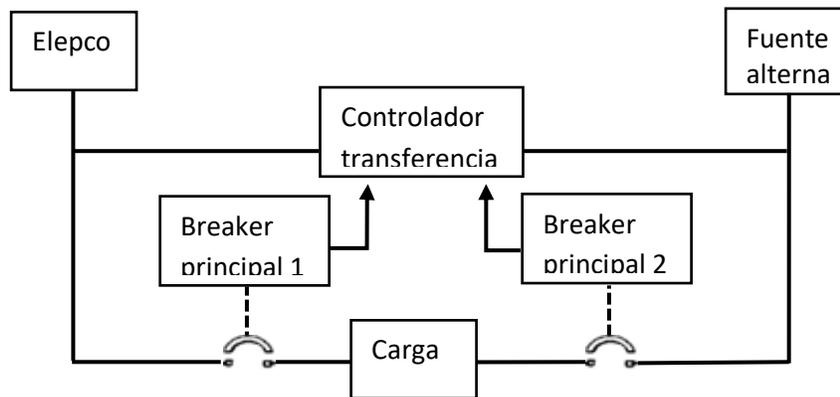
Se instalará una UPS en paralelo al consumo con la finalidad de evitar cortes de energía simultáneos al momento de realizar la transferencia, equipo que poseerá las siguientes características;

- Marca: UPS forza
- Voltaje: 120V AC
- Corriente: 5A
- Potencia: 1000VA/500W.
- Tiempo de respaldo: 32 Minutos

La energía que suministra la UPS, también se utilizará como fuente de alimentación para activar los dispositivos de control de tableros de transferencia automática, también servirá como apoyo para encendido de equipo alternativo de suministro de energía.

**Figura 25**

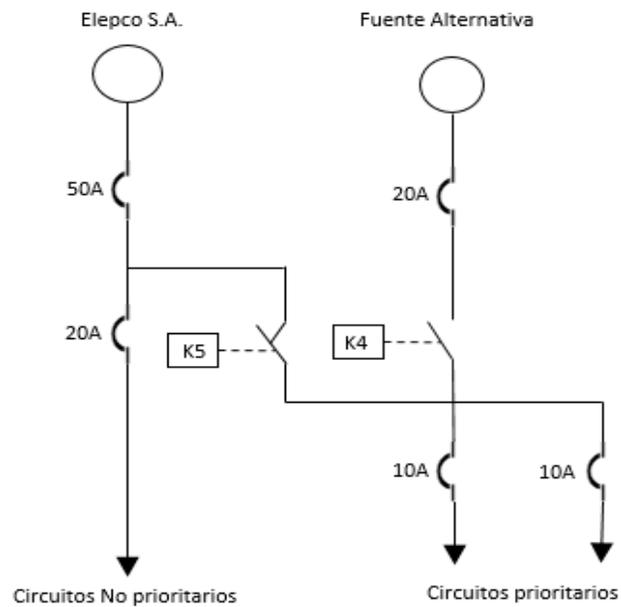
*Sistema de transferencia de energía.*



*Diagrama de fuerza y control de sistema de transferencia automática*

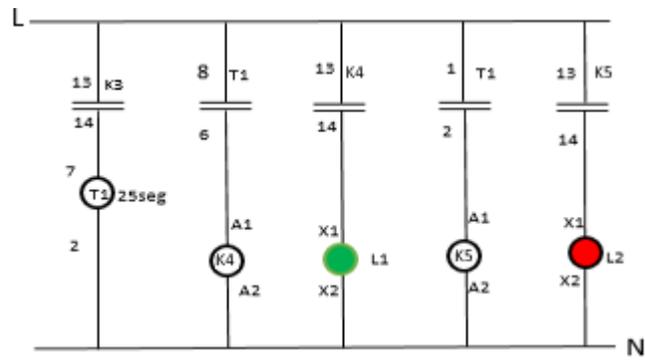
**Figura 26**

*Diagrama de potencia de sistema de transferencia.*



**Figura 27**

Diagrama de control de sistema de transferencia.



## Capítulo IV

### Pruebas y resultados

En este capítulo se realizó el análisis del comportamiento práctico del sistema de seguridad con sus diferentes componentes que lo integran y el análisis de los macrodatos que genera el proyecto.

Para las pruebas de funcionamiento se realizó la instalación de equipos en la unidad educativa los mismos que esta ubicados en puntos estratégico para su mejor funcionamiento. El sistema cuenta con los siguientes equipos:

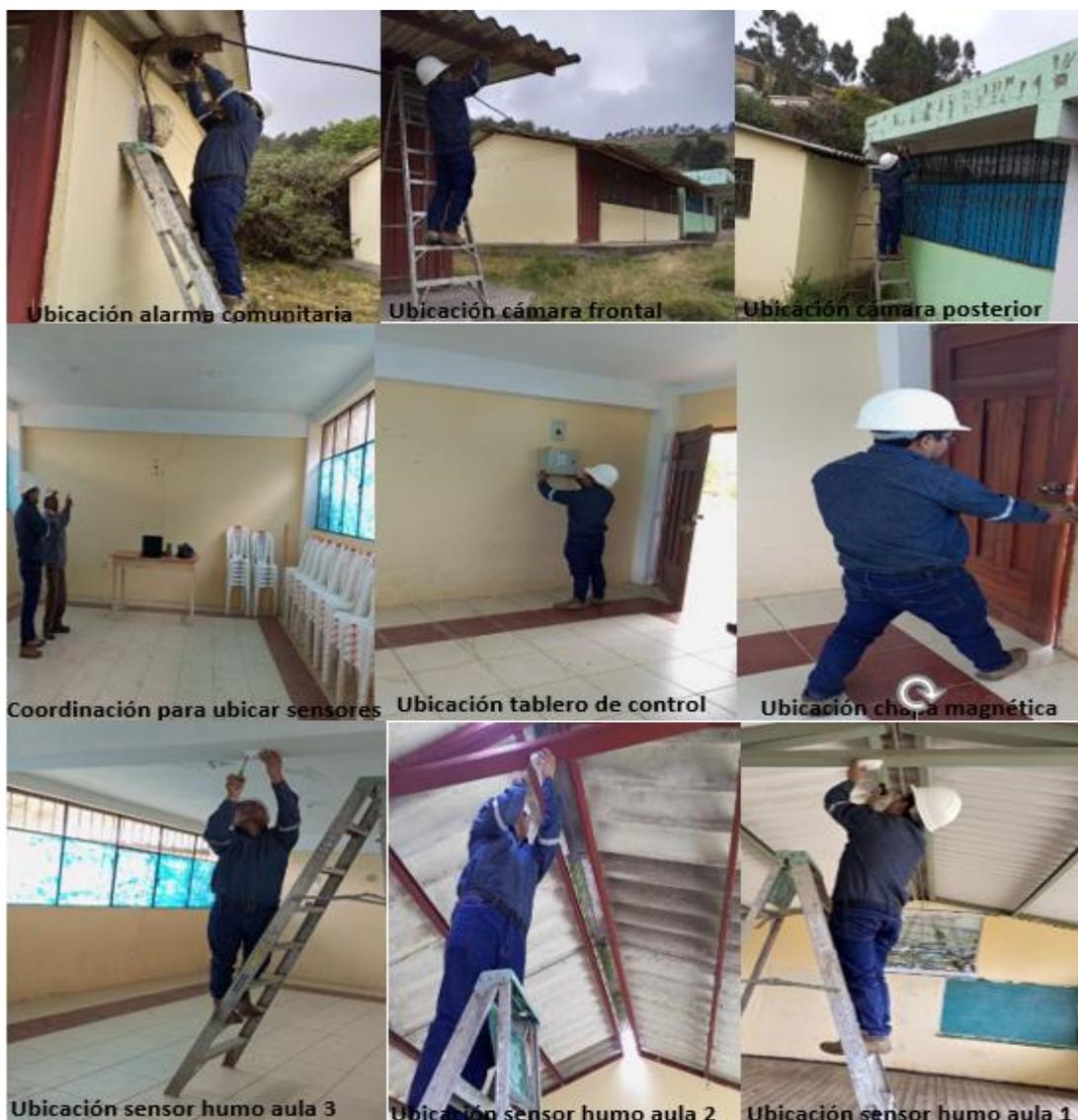
- Un Raspberry PI 4B, este módulo cuenta con un puerto de comunicación Ethernet RJ45 para la comunicación con la computadora y un puerto USB 2.0, dispone dos filas de 20 pines GPIO (General Purpose Input/Output), además cuenta con 12 pines de alimentación 5V, 3.3V y tierra, este equipo está ubicado en el aula 3 en el tablero principal del sistema.
- Un Kit de alarma AXPRO de hikvision, el cual cuenta con una central Ax pro inalámbrica, contacto magnético inalámbrico, detector de movimiento inalámbrico, control remoto para activar/desactivar. La central Ax-pro está ubicado en el aula 3 en el tablero principal del sistema, mientras que el contacto magnético, y detector de movimiento se ubica en zonas específicas de aula 1 y 2 respectivamente.
- Tres Sensor de humo fotoeléctrico, una para cada aula.
- Dos cámaras Onvif ubicadas de manera que una enfoca la parte frontal de las aulas mientras que la otra cámara enfoca la parte posterior.
- Chapa magnética, ubicada en la puerta de acceso al aula tres que se utiliza como bodega de la institución.
- Alarma comunitaria, la cual se encuentra en la parte frontal de la cancha deportiva de la institución.

En el Anexo se encuentra el datasheet de cada equipo.

En la Figura 28 se observa los equipos instalados en la Unidad Educativa.

**Figura 28**

*Diagrama de control de sistema de transferencia.*

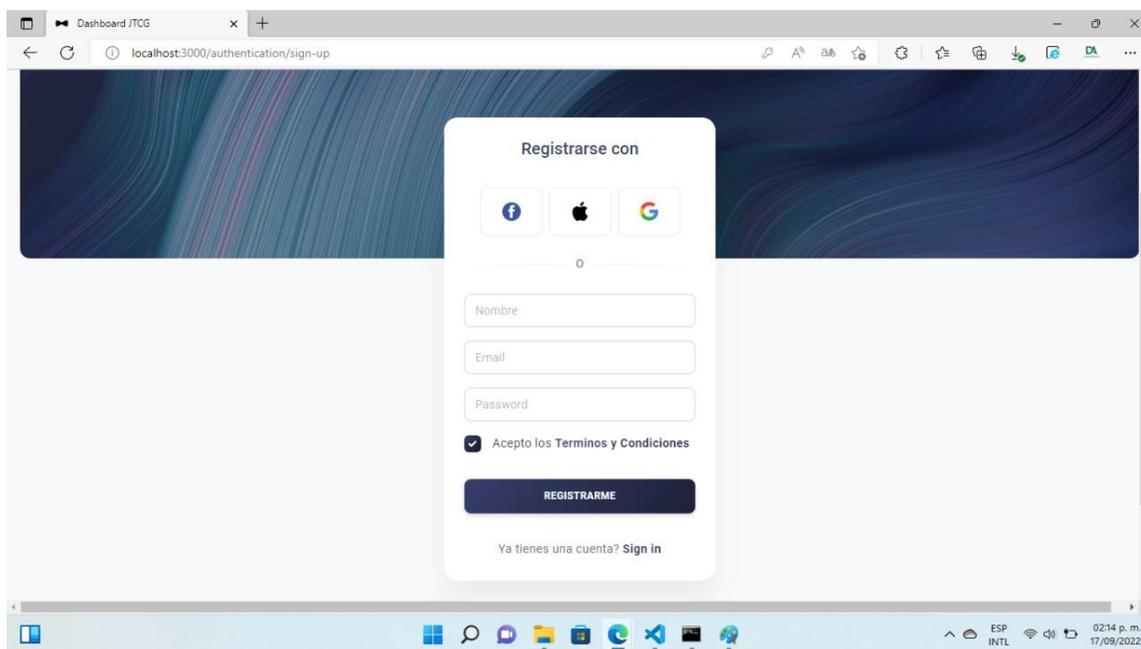


### **Creación de nuevo usuario desde el sistema de seguridad.**

El sistema de seguridad permite crear nuevos usuarios con distintas jerarquías, tanto para dirigentes del barrio, así como para familias pertenecientes a la comunidad, se realizó la prueba creando un nuevo usuario para administrador y operador como se puede observar en la figura 29, esta operación solo lo puede hacer el que tiene mayor jerarquía, en este caso el presidente o el secretario.

**Figura 29**

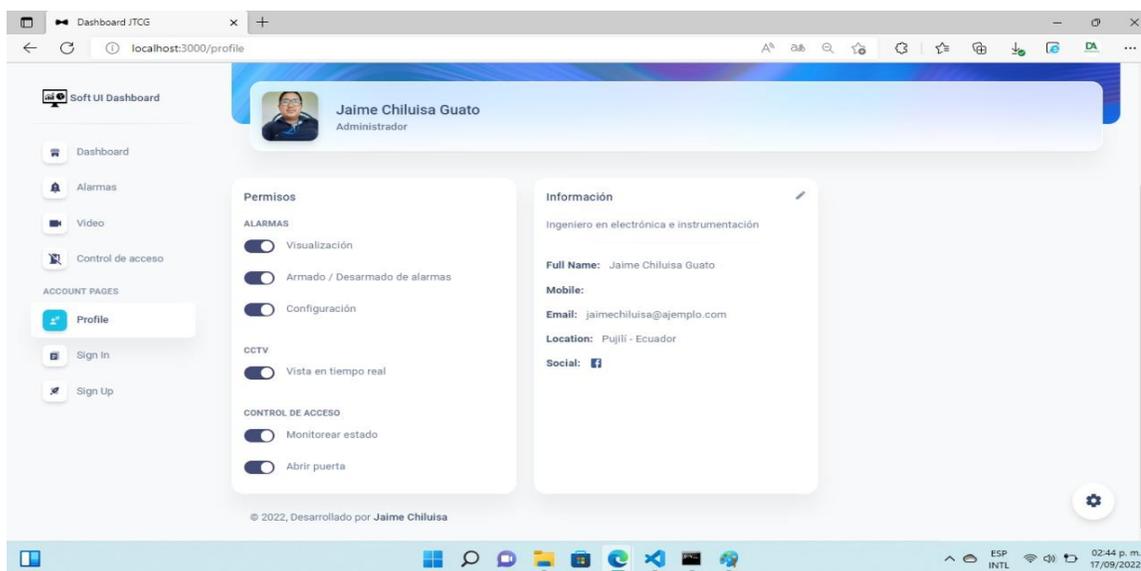
Registro en el sistema de seguridad.



Al registrar datos como administrador se tiene el control total del sistema, es decir se puede realizar cualquier operación de acuerdo con la necesidad.

**Figura 30**

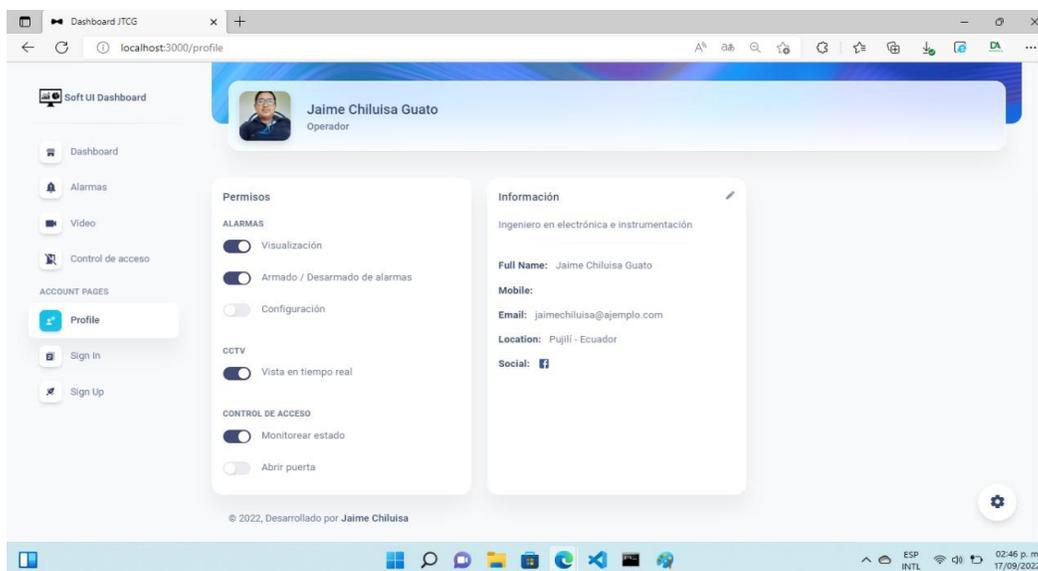
Plataforma principal, ingreso como administrador.



Por otro lado, si se realiza el registro como operador este tiene algunas restricciones, tales como: No se podrá realizar cualquier configuración, no se podrá habilitar o cerrar la puerta de bodega principal.

### **Figura 31**

*Plataforma principal, ingreso como administrador.*

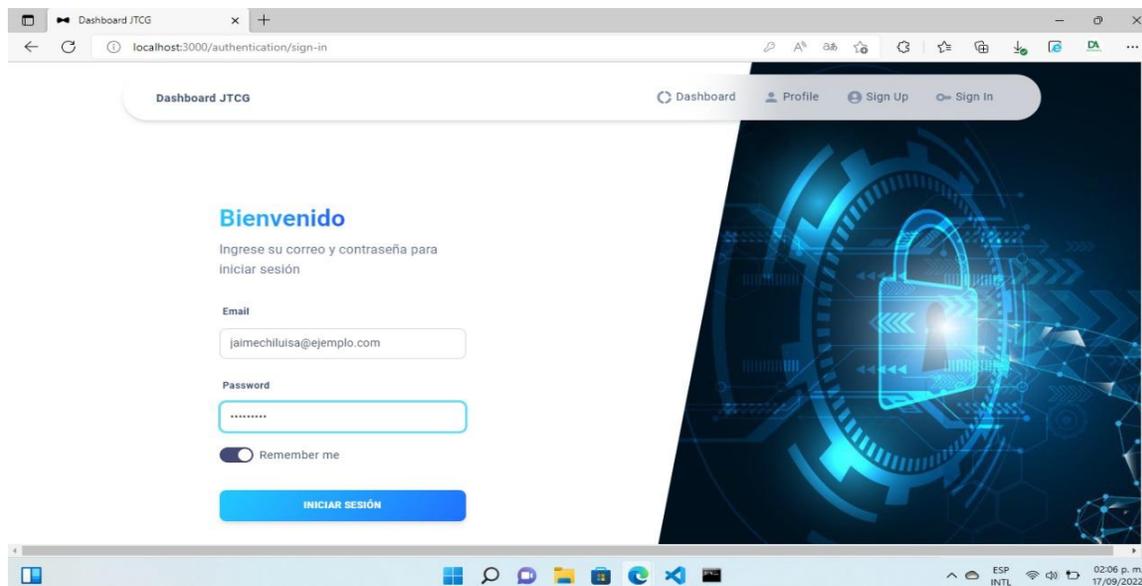


### **Ingreso a la página web de sistema de seguridad de la Unidad Educativa.**

En un navegador de cualquier dispositivo electrónico se ingresa el link <https://dashboardjt.onrender.com> el cual direcciona a la plataforma del sitio web del sistema de seguridad, en el entorno se tiene una pantalla principal que solicita usuario y contraseña, si los datos ingresados son correctos, se muestra automáticamente el tipo de usuario que es y se visualiza el botón para ingresar al sistema, como se puede observar en la figura 32.

**Figura 32**

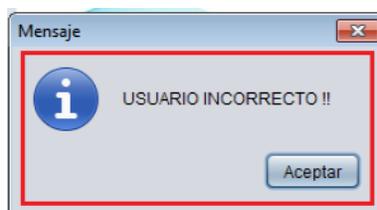
Plataforma principal de ingreso.



Al ingresar datos de usuario y contraseña el sistema de seguridad lee la base de datos, al no encontrar coincidencias con datos ingresados, este envía un mensaje de usuario incorrecto.

**Figura 33**

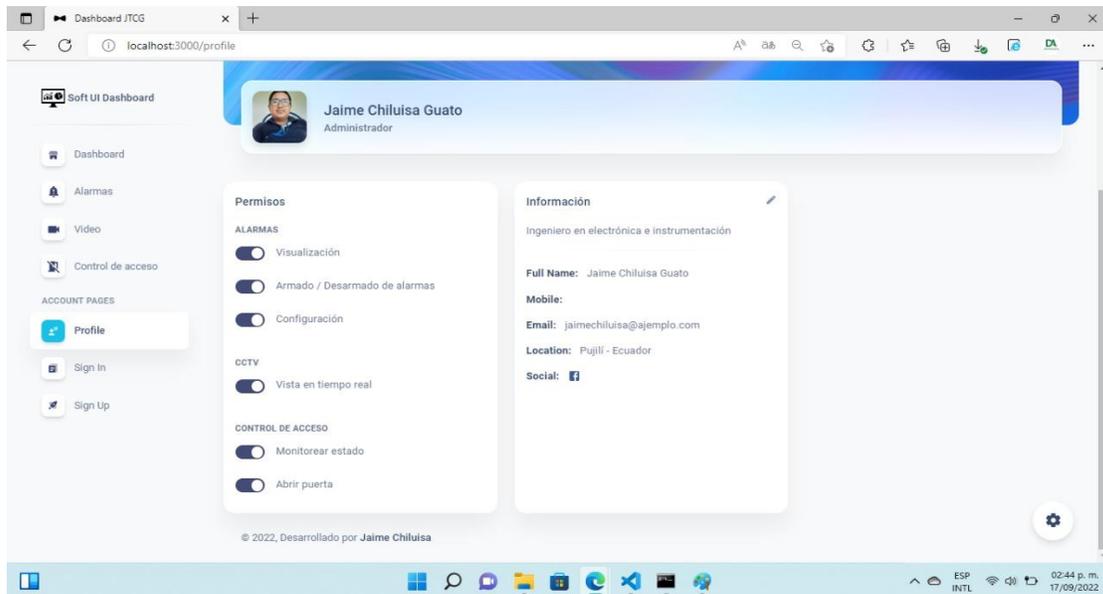
Datos incorrectos al ingresar al sistema de seguridad.



Una vez que el Usuario y contraseña sean las correctas se puede ingresar al entorno web que contiene todo el sistema de seguridad, en esta pantalla se tiene subpantallas a la que se puede acceder y modificar sus parámetros de acuerdo con privilegios programados y guardados.

**Figura 34**

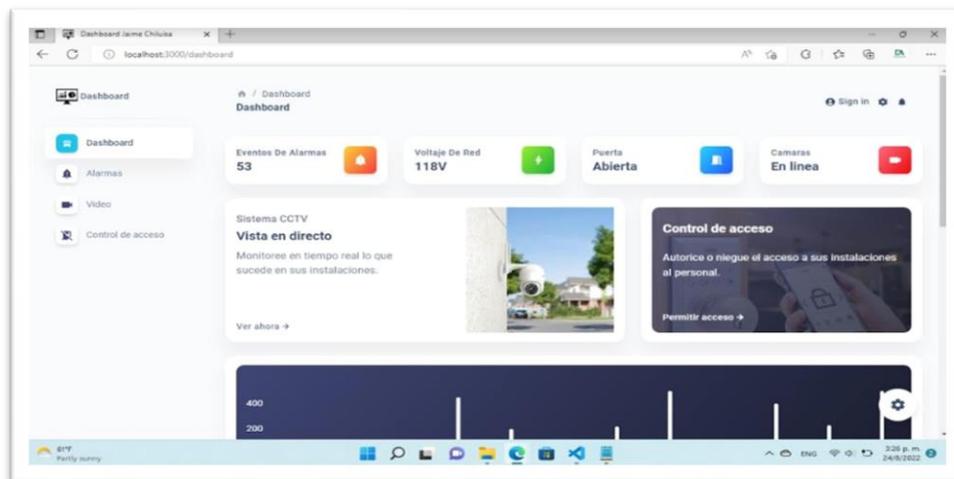
*Plataforma principal de sistema de seguridad.*



Al ingresar a la plataforma web del sistema de seguridad, se tendrá un set de botoneras que nos direccionan a subpantallas tales como; Dashboard, alarmas, video, control de acceso. Al presionar Dashboard este direcciona a la pantalla principal de mandos, donde se tiene Eventos de alarmas, Voltaje de Red, Accesos de puerta a bodega, y las pantallas de visualización de cámaras.

**Figura 35**

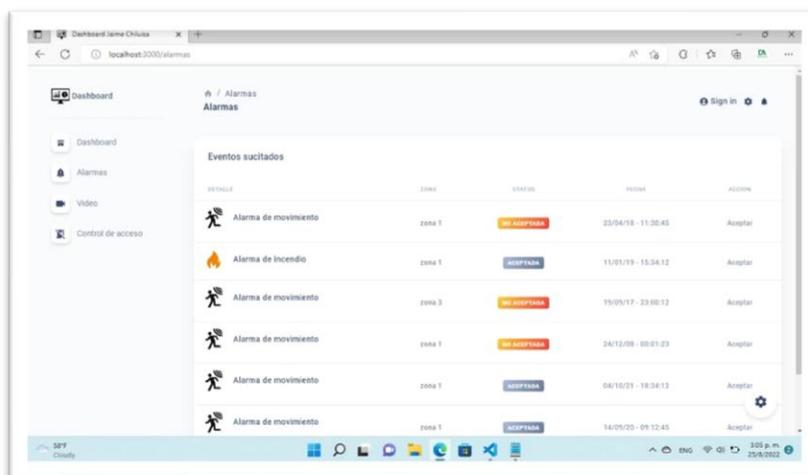
*Pantalla de mandos de sistema de seguridad.*



Al presionar el botón Alarmas este nos direcciona a la siguiente pantalla que contiene toda la información de todas las alarmas que se activaron.

**Figura 36**

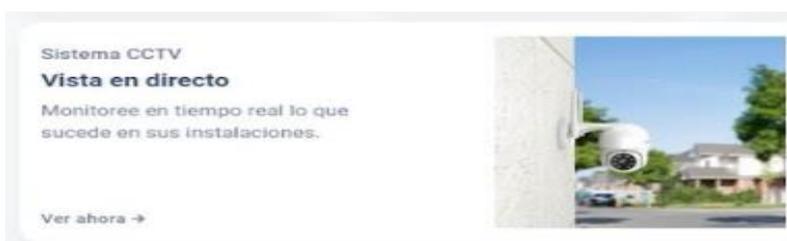
*Pantalla de eventos de sistema de seguridad.*



Al presionar el botón Videos se ingresa a la pantalla que indica en tiempo real lo que está sucediendo en la Unidad Educativa mediante dos cámaras instaladas en puntos estratégicos.

**Figura 37**

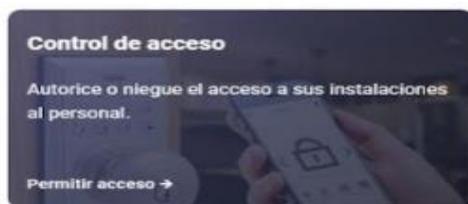
*Pantalla de videos de sistema de seguridad.*



Al presionar el botón Control de Accesos, rebota un cuadro de dialogo el cual indica ingresar usuario y contraseña, si los datos ingresados son correctos, de acuerdo con los privilegios programados, permite el acceso donde se puede activar la chapa magnética que está ubicado en puerta del aula número 3, aula que es utilizada como bodega principal.

**Figura 38**

*Pantalla de control de acceso de seguridad.*



### ***Pruebas de envío de peticiones desde el sistema de seguridad hacia los equipos***

Al tener diseñado las ventanas del sistema de seguridad y la comunicación mediante las API, se procede a enviar peticiones desde el sistema hacia los equipos, para probar la comunicación y las distintas salidas de accionamientos que se tiene en el sistema.

### **Pruebas de comunicación módulo sonoff con backend.**

El testing para la comunicación mediante un api al backend se realizó para comprobar que los componentes desarrollados mediante REACTJS se actualicen de manera correcta sobre el navegador para monitorear el sistema de alimentación de los equipos instalados en la Unidad Educativa, en la figura 39, se observa la respuesta indicando el valor en tiempo real de energía que alimenta a la infraestructura, en este caso nos da un valor de 118V AC, indicando la fecha y la hora de la toma de datos.

**Figura 39**

*Respuesta de testing módulo sonoff con backend*



## Pruebas de comunicación con los dispositivos en formato JSON

La comunicación entre el frontend y el backend se realiza a través de solicitudes y respuestas.

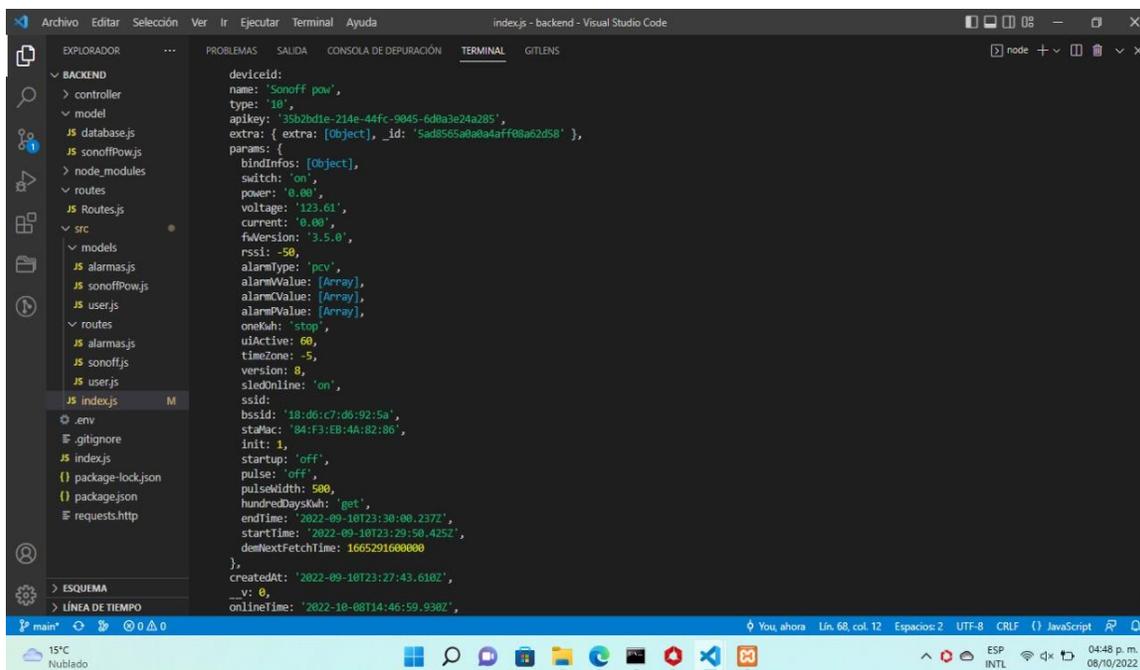
Estas solicitudes y respuestas se envían y reciben en formato JSON, en la figura 40 podemos observar la respuesta de los equipos instalados, las cuales se actualizan cada 500mSeg.

Las respuestas del servidor están definidas como:

- Respuesta exitosa: success: true, content.
- Respuesta fallida: success: false, content.
- Solicitud no autorizada: success: false, content.

### Figura 40

*Respuesta testing comunicación frontend y backend.*



```

deviceid:
  name: 'sonoff pow',
  type: '10',
  apikkey: '35b2bd1e-214e-44fc-9945-6d8a3e24a285',
  extra: { extra: [Object], _id: 'Sad855a8a8a4aff08a62d58' },
  params: {
    bindInfos: [Object],
    switch: 'on',
    power: '0.00',
    voltage: '123.61',
    current: '0.00',
    fwVersion: '3.5.0',
    rssi: -50,
    alarmType: 'pcv',
    alarmValue: [Array],
    alarmValues: [Array],
    alarmPValue: [Array],
    oneKwh: 'stop',
    uiActive: 60,
    timeZone: -5,
    version: 8,
    sledOnline: 'on',
    ssid:
      bssid: '18:d6:c7:d6:92:5a',
      staMac: '84:F3:EB:4A:82:86',
      init: 1,
      startup: 'off',
      pulse: 'off',
      pulseWidth: 500,
      hundredDaysKwh: 'get',
      endTime: '2022-09-18T23:39:00.237Z',
      startTime: '2022-09-18T23:29:50.425Z',
      demNextFetchTime: 1665291600000
    },
    createdAt: '2022-09-18T23:27:43.618Z',
    v: 0,
    onlineTime: '2022-10-08T14:46:59.930Z',
  }

```

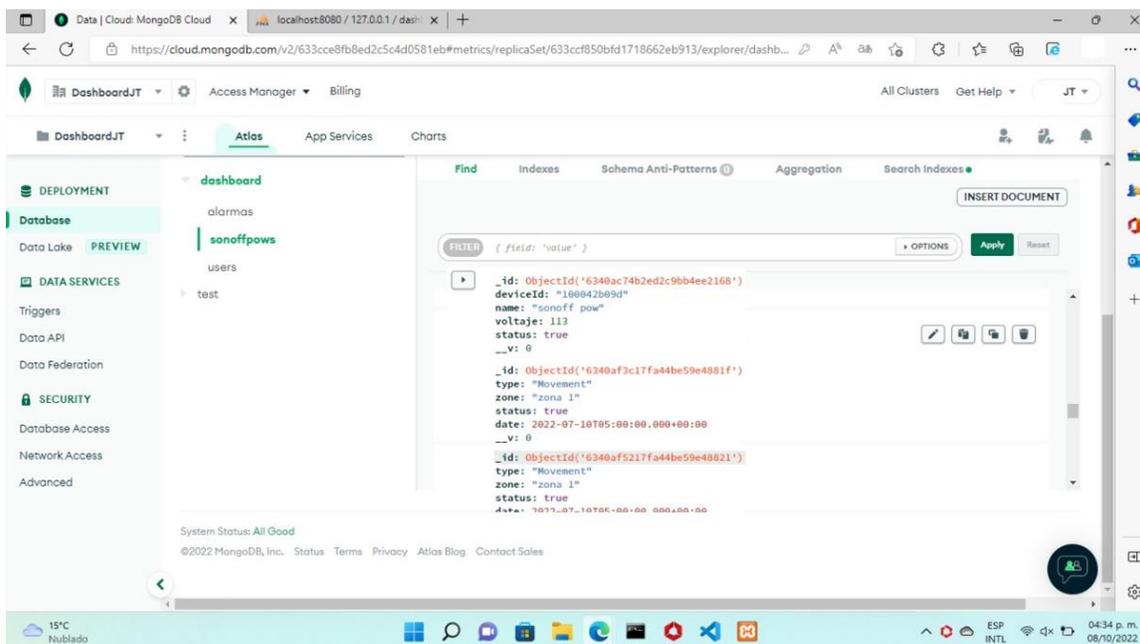
## Prueba de recolección de Datos utilizando MongoDB.

Para crear la base de datos se utiliza MongoDB que es orientada a documentos, es decir que, en lugar de guardar los datos en registros, guarda los datos en documentos, este permite el almacenamiento de

información en una estructura tipo JSON con un esquema dinámico, en la figura 41 se observa el almacenamiento de información tomado cada 500ms.

**Figura 41**

*Prueba de lectura de base de datos con MongoDB.*

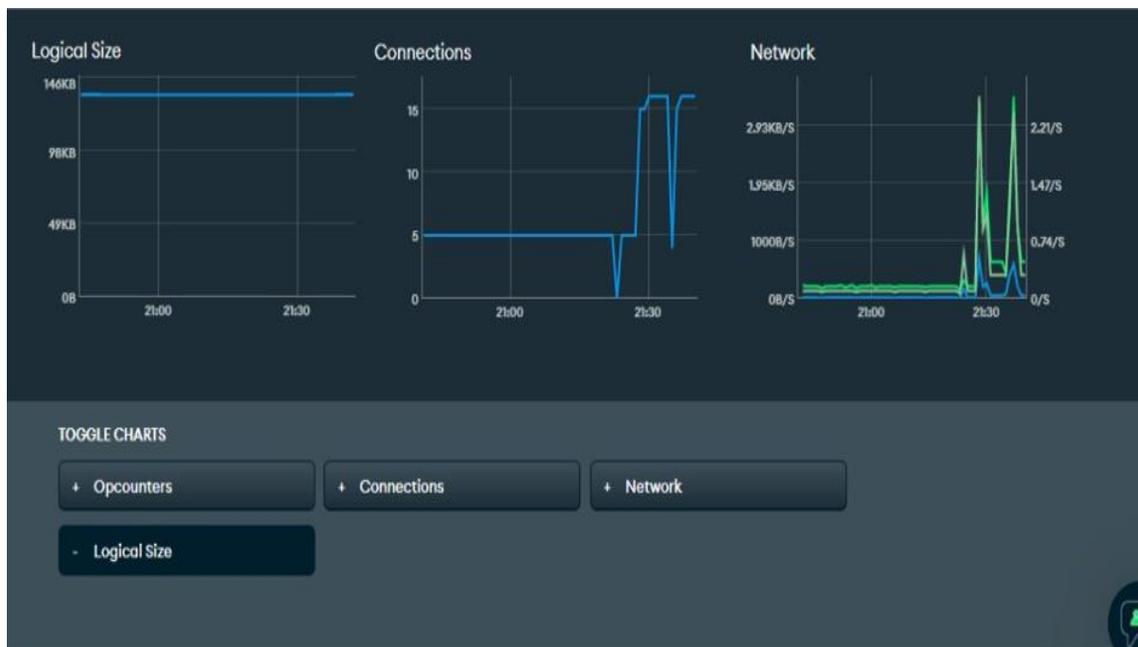


### Pruebas de rendimiento del sistema al almacenar Datos.

MongoDB realiza un consumo moderado del procesador lo cual ayuda a mejorar el rendimiento del sistema, durante las pruebas de los distintos escenarios en la Unidad Educativa se verificó el estado de los recursos del computador donde se almacena la base de datos, en la figura 42 se muestran un resumen de los resultados obtenidos correspondientes al consumo de recursos del computador por parte de las bases de datos.

**Figura 42**

*Tendencia de rendimiento del sistema al almacenar datos*



### Opcounters

Muestra el número de las siguientes operaciones por segundo ejecutadas en un proceso de MongoDB desde el último inicio del proceso:

- Orden
- Consulta
- Insertar
- Borrar
- Actualizar
- Obtener más

### Logical size

Muestra el tamaño total de todos los índices de la base de datos. Esta métrica incluye la sobrecarga en la que incurren los índices además de los datos reales del documento en los que se basan los índices.

## Connections

Muestra el número total de conexiones activas al despliegue de la base de datos, Supervisa las conexiones para determinar si los límites de conexión actuales son suficientes. Caso contrario si es necesario actualizar el nivel de clúster.

## Network

Supervise las métricas de la red para hacer un seguimiento del rendimiento de la misma.

- **BytesIn** muestra la tasa media de bytes físicos (después de cualquier compresión de cable) enviados a este servidor de base de datos por segundo durante el período de muestra seleccionado.
- **BytesOut** muestra la tasa media de bytes físicos (después de cualquier compresión de cable) enviados desde este servidor de base de datos por segundo durante el período de muestra seleccionado.
- **NumRequests** muestra la tasa media de solicitudes enviadas a este servidor de base de datos por segundo durante el período de muestra seleccionado.

## Capítulo V

### Conclusiones y Recomendaciones

#### Conclusiones

A continuación, se presentan las conclusiones a las que se llegó durante el desarrollo de este trabajo, las cuales dan a conocer el funcionamiento del sistema de seguridad implementado en la Unidad Educativa, así como la generación de macrodatos.

- El diseño e implementación del sistema de seguridad fue exitosa, obteniendo los resultados esperados, tanto para el investigador como para los dirigentes del barrio que solicitaron la implementación, se han verificado todos los requisitos de funcionalidad y cumplimiento propuestos en el diseño preliminar y la arquitectura del sistema.
- La investigación y revisión del uso de los dispositivos para sistemas de seguridad, y la integración a través de API con la aplicación web otorgó pautas para la implementación del proyecto, en cuanto a la instalación de dispositivos sensor actuador, fueron las últimas actividades para realizar las pruebas donde se verificó el correcto funcionamiento del sistema, obteniendo monitoreo y control en tiempo real de la Unidad Educativa.
- La generación de macrodatos se realiza en sistema de base de datos orientado a documentos ya que este admite datos en formato JSON las mismas que se utilizó para la comunicación mediante API de los equipos utilizados para la implementación del sistema de seguridad.
- La creación de base de datos de sistema de seguridad consume gran cantidad de espacio en disco por lo cual se debe realizar un backup cada 3 meses, los macrodatos generados servirán a futuro para diseñar un plan de seguridad ciudadana.
- La creación de la interfaz amigable con el usuario permite visualizar, administrar y gestionar la plataforma de una manera sencilla ya que la aplicación está orientado a usuarios que no tienen conocimiento en manejo de sistemas de gestión de edificios (BMS).

- El desarrollo del sistema de seguridad en la Unidad Educativa ayudó a entender y poner en práctica la integración entre la tecnología de desarrollo Node.js y la base de datos MongoDB.

## Recomendaciones

A continuación, se presenta las recomendaciones más importantes del sistema de seguridad desarrollado.

- Se recomienda realizar backup de la base de datos cada 3 meses ya que se almacena gran cantidad de información en disco.
- Se recomienda tener en cuenta que una vez que el sistema de seguridad desarrollado sea usado en trabajos similares, se debe realizar un respaldo de la base de datos y archivos del servidor para asegurar la disponibilidad de los recursos.
- Se recomienda fomentar la implementación de estos sistemas de seguridad en establecimientos como universidades, centros comerciales, hoteles, casas, conjuntos residenciales, condominios, etc. permitiendo a las personas que habitan o trabajan en esas instalaciones dispongan de un nivel de confort y seguridad adecuado.

## Bibliografía

- Carrillo, I. A. (2016). *Aplicación Web de gestión Financiera Personal desarrollado por Mean.js*. Obtenido de <https://1library.co/document/zp0wo37q-aplicación-web-gestión-financiera-personal-desarrollada-mean-js.html>
- CASTILLO, B. L. (2019). *DISEÑO DE UN SISTEMA DE GESTIÓN DE EDIFICIOS BMS*.
- Chentao Li, K. D. (2019). Design of Smart Home Monitoring and Control System Based on.
- Delía, L. N. (2009). *Framework para el desarrollo ágil de sistemas web*. Obtenido de Framework para el desarrollo ágil de sistemas web: <http://sedici.unlp.edu.ar/handle/10915/4000>
- Domínguez, C. G. (2015). *Aplicaciones orientadas a la domótica con Raspberry Pi*.
- Domóticas Integradas, SL. (2021). *Domoticaintegrada.com*. Obtenido de <https://domoticaintegrada.com/>
- GSL, I. (2022). *Industrias GSL*. Obtenido de <https://industriasgsl.com/blogs/automatización/sistema-bms>
- Hat, R. (31 de Octubre de 2017). *Hat Red*. Obtenido de Hat Red: <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>
- Lita I., V. D. (2016). Indoor environmental parameters monitoring for building automation systems. .
- Marques, G. &. (2017). Monitoring and control of the indoor environment. .
- Melendez, J. C. (2014). *MÓDULO WEB FRONT-END PARA EL DESARROLLO DE SIMULACIÓN A PARTIR DE WEIBULL, JI CUADRADO Y BETA*. Obtenido de <https://repository.ucatolica.edu.co/bitstream/10983/1283/1/Trabajo%20de%20grado%20final.pdf>
- Navarrete, J. M. (2017). *Diseño e implementación de un entorno web con nodeJs*. Obtenido de Diseño e implementación de un entorno web con nodeJs: <https://uvadoc.uva.es/bitstream/handle/10324/38684/TFG-G3692.pdf?sequence=1>
- Navarro, G. (2022). *Sistemas de seguridad electrónica*. Obtenido de <https://gruponavarro.pe/sistemas-seguridad/que-es-la-seguridad-electronica/>
- Nestrategia. (2022). *nestrategia.com*. Obtenido de [nestrategia.com: https://nestrategia.com/desarrollo-web-back-end-front-end/](https://nestrategia.com/desarrollo-web-back-end-front-end/)
- Nicolay Valov, I. V. (2020). Home Automation System with Raspberry Pi.
- OJEDA, J. C. (2008). *ESTUDIO DE FRAMEWORKS PARA EL DESARROLLO DE APLICACIONES*. Obtenido de ESTUDIO DE FRAMEWORKS PARA EL DESARROLLO DE APLICACIONES:

[https://repository.unab.edu.co/bitstream/handle/20.500.12749/1230/2008\\_Tesis\\_Prada\\_Ojeda\\_Juan\\_Camilo.pdf?sequence=1&isAllowed=y](https://repository.unab.edu.co/bitstream/handle/20.500.12749/1230/2008_Tesis_Prada_Ojeda_Juan_Camilo.pdf?sequence=1&isAllowed=y)

Omorogiuva, E. (2019). Review of Smart Based Building Management System.

Pérez, J. E. (2019). *Introducción a Javascript*. Obtenido de [http://190.57.147.202:90/jspui/bitstream/123456789/430/1/introduccion\\_javascript%20%281%29.pdf](http://190.57.147.202:90/jspui/bitstream/123456789/430/1/introduccion_javascript%20%281%29.pdf)

Portilla, W. A. (2021). *Implementación de nueva tecnología Front-end para mejorar rendimiento de sitios web*. Obtenido de [https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/4432/Walter\\_Brice%C3%B1o\\_Trabajo\\_de\\_Suficiencia\\_Profesional\\_Título\\_Profesional\\_2021.pdf?sequence=1&isAllowed=y](https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/4432/Walter_Brice%C3%B1o_Trabajo_de_Suficiencia_Profesional_Título_Profesional_2021.pdf?sequence=1&isAllowed=y)

Pranav Kumar Madupu, B. K. (2018). Automatic Service Request System for Security in.

Raghad Al-Syouf, B. A.-D. (2021). Towards a Secure Web-Based Smart Homes.

React. (2021). *Biblioteca de JavaScript para construir interfaces de usuario*. Obtenido de <https://es.reactjs.org/>

Rossell, D. (2014). Macrodatos y Estadística. *Macrodatos y Estadística*.

Sendari S., L. D. (2016). Developing prototype of web-based home controlling for teaching trainer. .

Shane Hamilton, M. K. (2019). El imperio de los macrodatos. *El imperio de los macrodatos*.

Stangaciu V., O. V. (2017). Scalable interconnected home automation system.

Tonarelli, S. M. (2021). *Tecnologías de software para interfaz de usuario adaptativa y multiplataforma*. Obtenido de <https://dspace.ort.edu.uy/bitstream/handle/20.500.11968/4497/Material%20completo.pdf?sequence=1&isAllowed=y>

Tzuc, O. M. (2015). Sistema Domótico de control de luces mediante una aplicación Android basada en una tarjeta Raspberry-Pi.

# ANEXOS