Prototipo de gestión de eventos y alertas de dispositivos IoT en alarmas de seguridad en inmuebles

Cristian Camilo Pérez Ramírez

José María Cabrera Galvis

Código

10892125406

10892128386

Universidad Antonio Nariño

Programa Ingeniería Sistemas

Facultad de Ingeniería Sistemas y Computación

Bogotá, Colombia

2022

Plataforma para la gestión y monitoreo de dispositivos IoT de seguridad privada en alarmas de seguridad para inmuebles.

Cristian Camilo Pérez Ramírez

José Maria Cabrera Galvis

Proyecto de grado presentado como requisito parcial para optar al título de:

Especialista en ingeniería de software

Director (es):

Dianalin Neme Prada

Universidad Antonio Nariño

Especialización de ingeniería de software

Facultad de Ingeniería Sistemas y Computación

Bogotá, Colombia

2022

NOTA DE ACEPTACIÓN

El trabajo de grado titulado

_____,

Cumple con los requisitos para optar

Al título de _____.

_____

Firma del Tutor

_____

Firma Jurado

_____

Firma Jurado

Bogotá, 09 Mayo 2022.

Índice de Figuras

Índice de Tablas

Preliminares

(Dedicatoria)

A Dios primero por permitimos vivir esta oportunidad de aprendizaje, a los profesores de la Especialización por su compromiso y entrega demostrado en las horas de clase y asesorías, a la directora del programa Dianalin por su apoyo para la construcción del presente documento y a Camilo por su responsabilidad, gran compromiso y dedicación.

José Cabrera

El presente trabajo es dedicado a Dios, a mi esposa a mi familia, docentes, porque cada uno aporto y contribuyo por lograr este título. A los docentes aportando con su conocimiento y experiencia en la industria del software.

Camilo Pérez

1. Resumen

Este proyecto surge a partir del problema de inseguridad en las viviendas o comercios que afecta la tranquilidad e integridad de muchos de los habitantes de Bogotá, en la actualidad se cuentan con sistemas de seguridad que no son asequibles para todos. Es por esto que el presente proyecto desarrolló un prototipo IoT asequible y con hardware open source soportado a través de una plataforma web el cual se desarrolla con el fin de monitorear y notificar a las empresas de seguridad o propietarios del inmueble a través de diferentes medios de comunicación como mensajes de textos (SMS), correo electrónico y telegram.

Palabras clave: IoT, eventos, WiFi, plataforma web, microservicios, correo electrónico, mensaje de texto (SMS), MQTT, Hardware open source.

1. Abstract

This project arises from the problem of insecurity in houses or shops that affects the calm and integrity of many of us, considering that currently these places have security systems that are not affordable for everyone. This is why this project has developed an affordable IoT prototype with open source hardware supported by a web platform which is developed in order to monitor and notify security companies or property owners through different media such as text messages (SMS) and email.

Keywords: IoT, events, WiFi, web platform, microservices, email, text message (SMS), MQTT, open source hardware.

2. Introducción

Una de las mayores problemáticas de convivencia y seguridad en Bogotá, es la inseguridad en los inmuebles o comercios, generando zozobra y miedo a los habitantes dado a que ya no están tranquilos, ni en sus hogares. Cada vez es más llamativo para los delincuentes las viviendas o comercios para delinquir, en el año 2021 en la ciudad de Bogotá fue uno de los delitos con altos índices, según la policía nacional en ese año se perpetraron 20.644 hurtos a residencias o, en otras palabras, 85 casos cada día. Siendo una cifra preocupante no solo a nivel de Bogotá sino también a nivel nacional (*Oracle business intelligence sign in*, s/f).

Por lo anterior, se propone en el presente proyecto la construcción de un prototipo de sistema de gestión y monitoreo de dispositivos IoT en alarmas de seguridad para inmuebles de propiedad horizontal, que cuenten con un esquema de seguridad privado, que facilite a los usuarios de los inmuebles llevar un monitoreo y control de su inmueble. y así mejorar la respuesta del de seguridad. Este proyecto se desarrolla aplicando conceptos de ingeniería de software como la arquitectura, el desarrollo, la calidad y la seguridad, además de otras ramas de las ingenieras permitiendo diseñar un prototipo IoT open source y funcional.

Para la realización del prototipo IoT, se realiza un análisis de productos similares ya existentes en el mercado, y se realizaron versiones del prototipo que tomaron alrededor de 5 meses para tener una versión final y totalmente funcional, posteriormente se realizan los diseños de software correspondientes y planteamientos de posibles tecnologías a

implementar, la cual, es evaluada con los escenarios de rendimiento y de seguridad más relevantes del sistema con el objetivo de crear un proyecto aplicando una metodología de desarrollo ágil acorde con las con la naturaleza del proyecto y sus componentes, se desarrollara una aplicación web que permita la captura y gestión de eventos generados por el prototipo IoT, alertando a la empresa de seguridad privada y propietarios con mensajes de texto, correo electrónico y al vigilante de turno.

3. Formulación y descripción del problema

La inseguridad en inmuebles ubicados en la ciudad de Bogotá siempre ha sido una problemática de convivencia y seguridad en la ciudad, Entre el mes de enero y marzo se han cometido alrededor de 1429 delitos de hurtos a residencias y 551 delitos menos que el año anterior disminuyendo en un 27,8% respecto al 2021(*Oracle business intelligence sign in*, s/f), Sin embargo en años anteriores se esta problemática venía aumentando su índice de delitos, según el SIEDCO - DIJIN de la policía nacional "A *enero de 2018 el balance de 2016 y. 2017 en hurto a residencias y comercio mostraba aumentos del 5% y 13% respectivamente. Tres meses más tarde, el balance es de incrementos del 90% en hurto a residencias y del 64% en hurto a comercios"* (SIEDCO, 2018). Estos delitos no discriminan estrato ni horario ni el tipo de vivienda, sin embargo, estos delitos ocurren la gran mayoría en ausencia de los habitantes o propietarios del inmueble. "*De los hogares que informaron haber sufrido hurto a residencias en 2018 para total nacional, el 68,4%, habitaba en vivienda tipo casa, mientras que el 29,8% habitaba en apartamento. Frente a la hora en la que ocurre este delito, el rango es entre las 12:00 a.m. y las 5:59 a.m. presentó la prevalencia más alta con 25,4% para total nacional."* (ECSC, 2018).

La mayoría de los inmuebles no cuentan con sistemas de seguridad efectivos, sin embargo, los sistemas de seguridad que se implementan en la mayoría son rejas, concertinas, redes eléctricas, cerraduras de seguridad, trancas, candados y entre otras. Los sistemas de seguridad como lo son las alarmas o sensores de movimiento, entre otros circuitos cerrados son de difícil acceso para los propietarios o residentes de los inmuebles por su costo y manutención, según el ECSC – 2018, solo el **4%** de 1267 encuestados contaban con estos sistemas de seguridad. Sin embargo, el costo en promedio de alquiler de un sistema de alarma de seguridad es de $1.050.000 COP, y su mensualidad o monitoreo es de $123.000 COP. Hoy en día se cuenta con sistema de alarmas de seguridad pero muchos de ellos no se encuentran enlazados a una plataforma en línea o solo funcionan localmente sin un monitoreo, que centralice las peticiones de eventos en una respuesta de comunicación desde la central hacia la empresa de seguridad y sus supervisores de seguridad.

Los dispositivos existentes en el mercado representan tecnologías costosas para la implementación en inmuebles, es por esto que se plantea la necesidad de desarrollar una plataforma para la gestión y monitoreo de dispositivos IoT de seguridad privada donde se puedan centralizar las peticiones enviadas desde el dispositivo conectado en línea, e interpretar las peticiones y generar una respuesta en forma inmediata a los interesados.

4. Objetivo General

Desarrollar un prototipo para gestionar los eventos de alarmas de seguridad IoT a través de una plataforma que monitoriza los eventos y notifica a los interesados.

5. Objetivos específicos

1. Desarrollar una plataforma para la gestión y monitoreo de sistema alarmas de seguridad WiFi.
2. Integrar la plataforma Web con el prototipo de alarma de seguridad WiFi.
3. Interpretar los eventos generados por el hardware IoT, generando reportes a los interesados.
4. Probar el funcionamiento de la integración de la plataforma para la gestión y monitoreo del prototipo IoT de seguridad, con al menos 4 prototipos funcionando en campo.
5. Generar alertas de seguridad desde la plataforma para notificar a los interesados de los eventos del dispositivo de IoT.
6. Asegurar la recepción de eventos en un 90% cuando el prototipo se encuentre conectado a WiFi.

6. Marco de Referencia

6.1. Estado del Arte

Durante la última década, los avances en la tecnología han transformado la manera de comunicar al ser humano con casi absolutamente todo lo que lo rodea y con el surgimiento del internet de las cosas (IoT), la cantidad de periféricos que son conectados a la red global crece a una velocidad nunca antes vista, se espera que el número de dispositivos IoT alcance los 50 mil millones para 2020 (Bogue, 2014). Muchos de estos dispositivos los vemos en la

actualidad las manillas inteligentes, las bombillas inteligentes, nuestros móviles o diferentes dispositivos en áreas como el hogar, industrias, automotores, agricultura, salud y entre otras.

En la actualidad en el mercado se encuentran diferentes dispositivos de alarmas de seguridad que cuentan con un panel de control, sensores de movimientos, sensores magnéticos, sirenas y botones de pánico o activación, una de las alarmas en el mercado comercializada por Technoimport cuenta con estos dispositivos y su funcionamiento se basa en "*Cuando un sensor se activa, envía la señal al panel de control, activando su sirena. El panel conectado a Internet por cable o red celular notifica al servidor en la nube de WeBeHome. El servidor de WeBeHome guarda estos registros y envía la señal a los dispositivos móviles por mensaje PUSH a la aplicación de WebeHome, mensaje de texto, correo electrónico y llamada telefónica.*" (*Kit Básico LS-10*, s/f). Este dispositivo de Technoimport tiene un costo de $1.360.000 COP, con un año de suscripción a WeBeHome gratis. La mayoría de estos dispositivos su comunicación con el panel es inalámbrica con frecuencias 433, 868 o 915 MHz, y cada sensor tiene un único número de identificación parametrizado en el panel de control para su respectiva recepción de señales. Otro de los dispositivos comercializados es la alarma de seguridad por PROSEGUR la cual consta de un panel GPRS con teclado, una sirena, una botonera de pánico, sensor de movimiento, sensor magnético y llave de proximidad o tag la cual tiene un costo de alquiler por $1.089.000 COP con una mensualidad por monitoreo de $123.000 COP incluyendo el servicio de ayuda que brinda, "*Personal propio de vigilantes motorizados, que ante una emergencia de tu sistema de alarma se dirigen al domicilio para prestar apoyo y cooperar con las fuerzas de seguridad brindando asistencia."* y acceso a la app PROSEGUR SMART (*Alarma Monitoreada para el Hogar*, n.d.) en la que se puede revisar el historial de eventos

generados, activar o desactivar la alarma, consultar grabaciones o imágenes de las cámaras enlazadas.

A nivel internacional se encuentran varios servicios de monitoreo de sistemas de seguridad incluyendo la automatización del hogar y el ahorro de dinero en facturas mensuales, uno de estos es la empresa FSS Technologies, que cuenta con sistemas de seguridad incluyendo timbres con video, aplicación de monitoreo, termostato, cerraduras inteligentes, controles de luz desde la aplicación, cámaras de interior o exterior y detectores de agua, un kit básico que cubre tres puertas, un sensor de movimiento, un teclado y una aplicación de teléfono integrada tiene un costo de $300 USD y la mensualidad tiene un costo de $35 USD (*Residential Security Systems*, n.d.).

Dada la investigación que realizamos encontramos que muchos de estos sistemas de alarmas de seguridad son costosas dificultando la accesibilidad para los propietarios y su monitoreo en promedio es costoso.

6.2. Impacto

El impacto social que tiene el presente prototipo es generar seguridad y una respuesta rápida, el dispositivo permitirá la protección de inmuebles o incluso integrantes de su núcleo familiar en viviendas de propiedad horizontal donde cuenten con un esquema de seguridad, el cual el dispositivo notificara de la emergencia al vigilante de turno evitando procesos desde la central y respondieron de una manera rápida y oportuna al estar en el sitio de la emergencia.

6.3. Componente de innovación

El componente de innovación del proyecto es la integración de los prototipos de alarmas de seguridad IoT y el envío de notificaciones a los interesados de una manera rápida y oportuna el cual como primera medida de reacción ante el evento se le notificará al vigilante de turno de la propiedad horizontal, haciendo uso de los servicios brindados por la plataforma. Se busca ser transparentes ante las alertas o eventos presentados mejorando la comunicación, orientada a ser efectiva y eficiente ante los interesados.

6.4. Marco Teórico

En los últimos tiempos se ha visto un incremento de las estadísticas en los homicidios, hurtos y robos en la ciudad de Bogotá, según la revista semana, jueves 2 septiembre “los casos de robos han crecido un 21,9% en el mes de junio 2021 con respecto al mes anterior.

Debido a esta brecha en la inseguridad se ha visto un aumento de la instalación de circuitos de seguridad para empresas y hogares, para resguardar sus bienes de hurtos, poder confiar en un sistema de seguridad que garantice la protección de sus patrimonios como una posible solución.

Se ha observado que, con el paso de los años, el internet ha podido evolucionar y posicionarse como una herramienta de uso esencial para el desarrollo de las actividades cotidianas. Podríamos clasificar la evolución de la internet en 3 etapas.

La web 1.0 se caracterizaba por el broadcast, es decir, la existencia de grandes emisores de contenido y el usuario tenía un rol pasivo, las páginas por lo general eran estáticas, no se podían añadir comentarios.

Luego viene la web 2.0 donde se comenzaron a visualizar sitios dinámicos, pasa de ser un simple contenedor o fuente de información; la web en este caso se convierte en una plataforma de trabajo colaborativo. Según Tim O'Reilly (2005) Web 2.0 es la red como plataforma, extendiéndose a todos los dispositivos conectados.

En la Web 3.0 También conocida como Web Semántica, tiene que ver con el uso de un lenguaje más natural para acceder a los contenidos de Internet. En esta evolución, la web permite el uso de un lenguaje orgánico, común al momento de buscar contenidos.

Con la evolución de la Minería de Datos la Web pasa a tener una estructura más organizada, estandarizada, reconocible no sólo por navegadores de Internet, sino básicamente por cualquier dispositivo capaz de procesar datos. Esta es la base de la Inteligencia Artificial. De esta forma, cada elemento existente en la web es capaz de describirse a sí mismo mediante "folksonomia" o etiquetas que identifican a dicho elemento y permiten clasificarlo con palabras que pueden ser usadas por buscadores cuando introducimos, en lenguaje natural, nuestros requerimientos en las cajas de texto.

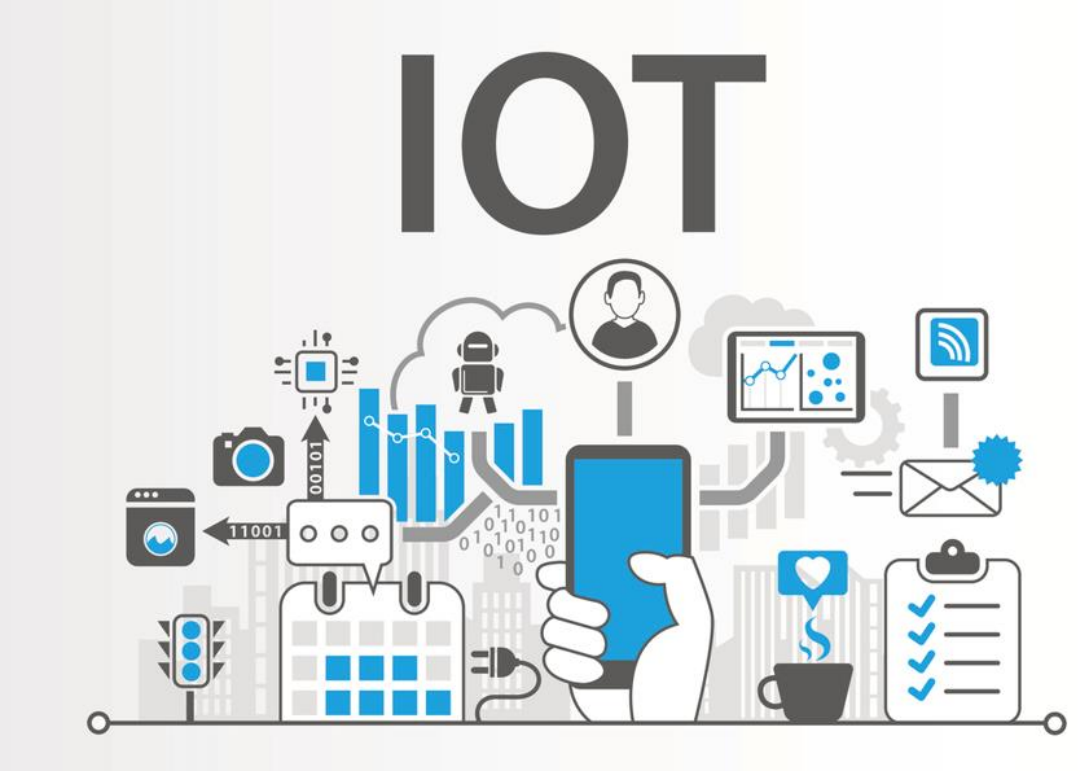
Respecto de la Web 4.0, se dice que consistirá en una profundización aún mayor del entendimiento que las máquinas tienen sobre el lenguaje natural de los humanos, por cuanto aspectos de inteligencia artificial y Web Semántica estarán presentes, así como reconocimiento de voz y experiencias 3D así como la supresión de barreras entre el mundo online y offline, algo semejante al "Internet de las cosas".

Internet de las cosas: El concepto de internet de las cosas fue propuesto en 1999, por Kevin Ashton, en el Auto-ID Center del MIT,7 en donde se realizaban investigaciones en el campo de la identificación por radiofrecuencia en red (RFID) y tecnologías de sensores.

Es un concepto que se refiere a una interconexión digital de objetos cotidianos con internet.3 4 Es, en definitiva, la conexión de internet más con objetos que con personas.2 También se suele conocer como internet de todas las cosas o internet en las cosas. Si los objetos de la vida cotidiana tuvieran incorporadas etiquetas de radio, podrían ser identificados y gestionados por otros equipos de la misma manera que si lo fuesen por seres humanos.

IoT: internet of Things (IoT) describe la red de objetos físicos (cosas) que incorporan sensores, software y otras tecnologías con el fin de conectar e intercambiar datos con otros dispositivos y sistemas a través de Internet. Estos dispositivos van desde objetos domésticos comunes hasta herramientas industriales sofisticadas.

Figura 1. Internet de las Cosas



Fuente: (Aytac, 2022)

Prototipo: es una representación aparente pero concreta de parte o la totalidad de una idea de negocio o sobre un producto o servicio.

Plataforma: es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible.4 Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una plataforma de software (incluyendo entornos de aplicaciones).5 Al definir plataformas se establecen los tipos compatibles de arquitectura, sistema operativo, lenguaje de programación o interfaz de usuario.

Seguridad inmobiliaria: es un factor determinante en la Compra de un Inmueble. La búsqueda de casas, oficinas o departamentos en venta, caen siempre ante el juicio de donde están ubicados.

Seguridad Privada: La seguridad privada son empresas que proveen servicios para mantener bienes e infraestructuras de una institución protegidos minimizando los riesgos de robo o intrusión.

La seguridad privada se refiere a la seguridad en términos de protección de una persona, empresa o evento y, privada se refiere a lo contrario de público, o sea, no es un servicio otorgado por el Estado por lo tanto no tiene los mismos poderes ni jurisdicción.

La seguridad privada se relaciona con guardias de seguridad y cámaras de vigilancia conectado con sistemas de alarmas. Dependiendo del tipo de bienes que se desea proteger, la seguridad privada otorgará servicios personalizados según el tipo de negocio.

Alarma de Seguridad: Un sistema de alarma consiste en la instalación de una serie de equipos electrónicos en los lugares de su hogar o empresa considerados estratégicos desde

desde el punto de vista de la seguridad y que están conectados hacia la Central de Monitoreo de ADT. Estos dispositivos pueden ser sensores de movimiento, contactos magnéticos, detectores de humo, botón de pánico, entre otros, y éstos envían señales en forma periódica a nuestra Central de Monitoreo.

MQTT: son las siglas MQ Telemetry Transport, aunque en primer lugar fue conocido como Message Queing Telemetry Transport. Es un protocolo de comunicación M2M (machine-to-machine) de tipo message queue.

Está basado en la pila TCP/IP como base para la comunicación. En el caso de MQTT cada conexión se mantiene abierta y se "reutiliza" en cada comunicación. Es una diferencia, por ejemplo, a una petición HTTP 1.0 donde cada transmisión se realiza a través de conexión.

MQTT fue creado por el Dr. Andy Stanford-Clark de IBM y Arlen Nipper de Arcom (ahora Eurotech) en 1999 como un mecanismo para conectar dispositivos empleados en la industria petrolera.

Aunque inicialmente era un formato propietario, en 2010 fue liberado y pasó a ser un estándar en 2014 según la OASIS (Organization for the Advancement of Structured Information Standards).

Protocolos de red: son un conjunto de reglas que gobiernan la comunicación entre dispositivos que están conectados a una red. Dichas reglas se constituyen de instrucciones que permiten a los dispositivos identificarse y conectarse entre sí, además de aplicar reglas de formateo, para que los mensajes viajen de la forma adecuada de principio a fin. Dichas reglas de formateo determinan si los datos son recibidos correctamente o si son rechazados o ha habido algún tipo de problema en la transferencia de la información.

TCP/IP: son protocolos o conjuntos de normas para formatos de mensaje y procedimientos que permiten a las máquinas y los programas de aplicación intercambiar información. Cada máquina implicada en la comunicación debe seguir estas normas para que el sistema principal de recepción pueda interpretar el mensaje. El *conjunto* de protocolos TCP/IP puede interpretarse en términos de capas (o niveles).

TCP/IP define cuidadosamente cómo se mueve la información desde el remitente hasta el destinatario. En primer lugar, los programas de aplicación envían mensajes o corrientes de datos a uno de los protocolos de la capa de transporte de Internet, UDP (User Datagram Protocol) o TCP (Transmission Control Protocolo). Estos protocolos reciben los datos de la aplicación, los dividen en partes más pequeñas llamadas paquetes, añaden una dirección de destino y, a continuación, pasan los paquetes a la siguiente capa de protocolo, la capa de red de Internet.

La capa de red de Internet pone el paquete en un datagrama de IP (Internet Protocol), pone la cabecera y la cola de datagrama, decide dónde enviar el datagrama (directamente a un destino o a una pasarela) y pasa el datagrama a la capa de interfaz de red.

La capa de interfaz de red acepta los datagramas IP y los transmite como tramas a través de un hardware de red específico, por ejemplo redes Ethernet o de Red en anillo.

OWASP IoT: Está diseñado para ayudar a los fabricantes, desarrolladores y consumidores a comprender mejor los problemas de seguridad asociados con el Internet de las Cosas y permitir a los usuarios en cualquier contexto tomar mejores decisiones de seguridad al construir, implementar o evaluar tecnologías IoT. A continuación se describen algunas de las prácticas que debemos tener en cuenta al momento de trabajar con dispositivos IoT (OWASP Internet of Things, s/f).

1. Contraseñas débiles, o codificadas: Uso de credenciales fácilmente forzadas, disponibles públicamente o inmutables, incluidas puertas traseras en firmware o software de cliente que otorgan acceso no autorizado a los sistemas implementados. (OWASP Internet of Things, s/f)

2. Servicios de red inseguros: Servicios de red innecesarios o inseguros que se ejecutan en el propio dispositivo, especialmente aquellos expuestos a Internet, que comprometen la confidencialidad, integridad/autenticidad o disponibilidad de la información o permiten el control remoto no autorizado (OWASP Internet of Things, s/f)
3. Falta de un mecanismo de actualización seguro: Esto incluye la falta de validación de firmware en el dispositivo, la falta de entrega segura (sin cifrar en tránsito), la falta de mecanismos anti retroceso y la falta de notificaciones de cambios de seguridad debido a actualizaciones (OWASP Internet of Things, s/f).
4. Uso de componentes inseguros u obsoletos: Uso de bibliotecas/componentes de software obsoletos o inseguros que podrían permitir que el dispositivo se vea comprometido. Esto incluye la personalización insegura de las plataformas del sistema operativo y el uso de software o componentes de hardware de terceros de una cadena de suministro comprometida (OWASP Internet of Things, s/f).
5. Protección de privacidad insuficiente: Información personal del usuario almacenada en el dispositivo o en el ecosistema que se usa de manera insegura, incorrecta o sin permiso (OWASP Internet of Things, s/f).
6. Transferencia y almacenamiento de datos inseguros: Falta de encriptación o control de acceso de datos confidenciales en cualquier lugar dentro del ecosistema, incluso en reposo, en tránsito o durante el procesamiento (OWASP Internet of Things, s/f).
7. Falta de gestión de dispositivos: Falta de soporte de seguridad en los dispositivos implementados en producción, incluida la gestión de activos, la gestión de

actualizaciones, el desmantelamiento seguro, el monitoreo de sistemas y las capacidades de respuesta (OWASP Internet of Things, s/f).

AWS CLOUD: Es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 200 servicios integrales de centros de datos a nivel global. (What is AWS, s/f)

AWS Cognito: Amazon Cognito proporciona autenticación, autorización y administración de usuarios para sus aplicaciones web y móviles. Tus usuarios pueden iniciar sesión directamente con un nombre de usuario y contraseña, o a través de un tercero como Facebook, Amazon, Google o Apple.

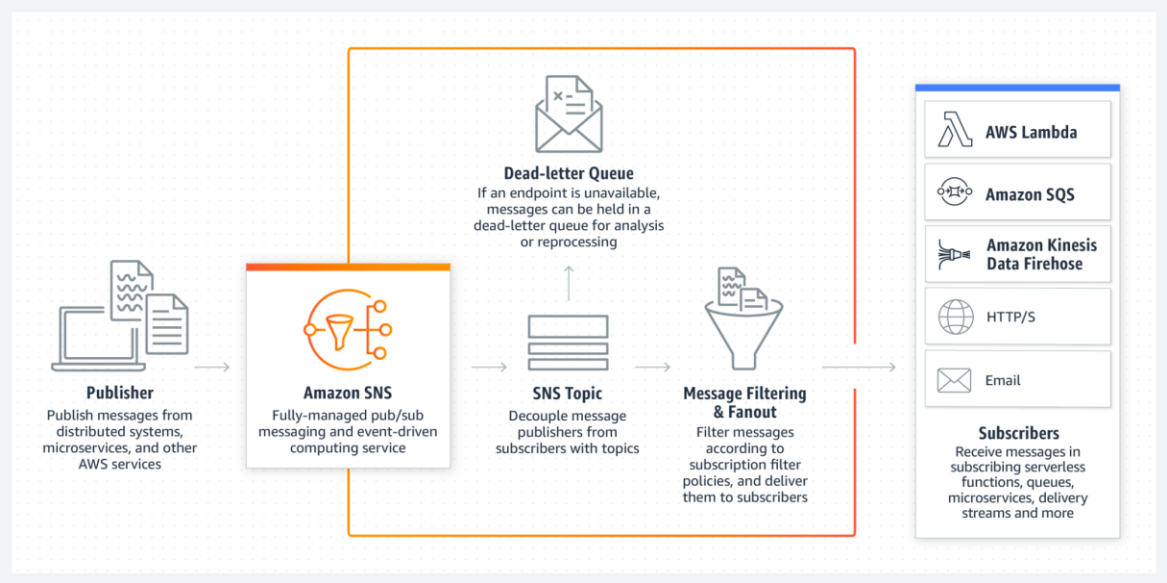
Los dos componentes principales de Amazon Cognito son los grupos de usuarios y los grupos de identidades. Los grupos de usuarios son directorios de usuarios que brindan opciones de registro e inicio de sesión para los usuarios de su aplicación. Los grupos de identidades le permiten otorgar a sus usuarios acceso a otros servicios de AWS. Puede usar grupos de identidades y grupos de usuarios por separado o juntos. (What is Amazon Cognito?, s/f).

AWS SNS: Es un servicio de mensajería completamente administrado para la comunicación aplicación a aplicación (A2A) y aplicación a persona (A2P).

La funcionalidad de publicación y suscripción A2A brinda temas para la mensajería de alto rendimiento, de muchos a muchos, basada en push entre sistemas distribuidos, microservicios y aplicaciones sin servidores controladas por eventos. Mediante el uso de temas de Amazon SNS, los sistemas de publicadores pueden distribuir los mensajes a una

gran cantidad de sistemas de suscriptores, entre otros, colas de Amazon SQS, funciones de AWS Lambda, puntos de enlace HTTPS y Amazon Kinesis Data Firehose para procesamiento paralelo. La funcionalidad A2P permite enviar mensajes a usuarios a escala a través de SMS, push móvil y correo electrónico. (Amazon SNS, s/f).

Figura 2. Diagrama Amazon SNS. Amazon SNS.



Fuente: (Amazon SNS, s/f)

AWS LAMBDA: Es un servicio informático sin servidor y basado en eventos que le permite ejecutar código para prácticamente cualquier tipo de aplicación o servicio backend sin necesidad de aprovisionar o administrar servidores (AWS Lambda, s/f).

Amazon Elastic Container Service (Amazon ECS): permite implementar fácilmente cargas de trabajo en contenedores en AWS. La potente simplicidad de Amazon ECS le permite pasar de un único contenedor Docker a administrar toda su cartera de aplicaciones empresariales. Ejecute y escale sus cargas de trabajo en contenedores a través de zonas de

disponibilidad, en la nube y en las instalaciones, sin la complejidad de administrar un plano de control o nodos. (Aws ECS, s/f)

7. Metodología

Para el desarrollo del proyecto, se usó la metodología ágil Scrumban, pues esta permite ejercer el papel de verificar las actividades propuestas, se caracteriza por permitir entregas parciales y con valor agregado, y al momento de encontrar dificultades se han analizadas y resueltas sin tener contratiempos que afecten los tiempos del proyecto.

Este proceso se hace repetitivo hasta finalizar el desarrollo del proyecto, cumpliendo los objetivos y solucionando la problemática plasmada.

Se conoce Kanban y Scrum como metodologías de gestión Agile. Scrumban combina las mejores características de ambos métodos. Reúne la naturaleza preceptiva de Scrum y la capacidad de mejora del proceso de Kanban, permitiendo a los equipos moverse hacia el desarrollo Agile y a mejorar constantemente sus procesos. Scrumban se está haciendo especialmente popular en industrias en las que el desarrollo del proyecto y el mantenimiento van juntos.

Scrumban evolucionó a partir de una instancia de Scrum complementado con prácticas básicas Kanban. Estas son:

1. Visualizar el flujo de trabajo. Esta visualización obliga a pensar en las fases de las tareas, sus límites y las personas que están implicadas en ellas.

2. Limitar el WIP (work in progress). Se trata de limitar el trabajo en progreso. Cuanto más trabajo hay en progreso, más tiempo se tarda en realizarlas. Hay que tener un flujo de trabajo pequeño.
3. Gestionar el flujo. Consiste en detectar cuellos de botella o embudos, para mejorar el flujo.
4. Establecer políticas explícitas. Se debe documentar los procedimientos de trabajo y estos deben ser de fácil acceso, para que las políticas sean fáciles de entender.
5. Mejora colaborativa y evolución empírica. Guarda relación con la metodología Lean Project Management. Además, hay que establecer revisiones cada cierto tiempo de los resultados.

Los principios básicos de implementación de Scrumban incluyen:

- Empieza con los tableros y labores que usas ahora.
- Está de acuerdo con perseguir la mejora hacia un proceso más efectivo.
- Respeta las labores y responsabilidades actuales mientras pretende mejorarlas fácilmente.

Una combinación de estos dos métodos tiene muchas ventajas. Puede ayudar al equipo de desarrollo a eliminar el elevado estrés, mejorar la eficiencia y la satisfacción general del cliente. Sin olvidar que los beneficios de la integración de estas metodologías:

- La entrega de productos de alta calidad.
- La mejora continua.
- La minimización de pérdidas.

- La reducción del tiempo de producción.

Las herramientas Scrumban - como Kanban Tool - son fáciles de aprender y usar. Kanban Tool proporciona Potenciadores que te ayudan a personalizar tus tableros. Permite la comunicación entre el grupo y colaborar en tiempo real, en cualquier momento y en cualquier lugar - compartiendo tareas, notas, documentos y comentarios.

El tablero Scrumban proporciona una excelente visión del flujo de trabajo de un proceso. Informa del número de cosas en las que el equipo está trabajando actualmente, así como del número de tareas ya finalizadas. Aumenta la rendición de cuentas, la responsabilidad, la comunicación y los resultados de rendimiento.

8. Procesos de Software

En la construcción del software, existen diferentes procesos que permiten analizar cada uno de los detalles a tener en cuenta para crear una plataforma web sin generar reprocesos y producir indeterminaciones en sus fases de creación. Cada etapa o proceso de desarrollo se describe con cierta documentación y diagramas que permiten dar claridad de la construcción de la plataforma web y su integración con el dispositivo IoT.

En la realización del proyecto se tuvieron dificultades para la integración del prototipo IoT con la plataforma web, dado a que no se había trabajado con estas tecnologías sin embargo la curva de aprendizaje afectó en las tareas programadas. Pero se fueron supliendo adquiriendo nuevos conocimientos y asimismo afianzando..

8.1. Requerimientos funcionales

El sistema debe permitir registrar información del apartamento o inmueble y seleccionar el tipo de notificaciones (SMS, Email).

Tabla 1. Requerimiento de registrar información del prototipo.

<table>
<tr><td>ID 1</td><td colspan="2">Registrar información del prototipo o apartamento</td></tr>
<tr><td>Versión</td><td colspan="2">1.0</td></tr>
<tr><td>Dependencias</td><td colspan="2"></td></tr>
<tr><td>Precondiciones</td><td colspan="2">● Se debe contar con un usuario con rol operador o administrador.
● Se debe contar con un prototipo libre y con un serial único.</td></tr>
<tr><td rowspan="4">Secuencia normal</td><td>Paso</td><td>Acción</td></tr>
<tr><td>1</td><td>Autenticarse con el usuario con rol operador o administrador</td></tr>
<tr><td>2</td><td>Ingresar al menú, y seleccionar la opción “Configurar dispositivos”</td></tr>
<tr><td>3</td><td>Seleccionar la opción “Crear apartamento”</td></tr>
</table>

	4	Ingresar los campos requeridos como, Serial del dispositivo, Nombre del propietario, Apellidos del propietario, Dirección, complemento, Celular del propietario, Correo del propietario, Ciudad, Seleccionar conjunto si aplica, Seleccionar tipo de notificaciones email y SMS.
	5	Clic en “Crear”
Postcondiciones	El sistema valida que el serial del dispositivo ingresado no se encuentre registrado, y la información ingresada se guarda en base de datos.	
Excepciones	Paso	Acción
	1	El sistema valida que el serial ingresado, si ya existe en el sistema, muestra un pop al usuario sobre el evento.
	2	Se habilita el botón “Crear”
	Paso	Acción
	1	El sistema valida que todo los campos se encuentren completamente diligenciados de lo contrario, muestra un mensaje de campo requerido.

El sistema debe permitir editar información del apartamento o inmueble y editar el tipo de notificaciones (SMS, Email).

Tabla 2. Requerimiento de editar información del prototipo.

ID 2	Editar información del prototipo o apartamento	
Versión	1.0	
Dependencias		
Precondiciones	1. Se debe contar con un usuario con rol operador o administrador. 2. Se debe contar con un prototipo configurado y con un serial único ya registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	Autenticarse con el usuario con rol operador o administrador
	2	Ingresar al menú, y seleccionar la opción “Configurar dispositivos”

	3	Ingresar el serial del dispositivo al campo de búsqueda, y dar clic.
	4	Seleccionar la opción “Editar apartamento”
	5	Editar los campos necesarios según el caso, como el Serial del dispositivo, Nombre del propietario, Apellidos del propietario, Dirección, complemento, Celular del propietario, Correo del propietario, Ciudad, Seleccionar conjunto si aplica, Seleccionar tipo de notificaciones email y SMS.
	6	Clic en “Editar”
	7	El sistema muestra un PopUp de confirmación “¿Está seguro de editar el dispositivo?” con las opciones Aceptar o Cancelar.
Postcondiciones	El sistema valida que al seleccionar el botón “Aceptar” del modal de confirmación la información editada se guarda en base de datos.	
Excepciones	Paso	Acción

	1	Al seleccionar el botón “Cancelar” del modal de confirmación, el sistema regresa a la pantalla de “Editar apartamento”.
	Paso	Acción
	1	El sistema valida que todo los campos se encuentren completamente diligenciados de lo contrario, muestra un mensaje de campo requerido.

El sistema debe permitir eliminar información del apartamento o inmueble.

Tabla 3. Requerimiento de Eliminar prototipo o apartamento.

ID 3	Eliminar información del prototipo o apartamento
Versión	1.0
Dependencias	
Precondiciones	1. Se debe contar con un usuario con rol operador o administrador.

<table>
<tr><td></td><td colspan="2">2. Se debe contar con un prototipo configurado y con un serial único ya registrado en el sistema.</td></tr>
<tr><td rowspan="7">Secuencia normal</td><td>Paso</td><td>Acción</td></tr>
<tr><td>1</td><td>Autenticarse con el usuario con rol operador o administrador</td></tr>
<tr><td>2</td><td>Ingresar al menú, y seleccionar la opción “Configurar dispositivos”</td></tr>
<tr><td>3</td><td>Ingresar el serial del dispositivo al campo de búsqueda, y dar clic.</td></tr>
<tr><td>4</td><td>Seleccionar la opción “Eliminar apartamento”</td></tr>
<tr><td>5</td><td>El sistema mostrará los campos diligencias, como el Serial del dispositivo, Nombre del propietario, Apellidos del propietario, Dirección, complemento, Celular del propietario, Correo del propietario, Ciudad, Seleccionar conjunto si aplica, tipo de notificaciones email y SMS.</td></tr>
<tr><td>6</td><td>Clic en “Eliminar”</td></tr>
</table>

	7	El sistema muestra un PopUp de confirmación "¿Está seguro de eliminar el dispositivo?" con las opciones Aceptar o Cancelar.
Postcondiciones	El sistema valida que al seleccionar el botón "Aceptar" del modal de confirmación la información se elimina en la base de datos.	
Excepciones	Paso	Acción
	1	Al seleccionar el botón "Cancelar" del modal de confirmación, el sistema regresa a la pantalla de "Eliminar apartamento".

El sistema debe recibir información de los eventos de rutina (armado y desarmado) y notificar a los interesados del apartamento o inmueble, ya sean propietarios o arrendatarios, además debe almacenar estos eventos en una tabla de la base de datos.

Tabla 4. Requerimiento de recibir eventos de rutina.

ID 4	Recibir eventos de rutina del dispositivo IoT
Versión	1.0
Dependencias	

<table>
<tr><td>Precondiciones</td><td colspan="2">Se debe contar con un prototipo configurado y con un serial único ya registrado en el sistema.</td></tr>
<tr><td rowspan="3">Secuencia normal</td><td>Paso</td><td>Acción</td></tr>
<tr><td>1</td><td>El sistema debe identificar los eventos (armado, desarmado) a través del backend, y guardarlos en una tabla de la base de datos</td></tr>
<tr><td>2</td><td>Si está habilitado el check de eventos de rutina, notificar a los interesados registrados</td></tr>
<tr><td>Postcondiciones</td><td colspan="2">El sistema recibe los eventos generados por el dispositivo IoT y lo almacena en base de datos, y notifica a los interesados si se encuentra esta opción seleccionada.</td></tr>
<tr><td rowspan="2">Excepciones</td><td>Paso</td><td>Acción</td></tr>
<tr><td>1</td><td>Al estar vacío el check de notificar eventos de rutina, el sistema guarda los eventos en base de datos pero no notifica a los interesados.</td></tr>
</table>

El sistema debe recibir información de los eventos de emergencia (alarma y pánico) y notificar a los interesados del apartamento o inmueble, ya sean propietarios o arrendatarios y empresas de seguridad privada, además debe almacenar estos eventos en una tabla de la base de datos.

Tabla 5. Requerimiento de notificar eventos de emergencia.

ID 5	Recibir eventos de emergencia del dispositivo IoT	
Versión	1.0	
Dependencias		
Precondiciones	Se debe contar con un prototipo configurado y con un serial único ya registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El sistema debe identificar los eventos (alarma, panico) a través del backend, y guardarlos en una tabla de la base de datos
	2	Por defecto se debe notificar a los interesados registrados y guardias de la empresa de seguridad privada.

Postcondiciones	El sistema recibe los eventos generados por el dispositivo IoT y lo almacena en base de datos, y notifica a los interesados si se encuentra esta opción seleccionada.

El sistema permite consultar los eventos de rutina (armado y desarmado) o eventos de emergencia por dispositivo mostrándose en una pantalla.

Tabla 6. Requerimiento para consultar reporte de eventos.

ID 6	Consultar reporte de eventos del dispositivo IoT	
Versión	1.0	
Dependencias		
Precondiciones	● Se debe contar con un prototipo configurado y con un serial único ya registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	Seleccionar dispositivo
	2	Consultar tipo de reporte, se puede seleccionar por rutina o por emergencia

Postcondiciones	El sistema muestra los eventos generados según lo seleccionado y permite seleccionar el tipo de eventos a consultar.

El sistema permite generar un excel con los eventos de rutina (armado y desarmado) o eventos de emergencia por dispositivo mostrados en pantalla.

Tabla 7. Requerimiento de generar reporte de eventos.

ID 7	Generar reporte de eventos del dispositivo IoT	
Versión	1.0	
Dependencias		
Precondiciones	Se debe contar con un prototipo configurado y con un serial único ya registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	Seleccionar dispositivo
	2	Consultar tipo de reporte, se puede seleccionar por rutina o por emergencia

	3	Seleccionar generar reporte.
Postcondiciones	El sistema muestra los eventos generados según lo seleccionado y permite seleccionar el tipo de eventos a consultar generando un reporte de tipo archivo excel.	

8.2. Requerimientos no funcionales

(https://sites.google.com/, 2022), define los requerimientos no funcionales que no se refieren a las funciones específicas del sistema, sino a otras características emergentes a partir de este como son la fiabilidad, la respuesta en tiempo y la capacidad de almacenamiento.

Los requerimientos funcionales surgen de la necesidad del usuario esto sumado a las restricciones que se generan a partir del presupuesto, a las mismas políticas que adopta el servicio ofrecido, y la necesidad de conexión e interoperabilidad entre los diferentes componentes, así como también factores externos como reglamentos de seguridad y políticas de privacidad, entre otros.

Los requisitos funcionales generan algunas limitantes para el proyecto en el diseño y en la implementación pues se pueden asociar con restricciones del diseño o estándares de calidad, son propiedades o cualidades del software para hacerlo funcionar de la manera deseada.

Clasificación de los requerimientos no funcionales

1. Requerimientos Organizacionales.

Tabla 8. Requerimiento organizacionales.

Requerimiento No funcional	Descripción	Prioridad
Respuesta	Tiempo de Respuesta de la Empresa de seguridad	Alta
Fuentes	La Aplicación deberá ser desarrollada en un lenguaje open source.	Alta

2. Requerimiento externo.

Tabla 9. Requerimiento externo.

Requerimiento No funcional	Descripción	Prioridad
Adaptación	Adaptar la solución para implementarla en ambientes que cuenten con conexión de red.	Alta
	Definir la empresa proveedora de conexión a red que mejor servicio tenga en la zona	Alta

3. Requerimientos Orientados al Usuario.

Tabla 10. Requerimientos orientados al usuario.

Requerimiento No funcional	Descripción	Prioridad
Navegabilidad	La interfaz de la solución deberá ser fácil de usar	Media
Registro	El sistema debe permitir el registro y acceso a la plataforma.	Alta

4. Requerimiento de soporte.

Tabla 11. Requerimientos de soporte.

Requerimiento No funcional	Descripción	Prioridad
Garantía	Ofrecer servicio de soporte ante incidentes que puedan llegar a presentarse después de la implementación de la solución.	Media

5. Requerimientos de disponibilidad.

Tabla 12. Requerimientos de disponibilidad.

Requerimiento No funcional	Descripción	Prioridad
Disponibilidad	El sistema deberá tener un disponibilidad del 90% cuando el usuario quiera acceder.	Alta
Mantenimientos	El promedio de duración de fallas e intermitencia no deberá superar los 30 minutos	Media
	El tiempo de ventana para mantenimiento o reinicio del sistema no podrá superar los 20 minutos.	Media

6. Requerimiento de Fiabilidad.

Tabla 13. Requerimientos de fiabilidad.

Requerimiento No funcional	Descripción	Prioridad
Seguridad	Consultar la información que se registre de manera confiable y en tiempo real.	Alta

Acceso	Permitir al usuario acceder a la plataforma con único usuario.	Alta

7. Requerimiento de Usabilidad.

Tabla 14. Requerimientos de usabilidad.

Requerimiento No funcional	Descripción	Prioridad
Usabilidad	La solución debe contar con interfaces intuitivas y recientes, que para el usuario final no sea compleja de interactuar.	Media
	La interfaz debe poder reducir los errores cometidos por el usuario, y alertar de estos, para su corrección..	Media

8. Requerimiento Orientado al Desarrollador.

Tabla 15. Requerimientos orientados al desarrollo.

Requerimiento No funcional	Descripción	Prioridad
	Deberá poder trabajar en equipo y contar con habilidades comunicativas.	Alta
	Deberá tener conocimientos en lenguajes de programación orientado a objetos.	Alta
	Deberá tener conocimientos en diseño web, HTML, JavaScript.	Alta
	Deberá tener conocimientos en metodologías de desarrollo ágil.	Alta

9. Requerimiento de Rendimiento y Escalabilidad.

Tabla 16. Requerimiento de rendimiento y escalabilidad.

Requerimiento No funcional	Descripción	Prioridad
	Adaptar la plataforma en base al crecimiento exponencial continuo sin	Alta

	perder la calidad y su capacidad de reaccionar.	
	Añadir nuevas capacidades de hardware para mejorar el rendimiento.	Media
	Combinar el escalamiento horizontal con el escalamiento vertical para tener mejores resultados, sin perder la funcionalidad y respuesta del sistema.	Alta

8.3. Diseño y arquitectura

A partir de la necesidad que se ha encontrado en Bogotá, enfocados en que se han ido incrementando los robos a inmuebles en los últimos años y apoyados también en que los sistemas integrados de seguridad y servicios de monitoreo son costosos, se ha podido discernir en el mercado de la seguridad en residencias una brecha de negocio para la oferta de servicios de seguridad en propiedades horizontales que ya cuentan con vigilancia privada, donde se puede ofrecer el servicio o solución que se propone, el cual es una plataforma de software asequible y de fácil instalación para la notificación de eventos disparados por sensores de alarmas de seguridad conectados a una red telefónica.

Para dar forma al diseño de la idea, se ha recogido en historias de usuarios, a través de entrevistas con los involucrados en el proceso, todos los detalles funcionales y se ha

entendido el modelo de comportamiento para poder generar la notificación de eventos de seguridad y notificar de manera automática el suceso a la persona interesada.

Desde la comprensión de las historias de usuario y la necesidad del mercado se ha definido los requerimientos funcionales y técnicos a partir de los cuales se construirá la solución tecnológica, la cual será basada en la nube para ser consumida fácilmente por todos los usuarios solo necesitando un dispositivo con conexión a red para acceder.

Se ha decidido trabajar la solución basados en un modelo de microservicios, esto se decide haciendo compartidas de las arquitecturas disponibles actualmente y basados en las que mejor se adaptaran a la solución planteada y que se integra con tecnología y dispositivos de IoT. El desarrollo del sistema se ha acoplado a diferentes soluciones de AWS para tener una mayor robustez y que la solución quede enfocada a tecnología de la nube.

8.4. Diagrama de despliegue

En la figura, se muestra el diagrama de despliegue el cual muestra cuándo y dónde se desplegará el sistema. En él se pueden visualizar cinco (5) partes físicas en forma de nodos y los componentes o artefactos que las componen.

1. Unidad de Usuario

En esta unidad el usuario podrá acceder desde cualquier dispositivo con conexión a la red al sistema en el cual podrá generar las alertas y consultar los reportes de estas.

2. AWS Services

En esta unidad se encuentran los servicios de Amazon Web Services que se han decidido consumir desde la solución para el manejo y envío de las notificaciones de alertas, en este nodo se cuentan con dos artefactos:

3. AWS SES :

Que es el servicio de correo electrónico rentable flexible y escalable que permite envío de correos desde cualquier aplicación de manera segura protegiendo la entrega del mensaje.

4. AWS SNS:

Como sus siglas en inglés lo dicen, servicio de notificación simple o SNS es el servicio de Amazon Web Services para el envío de mensajes de texto, permitiendo a temas crear tópicos y además permite la implementación de funciones lambda para su personalización.

5. Server AWS

Se decide implementar la solución en la nube es por esto que se eligió los servicios de AWS para desplegar la plataforma que contendrá el monitoreo, registro y eventos generados de los dispositivos de IoT relacionados.

Se ha decidido también tener los datos de la plataforma y servicios de autenticación en la nube.

6. MQTT Server

Es el protocolo de comunicación M2M (Máquina a Máquina) de mensajería push con patrón publicador/suscriptor (pub-sub). Se decidió utilizar debido a la facilidad con la

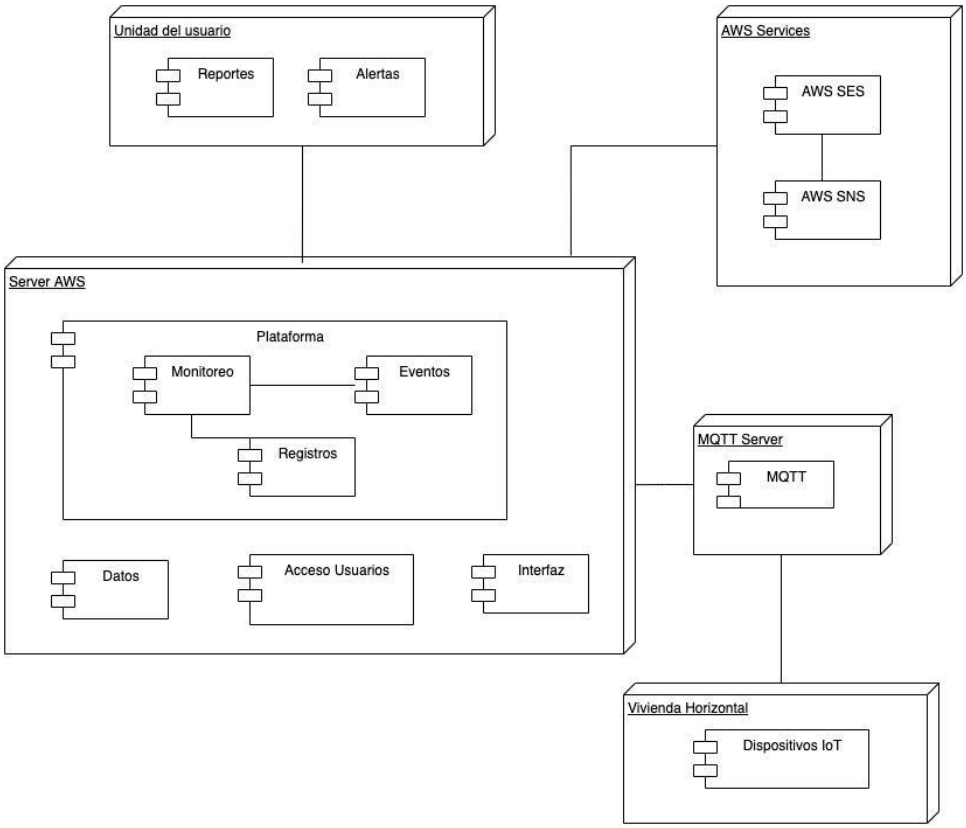
que la solución maneja la comunicación con los dispositivos IoT a través de un Broker y basados en la pila TCP/IP como base de la comunicación.

7. Vivienda Horizontal

Es donde finalmente se instalará el dispositivo de seguridad IoT conectado a la red y que generará los eventos de seguridad.

8.5. Caso de uso Arquitectura relevante

Figura 3. Diagrama de Componentes

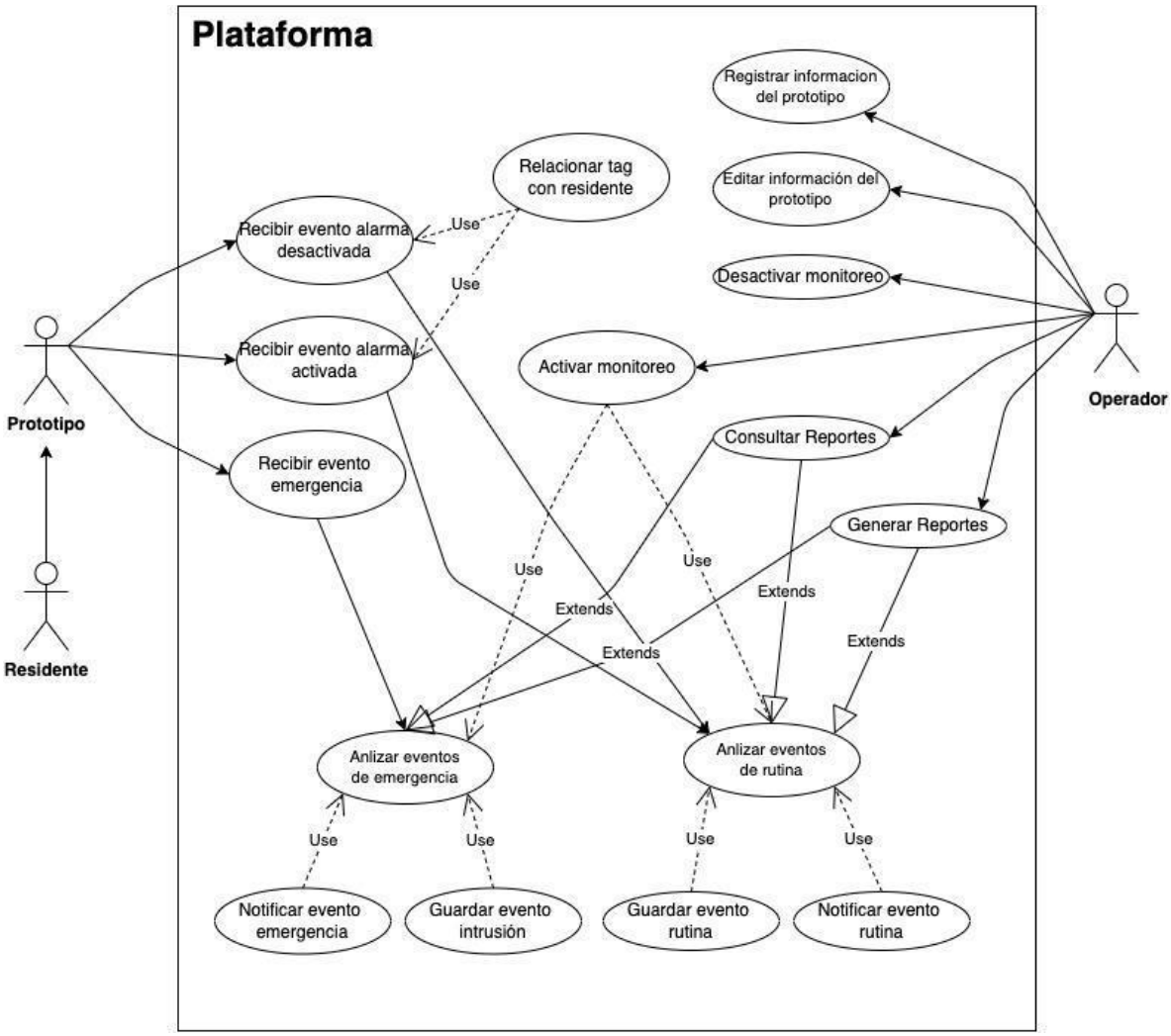


Fuente: Elaboración propia.

En este diagrama se presenta la descripción relevante de las actividades a llevar a cabo durante el proceso así como la identificación de cada uno de los actores y las relaciones

y dependencias que tienen los componentes de la solución. Como es de uso común en este se capturan los requisitos funcionales y se da una idea general del funcionamiento del sistema y como es cada uno de sus partes y que personas lo componen e interactúan en él, todo esto, es plasmado en un lenguaje de fácil comprensión que explica de manera clara el funcionamiento general del sistema.

Figura 4. Diagrama de Casos de Uso

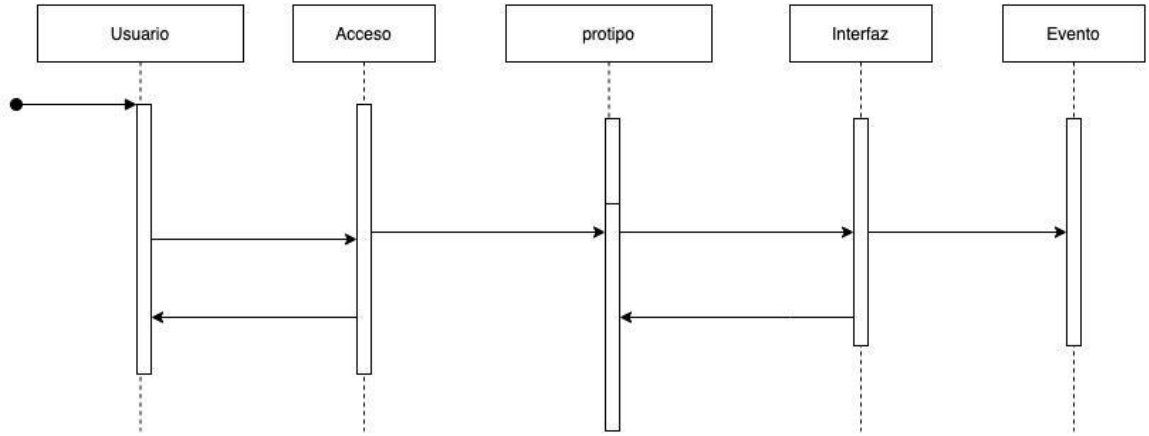


Fuente: Elaboración propia.

8.6. Diagrama de secuencia

Comprende el orden en el que los inventos de una secuencia interactúan, en otras palabras es una herramienta que representa la interacción de los objetos en un sistema en un orden secuencial, se obtiene una mejor visión del proceso de trabajo.

Figura 5. Diagrama de Secuencia



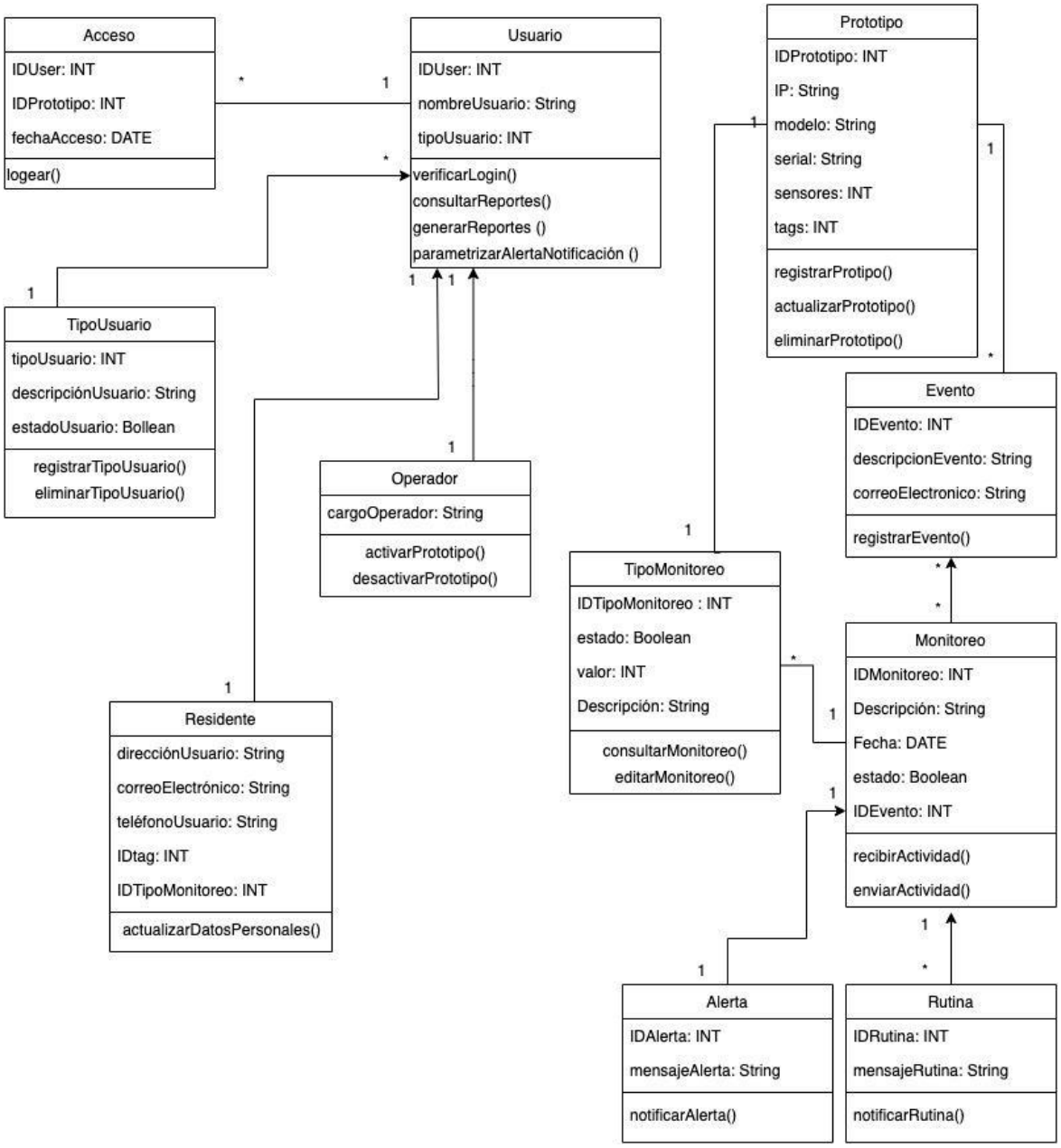
Fuente: Elaboración propia.

8.7. Diagrama de clases

Este diagrama es un tipo de diagrama de estructura, se implementa para representar las partes de un sistema en clases, desde un punto de vista estático, es por esto que este diagrama no incluye la forma como se comportan los elementos a lo largo de su ejecución.

El diagrama es puramente orientado al modelado de programación orientado a objetos, este está compuesto por las clases y representa la manera en la que se relaciónan las mismas.

Figura 6. Diagrama de Clases



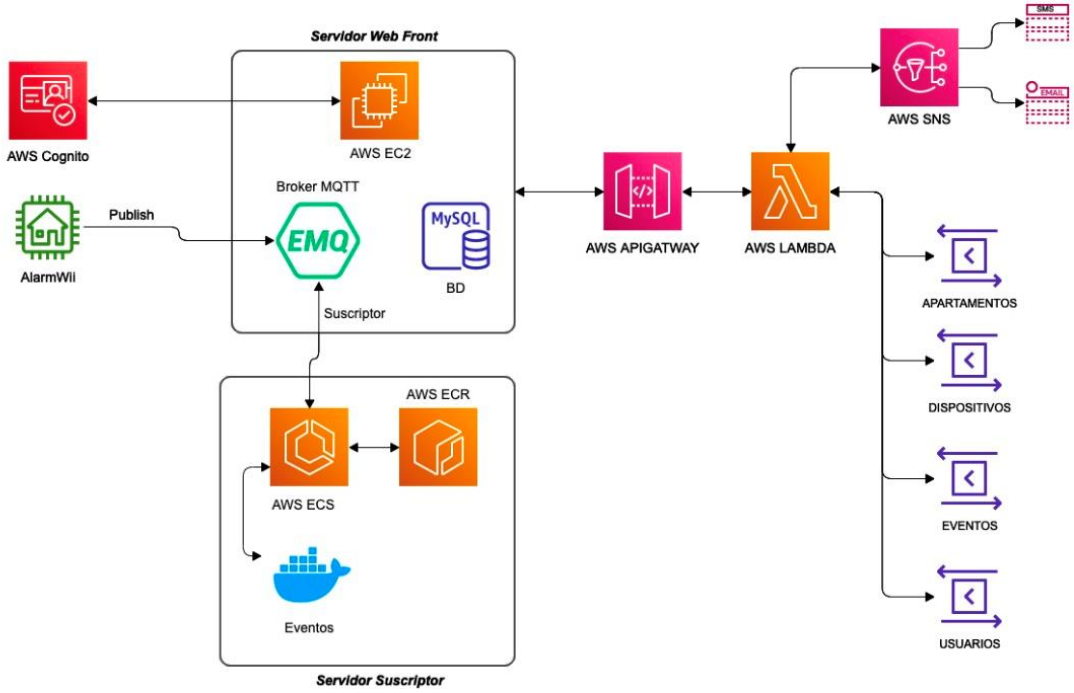
Fuente: Elaboración propia.

8.8. Arquitectura de alto nivel

Con la arquitectura de alto nivel de la figura, se define la estructura general del sistema

donde se identifican componentes clave para el funcionamiento, este diseño se implementó de acuerdo con las arquitecturas tradicionales de soluciones de IoT basados en AWS, se tienen dos servidores uno para la parte front donde está integrado el broker de MQTT y la base de datos, y el otro que es auto escalable para la suscripción de eventos y notificación de estos, haciendo el consumo de las apigateway. Los componentes de SMS y email están contenidos en funciones lambda así como la interacción de la parte backend y almacenamiento en base de datos.

Figura 7. Diagrama Alto Nivel



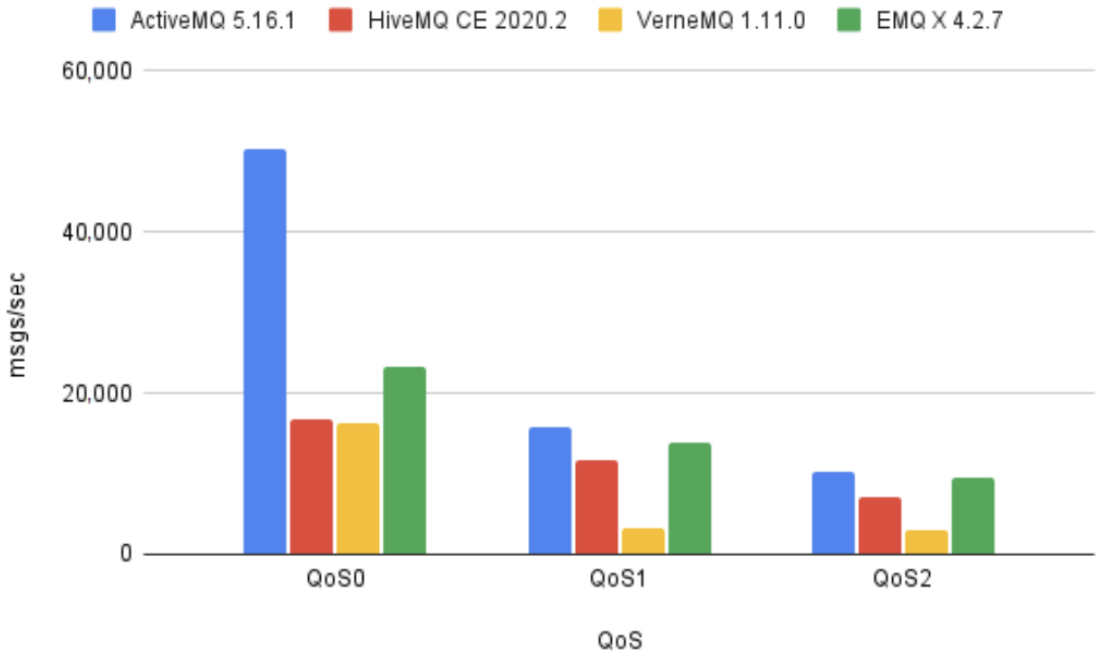
Fuente: Elaboración propia.

9. Construcción

Para abordar la construcción de la plataforma, se inicia por evaluar cómo va a ser la comunicación con los dispositivos o prototipos IoT, hoy en día los dispositivos IoT utilizan el protocolo de comunicación MQTT dado a que es ligero, se basa en una publicación y suscripción, de máquina a máquina, logrando una comunicación rápida, robusta, y sobre todo ligera.

Se realiza la investigación de tecnologías que utilizan los dispositivos IoT con este protocolo y con ayuda del articulo “Stress-Testing MQTT Brokers: A Comparative Analysis of Performance Measurements” (Mishra et al., 2021), se concluye que los dos mejores broker para implementar son ActiveMQ y EMQ X, sin embargo en un ambiente restringido y limitado Mosquito tenía sus ventajas respecto a los dos anteriormente mencionado, pero tenía sus desventajas al momento de ser implementado en un ambiente en la nube, por ello se decide implementar ActiveMQ, pero al momento de hacerlo el tema de configuración y comunicación con los prototipos IoT fue complejo teniendo problemas sin solucionar, se procede a configurar e instalar EMQX siendo exitosa la configuración y comunicación con el prototipo IoT.

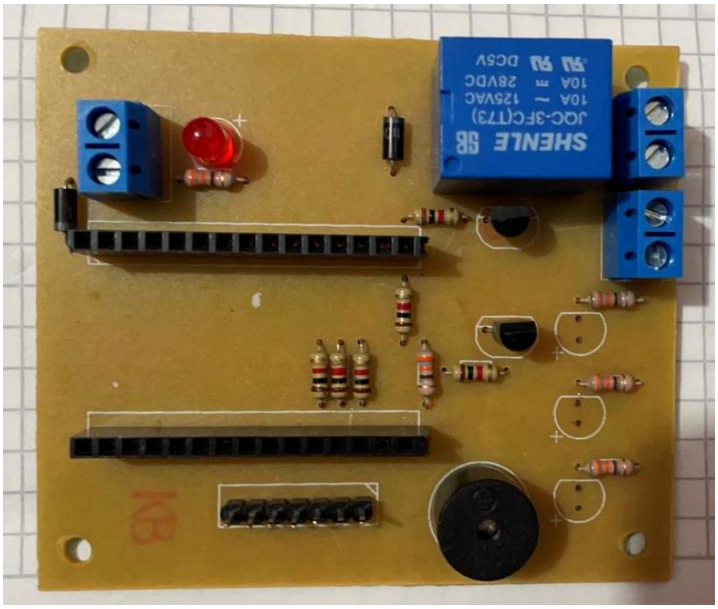
Figura 8. Grafica mensajes protegidos usando diferentes bróker en ambientes de nube..



Fuente: tomada de (Mishra et al., 2021).

Posteriormente, se desarrolla la construcción del prototipo, en una primera etapa se desarrolla con el chip ESP8266, para el armado y desarmado del sistema se utiliza de un teclado numérico y como limitación del prototipo los sensores deben ser cableados y para esta versión solo se contaba con un sensor de movimiento.

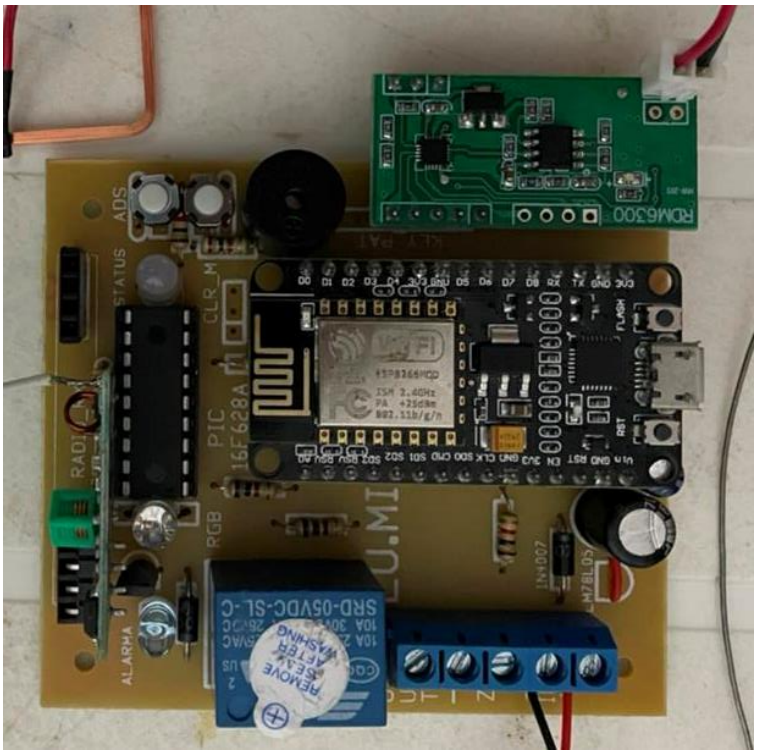
Figura 9. Prototipo IoT primera versión.



Fuente: Elaboración propia.

En una segunda etapa de la construcción del prototipo, se integra un PIC16F628 para la recepción de sensores inalámbricos con la posibilidad de almacenar los ID de los sensores en la memoria interna de este chip, así logrando tener más de un sensor y la recepción de sensores inalámbricos ya se han de movimiento o de apertura. Al tener problemas al momento de ingresar la clave en el teclado numérico, y al hacer la lectura en la ESP8266, se decide cambiar el método de armado y desarmado por un lector RFID de 125 KHz.

Figura 10. Prototipo Chip y memoria interna segunda versión.



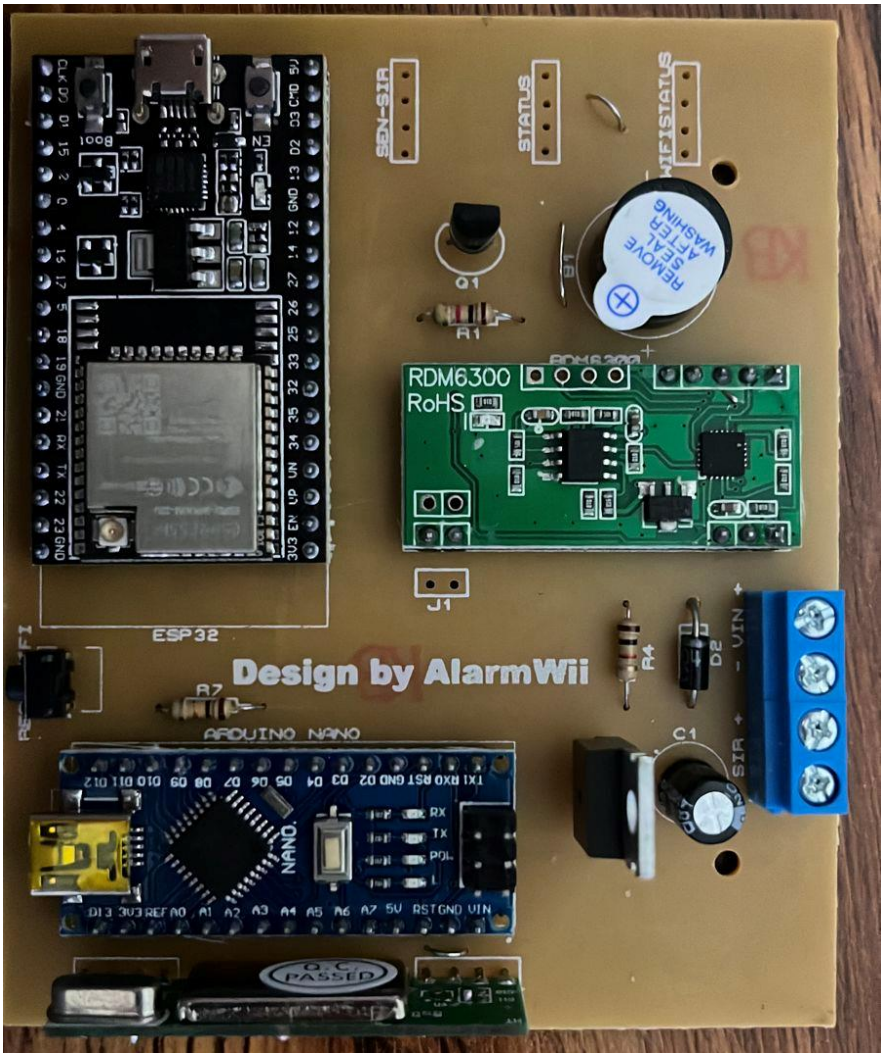
Fuente: Elaboración propia.

En una tercera etapa, en la construcción del prototipo, se decide cambiar chip ESP8266 al ESP32 por temas de rendimiento, se estabiliza la funcionalidad de conexión a WiFi, dado a que la anterior versión al momento de perder conexión con la red, no se restablecia. Se realiza la integración de 3 leds RGB para indicar el estado de la alarma, el estado de la conexión WiFi, y la detección de los sensores o zonas.

Se cambiar el chip PIC16F628 por un Arduino nano para que este solo funcione como receptor, y los transmita a la ESP32 con el fin de incluir funcionalidades que al momento de configurar el dispositivo se puedan escanear los sensores de detección, como integrar controles inalambricos de armado y desarmado, y a la vez funcionalidad de eventos de pánicos desde el control. Se incluye funcionalidad de mejorar al momento de tener

problemas con la red, y se guardan los estados de la alarma, ante un posible escenario de pérdida de energía, se realiza una nueva funcionalidad que le permita al usuario cambiar la red WiFi sin perder la configuración de sensores o tags .

Figura 11. Dispositivo IoT, sensores, tercera versión.



Fuente: Elaboración propia.

Al tener una versión del prototipo estable se procede a implementar el prototipo con el desarrollo de la plataforma. Iniciando por el tema de notificaciones de mensaje de texto (SMS), para esto ya en el servidor configurado en el momento de la instalación del EMQ X alojado en un servidor de AWS, con una configuración t2.micro, con la capacidad de 1 vCPU y 1 GB RAM con almacenamiento de 8 GB y un sistema operativo Ubuntu 18.4.

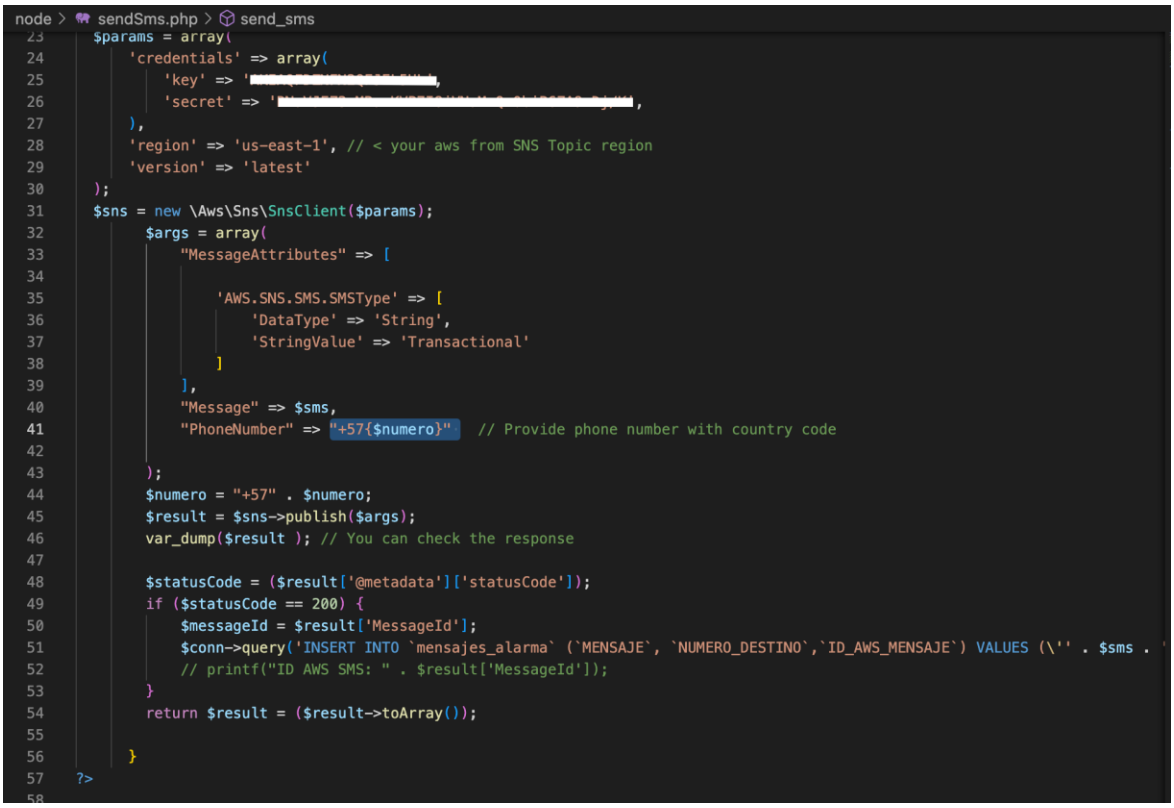
Figura 12. Intancias T2 AWS EC2.

Nombre	vCPU	RAM (GiB)
t2.nano	1	0,5
t2.micro	1	1,0
t2.small	1	2,0
t2.medium	2	4,0
t2.large	2	8,0
t2.xlarge	4	16,0
t2.2xlarge	8	32,0

Fuente: (Instancias T2 de Amazon EC2, n.d.)

Se realiza la configuración y construcción de la base de datos (MySQL 5.7.38), luego de esta configuración se implementa el servicio de AWS SNS de manera local, creando el rol y gestionando los permisos para el servicio de mensajería de texto.

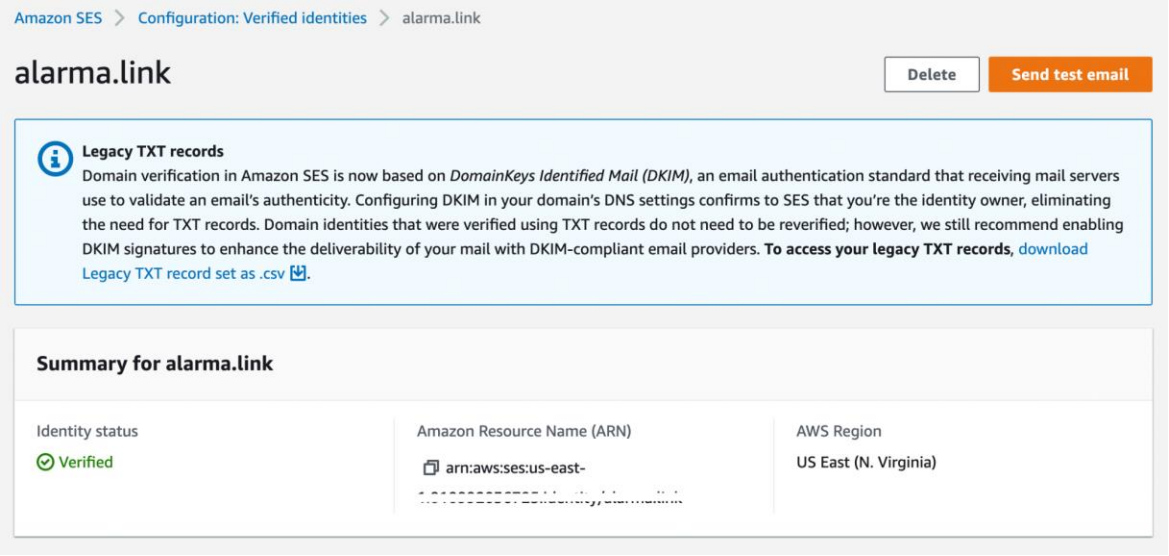
Figura 13. Envió de mensajes



Fuente: Elaboración propia.

Se procede a implementar la notificación por correo electrónico, haciendo uso de los servicios de SNS y habilitando un dominio de correos con el servicio de SES de AWS.

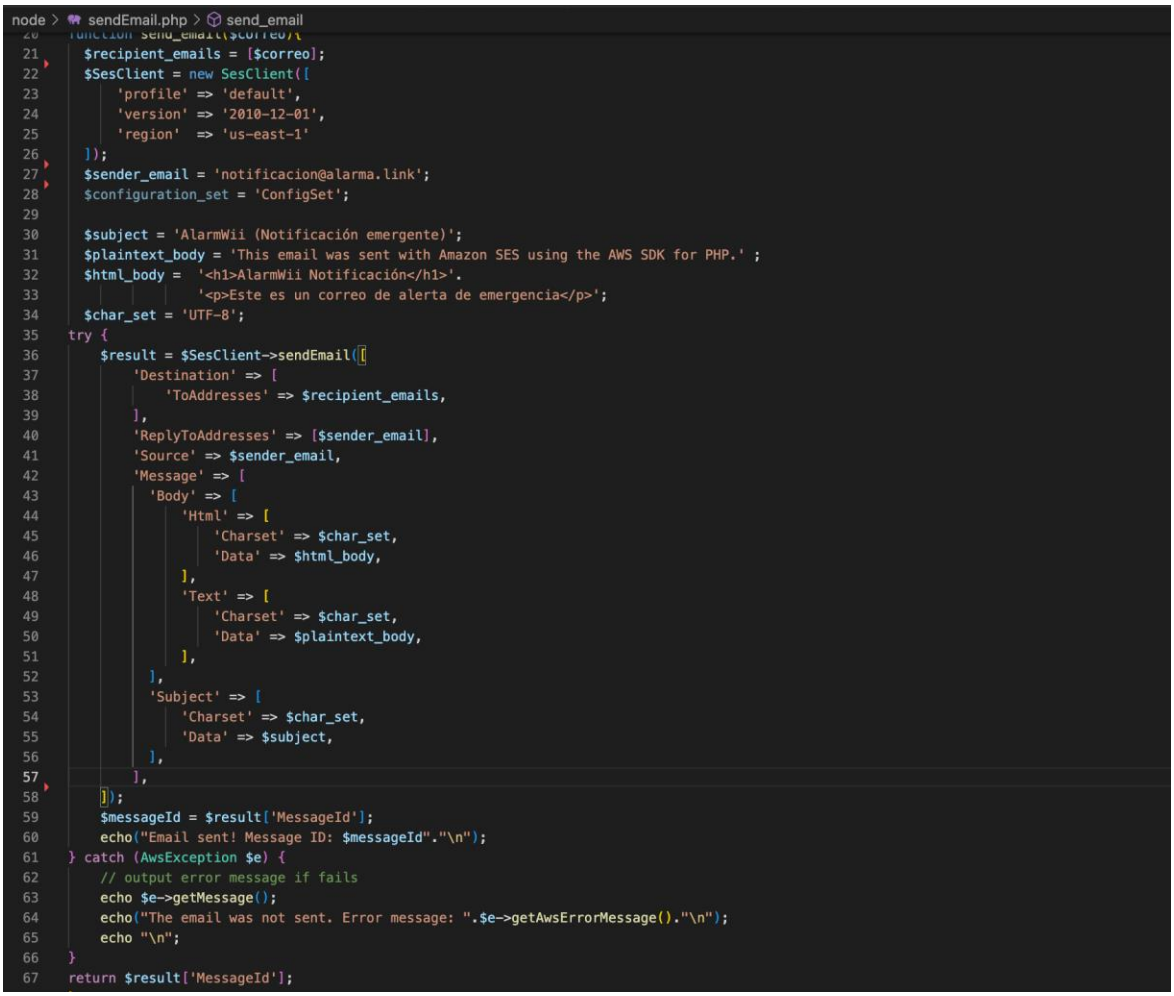
Figura 14. Dominio de correos con el servicio de SES de AWS



Fuente: Elaboración propia.

Se desarrolla el código para el envío de correos con el dominio ya aprobado por AWS SES, a continuación se muestra una imagen del código implementado.

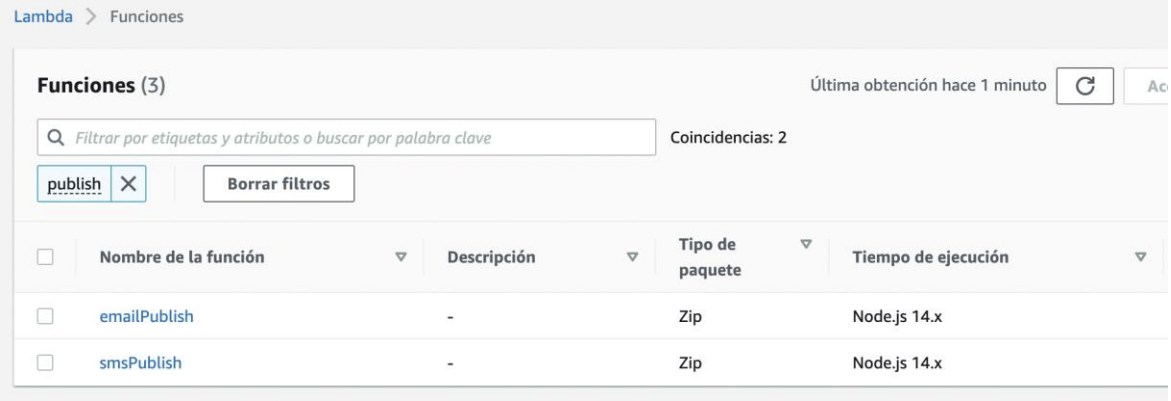
Figura 15. Envío correo



Fuente: Elaboración propia.

Al tener estas dos funcionalidades funcionando de manera local, se migran a AWS Lambda, se tiene dificultades al momento de la implementación dado a que no se tenían conocimientos acerca de Lambda.

Figura 16. Funciones Lambda



Fuente: Elaboración propia.

Se implementa con Node.js y se tiene un desencadenador que se comunica con el API GAWAY, en las siguientes imágenes se muestra el código implementado en lambda y el API.

Figura 17. Código para el envío de mensajes de texto (SMS) en AWS Lambda.

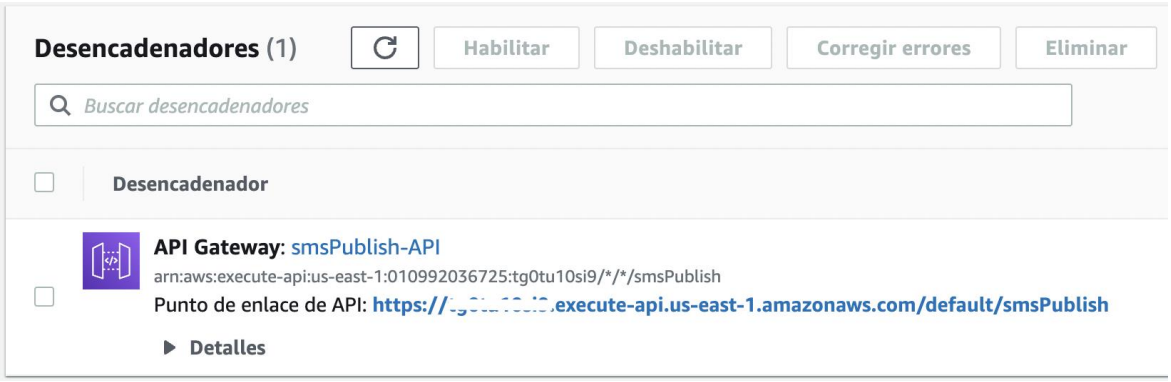
```
if (!body || !body.phoneNumber || !body.message) {
    const response = {
        statusCode: 400,
        body: JSON.stringify('Numero o mensaje vacio'),
    };
    return response;
}
var params = {
    PhoneNumber: body.phoneNumber,
    Message: body.message
};
var obj = { statusSMS: "", phoneNumber: params.PhoneNumber, messageSMS: params.Message, messageId: "" };
// Create promise and SNS service object
var publishTextPromise = new AWS.SNS({ apiVersion: '2010-03-31' }).publish(params).promise();

try {
    // Handle promise's fulfilled/rejected states
    await publishTextPromise.then(
        function(data) {
            console.log("Message ${params.Message} sent to the PhoneNumber ${params.PhoneNumber}");
            console.log("MessageID is " + data.MessageId);
            obj.statusSMS = "Mensaje enviado";
            obj.messageId = data.MessageId;
        }).catch(
        function(err) {
            console.error(err, err.stack);
            obj.statusSMS = err;
        });

    const response = {
        statusCode: 200,
        body: JSON.stringify(obj),
    };
    return response;
}
```

Fuente: Elaboración propia.

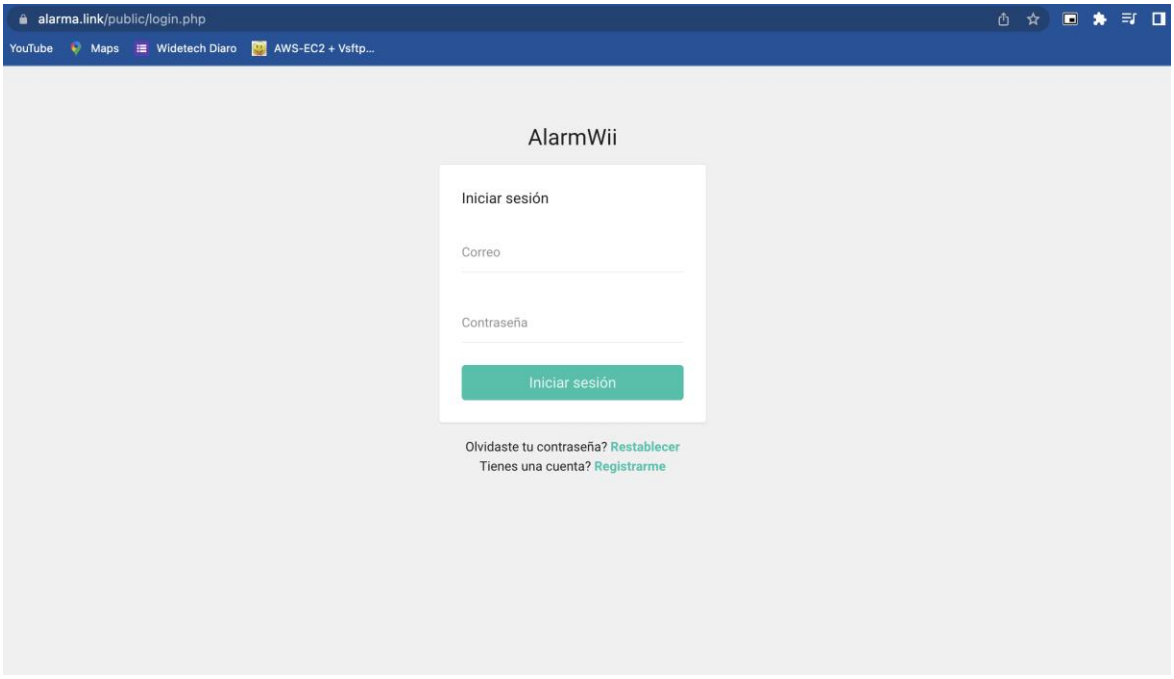
Figura 18. Api Gatway de la publicación de mensajes de texo (SMS).



Fuente: Elaboración propia.

Para la autenticación del sistema, se decide integra con el servicio que ofrece AWS Cognito, para obtener una mayor seguridad al momento de autenticarse y acceder a la plataforma.

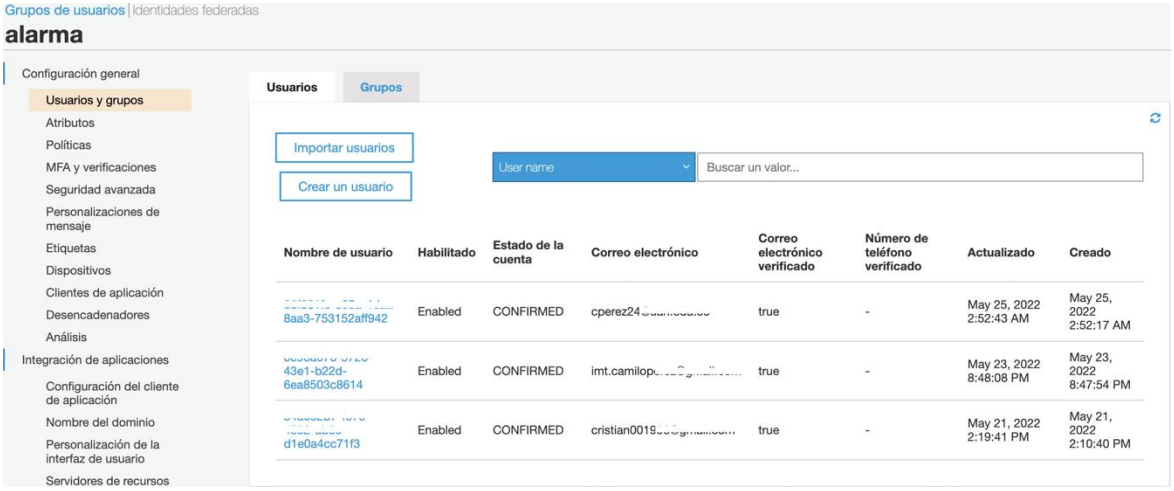
Figura 19. Login de la plataforma.



Fuente: Elaboración propia.

Desde la consola de Cognito se muestran los usuarios registrados y verificados con una autenticación de doble factor del correo registrado.

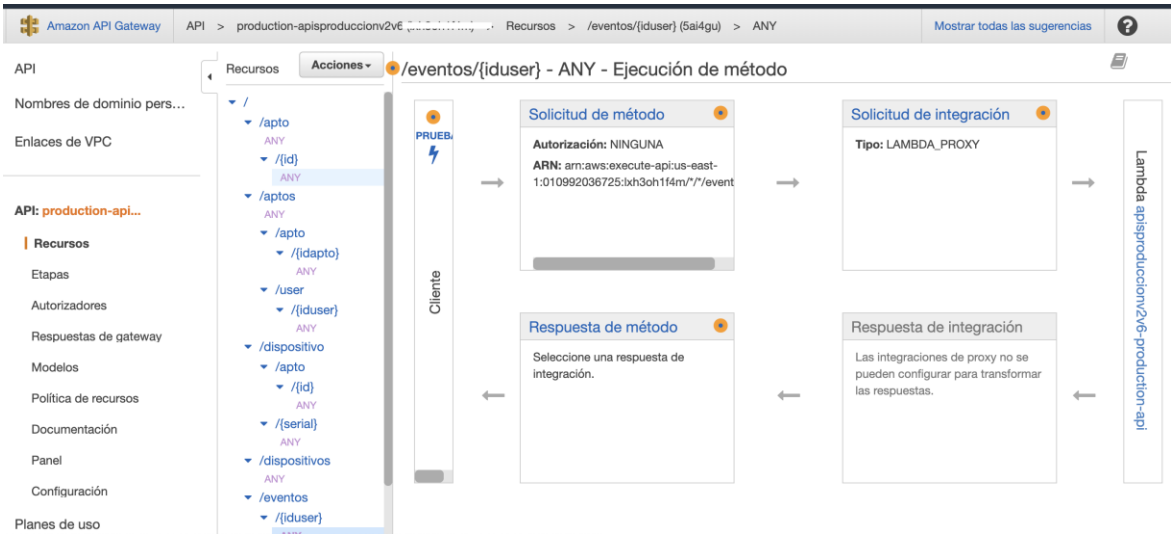
Figura 20. Consola de cognito.



Fuente: Elaboración propia.

Al haber implementado el servicio de cognito y realizar pruebas al módulo, se pasa a construir los microservicios de cada módulo con su lógica de negocio, los cuales son los CRUDs para el módulo de apartamentos, dispositivos, eventos y usuarios. Implentados en las funciones de Lambda, se tuvo problemas al momento de implementarlas en este servicio, dado a que de manera local se habían construido con Node js Expresso, el cual la solución con apoyo de Serverless permite subirlas a Lambda y enlazarlas con API Gateway.

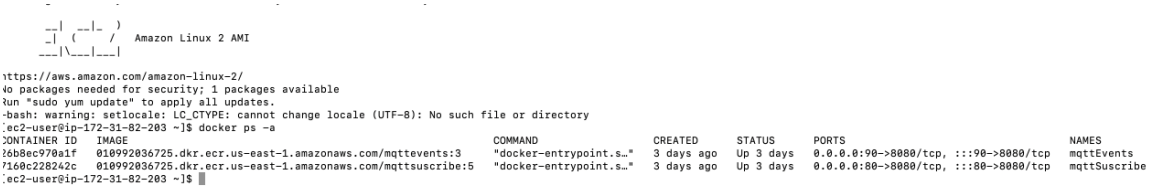
Figura 21. Api de producción para los módulos apartamentos, dispositivos y eventos.



Fuente: Elaboración propia.

Luego de implementar y crear las API, se procede a implementar los microservicios para la suscripción, utilizan las herramientas de Docker con servidores en la nube AWS EC2.

Figura 22. Terminal de la instancia ECS donde se encuentran los dockers.

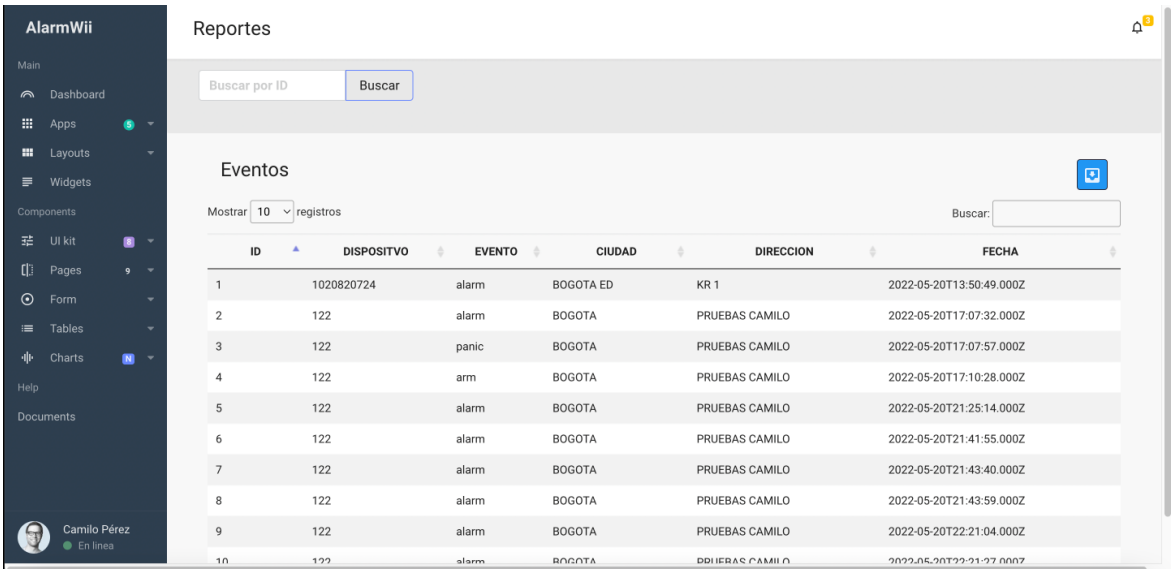


Fuente: Elaboración propia.

En la figura anterior se muestra los Docker containers corriendo, el cual el Docker mqttEvents su tarea es guardar los eventos generados por los dispositivos en la base de datos para realizar y consultar los reportes. Mientras que el Docker container mqttSuscribe se suscribe a los eventos de emergencia y hace el llamado a las funciones en lambda de SMS y correo electrónico.

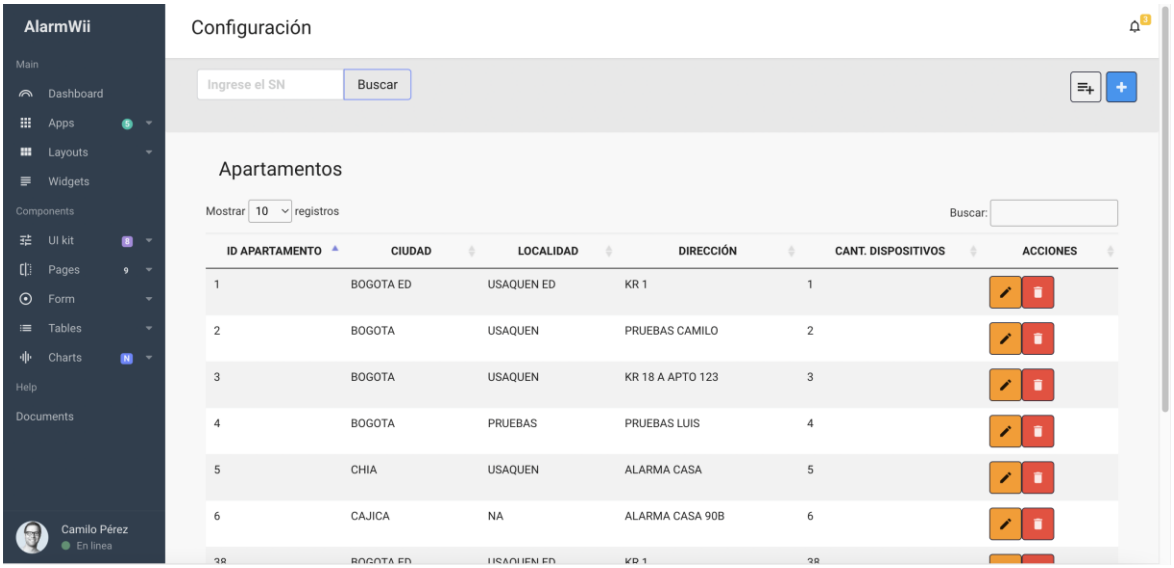
A continuación se muestra la plataforma web, las pantallas de login, crear apartamentos y consultar eventos.

Figura 23. Consola de reportes – Eventos de los dispositviso IoT



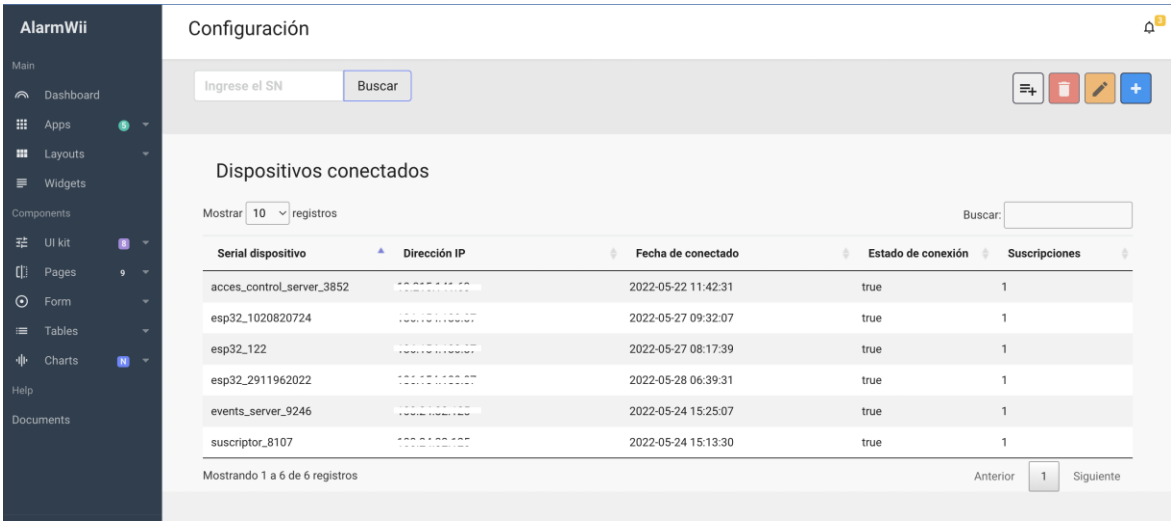
Fuente: Elaboración propia.

Figura 24. Configuración módulo de Apartamentos



Fuente: Elaboración propia.

Figura 25. Vista de dispositivos conectados.



Fuente: Elaboración propia.

10. Pruebas

Las pruebas de software es el proceso de evaluación y verificación de un producto o aplicación de software para saber si hace lo que se supone que debe hacer. Los beneficios de las pruebas incluyen la prevención de errores, la reducción de los costos de desarrollo y la mejora del rendimiento. ”*IBM - United States*. (2015, October 1). Ibm.Com. http://www.ibm.com

Las pruebas de aceptación de usuario son útiles pues ayudan a verificar si todo el sistema funciona de acuerdo a lo previsto, es decir, establecer el grado de confianza en un sistema, en partes del mismo o en sus características no funcionales.

En la medida que las historias de usuario, requerimientos y procesos de negocio solicitados se hayan interpretado y adaptado de manera correcta, más confiable será el software.

Esto quiere decir que a medida que se van liberando los springs de acuerdo al cronograma inicial, se va generando incrementos en el entregable que pueden irse probando por el usuario final, ya que él puede validar si la entrega se adecua a los requerimientos del cliente.

Para este paso se implementó un Script de casos de prueba donde se indicaba al usuario el paso a paso a seguir y la respuesta esperada.

A continuación en la imagen se aprecia el formato implementado para documentar las pruebas de aceptación de usuario.

Figura 26. Formato de documentación de las pruebas de aceptación.

Alarma

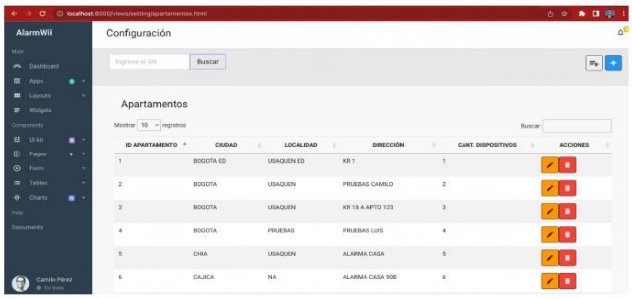
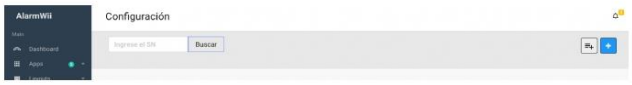
1. Registrar información del apartamento

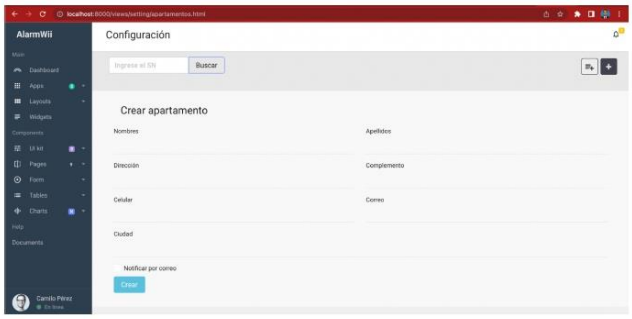
DESAM 05	Workflow Ingreso Puestos de Trabajo
Preparado por:	**Jose Maria Cabrera Agelvis** **Cargo:** **Desarrollador**
Descripción:	Realizar el proceso de registro en la plataforma para apartamentos
Precondiciones:	• Usuario Habilitado en Cognito • Datos del apartamento.
Rol con el que se ejecuta la prueba:	Administrador del Sistema

Datos de entrada: Ciudad, Localidad, Dirección, Conjunto, Celular Notificación, Correo Notificación, Celular Notificación Seguridad

Datos de puesto de trabajo

PASOS Y CONDICIONES DE EJECUCIÓN

Paso	Acción	Resultados esperados
1.	Ingrese al Módulo de Apartamentos 	Ingresar a Apartamentos
1.	Dar clic en el boton azul signo más, para agregar un nuevo apartamento 	Abrir el formulario para el registro del apartamento
1.	Debe diligenciar los datos solicitados por el formulario	Ingresar los datos del apartamento

		
1.	Dar clic en el boton azul “crear”	Registrar los datos del apartamento
COMENTARIOS		
<Observaciones generales del consultor o usuario sobre la ejecución de la prueba>		
Resultado de la prueba	Estado de la prueba	Marque con X el estado que corresponda
	Falla en la ejecución del caso de prueba	
	Test ejecutado con éxito	
	Erróneo, test posterior necesario	
	Test posterior O.K.	
	O.K con mejoras	

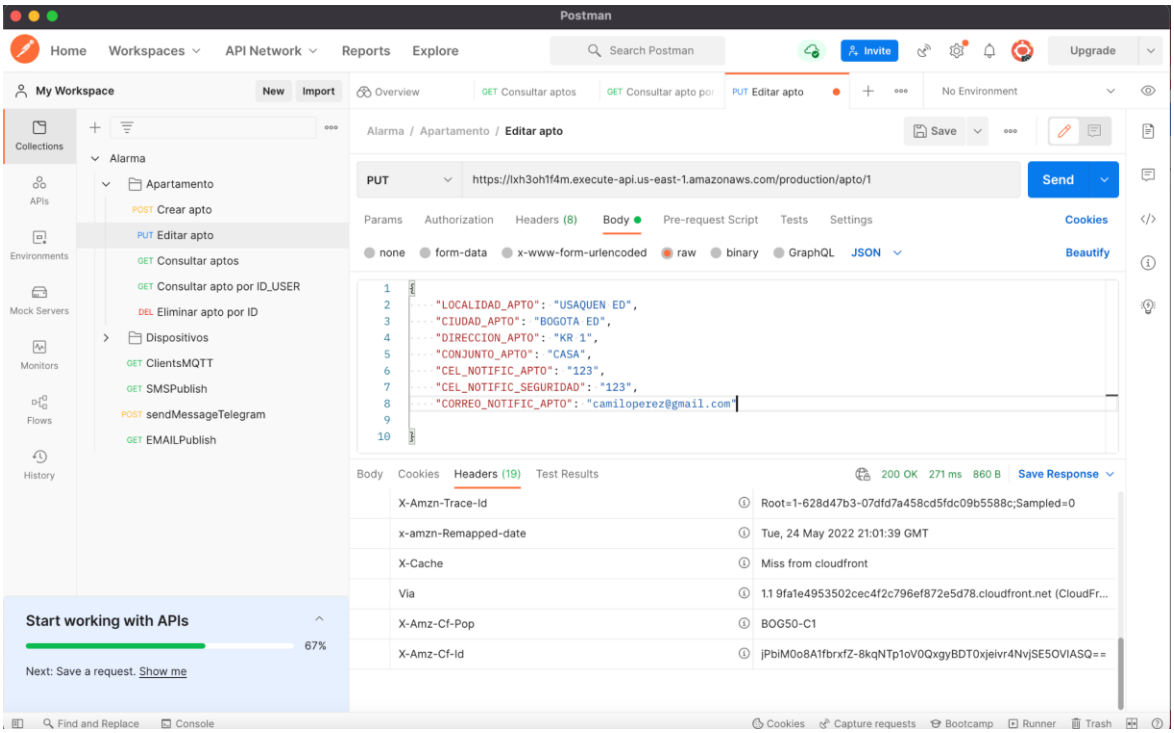
Fuente: Elaboración propia.

Pruebas de integración prueban componentes funcionales del software para comprobar que funcionen correctamente. Las pruebas de integración dentro del software testing chequean la integración o interfaces entre componentes, interacciones con diferentes partes del sistema, como un sistema operativo, sistema de archivos y hardware o interfaces entre sistemas. Las pruebas de integración son un aspecto clave del software testing.

Es esencial que un probador de software tenga una buena comprensión de los enfoques de prueba de integración, para lograr altos estándares de calidad y buenos resultados.

Las pruebas de integración de software se aplican al proyecto para hacer testing de los servicios API que se han implementado bajo funciones lambda en la nube de AWS, para ello se implementa la herramienta POSTMAN. Se ha podido verificar el correcto funcionamiento de las funciones al devolver los mensajes de HTTP requests Estado 200, que es el código de respuesta de estado satisfactorio, esto varía de acuerdo al método de solicitud implementado.

Figura 27. Edición de un apartamento desde postman.



Fuente: Elaboración propia.

Las pruebas de usabilidad están enfocadas en el usuario final y te ayudan a crear la mejor experiencia de usuario (también conocida como UX), en cuanto que obtendrá sus comentarios de forma directa.

La usabilidad indica la facilidad de uso de una herramienta. Abarca tanto la experiencia del usuario como la sencillez para lograr un objetivo por medio de un sistema o dispositivo, por lo que es aplicable tanto a software como a hardware.

basado en los principios de usabilidad se presenta el siguiente cuadro comparativo para determinar el grado de usabilidad de la solución.

Tabla 17. Modelo comparativo de grado de usabilidad.

MODULO:	**INGRESO APARTAMENTO**	
Principio de usabilidad	**Errores**	**Dificultad**
Visibilidad del Estado del Sistema		
Relación entre el sistema y el mundo real		
Control y libertad del usuario		
Estándares y consistencia		
prevención de errores		
Reconocimiento en vez de recuerdo		
Flexibilidad y eficiencia en el uso		
Diseño minimalista y		

estético		
Asistencia a los usuarios para reconocer, diagnosticar y corregir errores		
Ayuda y documentación		

La prueba de usabilidad se aplica en la solución seleccionando un usuario administrador de la aplicación y solicitando que lleve a cabo la tarea que ejecuta en sus competencias laborales en la plataforma mientras que alguien del equipo de diseño y/o desarrollo toma nota de la interacción, particularmente de los errores y dificultades que presente el usuario, esta prueba es aplicable no solo al producto terminado sino que también puede aplicarse a un prototipo del mismo.

11. Instalación y configuración

Para el despliegue del componente web se debe cumplir con los siguientes requerimientos en el servidor:

Requerimientos:

- Server versión >= Apache/2.4.29 (Ubuntu)
- PHP >= 7.2.24
- MySQL Versión del servidor >= 5.7.38
- EMQX versión = 4.1.5
- node versión = 8.10.0

Copiar el contenido del proyecto en las carpetas públicas configuradas en el servidor web, a través de un FTP o git.

1. Instalar VestaCP, el cual nos permite gran variedad de posibilidades de configuración, para poder manejar nuestro servidor web de la mejor manera y de forma gratuita.
2. Descargar script de instalación

   curl -O http://vestacp.com/pub/vst-install.sh

3. Ejecutar instalación sin servidor de correos

   bash vst-install.sh --nginx yes --apache yes --phpfpm no --named yes --remi yes --vsftpd yes --proftpd no --iptables yes --fail2ban yes --quota no --exim no --dovecot no --spamassassin no --clamav no --softaculous no --mysql yes --postgresql no --hostname cursoiot.ga --email correlectronico --password password

4. Configuración de puertos tanto en el firewall como el servidor web

1883,8883,8093,8094,8090,18083,8080

5. Generar certificados de seguridad a través de VestaCP y copiarlos a las siguientes rutas.

ln -s /home/username/conf/web/ssl.website.crt /usr/local/vesta/ssl/certificate.crt

ln -s /home/username/conf/web/ssl.website.key /usr/local/vesta/ssl/certificate.key

6. Descargar y descomprimir emqx

wget https://www.emqx.io/downloads/broker/v4.1.5/emqx-ubuntu18.04-v4.1.5.zip

unzip emqx-ubuntu18.04-v4.1.5.zip

7. Instalar NodeJS

sudo apt update

apt install nodejs

apt install npm

Para el hardware se debe contar con el IDE de Arduino y los códigos fuente para el ESP32 y el arduino nano para la recepción de los sensores de radiofrecuencia y las librerías.

12. Conclusiones

Se desarrolló la plataforma para la gestión y monitoreo de sistema alarmas de seguridad WiFi, con deudas técnicas para la gestión

Se integró la plataforma Web con el prototipo de alarma de seguridad WiFi.

Se recibieron los eventos generados por el hardware IoT, reportando alertas a los interesados.

Se probó satisfactoriamente el funcionamiento de la integración de la plataforma para la gestión y monitoreo del prototipo IoT de seguridad, con al menos 4 prototipos funcionando en campo.

Se generaron alertas de seguridad desde la plataforma para notificar a los interesados de los eventos del dispositivo de IoT.

Se comprobó que la plataforma envía las notificaciones, donde se recibieron 90% de las alertas cuando el prototipo se encontraba conectado a WiFi.

13. Referencias Bibliográficas

Alarma Monitoreada para el Hogar. (s/f). Com.co. Recuperado el 4 de mayo de 2022, de https://www.prosegur.com.co/alarma/monitoreada-hogar-inalambrica-sistema-seguridad

Bogue, R. (2014). Towards the trillion sensors market. *Sensor Review*, *34*(2), 137–142. https://doi.org/10.1108/sr-12-2013-755

Kit Básico LS-10. (s/f). Technoimport. Recuperado el 4 de mayo de 2022, de https://www.technoimport.com.co/pagina-del-producto/kit-b%C3%A1sico-ls-10

Oracle business intelligence sign in. (s/f). Gov.Co. Recuperado el 8 de mayo de 2022, de https://analitica.scj.gov.co/analytics/saw.dll?Portal

Residential Security Systems. (s/f). FSS Technologies (Staging). Recuperado el 4 de mayo de 2022, de https://www.fsstechnologies.com/security/residential

Metodología Gestión de Requerimientos. (s/f).Recuperado el 5 de mayo de 2022, de https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales

Aytac, H. (2022, abril 14). *IoT - What is the Internet of Things*. MD - Best Engineering Blog; Mechanicalland Content Editor. https://mechanicalland.com/iot-what-is-the-internet-of-things/

OWASP Internet of Things. (s/f). Owasp.Org. Recuperado el 27 de mayo de 2022, de https://owasp.org/www-project-internet-of-things/

What is AWS. (s/f). Amazon.com. Recuperado el 14 de mayo de 2022, de https://aws.amazon.com/es/what-is-aws/

Amazon SNS. (s/f). Amazon.com. Recuperado el 28 de mayo de 2022, de https://aws.amazon.com/es/sns

Aws ECS. (s/f). Amazon.com. Recuperado el 28 de mayo de 2022, de https://aws.amazon.com/es/ecs/features/

AWS Lambda. (s/f). Amazon.com. Recuperado el 28 de mayo de 2022, de https://aws.amazon.com/es/lambda/

What is Amazon Cognito? (s/f). Amazon.com. Recuperado el 28 de mayo de 2022, de https://docs.aws.amazon.com/cognito

¿Qué son las pruebas de integración de software(s/f). Trans Ti. Recuperado el 28 de mayo de 2022, de https://trans-ti.com/2020/12/14/que-son-las-pruebas-de-integracion-en-el-sofware-testing/