



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**DISEÑO E IMPLEMENTACION DE LA APLICACIÓN ALERTAUSAC PARA PUBLICACIÓN
DE EVENTOS Y ALERTAS EN EL CENTRO UNIVERSITARIO METROPOLITANO,
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, ACCESIBLE DESDE TELÉFONOS CON
PLATAFORMA ANDROID**

Manuel Antonio Fuentes Fuentes

Asesorado por el Ing. Herman Igor Veliz Linares

Guatemala, febrero de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO E IMPLEMENTACION DE LA APLICACIÓN ALERTAUSAC PARA
PUBLICACION DE EVENTOS Y ALERTAS EN EL CENTRO UNIVERSITARIO
METROPOLITANO, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA,
ACCESIBLE DESDE TELEFONOS CON PLATAFORMA ANDROID**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

MANUEL ANTONIO FUENTES FUENTES
ASESORADO POR EL ING. HERMAN IGOR VELIZ LINARES

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, FEBRERO DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Vladimir Armando Cruz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. César Rolando Batz
EXAMINADORA	Inga. Devora Emperatris Meza
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO E IMPLEMENTACION DE LA APLICACIÓN ALERTAUSAC PARA PUBLICACIÓN DE EVENTOS Y ALERTAS EN EL CENTRO UNIVERSITARIO METROPOLITANO, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, ACCESIBLE DESDE TELÉFONOS CON PLATAFORMA ANDROID

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha septiembre de 2020.



Manuel Antonio Fuentes Fuentes

Guatemala, 30 de agosto de 2020

Ing. Carlos Azurdia
Escuela de Ciencias y Sistemas
Facultad de Ingeniería USAC

Respetable Ing. Azurdia:

Por medio de la presente hago de su conocimiento que el estudiante **Manuel Antonio Fuentes Fuentes** con Registro Académico **201213580** y documento personal de identificación **2433436551202**, de la carrera de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería, USAC. Culmino satisfactoriamente el contenido de su trabajo de graduación titulado "**DISEÑO E IMPLEMENTACION DE LA APLICACIÓN ALERTAUSAC PARA PUBLICACION DE EVENTOS Y ALERTAS EN EL CENTRO UNIVERSITARIO METROPOLITANO, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, ACCESIBLE DESDE TELEFONOS CON PLATAFORMA ANDROID**", por lo tanto, considero que se puede proseguir con la siguiente fase del proceso de graduación.

Sin otro particular, me es grato suscribirme.

Atentamente,



Ing. Herman Igor Veliz Linares
COLEGIADO No. 4836

-Ing. Herman Igor Veliz Linares
Docente Escuela de Ciencias y Sistemas
Facultad de Ingeniería, USAC
Colegiado No. 4836
Herman.veliz@gmail.com
5402-2579



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 1 de septiembre de 2020

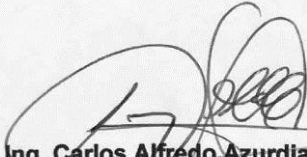
Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **MANUEL ANTONIO FUENTES FUENTES** con carné **201213580** y CUI **2433 43655 1202** titulado **“DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN ALERTAUSAC PARA PUBLICACIÓN DE EVENTOS Y ALERTAS EN EL CENTRO UNIVERSITARIO METROPOLITANO, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, ACCESIBLE DESDE TELEFONOS CON PLATAFORMA ANDROID”** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **"DISEÑO E IMPLEMENTACION DE LA APLICACIÓN ALERTAUSAC PARA PUBLICACIÓN DE EVENTOS Y ALERTAS EN EL CENTRO UNIVERSITARIO METROPOLITANO, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, ACCESIBLE DESDE TELÉFONOS CON PLATAFORMA ANDROID"**, realizado por el estudiante, MANUEL ANTONIO FUENTES FUENTES aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAR A TODOS"

A handwritten signature in black ink over a circular official stamp. The stamp contains the text 'UNIVERSIDAD DE SAN CARLOS DE GUATEMALA' and 'ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS'.

Msc. Carlos Gustavo Alonzo
Director

Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 18 de febrero de 2021



**Decanato
Facultad de Ingeniería
24189101- 24189102**

DTG. 050.2021.

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DISEÑO E IMPLEMENTACION DE LA APLICACIÓN ALERTAUSAC PARA PUBLICACIÓN DE EVENTOS Y ALERTAS EN EL CENTRO UNIVERSITARIO METROPOLITANO, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, ACCESIBLE DESDE TELÉFONOS CON PLATAFORMA ANDROID**, presentado por el estudiante universitario: **Manuel Antonio Fuentes Fuentes**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Anabela Cordova Estrada
Decana

Guatemala, febrero de 2021.

AACE/asga

ACTO QUE DEDICO A:

Mis padres

Victor Fuentes y Gabriela Fuentes, por apoyarme en todo lo que he realizado, por ser un ejemplo a seguir, por sus consejos y cariño en cada una de las etapas de mi vida.

Mis hermanos

Andrea y Carlos Fuentes, por los consejos, risas, regaños y buenos momentos que hemos pasado juntos.

Mi sobrina

Andrea Valentina Calderón, por esos momentos de diversión y aprendizaje.

Mis abuelos

Por guiarme a ser una persona de bien, por los consejos que me dieron y el apoyo en todo momento.

Mis tíos

Por brindarme su apoyo, consejos a lo largo de mi vida.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por haberme proporcionado los conocimientos y experiencia para mi desarrollo personal y profesional.
Facultad de Ingeniería	Por ser una importante influencia en mi carrera, entre otras cosas.
Mis amigos	Brayan González, Ricky Yancor, Fernando Santos, Brian Gonzalez, por compartir buenos momentos y tener gratos recuerdos.
Herman Veliz	Por su apoyo para llevar a cabo este proyecto.
Juan Orlando López	Por apoyarme en mi formación académica.
Milton Fuentes	Por brindarme su apoyo y confianza para llevar a cabo este proyecto.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS.....	IX
GLOSARIO.....	XI
RESUMEN.....	XIII
OBJETIVOS	XV
INTRODUCCIÓN.....	XVII
1. MARCO TEORICO.....	1
1.1. Aplicación móvil.....	1
1.1.1. Orígenes de las aplicaciones móviles	1
1.1.2. Sistemas operativos para dispositivos móviles.....	2
1.1.2.1. Sistema operativo Android	2
1.1.2.2. Sistema operativo IOS	3
1.1.3. Evolución	3
1.1.4. Flutter Framework.....	4
1.2. Métodos de comunicación	6
1.2.1. Web services	7
1.2.2. RESTFUL Web Services.....	8
1.2.3. NodeJS Framework	10
1.3. Servicio en la nube	13
1.3.1. Google Cloud.....	14
1.3.2. Play Store	15
1.4. Sistemas de localización.....	17
1.4.1. Sistemas de corto alcance Bluetooth	17
1.4.2. Sistemas de largo alcance	18

1.4.3.	GPS.....	18
1.5.	Metodología de desarrollo	20
1.5.1.	Proceso scrum	20
1.5.2.	Beneficios de las metodologías de desarrollo.....	22
2.	DESCRIPCIÓN DEL PROBLEMA Y PLANTEAMIENTO DE LA SOLUCIÓN.....	25
2.1.	Descripción del problema	25
2.1.1.	Antecedentes de la solución	26
2.2.	Planteamiento de la solución.....	31
2.2.1.	Descripción de la aplicación	31
2.2.1.1.	Alertas y eventos	32
2.2.1.2.	Descripción de los roles.....	33
2.2.1.3.	Alcances de la aplicación.....	34
2.2.1.4.	Límites de la aplicación.....	35
3.	DISEÑO E IMPLEMENTACIÓN DE LA SOLUCION	37
3.1.	Diseño de pantallas.....	37
3.2.	Arquitectura del sistema	50
3.3.	Diseño de la base de datos	51
3.3.1.	Conjunto de relaciones.....	52
3.4.	Diseño de comunicación entre componentes	54
3.5.	Usabilidad	55
4.	DOCUMENTACIÓN DEL SISTEMA	57
4.1.	Manual de usuario.....	57
4.1.1.	Descargar aplicación	57
4.1.2.	Registro en el sistema	58
4.1.3.	Inicio de sesión.....	59
4.1.4.	Recuperar contraseña	60
4.1.5.	Publicación de alertas y eventos	63
4.1.6.	Administración de publicaciones.....	67

4.1.7.	Números de emergencia	70
4.1.8.	Perfil usuario	72
4.1.9.	Estadísticas	73
4.2.	Manual técnico	76
4.2.1.	Requerimientos mínimos	76
4.2.1.1.	Requerimientos recomendados para la instalación de la aplicación	76
4.2.1.2.	Versión de herramientas	76
4.2.1.3.	Paquetes instalados flutter	77
4.2.1.4.	Servidor Google Cloud	77
4.2.2.	Gestión de archivos	78
4.2.3.	Servicios principales	79
CONCLUSIONES		83
RECOMENDACIONES		85
BIBLIOGRAFÍA.....		87
APÉNDICE		89

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Arquitectura flutter	5
2.	Canales de comunicación flutter	6
3.	Arquitectura nodejs	10
4.	Arquitectura convencional síncrona	11
5.	Arquitectura servidor nodejs.....	12
6.	Clases de bluetooth	18
7.	Distribución de satélites gps.....	19
8.	Ciclo de vida metodología scrum	22
9.	Botón emergencia	27
10.	Mapa estaciones de emergencia.....	27
11.	Sitio web app usacenlinea.....	28
12.	Banner publicidad app espanta cacos.....	29
13.	Actividad académica eras	30
14.	Pantalla inicio sesión.....	37
15.	Pantalla de registro	38
16.	Pantalla mapa interactivo	39
17.	Pantalla crear evento	40
18.	Pantalla crear alerta	41
19.	Pantalla perfil usuario.....	42
20.	Pantalla editar usuario	43
21.	Pantalla lista de publicaciones	44
22.	Pantalla lista números de emergencia.....	45
23.	Menú usuario administrador	46

24.	Menú usuario registrado	47
25.	Menú usuario invitado.....	48
26.	Estadísticas	49
27.	Arquitectura del sistema	51
28.	Modelo entidad relación de la aplicación ALERTAUSAC.....	53
29.	Pantalla principal	58
30.	Pantalla registro.....	59
31.	Pantalla principal	60
32.	Pantalla principal	61
33.	Pantalla recuperar	62
34.	Pantalla mapa.....	63
35.	Pantalla mapa.....	64
36.	Pantalla evento.....	65
37.	Pantalla tipo evento	65
38.	Pantalla alerta.....	66
39.	Pantalla tipo alerta	66
40.	Menú aplicación.....	67
41.	Pantalla publicaciones	68
42.	Pantalla ver publicación.....	69
43.	Menú aplicación.....	70
44.	Pantalla emergencia	71
45.	Menú aplicación.....	72
46.	Pantalla perfil.....	73
47.	Menú aplicación.....	74
48.	Pantallas estadísticas	75
49.	Gestión de archivos	78
50.	Servicio para obtener publicaciones generales.....	80
51.	Servicio obtener publicaciones usuarias	80
52.	Servicio foto perfil	81

53.	Servicio registrar publicación	81
-----	--------------------------------------	----

TABLAS

I.	Entidades aplicación ALERTAUSAC.....	53
II.	Estándar rutas servicio usuario	54
III.	Estándar rutas servicio publicaciones	55
IV.	Estándar rutas servicios especiales	55
V.	Paquetes instalados flutter	77

LISTA DE SÍMBOLOS

Símbolo	Significado
XML	Extensible Markup Language
JSON	Java Script Object Notation
REST	Representational State Transfer
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

GLOSARIO

<i>Backend</i>	Capa de acceso a datos que lleva a cabo toda la lógica de negocio.
CUM	Centro Universitario Metropolitano de la Universidad de San Carlos de Guatemala.
<i>Framework</i>	Entorno de trabajo con una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software.
<i>Frontend</i>	Se dará si el concepto es demasiado largo por lo que deberá corregir los tabuladores o utilizar su ingenio.
GPS	Sistema de posicionamiento global (en inglés, GPS: global positioning system).
Json	Java Script Object Notation, el cual sirve para el envío y recepción de datos.
SDK	Es un kit de desarrollo de software (Software Development Kit) que reúne diferentes herramientas de desarrollo para uso de los desarrolladores.

<i>Tracking</i>	Sistema de seguimiento para observar personas u objetos en movimiento y proporcionar una secuencia ordenada y oportuna de datos de ubicación, para su posterior procesamiento.
Uris	Uniform Resource Identifier es una cadena de caracteres que identifica los recursos de una red de manera única.
Url	Uniform Resource Locator es una secuencia estándar de caracteres que identifica y permite localizar y recuperar información de internet.
USAC	Universidad de San Carlos de Guatemala.
<i>Widget</i>	Unidad básica para construir interfaces de usuario en el Framework Flutter.

RESUMEN

El principal objetivo de la aplicación es crear un medio de difusión de publicaciones para toda la comunidad estudiantil, claustro de catedráticos, personal administrativo y público en general, que permita mantenerlos al tanto de todas las alertas y eventos que se desarrollen en el Centro Universitario Metropolitano, podrán visualizarlas por medio de un mapa interactivo donde podrán interactuar con todas las actividades que ingresen, tanto académicas y recreativas como alertas que surjan dentro del centro universitario.

Los usuarios tendrán la capacidad de administrar las publicaciones que realicen y visualizar las publicaciones de otros usuarios; para ello tendrán que realizar el registro en la aplicación, lo que les permitirá tener acceso a todas las funcionalidades. Si un usuario no se registra podrá ingresar a la aplicación como un usuario invitado y podrá interactuar con las publicaciones, pero no crear una nueva publicación.

Dado que se centralizan todas las publicaciones se manejará un rol administrador. Este tendrá como función la gestión de las publicaciones, lo que le permitirá editarlas y borrarlas, según el criterio del mismo.

Al ser una aplicación que está planificada para ser escalable se detallan cada uno de los aspectos técnicos que permitirán dar un mejor contexto del desarrollo de la aplicación, como las tecnologías implementadas, así como la arquitectura que se utilizó para garantizar que se puedan agregar más funcionalidades.

OBJETIVOS

General

Diseñar e implementar un software innovador y funcional que permita mejorar la información y monitoreo de los eventos que surjan dentro del Centro Universitario Metropolitano (CUM) de la Universidad de San Carlos de Guatemala (USAC), así como ver, publicar distintas alertas y eventos a través de los teléfonos inteligentes con sistema operativo Android que permiten utilizar el sistema de posicionamiento global (GPS Global Positioning System).

Específicos

1. Diseñar un software en donde se visualice la ubicación de los eventos y alertas en el teléfono Android para tener una mejor perspectiva dentro del centro universitario.
2. Centralizar el flujo de comunicación de los eventos y alertas que suceden dentro del centro universitario.
3. Determinar la arquitectura óptima para implementar el backend de una aplicación que utilice el sistema de posicionamiento global.

INTRODUCCIÓN

La Universidad de San Carlos de Guatemala, al ser la universidad con la mayor cantidad de estudiantes de Guatemala, cuenta con diferentes centros distribuidos en distintos puntos del país, con el fin de cubrir las necesidades académicas que demandan los estudiantes. Uno de ellos es el centro universitario metropolitano (CUM), el cual se ubica en la zona 11 de la ciudad capital. Cuenta con una infraestructura utilizada por el área académica, administrativa, recursos estudiantiles, servicios, áreas deportivas, áreas de investigación, entre otros. Dentro de la misma hay módulos que son puntos de referencia, como los edificios y la biblioteca, los cuales albergan actividades culturales, actos académicos, de formación estudiantil y son de gran afluencia de personas.

Dado que el CUM es visitado por cientos de personas diariamente, es necesario que dé un paso hacia la tecnología y provea a quienes ingresan no solo de los servicios básicos que tiene, sino también de una herramienta que promueva las actividades y eventos que se realicen dentro de las instalaciones y permita crear alertas, para que las personas puedan conocer los eventos que están en proceso.

Es por ello que se presenta el desarrollo de la aplicación (app) ALERTAUSAC, que sea accesible gratuitamente a todas las personas que ingresan diariamente al CUM. En el presente documento se muestra una visión general del software a desarrollar, el cual consiste en una app para teléfonos con plataforma Android que le permita a cualquier estudiante, docente o personal administrativo, que posea un teléfono inteligente, ver y publicar un evento o alerta

que suceda dentro del CUM. Se podrá así notificar a las personas acerca de actividades culturales, actos académicos o de formación estudiantil en una sola app, así como alertar de incidentes y eventos que ocurran, como accidentes vehiculares, robos, emergencias médicas, todo de manera inmediata por medio de acceso a internet.

Esta aplicación no solo ayudará a los estudiantes, docentes y personal administrativo a la notificación de eventos y alertas, sino que permitirá también conocer la localización de los distintos módulos que existen dentro de la universidad y fortalecer la comunicación, información y seguridad de cada una de las personas.

1. MARCO TEORICO

En este capítulo se definirán conceptos claves que nos permitirán definir una base teórica para el desarrollo de la aplicación ALERTAUSAC; así como presentar la fundamentación para plantear la solución y su posterior implementación.

1.1. Aplicación móvil

Los dispositivos móviles han tenido una gran evolución e importancia en la última década, por lo que vamos a analizar la evolución que han tenido.

1.1.1. Orígenes de las aplicaciones móviles

Los orígenes de las aplicaciones móviles se remontan a la década de los años noventa cuando se desarrollaron las primeras aplicaciones para teléfonos celulares o móviles las cuales se centraban en videojuegos, calendario, agendas o tonos de llamadas. Estos teléfonos móviles se caracterizaban por tener pantallas reducidas y la mayoría con botones para interactuar con el contenido. La primera aplicación para teléfonos celulares se desarrolló en 1994 a cargo de una manufactura danesa, la cual se trataba del popular juego Tetris. En los años posteriores se fueron desarrollando distintas aplicaciones creando una competencia entre las compañías líderes en desarrollo y producción de teléfonos móviles, tales como Nokia, Ketchapp. Pero el mayor avance se logró en el año 2008 con el lanzamiento del App Store de Apple, la publicación del primer SDK para Android y el lanzamiento de Android Market (Google Play Store), teniendo a la cabeza en el desarrollo a empresas como Apple y Google, los cuales, hasta

el día de hoy son los principales referentes en el desarrollo de aplicaciones móviles.

1.1.2. Sistemas operativos para dispositivos móviles

Hoy existe un gran número de sistemas operativos desarrollados para dispositivos móviles. Algunos han tenido éxito y otros no han logrado tener una base de usuarios grande para mantener el número de nuevas aplicaciones. Entre las empresas que hoy tienen la mayor cantidad de dispositivos con sus sistemas operativos podemos mencionar a Google, con el sistema operativo Android, y Apple con IOS. Estas dos empresas abarcan más del 90 % de dispositivos, teniendo cada uno de ellos ventajas y desventajas.

1.1.2.1. Sistema operativo Android

El sistema operativo Android es un proyecto de código abierto para teléfonos móviles y otros dispositivos. Está desarrollado con el fin de proveer a desarrolladores una plataforma abierta para la creación de aplicaciones móviles. Este proyecto se originó por medio de un grupo de compañías conocidas como Open Handset Alliance, las cuales son lideradas por Google. Este sistema operativo tiene una gran cantidad de características y especificaciones. Las características principales van desde mensajería, soporte multimedia, soporte para streaming hasta características de comunicación como videollamadas, GPS, bluetooth. Estas permiten una combinación que dota a los desarrolladores y a los usuarios finales de un sistema operativo robusto y flexible. Una de las principales características de este sistema operativo es la actualización constante que se da al entorno, es por esto que se tiene un catálogo de las distintas versiones que tiene el sistema operativo. Actualmente, la última versión oficial es

Android 9 con nombre clave Pie. Desde que salió la primera versión oficial de Android ha habido en total 9 versiones del sistema operativo.

1.1.2.2. Sistema operativo IOS

El sistema operativo IOS es un proyecto de la empresa multinacional Apple, es un proyecto privado que originalmente se desarrolló para el teléfono móvil iPhone, pero se amplió para dispositivos como iPod touch y iPad. Al ser un software privado no es posible instalarlo en dispositivos que no pertenezcan a la empresa Apple. Este sistema operativo tiene ciertas características que lo vuelven único, como la autenticación por medio del sistema de almacenamiento de Apple iCloud, el cual permite deshabilitar el dispositivo en caso de robo; además, para acceder a los datos de la cuenta se necesita una confirmación. Existen diferentes aplicaciones que son exclusivas del sistema operativo IOS. Una de ellas es el navegador Safari, así como el reproductor de música iTunes. Además de tener todas las aplicaciones que cubren las funcionalidades básicas para un teléfono móvil como llamadas, transferencia de archivos por bluetooth, GPS, soporte para streaming, soporte multimedia, soporte multitarea, este sistema operativo ofrece diferentes versiones para los usuarios y desarrolladores. La versión más actual es IOS3, la cual fue lanzada en junio de 2019. Al ser un proyecto de una empresa privada tiene varias limitaciones con respecto a desarrollo de aplicaciones, para utilizar el SDK es necesario una computadora Mac de Apple para su desarrollo.

1.1.3. Evolución

Las aplicaciones móviles tienen una responsabilidad al tener que cubrir una demanda en auge, la cual es proporcionar aplicaciones que proveen distintos servicios para los usuarios. Al tener dos sistemas operativos líderes en el

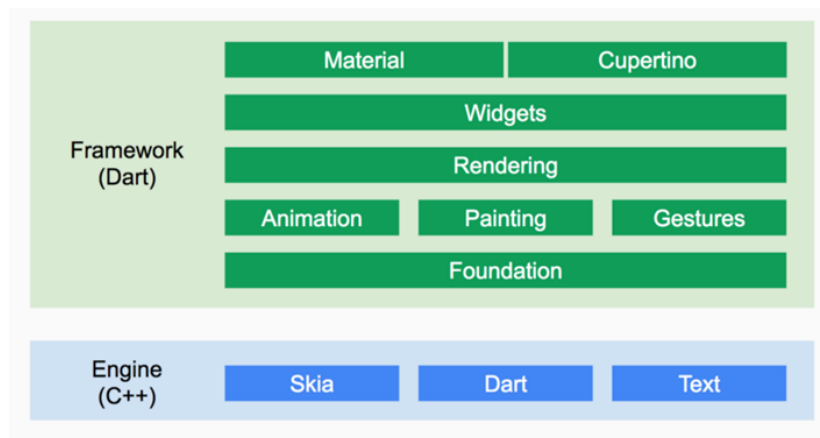
mercado, ha tenido que evolucionar. Esta evolución se centralizó en dar a los desarrolladores herramientas que permitan facilitar la creación de aplicaciones móviles en menor tiempo y con la menor cantidad de recursos. Es con base en esta necesidad donde surgen las aplicaciones híbridas. Este tipo de aplicaciones se basan en un lenguaje en común que permite generar aplicaciones para teléfonos móviles con sistema operativo Android y IOS. Esta es una tecnología que ha tenido una gran aceptación entre las comunidades de desarrollo por su versatilidad y facilidad de aprendizaje. Una de las primeras herramientas que surgió para suplir esa necesidad es phonegap, que permite crear aplicaciones para teléfonos móviles utilizando herramientas genéricas tales como JavaScript, Html5 y CSS3, las cuales tienen como base las vistas web que se presentan en la aplicación. Este es uno de los primeros frameworks que fueron popularmente utilizados para la generación de aplicaciones híbridas. Hoy existe una diversidad muy grande de frameworks para la generación de aplicaciones híbridas, tales como Reac Native, Native Script, Flutter, etc. Cada una de ellas con sus ventajas y desventajas, pero centrándose en el objetivo principal de generar aplicaciones móviles para dispositivos con sistema operativo Android y IOS, en base a un desarrollo común.

1.1.4. Flutter Framework

Flutter es un framework de código abierto desarrollado por Google que permite a desarrolladores crear aplicaciones móviles para teléfonos con sistema operativo Android y IOS, teniendo como base el lenguaje de programación Dart. Este lenguaje utiliza el paradigma orientado a objetos, pero integra funcionalidades asíncronas similares a Javascript. Este framework permite utilizar los componentes nativos del teléfono como los sensores, contactos, información de red, etc. Además, permite una integración con código nativo de los sistemas operativos Android y IOS por medio de canales. Uno de los

componentes claves de este framework es la utilización de widgets. Estos son componentes gráficos que permiten ser configurados según las necesidades que se tengan al momento de la creación de una aplicación. Los canales de plataforma y los widgets son las principales características que diferencian a flutter de otros frameworks que permiten desarrollar aplicaciones para los distintos sistemas operativos.

Figura 1. **Arquitectura Flutter**



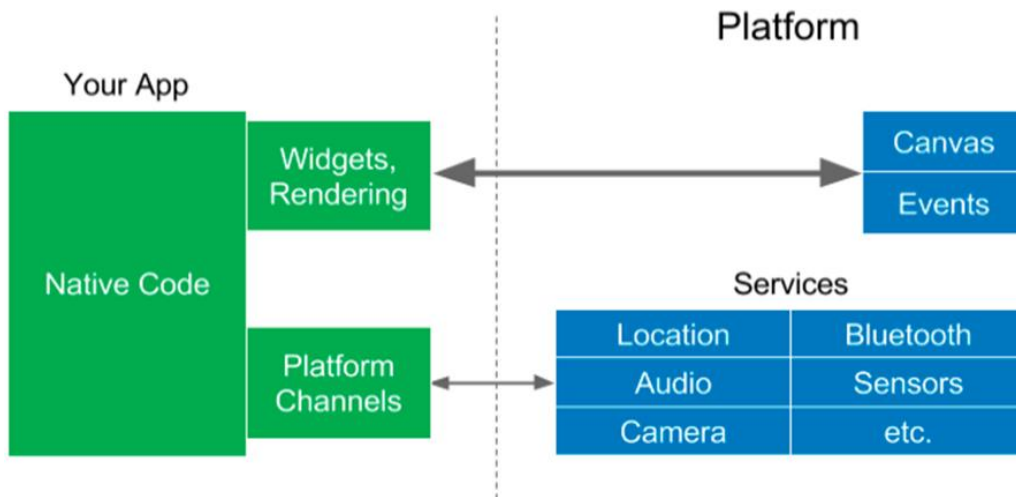
Fuente: VEGA, Cesar. *Lo revolucionario de Flutter*. <https://medium.com/comunidad-flutter/lo-revolucionario-de-flutter-425d4e43de8d>. Consulta: 19 agosto 2019.

Flutter tiene un catálogo de widgets estándar que nos permite avanzar de manera rápida en el desarrollo de la aplicación, ya que estos cubren los diseños básicos de aplicaciones para que los desarrolladores tengan una curva de aprendizaje más corta. Además, permite crear widgets personalizados según las necesidades y distribuirlos a las distintas comunidades de desarrolladores de Flutter.

Una de las principales características de flutter es la recarga en caliente (Hot Reload), ya que permite realizar cambios a la aplicación mientras se está ejecutando, para visualizarlos sin perder el estado actual de la aplicación. Esta

es una gran ventaja, ya que permite acortar los tiempos de compilación al momento de encontrar un error en el desarrollo y poder modificarlo para ver el resultado de manera inmediata. Esto se logra gracias a que los widgets se generan con base en el estado que tienen guardado y permite presentar al usuario los cambios que se realicen en el código.

Figura 2. **Canales de comunicación Flutter**



Fuente: VEGA, Cesar. *Lo revolucionario de Flutter*. <https://medium.com/comunidad-flutter/lo-revolucionario-de-flutter-425d4e43de8d>. Consulta: 19 agosto 2019.

1.2. **Métodos de comunicación**

Hoy existen distintos medios de comunicación entre los servicios backend y frontend debido a las distintas características que puede tener cada uno de ellos, por lo que es necesario conocer las distintas tecnologías para la implementación de la aplicación de alertas.

1.2.1. Web services

Es una tecnología que utiliza estándares de comunicación y protocolos para lograr una comunicación entre distintos dispositivos o servicios. Esta comunicación se define por medio de mensajes SOAP. El uso de web services no se restringe a un lenguaje de programación ni a una plataforma en específico; esta característica permite darle una gran flexibilidad y robustez. Los estándares y protocolos empleados para la implementación de un web services son los siguientes:

- XML: Extensible Markup Language es un lenguaje de marcado de propósito general. La característica principal es la definición propia de etiquetas, la cual permite personalizar el mensaje que se va a transmitir y dota de una gran flexibilidad a la comunicación. Es usado como estándar para la comunicación en web services.
- SOAP: Simple Object Access Protocol es un protocolo estándar que se encarga de la definición de comunicación entre dos entidades por medio del intercambio de datos utilizando un modelo XML. Este protocolo cuenta con tres características principales, que son la extensibilidad, la neutralidad y la independencia. La extensibilidad se basa en la seguridad que provee al ser implementado y que se aplica al momento del desarrollo. La neutralidad nos permite implementarlo en protocolos de comunicación de internet como HTTP, SMTP o JMS.

- WSDL: Web Services Description Language es un estándar que se basa en XML. Es el encargado de la descripción de la interfaz pública del web services, por lo que podremos saber el protocolo de comunicación y el formato de mensajes que se requieren para una interacción con el servicio. Este estándar se implementa cuando un servicio quiera conectarse y necesite configurar los mensajes que enviará.
- WS-Security: Web Services Security es un protocolo de comunicación que permite configurar la seguridad de los Web Services para contrarrestar ataques externos. Este protocolo es flexible para acomodar nuevos mecanismos de seguridad y ofrecer mecanismos alternos.

1.2.2. RESTFUL Web Services

Representational State Transfer es una alternativa al protocolo SOAP con la diferencia que no utiliza Web Services Description Language - WSDL para definir los protocolos de comunicación. REST define una serie de principios los cuales se desarrollan de la siguiente manera.

- Implementación de métodos HTTP: REST hace uso de los métodos HTTP, los cuales se definen como operaciones básicas que se pueden realizar en un sistema. Estos son definidos con palabras representativas de las acciones que realizan como crear, leer, actualizar, borrar. Cada una de esas tiene una asociación a los métodos HTTP.
 - POST: método para la creación de un recurso.
 - GET: método para obtener información de un recurso.
 - PUT: método que permite cambiar el estado de un recurso o actualizarlo.
 - DELETE: método que permite eliminar un recurso.

- No mantiene estado: este principio se basa en que REST necesita recibir los parámetros para realizar las operaciones, ya que no almacena ninguna información de las operaciones. De esta manera efectúa una tarea eficiente y reduce la complejidad de las tareas que debe realizar.
- URIs con forma de directorios: este principio se centra en definir la estructura de las URIs de manera que sean intuitivas, basándose en una jerarquía similar a las carpetas. Implementan una única ruta raíz y esta se desglosa en subrutas que se van a utilizar, llegando a un punto donde se puede predecir la ruta de un servicio al momento de implementar de manera correcta este principio. Las URIs son estáticas, ya que, al momento de modificarlas, la lógica de negocio se ve reflejada en los datos que retorna y no en la estructura de la definición de la ruta.
- Transferencia XML, JSON: este principio se enfoca en el formato de los datos que va a recibir, manipular y retornar el servicio, por lo que XML y JSON se vuelven un elemento clave de la comunicación en los servicios REST.
- JSON: es un formato de intercambio de datos que se basa en estructuras como colección de pares de nombre valor y listas ordenada de valores. Este formato maneja distintos tipos de datos, entre ellos números, cadenas, booleanos, nulos, arreglos, objetos. Cada uno tiene un formato específico que permite construir la estructura de manera intuitiva.
- Tanto XML como JSON tienen ventajas que permiten utilizar una estructura de transporte y almacenamiento de datos personalizables según las necesidades. En los servicios REST se puede definir el formato que necesitamos para enviar o retornar la información que envía el servicio. Para esto se tiene definición de Content-type, que envía en los encabezados de las

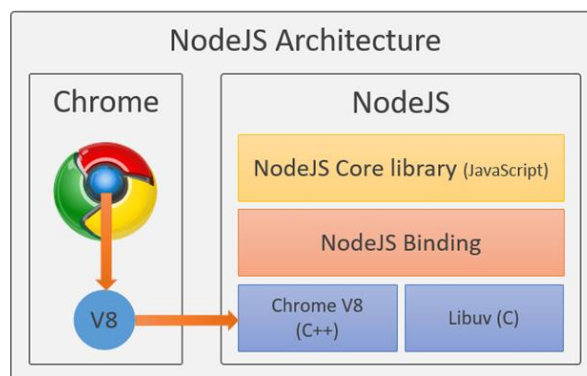
URIs el tipo de dato que va a manipular. Los tipos de Content-type son los siguientes:

- JSON - application/json
- XML - aplicacion/xml

1.2.3. NodeJS Framework

NodeJS es un entorno de ejecución para el lenguaje de programación Javascript del lado del servidor. Este entorno nace de la toma del motor V8 de JavaScript monta como el core de NodeJS, por lo que permite la compatibilidad en los servidores gracias a que JavaScript está pensado para su ejecución en la interfaz y no como lenguaje base del backend. La arquitectura de NodeJS es en realidad el motor V8 con una capa superior de librerías de NodeJS, las cuales se encargan de la comunicación entre la API de NodeJS y el motor V8, Adicionalmente, se apoya en la librería Libuv, la cual tiene a su cargo el procesamiento de entradas y salidas asíncronas.

Figura 3. **Arquitectura NodeJS**

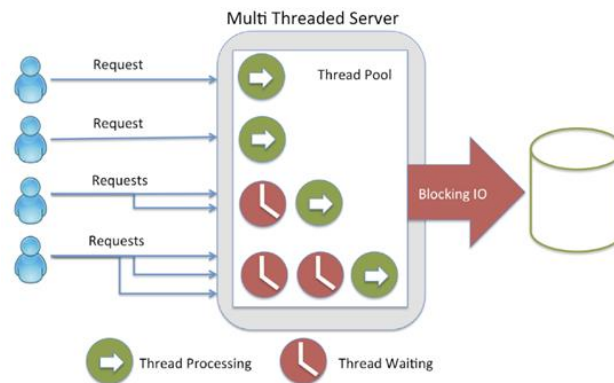


Fuente: BLANCARTE, Oscar. *Introducción a NodeJS*.

<https://www.oscarblancarteblog.com/2017/05/29/introduccion-a-nodejs-2/>. Consulta: 28 agosto 2019.

NodeJS utiliza el paradigma orientado a eventos, lo que permite que el flujo de ejecución sea asíncrono y evita los bloqueos que suelen suceder en otros lenguajes de programación.

Figura 4. **Arquitectura convencional síncrona**

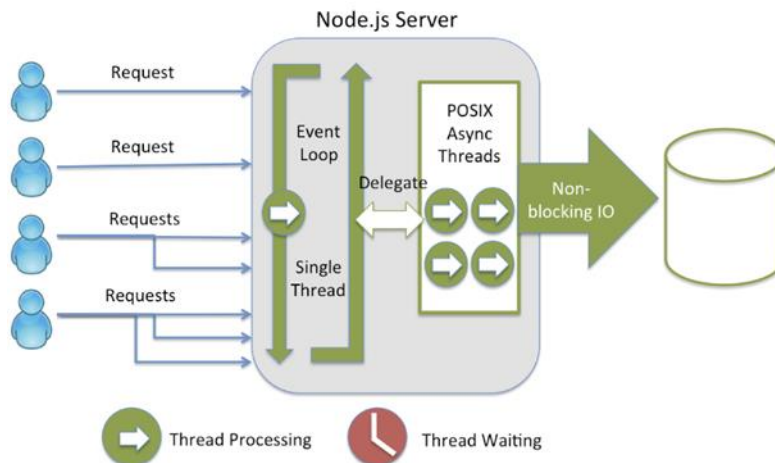


Fuente: PAUL, Javin. *Why we use Threads in Java?*.

<https://javarevisited.blogspot.com/2017/03/why-we-use-threads-in-java.html>. Consulta: 25 agosto 2019.

Cuando comparamos una arquitectura que implementa un flujo de ejecución asíncrono notamos las debilidades al resolver bloqueos que se dan al momento de la ejecución.

Figura 5. **Arquitectura servidor NodeJS**



Fuente: MATATHAMANA, Manjunath, *Building RESTful APIs using Node JS, Express JS, and MS SQL Server*. <https://www.sqlservercentral.com/articles/building-restful-apis-using-node-js-express-js-and-ms-sql-server>. Consulta: 28 agosto 2019.

La diferencia principal de las dos arquitecturas de ejecución es que la arquitectura síncrona, al momento de ejecutar una función o proceso y ocurre un error, rompe todo el proceso principal de ejecución. Mientras que en NodeJS, al estar basado en una arquitectura asíncrona, permite tener procesos separados en distintas ejecuciones. Esto significa que, al momento de un error, este seguirá con la ejecución de los otros procesos.

NodeJS cuenta con un gestor de paquetes, el cual tiene como nombre NPM. Este gestor se encarga de la descarga de las dependencias y ejecutar instrucciones como iniciar o apagar NodeJS.

Dentro de los paquetes y dependencias que se pueden integrar con NodeJS existe una en particular que permite desarrollar de manera sencilla los servicios REST. Este es un framework de desarrollo de aplicaciones web flexible

el cual tiene como nombre Express.js. Está desarrollado para ofrecer las operaciones básicas de REST como GET, POST, PUT, DELETE, entre otros.

Además, permite la integración con motores de renderización de vistas para generar respuestas mediante la introducción de datos en plantillas. Dentro de las librerías que tiene implementadas existen las que pueden trabajar con cookies, sesiones, inicios de sesión de usuario, parámetros URL, cabeceras de seguridad, etc. Es un framework robusto que permitirá una implementación limpia de un servicio de comunicación REST.

1.3. Servicio en la nube

Para definir un servicio en la nube es necesario conocer el concepto de nube. La nube es una referencia a los servicios que funcionan a través de internet los cuales se encuentran interconectados por medio de una red mundial y tienen una función en específica. La nube no es una entidad física, sino se compone de una red de servidores remotos. Estos están diseñados para almacenar y administrar datos, administrador recursos y entregar contenido o servicios.

Un servicio en la nube (Cloud Computing) es todo aquel programa que usamos y no está físicamente instalado en nuestro entorno de trabajo. Se acceden a él por medio de una conexión a internet. Existen distintos tipos de servicios en la nube:

- **Software as a Service (SaaS):** este tipo de servicio en la nube se caracteriza porque el proveedor permite alojar el software que se desarrolla en sus servidores, y que el cliente acceda a los recursos para administrarlos.

- Platform as a Service (PaaS): en este tipo de servicio, el proveedor ofrece acceso a un entorno en el cual el usuario puede crear y distribuir los sistemas necesarios, sin la necesidad de administrar los servicios.
- Infrastructure as a Service (IaaS): permite al usuario acceder a soluciones planteadas basadas en software y servicios en la nube que el proveedor administra. El usuario solo se comunica por medio de api's que el proveedor establece.

Hoy existe una gran cantidad de proveedores de servicios en la nube, los cuales destacan Amazon Web Services, Azure Microsoft y Google Cloud. Estos ofrecen distintas formas de administrar servicios de despliegue de sistemas. Hay otro tipo de proveedores de servicios que nos permiten almacenar nuestras aplicaciones para distribuirlas, como PlayStore, AppStore.

1.3.1. Google Cloud

Es una plataforma que contiene diversos servicios que funcionan bajo la misma infraestructura que utiliza Google para sus servicios. Esta provee a los usuarios un conjunto de herramientas que abarcan Cloud Computing, Networking, Data Storage, entre otros.

Esta infraestructura se divide en regiones y zonas, distribuidas a nivel mundial, con el fin de prevenir desastres naturales que puedan afectar la disponibilidad de los servicios y tener una replicación de los datos de los usuarios en diferentes partes.

Uno de los principales servicios que ofrece la plataforma Google Cloud es Computing, que se divide en varios productos que se pueden implementar, como

App Engine y Compute Engine. El primero está enfocado en la creación y despliegue de aplicaciones web, Android, entre otros. Se caracteriza por la capacidad de desligar al usuario de toda responsabilidad en la gestión de recursos y permite que se enfoque en el desarrollo. App Engine tiene soporte para una gran cantidad de lenguajes de programación como Java, Go, PHP, Python, entre otros. Compute Engine maneja un enfoque diferente en cuanto a la gestión de recursos por parte del desarrollador, ya que brinda un entorno virtualizado el cual vamos a gestionar los recursos del servidor. Dentro de estos recursos se encuentra la memoria ram, el disco duro, los procesadores, entre otros. Este servicio tiene un costo por hora que varía dependiendo de la configuración y uso que se le dé al servidor virtualizado; el uso mínimo es de 10 minutos. Los servidores virtuales disponen de varias distribuciones de sistemas operativos, como linux, incluidas distribuciones como Debian y CentOS; además, permite el uso de máquinas con software privado como Windows server, que tiene un costo adicional por el uso de las licencias.

1.3.2. Play Store

Play Store o Google Play es una plataforma de distribución de aplicaciones móviles para dispositivos con sistema operativo Android. Es la tienda oficial en línea desarrollada y administrada por Google. Permite a los usuarios distribuir y descargar de manera segura las aplicaciones que se desarrollen mediante el Android sdk. Dentro de las categorías que se encuentra dentro de Google Play Store están juegos, música, libros, revistas y películas. Existen distintos tipos de descargas. Las gratuitas y las que son pagadas para instalar las aplicaciones en nuestro teléfono. Hoy en día es uno de los principales medios de difusión de aplicaciones para teléfonos con sistema operativo Android y tablets. Uno de los puntos básicos al momento de la clasificación de las aplicaciones es el sistema

de normas que utiliza Google el cual están basados en 4 tipos, dentro de los cuales tenemos los siguientes:

- Para todos: contenido sin restricciones de edad
- Madurez baja: recomendado para mayores de 6 o 7 años
- Madurez media: recomendado para mayores de 12 años
- Madurez alta: recomendado para mayores de 15 años

Existe una serie de herramientas las cuales son exclusivas de usuarios que se registren como desarrolladores en la plataforma, Google Play Store ofrece a los desarrolladores pruebas y publicación de las aplicaciones, ya que permite tener a un grupo reducido de usuarios para realizar pruebas en la aplicación y poder verificar las funcionalidades de la misma. Para registrarse como un usuario desarrollador se necesita un pago único de \$ 25 dólares americanos, los cuales permiten subir una cantidad ilimitada de aplicaciones, una gran ventaja en comparación a otras plataformas.

Para realizar una publicación existen dos métodos: una publicación estándar y una programada. La diferencia radica en que las publicaciones programadas se enfocan en las actualizaciones que van a recibir los usuarios que tengan descargada la aplicación, y la publicación estándar es para que los cambios se vean reflejados en el menor tiempo posible. Para cualquiera de los tipos de publicaciones se tiene que pasar por una revisión de la aplicación. Este es un tema central al momento de hacer referencia a la seguridad que tiene la Google Play Store, ya que permite verificar el contenido de la aplicación, así como estándares que indica Google para desarrollar todo tipo de aplicaciones, con el fin de proveer el usuario de una mejor experiencia al momento de utilizar las aplicaciones que descargue. Este periodo de aprobación de la aplicación puede

tardar un periodo de 5 horas; en casos específicos puede tardar un periodo de 7 días o más.

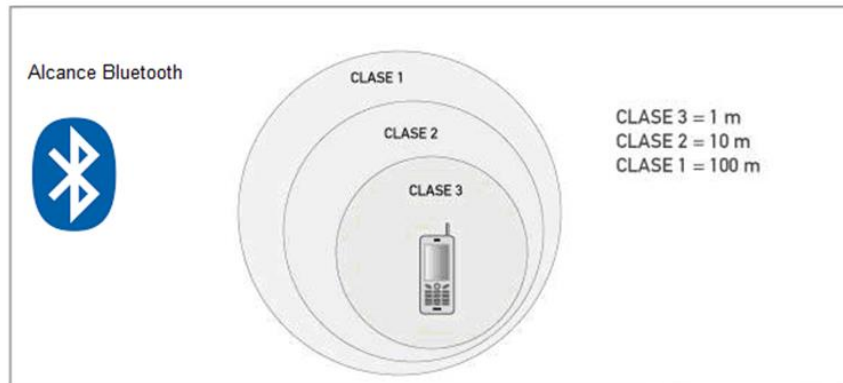
1.4. Sistemas de localización

Un sistema de localización es un protocolo que permite ubicar de manera inmediata un objeto móvil en un espacio cerrado como un edificio, casa, entre otros, o un espacio abierto como una calle, campo, entre otros. El proceso de ubicación del objeto se denomina seguimiento o tracking. Existen distintos tipos de sistemas de localización en tiempo real, los cuales se clasifican dependiendo del alcance que tenga cada uno de ellos:

1.4.1. Sistemas de corto alcance Bluetooth

Estos sistemas son comúnmente utilizados para comunicar dos dispositivos que se encuentran muy cerca, ya que tiene un alcance reducido. Existen distintas categorías de bluetooth, cada una tiene alcances diferentes. La clase 3 tiene un alcance de 1 metro; la clase 2, de 10 metros y la clase 1, que es la de mayor alcance, de 100 metros.

Figura 6. **Clases de Bluetooth**



Fuente: AGPS. *Sistemas de Localización*. <https://www.agps.es/sistemas-de-localizacion/>.
Consulta: 28 agosto 2019.

1.4.2. Sistemas de largo alcance

Estos sistemas se basan en radiofrecuencia ya que necesitan un equipo emisor de la señal y un equipo receptor. Tiene la característica que los equipos emisores son ligeros y pequeños en contraparte los equipos receptores son grandes y pesados. Este tipo de sistemas tiene varias dificultades para implementar, ya que necesita que no exista interferencia entre el emisor y el receptor de la señal. Una de las limitantes de este sistema es la curvatura de la tierra, ya que se perdería la señal al ser interrumpida. Una de las aplicaciones de este sistema es en las estaciones de radio FM, ya que se colocan antenas en las montañas para tener un mayor alcance.

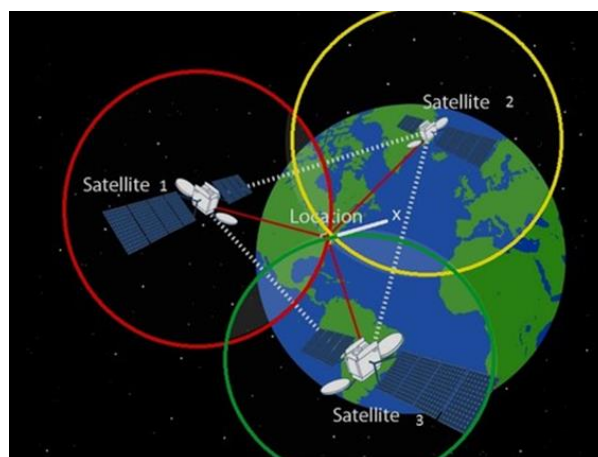
1.4.3. GPS

El sistema de posicionamiento global es un sistema basado en radiofrecuencia que, a diferencia de los sistemas de corto y largo alcance,

necesita de satélites para poder enviar las señales y obtener la posición exacta del objeto. La principal ventaja de la localización por gps es el alcance, ya que al estar utilizando satélites no tiene un límite; además, la precisión que ofrece y su bajo coste hacen del gps una herramienta muy versátil.

El sistema de posicionamiento global funciona mediante una red de satélites que orbitan sobre el planeta Tierra. Existe un requerimiento para que pueda funcionar, ya que es necesario que por lo menos 4 satélites estén visibles en cualquier punto para determinar la posición del objeto. Es por esto que, para tener una cobertura total a nivel mundial, se necesitan como mínimo 24 satélites, cada uno distribuido en distintas órbitas. Al tener los 4 satélites localizados por el receptor se procede a recibir unas señales indicando la identificación y la hora del reloj de cada uno de los satélites. Con base en estas señales se procede a sincronizar con el reloj propio del receptor y se calculan los tiempos que tardan en llegar las señales, por lo que se puede determinar la posición del objeto con un margen de error relativamente bajo.

Figura 7. **Distribución de satélites GPS**



Fuente: FERNANDEZ, Yúbal. *Gps en android: cómo funciona, cómo mejorar su precisión y cómo decidir qué aplicaciones lo usan*. <https://www.xataka.com/basics/gps-android-como-funciona-como-mejorar-su-precision-como-decidir-que-aplicaciones-usan>. Consulta: 29 agosto 2019.

1.5. Metodología de desarrollo

Una metodología de desarrollo es un marco de trabajo utilizado para documentar de manera formal los pasos referentes a los procesos, las políticas y los procedimientos que intervienen en las distintas fases del desarrollo de software. Uno de los principales fines de las metodologías de desarrollo es la eficiencia en el cumplimiento de los requisitos planteados al inicio y realizar de manera eficiente el proceso hasta llegar a su fase final. Dentro de las metodologías de desarrollo encontramos las metodologías ágiles, que se caracterizan por la comunicación cara a cara y porque se centran en la interacción de las partes involucradas en el proceso de establecimiento de los requerimientos y alcances del desarrollo. Estas partes involucradas son los clientes, desarrolladores, inversores, facilitadores, entre otros. Esto va a depender de la complejidad y el tamaño del proyecto. Las metodologías ágiles permiten una gran flexibilidad al momento de desarrollar el software, además de una retroalimentación por parte de los clientes. El nivel de satisfacción mejora, ya que se sienten involucrados y se comprometen a lo largo del proyecto. Otro factor por considerar es el involucramiento del equipo de desarrollo, ya que permite opinar sobre la toma de decisiones, por lo que se sienten más identificados con el proyecto.

Hoy existen diferentes metodologías ágiles de desarrollo, dentro de las cuales podemos mencionar Programación Extrema XP, Scrum y Kanban.

1.5.1. Proceso scrum

El ciclo de vida de scrum consta de varias fases, las cuales permiten tener retroalimentación para identificar posibles fallas o aciertos al momento del desarrollo. Se ejecuta en iteraciones temporales cortas que duran

aproximadamente 2 semanas. Cada iteración debe proporcionar un resultado completo, un incremento en el producto final que sea susceptible a la vista del cliente.

Se crea una lista de objetivos/requisitos del producto, que va a tomar el rol de plan de proyecto. Esta lista se prioriza por parte del cliente, que será nombrado como Product Owner. Posteriormente se realiza la planificación de la iteración, en la cual está involucrada todo el equipo de desarrollo. Permite darle una estimación de esfuerzo a cada uno de los elementos de la lista de objetivos/requisitos, que permite tener varios puntos de vista y un consenso en el equipo con respecto a los tiempos de desarrollo. Terminada la fase anterior se procede a la ejecución de la iteración, la cual tiene como base una reunión diaria, la cual se denomina de sincronización. Esta reunión no debe de exceder los 15 minutos de duración y su objetivo es realizar 3 preguntas directas a cada miembro del equipo:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a realizar a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener que impidan completarlos?

Estas tres simples preguntas abarcan una parte significativa en el proceso de desarrollo, ya que permite identificar los avances y comprometer a cada miembro a realizar una tarea que agregue valor al desarrollo.

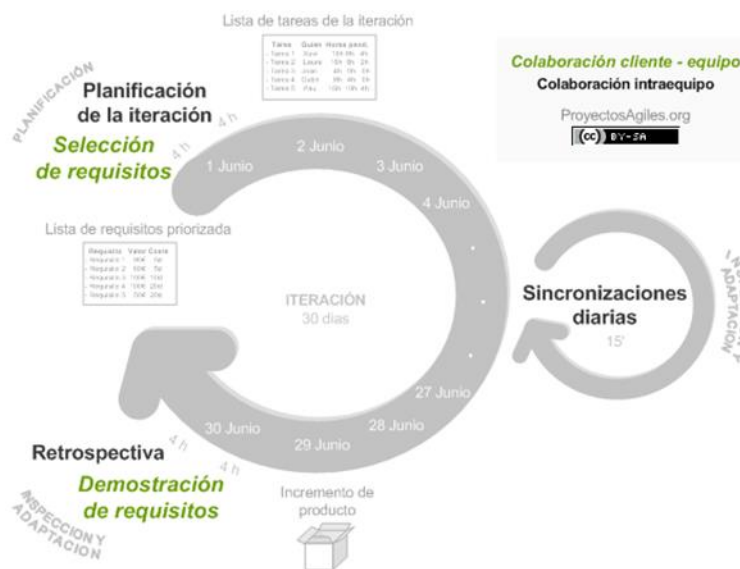
Al finalizar la iteración se realiza una reunión para revisar los avances de la misma. Esta consta de dos partes:

- Revisión: consiste en presentar los requisitos finalizados al cliente, lo cual permite crear un ambiente donde al cliente se le hacen entregas continuas.

Puede ver de manera continua el avance del proyecto y sugerir cambios, dependiendo del impacto que tendrían sobre el proyecto.

- Retrospectiva: esta reunión es interna y está involucrado el equipo de desarrollo, en el cual se analiza la manera de trabajar y cuáles son los problemas que podrían interrumpir en el desarrollo de los requisitos.

Figura 8. **Ciclo de vida metodología Scrum**



Fuente: Albaladejo, Xavier. ¿Qué es Scrum? <https://proyectosagiles.org/que-es-scrum/>.
Consulta: 29 agosto 2019.

1.5.2. Beneficios de las metodologías de desarrollo

Los proyectos que se llevan a cabo implementando una metodología de desarrollo tienen una alta tasa de éxito, ya que ponen el foco en las personas encargadas del desarrollo y del buen manejo del tiempo. Las metodologías de desarrollo permiten fomentar un ambiente de responsabilidad dentro del equipo y aumentan la autonomía, ya que el equipo está involucrado en la toma de

decisiones y en la definición de los tiempos de desarrollo, lo que propicia tener distintos criterios.

Permite una mejor visión del proyecto, ya que minimiza los riesgos al estar recibiendo retroalimentación por parte del cliente. Esto mejora la experiencia del cliente y permite una comunicación continua. Dividir las fases de desarrollo y las funcionalidades o requisitos da la oportunidad de un mejor manejo en los tiempos de desarrollo.

2. DESCRIPCIÓN DEL PROBLEMA Y PLANTEAMIENTO DE LA SOLUCIÓN

En este capítulo abordaremos los distintos puntos por considerar al momento de identificar el problema y el planteamiento de la solución.

2.1. Descripción del problema

El Centro Universitario Metropolitano (CUM) de la Universidad de San Carlos de Guatemala es un centro de formación académica en donde diariamente se alberga a un conglomerado de estudiantes, docentes, personal administrativo y personas en general, que participan de una serie de actividades que se dan dentro del centro regional, como eventos científico-académicos, investigación, culturales, sociales y de formación general. Esto crea la necesidad de estar informados de los acontecimientos que surgen en la universidad y que son de suma importancia para velar por que los usuarios tengan conocimiento de las emergencias y eventos que se puedan dar en determinado momento que ameriten la respuesta rápida de las personas encargadas de velar por el orden y el buen funcionamiento de las instalaciones. Al carecer de un sistema que centralice estos eventos se puede dar paso a una desinformación o a un total desconocimiento con repercusiones relacionadas la pérdida de tiempo, aspectos de seguridad, embotellamientos entre otros.

La Universidad de San Carlos de Guatemala (USAC) en los últimos años ha desarrollado distintos medios, tanto para la publicación de eventos relacionados con actividades que se realizan dentro de la USAC, así como distintos planes para mejorar la seguridad. Sin embargo, estos planes se han

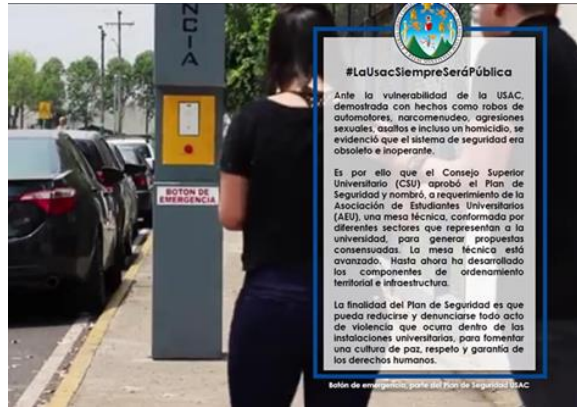
centralizado en el campus central de la universidad, por lo que centros regionales y el centro metropolitano carecen de proyectos que puedan satisfacer esta necesidad.

2.1.1. Antecedentes de la solución

Al momento de plantear la problemática se lograron identificar soluciones anteriores que buscan solventar de manera parcial el problema, por lo que vamos a describirlos.

- Estaciones de emergencia: actualmente, el plan de seguridad de la USAC es centralizado en el campus central, en el cual implementaron estaciones de emergencia que cuentan con un botón de pánico y un intercomunicador. Este se redirige directamente con el personal de seguridad para atender la emergencia. El problema es que este proyecto está implementado únicamente en el campus central y al estar en puntos fijos se depende mucho de que las personas se movilicen hasta la estación más cercana para realizar la alerta. Además, se pone en riesgo que una de las estaciones no esté disponible por desperfectos, esté en mantenimiento o, en un caso extremo, que se desconozca la ubicación.

Figura 9. Botón emergencia

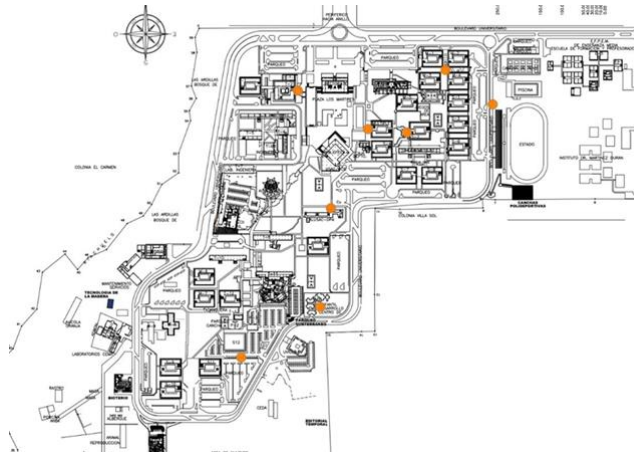


Fuente: Unidad de Difusión USAC. *Botón de emergencia.*

<https://www.facebook.com/idhusac/posts/d41d8cd9/1711761605631119/>. Consulta: 10 septiembre 2019.

Estas 8 estaciones de emergencia están ubicadas en distintos puntos de la universidad.

Figura 10. Mapa estaciones de emergencia



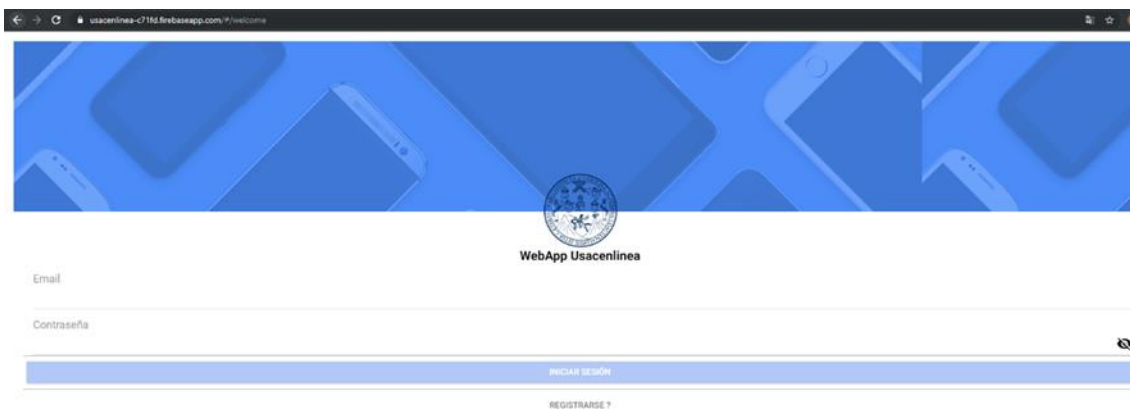
Fuente: Cruz Kevin. Mapa Estaciones de Emergencia.

<https://www.facebook.com/KevinCruzPaginaOficial/photos/a.534145217093935/588861368288986>. Consulta: 11 septiembre 2019.

Actualmente, este proyecto está solo desarrollado en el campus central de la universidad, por lo que el centro universitario metropolitano y los centros regionales carecen de este tipo de implementaciones.

- Plan de denuncia: la universidad tiene implementado una plataforma de denuncias por medio de la página web <https://usacenlinea-c71fd.firebaseio.com/#/welcome>, la cual permite realizar la denuncia. Esta plataforma está pensada en denuncias de anomalías de procesos o situaciones puntuales dentro de la universidad, como cobros, acoso, entre otros.

Figura 11. **Sitio web app usacenlinea**



Fuente: elaboración propia.

- App Espanta Cacos: esta app fue implementada por la Asociación Jóvenes Contra la Violencia para funcionar en la ciudad de Guatemala. Permite, por medio de un mapa, ver las distintas zonas que son peligrosas o donde han ocurrido asaltos o hechos delictivos. La idea de la app es tener un mejor monitoreo de los sectores vulnerables. Esto se logró con apoyo de las autoridades para tener un mejor control de estas zonas.

Para realizar la publicación del lugar peligroso se necesita de un registro previo para su posterior uso. La app está desarrollada para teléfonos inteligentes con sistema operativo Android y IOS.

Dentro de los puntos claves de la implementación es que se logró determinar los puntos más peligrosos de la ciudad, como: zona 1, Calzada Roosevelt, Anillo Periférico, Villa Linda Zona 7, entre otros.

Figura 12. **banner publicidad app espanta cacos**



Fuente: Espanta Cacos. *Espanta Cacos*.

<https://www.facebook.com/espantacacos/photos/a.518188968195738/1282394601775167>.

Consulta: 12 septiembre 2019.

- Publicación de eventos de unidades académicas: la mayor parte de las unidades académicas que están ubicadas en la universidad y en el centro metropolitano, tiene una página en la red social Facebook. Esta permite enviar información y publicación de eventos que se suscitan en el día a día; además, cuentan con espacios físicos disponibles para colocar carteles o afiches. Dentro de las unidades académicas que hacen uso de

estas herramientas están: las facultades, escuelas no facultativas, unidades administrativas y de servicio.

La facultad de medicina ubicada en el Centro Universitario Metropolitano (CUM) hace uso de estas herramientas para informar a los estudiantes y población en general sobre estas actividades.

Figura 13. **Actividad académica ERAS**



Fuente: Facultad de Ciencias Médicas USAC. *Sesión Científica*.

<https://www.facebook.com/facmedusac/photos/2379465768969653>. Consulta: 13 septiembre

2019

Estos son varios ejemplos de los intentos por mejorar la comunicación y la seguridad en la universidad y en la ciudad, tomando en consideración que cada una tiene sus puntos débiles los cuales se pretenden solucionar, así como mejorar el flujo de los procesos por medio de la centralización de las características en una sola aplicación.

2.2. Planteamiento de la solución

Al tener claro los antecedentes y la visión general del problema podemos plantear una solución que permita satisfacer con todas las necesidades.

2.2.1. Descripción de la aplicación

Se plantea el desarrollo de una aplicación para teléfonos móviles con sistema operativo Android que permita la centralización de alertas y eventos que sucedan dentro del centro universitario metropolitano (CUM). Esta aplicación permitirá registrarse dentro de la aplicación e ingresar al contenido de la misma como usuario invitado.

Para ingresar a la aplicación es necesario ingresar el Código Único de Identificación (CUI) y una contraseña. Con este proceso se podrá determinar si la persona que va a ingresar está registrada o si podrá acceder como usuario invitado.

Las alertas y eventos que sucedan dentro de las instalaciones del CUM se registrarán por medio de la aplicación, la cual obtendrá la ubicación exacta del dispositivo por medio del GPS que tiene integrado, por lo que se contará con la posición exacta de las alertas y eventos. Las alertas y eventos tienen diferentes tipos, los cuales vamos a detallar en la siguiente sección.

2.2.1.1. Alertas y eventos

Las alertas y eventos son la base central de la aplicación, ya que permitirán informar a la población estudiantil, personal de las distintas unidades académicas y población en general de los sucesos que ocurran dentro del CUM.

Las alertas se distribuyen en diferentes tipos, los cuales se pueden clasificar en:

- **Emergencias médicas:** son emergencias que puedan suceder en cualquier momento y ubicación del CUM. Tiene como objetivo alertar a las personas cercanas para obtener atención médica.
- **Reporte de asaltos:** permite reportar de manera anónima un asalto. Esta alerta podrá ser visualizada por el usuario administrador, el cual se encargará de coordinar al personal de seguridad para atender la emergencia.
- **Reporte de accidentes automovilísticos:** permite reportar un accidente automovilístico dentro de las instalaciones del CUM. Permitirá alertar a los usuarios de posibles embotellamientos, para así evitarlos.

Los eventos se distribuyen en las diferentes ubicaciones dentro del CUM. Se pueden clasificar en:

- **Actividades académicas:** conferencias, congresos, foros, simposios.
- **Actividades culturales:** conmemoración, aniversario.

Estas actividades se desarrollan dentro de las instalaciones del CUM y son de importancia para la comunidad estudiantil, personal de las unidades académicas y población en general.

Las alertas y eventos requieren información específica para describirlas. Al momento de crear una de ellas es necesario ingresar la siguiente información:

- Ubicación (detecta la ubicación por medio del GPS del teléfono móvil).
- Nombre de la persona que crea la alerta o evento
- Descripción general
- Fecha y hora
- Tipo

Estos datos van a ser completados por los usuarios, dependiendo de si es una alerta o un evento, ya que tienen diferentes usos dependiendo de la situación en la que se encuentre.

2.2.1.2. Descripción de los roles

Se definirán las acciones que podrán realizar cada uno de los usuarios que existen en la aplicación, para determinar el alcance de cada uno de ellos.

- Usuario administrador: este usuario tiene a su cargo la administración de las publicaciones, por lo que podrá crear, modificar, ver y eliminar cada una de ellas. Además, tendrá la visualización de las alertas que puedan suceder sin importar el tipo. Tiene una importancia grande, ya que es el encargado de la moderación de la aplicación. Los datos que se requieren para el manejo del usuario administrador son:

- Código Único Administrador (CUA)
- Contraseña administrador

- Usuario registrado: este tipo de usuario podrá realizar publicaciones de alertas y eventos. Podrá administrar sus publicaciones y observar las de otros usuarios. Los datos que se requieren para el manejo del usuario registrado son:
 - CUI usuario
 - Contraseña usuario

- Usuario invitado: este usuario, al no tener un registro en la aplicación, solo se le permitirá visualizar las alertas y eventos. No podrá crear, modificar o eliminar alguna de ellas. Este mecanismo es para evitar las publicaciones engañosas de usuarios de quienes no se tiene la información definida. Este usuario no requiere de datos para ingresar.

2.2.1.3. Alcances de la aplicación

El desarrollo e implementación de la aplicación propuesta tiene como alcance mejorar y optimizar el control y monitoreo de eventos y alertas en el centro universitario metropolitano CUM de la Universidad de San Carlos de Guatemala.

Los aspectos puntuales que comprenden esta aplicación son dotar a los estudiantes, docentes, personal administrativo y usuarios en general de una herramienta que permita publicar y consultar información inmediata, confiable y útil de temas concernientes a la seguridad, eventos académicos, culturales e incidentes que puedan suscitarse en la jornada diaria dentro del CUM.

Este software será implementado y desarrollado como un plan piloto para el CUM y quedará prevista la arquitectura y codificación para la implementación en otras unidades académicas, como el campus central y centros regionales que pertenezcan a la USAC. Se puede replicar en cada uno de ellos haciendo un estudio y desarrollo para cubrir las necesidades que esto conlleva.

2.2.1.4. Límites de la aplicación

La aplicación se limitará a mostrar los eventos y alertas que puedan surgir dentro del centro universitario metropolitano (CUM) de la Universidad de San Carlos de Guatemala (USAC).

Se enfoca el desarrollo e implementación del software para teléfonos móviles con sistema operativo Android. El software se limitará para uso de estudiantes, docentes y personal administrativo que pertenezca a la Universidad de San Carlos. Así también se puede considerar a personas que se desempeñan dentro del CUM y que estén diariamente en las instalaciones, como personal de servicios comerciales que se ubican en los diferentes módulos. (cafeterías, librerías, entre otros.).

Para el uso del software el usuario deberá de contar con una conexión a internet, ya sea por datos móviles o por WIFI.

3. DISEÑO E IMPLEMENTACIÓN DE LA SOLUCION

En este capítulo se desarrollarán el diseño de las pantallas, definición de la arquitectura del sistema, así como el modelo de la base datos, incluyendo en cada una de ellas las bases necesarias para su implementación.

3.1. Diseño de pantallas

En esta sección se definen los diseños de las pantallas que tendrá la aplicación ALERTAUSAC. Estas pantallas están distribuidas con base en el uso general y específicas del usuario registrado y administrador.

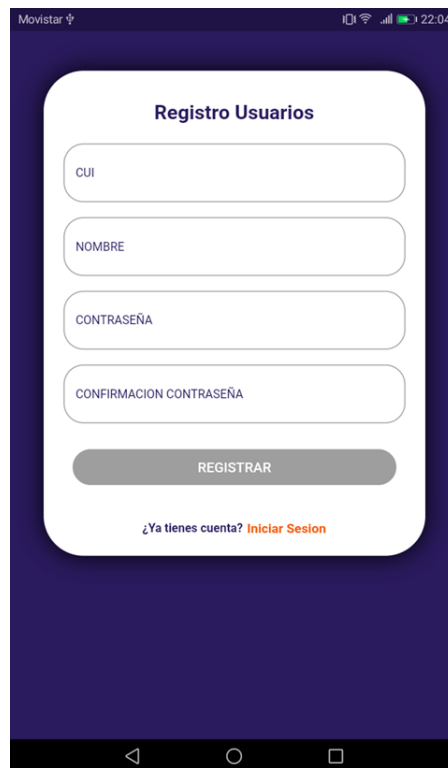
Figura 14. Pantalla inicio sesión



Fuente: elaboración propia.

La pantalla de inicio de sesión permitirá a los usuarios administrador y registrado a ingresar al mapa interactivo de la aplicación. Deben ingresar su CUI y contraseña. Tiene dos botones adicionales: el botón de invitado, que permitirá navegar en el mapa interactivo como usuario invitado, y el botón registro, el cual permitirá registrar a los usuarios nuevos que quieran utilizar la aplicación.

Figura 15. **Pantalla de registro**

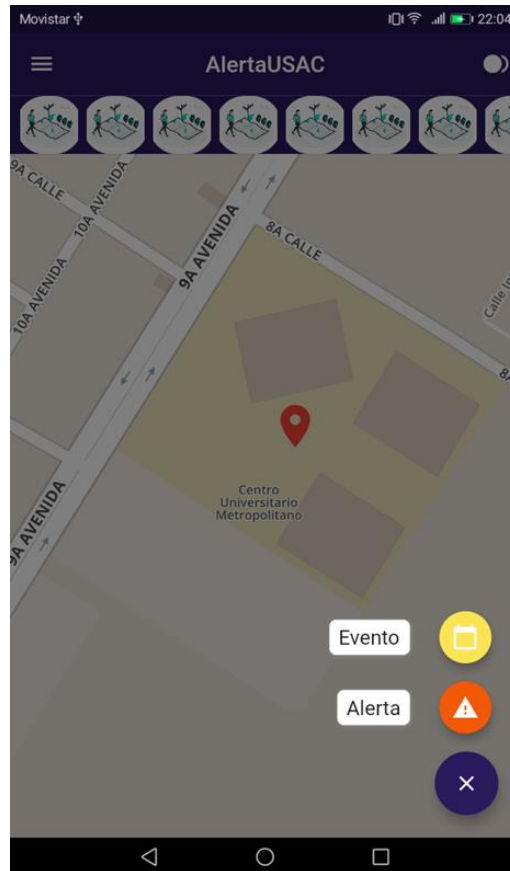


The image shows a mobile application interface for user registration. The screen is titled "Registro Usuarios" and features a white rounded rectangle on a dark blue background. Inside the rectangle, there are four input fields labeled "CUI", "NOMBRE", "CONTRASEÑA", and "CONFIRMACION CONTRASEÑA". Below these fields is a grey button labeled "REGISTRAR". At the bottom of the form, there is a link that says "¿Ya tienes cuenta? Iniciar Sesión". The top of the screen shows the status bar with "Movistar" and the time "22:04".

Fuente: elaboración propia.

La pantalla de registro permitirá a los usuarios ingresar los datos CUI, nombre, contraseña y una confirmación de contraseña, para ingresar al sistema como usuarios registrados. Esta pantalla no podrá registrar usuarios administradores.

Figura 16. **Pantalla mapa interactivo**



Fuente: elaboración propia.

La pantalla del mapa interactivo es la parte central de la aplicación, ya que consta de tres elementos clave, el carruaje de historias de alertas y eventos, en el cual se podrá visualizar cada una de las publicaciones que se realicen en el mapa. En el mapa interactivo se podrán visualizar las publicaciones mostrando su ubicación, estas se podrán diferenciar de una alerta y un evento por el color del icono. Contará con un botón desplegable, el cual permitirá crear un evento o una alerta. Esta pantalla constará de un botón de pánico, el cual permitirá enviar una alerta cuando se presione por un periodo de 2 segundos.

Figura 17. **Pantalla crear evento**

The screenshot displays the 'AlertaUSAC' mobile application interface for creating an event. The screen features a dark blue background with a white rounded rectangle containing the form. At the top, the title 'Tipo de Evento' is displayed, followed by a dropdown menu currently set to 'Academico'. Below this, there are four input fields: 'Fecha Evento', 'Hora Evento', 'Titulo Evento', and 'Descripcion Evento'. At the bottom of the form is a 'Publicar' button. The status bar at the top of the phone shows 'Movistar', signal strength, Wi-Fi, and the time '22:04'.

Fuente: elaboración propia.

Esta pantalla permitirá crear un evento en el cual el usuario debe ingresar el tipo de evento, fecha, hora, título y descripción. Al ingresar los datos necesarios la aplicación detectará la ubicación por medio del GPS del teléfono y permitirá crear el evento.

Figura 18. **Pantalla crear alerta**

The screenshot shows a mobile application interface for creating an alert. At the top, there is an orange header with a back arrow and the text 'AlertaUSAC'. Below this, a white rounded rectangle is centered on a dark blue background. Inside the rectangle, the title 'Tipo Alerta' is displayed. Below the title is a dropdown menu with a red cross icon and the text 'Emergencia Medica'. Underneath the dropdown are two text input fields: 'Titulo Alerta' and 'Descripcion Alerta'. At the bottom of the white rectangle is a grey button labeled 'Publicar'. The top of the phone screen shows the status bar with 'Movistar', signal strength, Wi-Fi, and the time '22:05'. The bottom of the screen shows the Android navigation bar.

Fuente: elaboración propia.

Esta pantalla permitirá crear una alerta, cual el usuario deberá de ingresar el tipo de alerta, título y descripción. La aplicación automáticamente detectará la hora, fecha y ubicación de la alerta.

Figura 19. **Pantalla perfil usuario**



Fuente: elaboración propia.

El usuario podrá visualizar la siguiente información: CUI, nombre, contraseña y foto de perfil. Esta pantalla contará con un botón, el cual permitirá actualizar la información del usuario.

Figura 20. **Pantalla editar usuario**

The screenshot displays a mobile application interface for updating user data. The title is "Actualizar datos". The form contains the following fields:

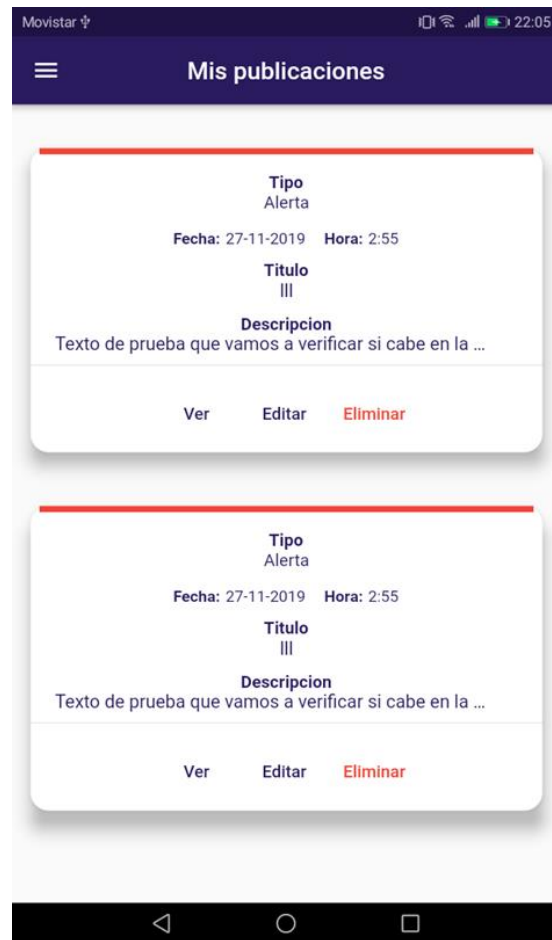
- Cui: 1000
- Nombre: manuel
- Contraseña:
- Confirmacion contraseña:

At the bottom center, there is a blue icon of a jar with a white 'C' on it. The status bar at the top shows "Movistar" and the time "22:30".

Fuente: elaboración propia.

Esta pantalla permitirá al usuario actualizar su información, y contará con los campos CUI, nombre, contraseña y una confirmación de contraseña. Contendrá un botón de actualizar datos.

Figura 21. **Pantalla lista de publicaciones**



Fuente: elaboración propia.

En esta pantalla, el usuario podrá visualizar las publicaciones que se han realizado en la aplicación, editarlas o eliminarlas. Las publicaciones tienen una vista preliminar de los datos, título, fecha, hora, descripción.

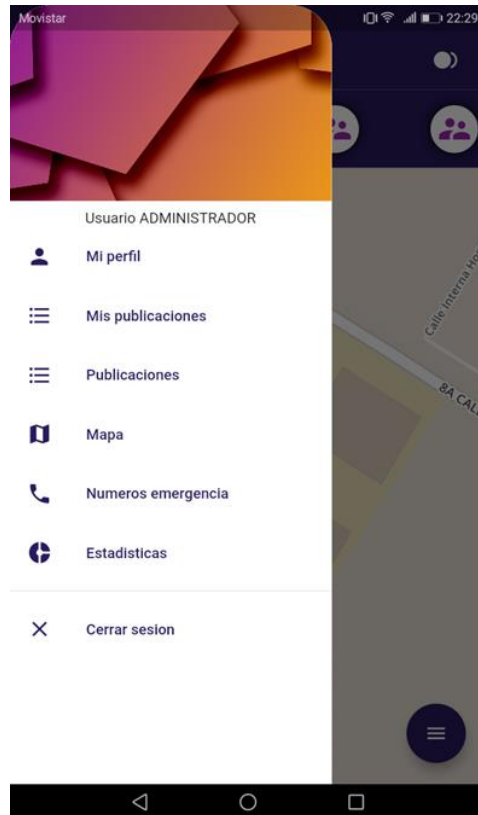
Figura 22. Pantalla lista números de emergencia



Fuente: elaboración propia.

Esta pantalla desplegará una lista con los principales números de emergencia, incluidos instituciones como Policía Nacional Civil, Bomberos Municipales y Voluntarios, y las autoridades de la universidad.

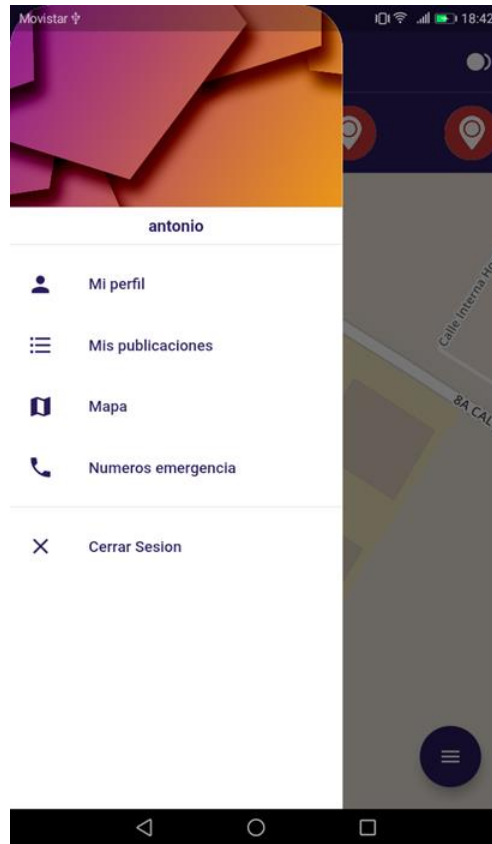
Figura 23. **Menú usuario administrador**



Fuente: elaboración propia.

El usuario administrador tiene el control de las publicaciones que realizan todos los usuarios y el administrador. Puede visualizar su información de perfil y la lista de los números de emergencia. Cuenta con la opción de visualizar el mapa de publicaciones, una sección de estadísticas y cerrar sesión.

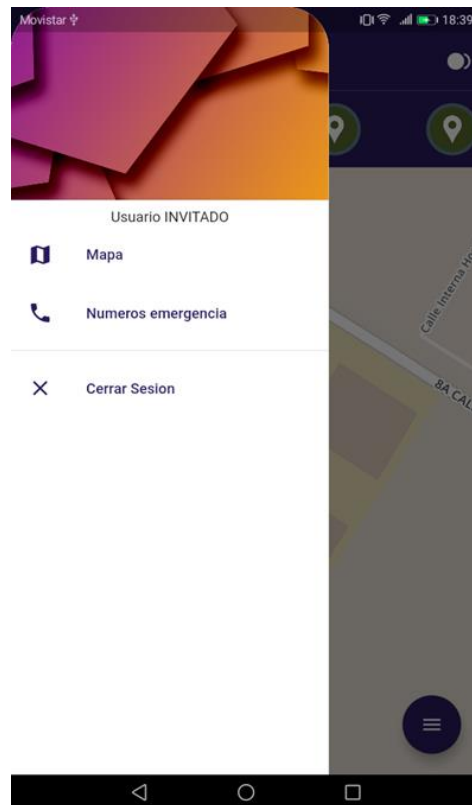
Figura 24. **Menú usuario registrado**



Fuente: elaboración propia.

El usuario registrado podrá visualizar en el menú las siguientes opciones: mi perfil, mis publicaciones, mapa y cerrar sesión. Cada una de ellas permitirá visualizar diferentes opciones y navegar en cada una de ellas.

Figura 25. **Menú usuario invitado**



Fuente: elaboración propia.

El usuario invitado, al tener las funciones únicamente de visualización, solamente tendrá las opciones de visualización de mapa, cerrar sesión y números de emergencia.

Figura 26. Estadísticas



Fuente: elaboración propia.

El usuario administrador podrá visualizar las estadísticas que genere la aplicación como el número de publicaciones que se han realizado, así como la cantidad de alertas de pánico que han sido activadas.

3.2. Arquitectura del sistema

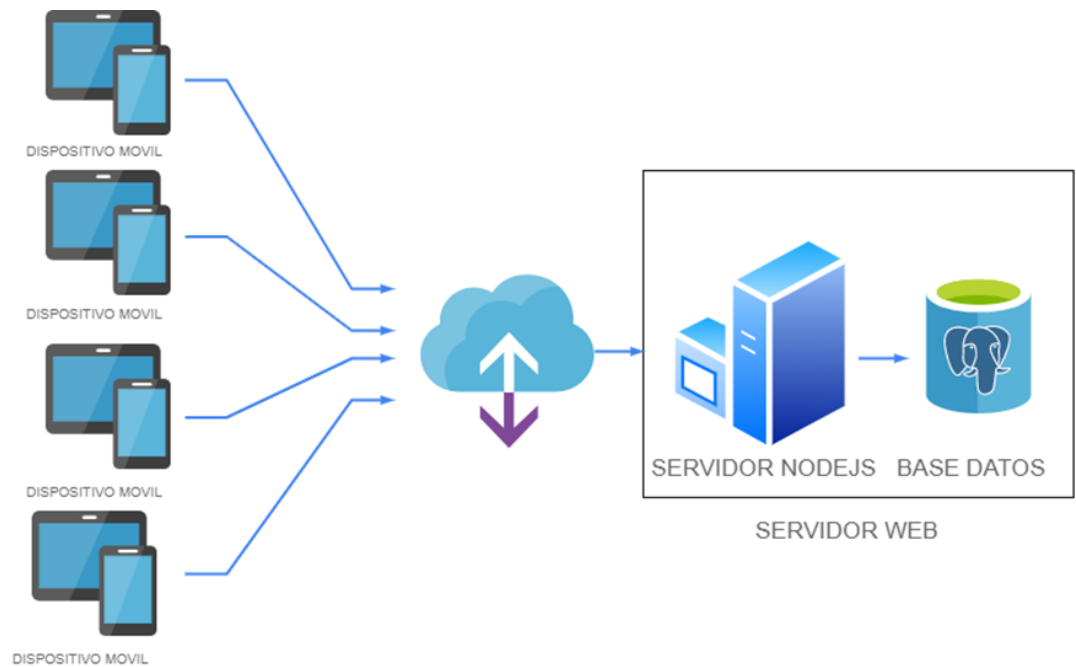
La arquitectura que se plantea para el desarrollo de la aplicación se basa en el modelo cliente – servidor. Esta arquitectura reparte la carga de tareas entre los proveedores de recursos o servicios y los consumidores. Estos dos roles los conocemos como servidor y cliente.

Esta arquitectura se basa en el consumo de los recursos por parte de los clientes hacia el servidor por medio de peticiones, las cuales el servidor se encarga de administrar y retornar una respuesta con base en los parámetros de la petición.

Una de las características que tiene esta arquitectura es que permite tener a muchos clientes enviando peticiones a un servidor central, por lo que la escalabilidad del sistema es muy grande al permitir agregar clientes de manera dinámica.

Para el desarrollo de la aplicación están los teléfonos móviles con sistema operativo Android como los clientes que van a consumir, el servidor que estará alojado en una máquina virtual alojada en el servicio en la nube de Google. Esta máquina virtual tomará el rol de servidor web, el cual estará desarrollado con el Framework NodeJS y tendrá un gestor de bases de datos Postgresql. El modelo de la base de datos se detalla en la sección 2.3.3.

Figura 27. **Arquitectura del sistema**



Fuente: elaboración propia, empleando web draw.io.

3.3. **Diseño de la base de datos**

Para el diseño de la base de datos se utilizará el modelo entidad relación. Esta es una herramienta que permite representar entidades en una base de datos.

Una entidad es una cosa, objeto o concepto del mundo real que tiene una existencia independiente; es decir, que se diferencia de otro objeto o cosa. Esta independencia se logra gracias a los atributos que tiene cada una de ellas.

Los atributos son características que definen a una entidad, estas no tienen un número específico ni son exclusivas de la entidad, por lo que puede existir entidad con atributos iguales. Estos atributos tienen tipos de datos, los cuales van a depender de las características de la entidad.

3.3.1. Conjunto de relaciones

Las relaciones se basan en cardinalidades, las cuales se dan entre relaciones binarias. Las cardinalidades que existen son las siguientes:

- Uno a uno: esta es una entidad A que se relaciona con otra entidad B con un solo registro.
- Uno a varios: es un registro de una entidad A que se relaciona con uno o muchos registros de una entidad B.
- Varios a uno: una entidad A se puede relacionar con un solo registro de la entidad B, pero un registro de la entidad B puede tener muchas relaciones con la entidad A.

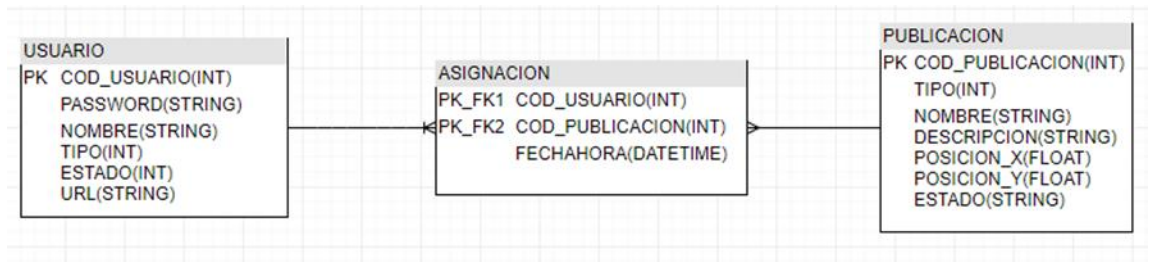
Para el diseño del modelo entidad relación de la aplicación ALERTAUSAC, se identificaron las siguientes entidades:

Tabla I. **Entidades Aplicación ALERTAUSAC**

Entidad	Descripción	Atributos
Usuario	Esta entidad representa los roles de usuario administrador y usuario registrado	Código usuario Código Único Identificación Contraseña Tipo de usuario Nombre usuario Estado usuario Url fotografía
Publicación	Representa la entidad encargada de almacenar la información de los eventos y publicaciones.	Código publicación Tipo de publicación Descripción Coordenada X Coordenada Y Nombre publicación Estado publicación
Asignación	Representa la entidad que relaciona las publicaciones y los usuarios.	Código Publicación Código Usuario Fecha publicación Hora publicación

Fuente: elaboración propia, empleando Software Microsoft Word 2019.

Figura 28. **Modelo entidad relación de la aplicación ALERTAUSAC**



Fuente: elaboración propia, empleando página web draw.io.

3.4. Diseño de comunicación entre componentes

Los componentes que serán implementados son los servicios que tendrá expuesto el servidor para poder comunicarse con la aplicación móvil. Para esto se usará RESTFUL Services. A continuación, se establece un estándar en la forma de declarar los servicios que serán expuestos, de los cuales se tendrá la definición de la acción al principio de la ruta:

- usuarioAU: este servicio proveerá las operaciones básicas para administrar los usuarios, y son los siguientes: obtener (GET), crear usuario (POST), actualizar (PUT), eliminar (DELETE).

Tabla II. Estándar rutas servicio usuario

Nombre	Ruta	Tipo Comunicación
Obtener Usuario	/get/usuarioAU	GET
Crear Usuario	/post/usuarioAU	POST
Actualizar Usuario	/put/usuarioAU	PUT
Eliminar Usuario	/delete/usuarioAU	DELETE

Fuente: elaboración propia, empleando Software Microsoft Word 2019.

Cada una de las rutas va definida con los distintos parámetros que necesiten enviar para realizar el filtrado de datos.

- publicacionAU: este servicio proveerá las operaciones básicas para administrar las publicaciones, las cuales son las siguientes: obtener (GET), crear publicación (POST), actualizar (PUT), eliminar (DELETE).

Tabla III. **Estándar rutas servicio publicaciones**

Nombre	Ruta	Tipo Comunicación
Obtener Publicación General	/get/publicacionGAU	GET
Obtener Publicación Usuario	/get/publicacionUAU	GET
Crear Publicación	/post/publicacionAU	POST
Actualizar Publicación	/put/publicacionAU	PUT
Eliminar Publicación	/delete/publicacionAU	DELETE

Fuente: elaboración propia, empleando Software Microsoft Word 2019.

Definición rutas especiales:

Estas rutas nos ayudarán a la gestión de operaciones dentro de la aplicación, Son de uso común y es en base a la interacción que se dé en la aplicación.

Tabla IV. **Estándar rutas servicios especiales**

Nombre	Ruta	Tipo Comunicación
Iniciar Sesión	/post/iniciar sesión	POST
Registro	/post/registro	POST

Fuente: elaboración propia, empleando Software Microsoft Word 2019.

3.5. Usabilidad

Al momento de desarrollar una aplicación para teléfonos móviles se puede llegar a descuidar el factor de la usabilidad, por lo que este es uno de los puntos centrales de la aplicación ALERTAUSAC, ya que busca ser una aplicación robusta, eficiente y eficaz en los procesos que tiene.

4. DOCUMENTACIÓN DEL SISTEMA

En este capítulo vamos a describir una parte muy importante en el diseño e implementación del sistema y es la documentación, la cual permitirá a los usuarios tener una idea general del uso de la aplicación, así como conocer las funcionalidades y tener un registro de los procesos y servicios de la misma aplicación.

4.1. Manual de usuario

En esta sección vamos a detallar la documentación para el usuario, donde se describirán todas las funciones principales que tiene el sistema para que el usuario lo utilice como una guía.

4.1.1. Descargar aplicación

Para que el usuario tenga acceso a la aplicación debe descargarla en la tienda de aplicaciones de Google, la cual se encuentra instalada en cualquier teléfono con sistema operativo Android.

Tras ingresar a la tienda de aplicaciones, en la sección de búsqueda, se debe ingresar el nombre de la aplicación, "ALERTAUSAC".

Al encontrar la aplicación se procede a la descarga e instalación en el teléfono.

4.1.2. Registro en el sistema

Posteriormente a la instalación de la aplicación en el dispositivo, procederemos al registro en el sistema, por lo que seguiremos los siguientes pasos:

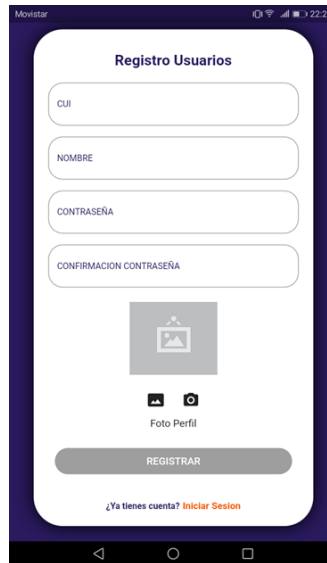
- Al momento de abrir la aplicación nos dará distintas opciones como menú principal. Vamos a la sección de “¿Aún no tienes cuenta? Regístrate”. Al presionar esta opción despliega el formulario que se debe completar para registrarnos en la aplicación.
- Este formulario tiene los siguientes campos: CUI, nombre, contraseña, confirmación de contraseña y foto de perfil. Para poder registrar una foto de perfil hay dos opciones: elegir una fotografía de la galería o tomar una fotografía en ese momento. Todos los campos son obligatorios, por lo que se deberán completar para realizar el registro de manera exitosa.

Figura 29. **Pantalla principal**



Fuente: elaboración propia.

Figura 30. **Pantalla registro**



Fuente: elaboración propia.

4.1.3. Inicio de sesión

Para iniciar sesión se ingresa el CUI y contraseña con el cual nos registramos previamente. Si existiera algún campo incorrecto, la aplicación nos dará una alerta, y si el ingreso es exitoso, permitirá ingresar a las funcionalidades. Para ingresar existe la opción de ingresar como invitado, pero esta opción solo permite visualizar las publicaciones y no realizar publicaciones dentro de la aplicación.

Figura 31. **Pantalla principal**

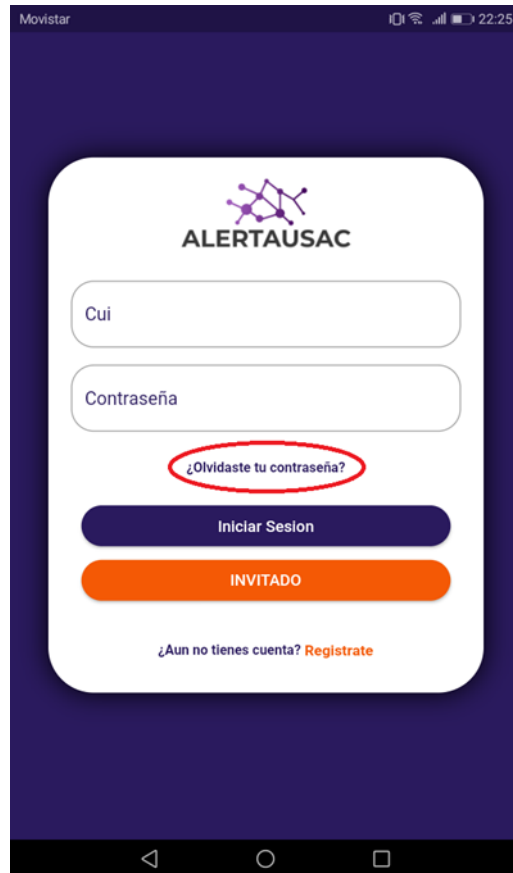


Fuente: elaboración propia.

4.1.4. **Recuperar contraseña**

En la sección de inicio de sesión encontramos una opción de “¿Olvidaste tu contraseña?”. Esta nos permitirá recuperar nuestra contraseña al ingresar el CUI, por lo que mostrará una alerta con los datos del usuario. Si el CUI no existe, mostrará una alerta con el mensaje de error.

Figura 32. **Pantalla Principal**



Fuente: elaboración propia.

Figura 33. **Pantalla recuperar**



Fuente: elaboración propia.

4.1.5. Publicación de alertas y eventos

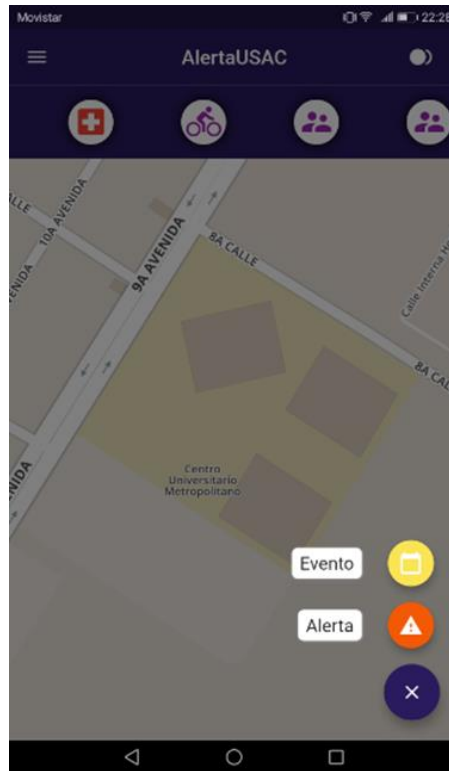
Para publicar una alerta o evento debemos tener sesión iniciada. En la parte del mapa hay un botón en la esquina inferior derecha, el cual al momento de pulsarlo despliega dos opciones para realizar una publicación.

Figura 34. Pantalla mapa



Fuente: elaboración propia.

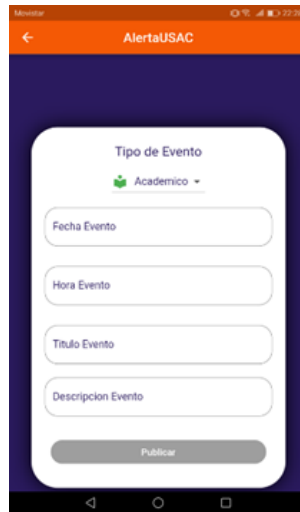
Figura 35. **Pantalla mapa**



Fuente: elaboración propia.

Para publicar un evento, nos mostrará el formulario donde vamos a ingresar el tipo, fecha, hora, título y descripción. Al completar los datos habilitará el botón para crear la publicación. Para la creación de una alerta, los campos del formulario son tipo, título y descripción. Para crear un evento o una alerta tenemos que activar el GPS de nuestro teléfono, ya que reconoce la ubicación de manera automática.

Figura 36. **Pantalla evento**



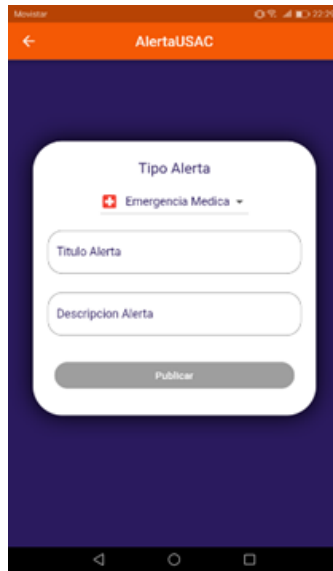
Fuente: elaboración propia.

Figura 37. **Pantalla tipo evento**



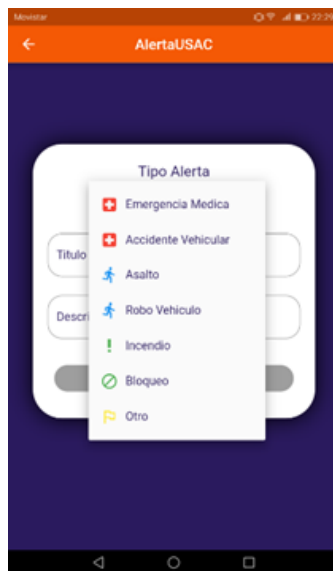
Fuente: elaboración propia.

Figura 38. **Pantalla alerta**



Fuente: elaboración propia.

Figura 39. **Pantalla tipo alerta**

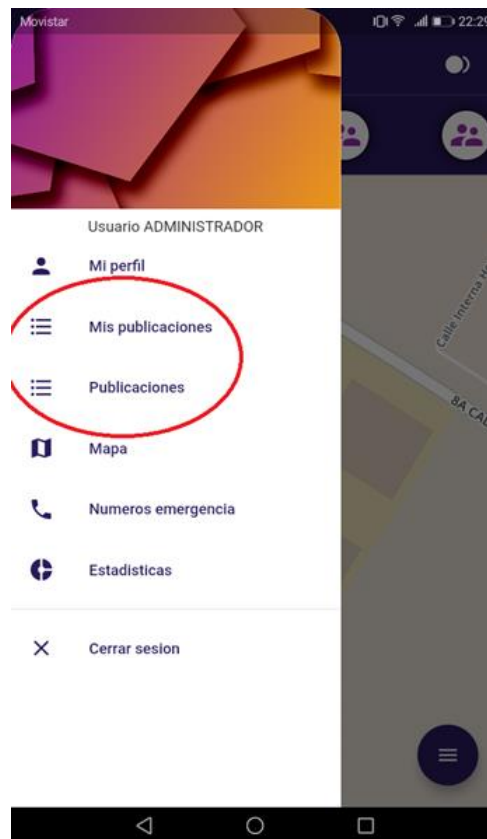


Fuente: elaboración propia.

4.1.6. Administración de publicaciones

Para administrar las publicaciones que se realicen, ya sean propias o por parte del administrador, vamos a la sección del menú de la aplicación donde están las secciones de publicaciones y mis publicaciones. En ambos casos mostrará una lista de las publicaciones que se han realizado hasta el momento. Cada una contará con todos los datos ingresados y permitirá ver, editar y eliminar las publicaciones.

Figura 40. Menú aplicación



Fuente: elaboración propia.

Figura 41. **Pantalla publicaciones**

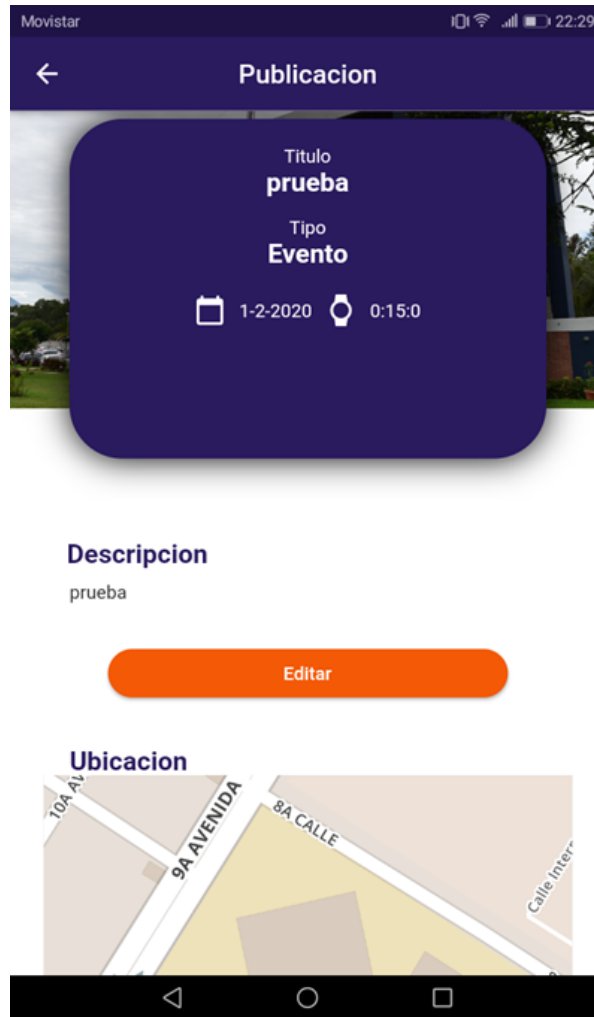


Fuente: elaboración propia.

Al momento de ver la publicación mostrará la información completa y un mapa con la ubicación de la publicación.

Para editar una publicación nos desplegará el formulario para modificar los datos que sean necesarios. Para eliminar una publicación nos pedirá una confirmación y la eliminará del listado.

Figura 42. Pantalla ver publicación

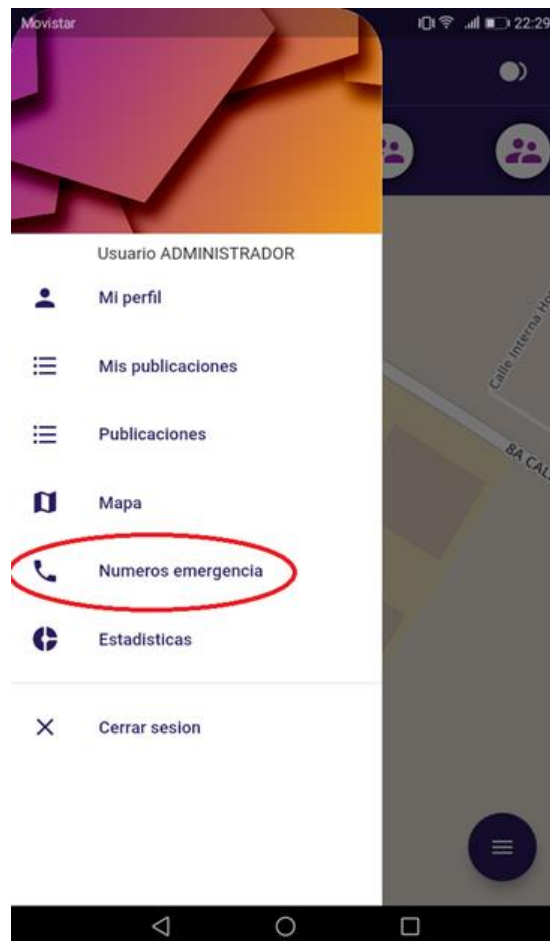


Fuente: elaboración propia.

4.1.7. Números de emergencia

Para ver los números de emergencia hay una opción en el menú para los tres tipos de usuarios, en el cual se detalla una lista con los números telefónicos. Al momento de presionar el icono para llamada, se desplegará el número para realizarla.

Figura 43. Menú aplicación



Fuente: elaboración propia.

Figura 44. **Pantalla emergencia**

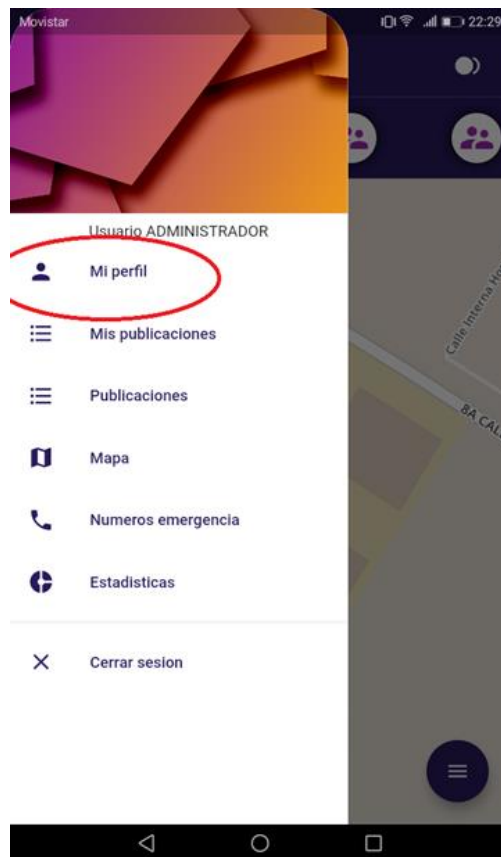


Fuente: elaboración propia.

4.1.8. Perfil usuario

Para visualizar la información del usuario, vamos a la sección de mi perfil en el menú de la aplicación. En esta pantalla nos mostrará toda la información del usuario y permitirá modificarla por medio de un formulario.

Figura 45. Menú aplicación



Fuente: elaboración propia.

Figura 46. **Pantalla perfil**

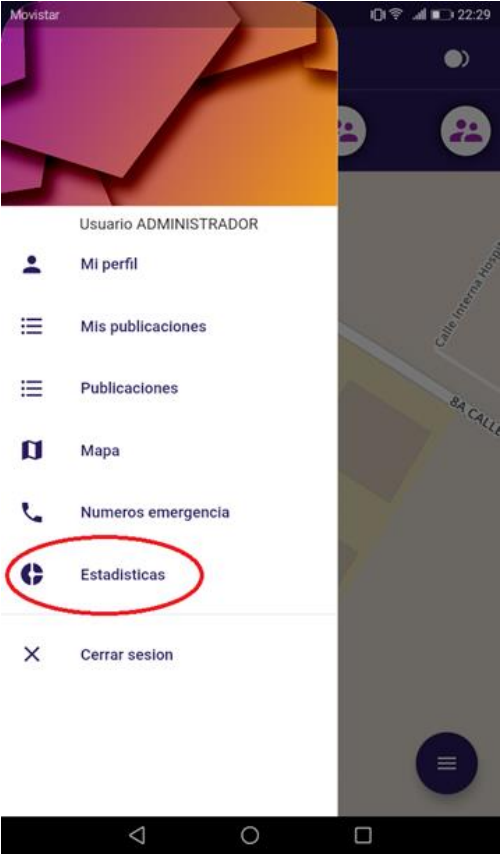


Fuente: elaboración propia.

4.1.9. Estadísticas

Para visualizar las estadísticas de la aplicación, vamos a la sección estadísticas del menú de la aplicación, en esta nos mostrará dos datos, el número de publicaciones por tipo y el número de alertas por medio del botón de pánico.

Figura 47. Menú aplicación



Fuente: elaboración propia.

Figura 48. **Pantallas estadísticas**



Fuente: elaboración propia.

4.2. Manual técnico

En esta sección vamos a detallar las generalidades técnicas de la aplicación, abordando temas de requerimientos, servicios que se tiene para futuras mejoras.

4.2.1. Requerimientos mínimos

Los requerimientos mínimos para utilizar el sistema es una parte importante, ya que nos permite proyectar si los recursos con los que contamos son suficientes o si tenemos que adquirir más para suplir las necesidades.

4.2.1.1. Requerimientos recomendados para la instalación de la aplicación

- Sistema Operativo: Android 5.1.1
- Espacio Disponible: 35 mb.

4.2.1.2. Versión de herramientas

- Flutter: 1.12.13+hotfix.5
- Dart: 2.7.0
- NodeJs: 10.13-alpine
- Postgresql: 12
- Visual Studio Code: 1.41.1

4.2.1.3. Paquetes instalados flutter

Para el desarrollo de la aplicación se instalaron de los siguientes paquetes:

Tabla V. Paquetes instalados flutter

Paquete	Versión
cupertino_icons:	^0.1.2
flutter_circular_chart:	^0.1.0
mime_type:	^0.2.4
http:	^0.12.0+2
image_picker:	^0.6.3+1
url_launcher:	^5.4.1
shimmer:	^1.0.1
flutter_speed_dial:	^1.2.5
geolocator:	^5.1.5
shared_preferences:	^0.5.6
flutter_swiper:	^1.1.6
rxdart:	^0.22.6
latlong:	^0.6.1
flutter_map:	^0.8.2

Fuente: elaboración propia, empleando Software Microsoft Word 2020.

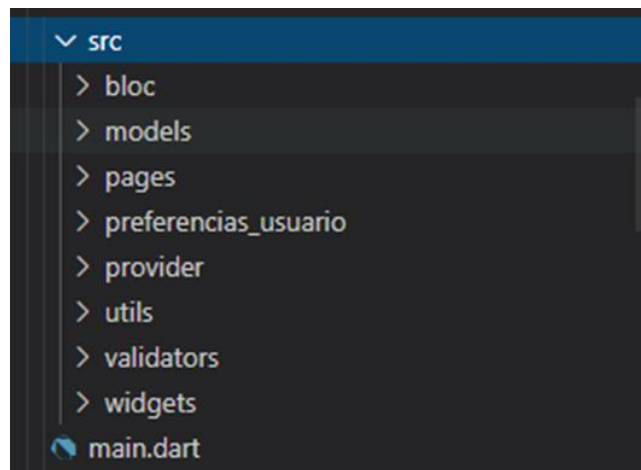
4.2.1.4. Servidor Google Cloud

- Tipo: N1-Estandar
- Almacenamiento: 10 Gb
- Memoria Ram: 3.75 GB
- Procesador: 1vCPU Intel Haswell
- Sistema Operativo: Ubuntu 18.04

4.2.2. Gestión de archivos

Se implementó una distribución de archivos y carpetas con la finalidad de desacoplar todos los servicios involucrados en el desarrollo de la aplicación. Las carpetas principales son las siguientes:

Figura 49. **Gestión de archivos**



Fuente: elaboración propia.

- **Bloc:** esta carpeta centraliza todos los archivos que manejan el estado de los formularios, tanto de los usuarios como de las publicaciones.
- **Models:** en esta parte se almacenan los modelos de los tablas y consultas que se realizan a la base de datos para mapearlas y manipular los datos.
- **Pages:** esta parte almacena todos los archivos de las pantallas de la aplicación. No cuenta con lógica de negocio, sino únicamente con la interfaz de la aplicación.

- **Preferencias_usuario:** esta sección nos permite administrar los datos que van a persistir en la aplicación y que son necesarios para el funcionamiento correcto del servicio.
- **Provider:** esta parte contiene la lógica de negocio y la conexión con los servicios del servidor, por lo que es la capa de comunicación.
- **Utils:** son componentes reutilizables dentro de la aplicación como alertas, notificaciones.
- **Validators:** esta sección permite reunir todas las validaciones de los formularios.
- **Widgets:** permite almacenar componentes reutilizables o de uso común para el funcionamiento de la aplicación.

4.2.3. Servicios principales

En esta sección vamos a describir los servicios principales que componen la aplicación, dando a conocer de manera general su funcionalidad.

Obtener publicaciones generales: este servicio nos permite obtener las publicaciones generales que están en la base de datos, con el fin de obtener todos los datos necesarios para poder mostrar las publicaciones. Tiene varios filtros, ya que por medio de la paginación permite obtener resultados de 10 en 10 para no saturar los servicios de la aplicación.

Figura 50. Servicio para obtener publicaciones generales

```
async function listado_publicaciones_generales(paginacion){
  var query="SELECT usu.cui,usu.nombre,publi.cod_publicacion,publi.tipo,publi.nombre,publi.descripcion,publi.posicion_x,publi.posicion_y,publi.fechahora,publi.subtipo,publi.fechahora "+
  "FROM usuario usu "+
  "INNER JOIN asignacion asi "+
  "ON usu.cod_usuario = asi.cod_usuario "+
  "INNER JOIN publicacion publi "+
  "ON asi.cod_publicacion = publi.cod_publicacion ";
  query+=" ORDER BY publi.fechahora DESC ";
  if(paginacion==0){
    query+="LIMIT 10 OFFSET "+paginacion;
  }
  const respuesta=await conexionbd.client.query(query)
  .then(res=>{
    if(res.rowCount==0) return [];
    return res.rows;
  }).catch(e=>{
    console.log(e);
    return {"codigo":501,"mensaje":"Error al momento de buscar las publicaciones generales, intente nuevamente"};
  });
  return respuesta;
}
```

Fuente: elaboración propia.

Obtener publicaciones usuarios: Este servicio permite obtener las publicaciones de un usuario en específico por medio del CUI, además, tiene el filtro de paginación que permite retornar valores de 10 en 10 para no saturar los servicios de la aplicación.

Figura 51. Servicio obtener publicaciones usuarias

```
async function listado_publicaciones_usuario(cui,paginacion){
  var query="SELECT usu.cui,usu.nombre,publi.cod_publicacion, publi.tipo,publi.nombre,publi.descripcion,publi.posicion_x,publi.posicion_y,publi.fechahora,publi.subtipo "+
  "FROM usuario usu "+
  "INNER JOIN asignacion asi "+
  "ON usu.cod_usuario = asi.cod_usuario "+
  "INNER JOIN publicacion publi "+
  "ON asi.cod_publicacion = publi.cod_publicacion "+
  "WHERE usu.cui="+cui+" "+
  "ORDER BY publi.fechahora DESC "+
  "LIMIT 10 OFFSET "+paginacion;
  const respuesta=await conexionbd.client.query(query)
  .then(res=>{
    if(res.rowCount==0) return 0;
    return res.rows;
  }).catch(e=>{
    console.log(e);
    return {"codigo":501,"mensaje":"Error al momento de buscar las publicaciones del usuario, intente nuevamente"};
  });
  return respuesta;
}
```

Fuente: elaboración propia

Subir foto de perfil: este servicio permite subir la imagen de perfil del usuario que se registra y de la actualización del perfil. Almacena la imagen en

una carpeta y genera un código único para identificarla. Este servicio retorna el nombre de la imagen para consumirla desde la ip del servidor.

Figura 52. Servicio foto perfil

```
app.post('/post/imagenAU',function(req,res){
  try{
    if(!req.files){
      res.json({
        "codigo": 501,
        "mensaje": "Necesita agregar una fotografia"
      });
    }
    console.log(req.files);
    var UID =Math.floor(Math.random() * 999999);
    req.files.AVATAR.mv('./uploads/' +UID+'_'+req.files.AVATAR.name.replace(/ /g, "_"));
    res.json({"codigo":200,"mensaje":"correcto","url":UID+'_'+req.files.AVATAR.name.replace(/ /g, "_")});
  }catch(err){
    console.log(err);
    res.json({"codigo":500,"mensaje":"Ocurrio un error, al crear la fotografia "});
  }
});
```

Fuente: elaboración propia.

Registrar publicación: este servicio permite registrar una publicación en la aplicación. Recibe todos los datos de los formularios y permite verificar la validación de los campos.

Figura 53. Servicio registrar publicación

```
async function registrarPublicacion(tipo, nombre, descripcion, posicionX, posicionY, estado,cui,subtipo,fechahora){
  if(tipo==''||nombre==''||descripcion==''||posicionX==''||posicionY==''||estado==''||cui==''||subtipo==''||fechahora=='')return {"codigo":402,"mensaje":"Datos Incompletos para realizar la publicacion, intente nuevamente"};
  var query="select CREAM_PUBLICACION('"+tipo+"','"+nombre+"','"+descripcion+"','"+posicionX+"','"+posicionY+"','"+estado+"','"+cui+"','"+subtipo+"','"+fechahora+"')";
  console.log(query);
  const respuesta=await conexionbd.client.query(query)
  .then(async res=>{
    return {"codigo":200,"mensaje":"Publicacion Creada con Exito"};
  })
  .catch(e=>{
    console.log(e.toString());
    return {"codigo":501,"mensaje":"Error al momento de registrar la publicacion, intente nuevamente"};
  });
  return respuesta;
}
```

Fuente: elaboración propia.

CONCLUSIONES

1. Implementar un mapa interactivo que centraliza todas las publicaciones y permite ver cada una de ellas y brinda una perspectiva concisa de los puntos dentro del Centro Universitario Metropolitano que generan más publicaciones. Esto es gracias a la ubicación por medio del sistema de posicionamiento global (GPS) del teléfono inteligente, por lo que permite identificar las zonas con mayor riesgo de ocurrir alertas.
2. Por medio de la aplicación ALERTAUSAC podemos visualizar, crear y monitorear cada una de las publicaciones. Los usuarios tendrán acceso de manera inmediata a la información que se esté generando, por lo que permite crear un flujo de comunicación directo y brinda un medio de difusión masivo para todos los usuarios que interactúen con la aplicación.
3. Una de las características principales de la aplicación ALERTAUSAC es ser un medio que contiene información que está en constante actualización y que pueden visualizar los usuarios. Por esto, la arquitectura de la aplicación y servidor debe capaz de manejar la carga de trabajo y permitir escalar los servicios al momento de ampliar la cobertura de la aplicación. La arquitectura cliente servidor nos brinda los beneficios de comunicación y rendimiento, al ser una arquitectura robusta y altamente escalable.

RECOMENDACIONES

1. Para mantener la fidelidad de los datos que se ingresan en la aplicación se recomienda crear una conexión con la base de datos de usuarios de la Universidad de San Carlos, ya que permitiría verificar de manera confiable la información de las personas que se registran.
2. Tanto la aplicación como el servidor tienen una arquitectura escalable, la cual permitirá la implementación del sistema tanto en el campus central como en distintos centros regionales. Permitiría centralizar todos los datos para un mejor control y monitoreo de los acontecimientos que sucedan en la universidad.
3. Al ser un servicio de centralización de información permite integrar otros servicios de difusión oficiales para establecer un canal de comunicación confiable y que mantenga al día de los acontecimientos a toda la comunidad estudiantil y administrativa de la universidad.
4. Enlazar la aplicación con el sistema de seguridad interna de la universidad para tener una respuesta inmediata y un mejor control y monitoreo.
5. Ingresar publicaciones en la aplicación permite generar datos que pueden ser utilizados para un estudio estadístico y determinar zonas rojas dentro de la universidad y centros regionales.

BIBLIOGRAFÍA

1. Deloitte. *Metodologia Waterfall vs Agile*. [en línea] <<https://www2.deloitte.com/es/es/pages/technology/articles/waterfall-vs-agile.html#>> [Consulta: 15 de septiembre de 2019].
2. Flutter. *Flutter Documentation*. [en línea] <<https://flutter.dev/docs>> [Consulta: 30 agosto 2019].
3. Google Cloud. *Google Cloud Documentacion*. [en línea] <<https://cloud.google.com/docs/>> [Consulta: 12 septiembre 2019].
4. NodeJS. *Nodejs Documentation*. [en línea] <<https://nodejs.org/es/docs/>> [Consulta: 10 septiembre 2019].
5. PostgreSQL. *Postgresql Documentation*. [en línea] <<https://www.postgresql.org/docs/>> [Consulta: 10 septiembre 2019].
6. RIVAS, Carlos Ignacio, et al. *Metodologías actuales de desarrollo de software*. vol. 2, no 5. México. Revista de Tecnología e Innovación, 2015. p. 980-986.

APÉNDICE

Apéndice 1. **Presentación de la aplicación ALERTAUSAC, Asociación Estudiantes Medicina-AEM**



Fuente: elaboración propia.

