

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingeniero de Sistemas**

**TEMA:
DESARROLLO DE UNA
APLICACIÓN WEB QUE PERMITA EL MONITOREO DE INCIDENTES A
TRAVÉS DE UN GIS, UTILIZANDO UNA APLICACIÓN A SER
DESARROLLADA PARA DISPOSITIVOS MÓVILES CON SISTEMA
OPERATIVO ANDROID, QUE PERMITA REGISTRAR LA
INFORMACIÓN GEORREFERENCIADA.**

**AUTOR:
JHON XAVIER VALENCIA SÁNCHEZ**

**TUTOR:
ROBINSON DIMITRI LLERENA PAZ**

Quito, mayo del 2018

CESIÓN DE DERECHOS DE AUTOR

Yo Valencia Sánchez Jhon Xavier, con documento de identificación N° 1719808618, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy el autor del trabajo de titulación con el tema: “DESARROLLO DE UNA APLICACIÓN WEB QUE PERMITA EL MONITOREO DE INCIDENTES A TRAVÉS DE UN GIS, UTILIZANDO UNA APLICACIÓN A SER DESARROLLADA PARA DISPOSITIVOS MÓVILES CON SISTEMA OPERATIVO ANDROID, QUE PERMITA REGISTRAR LA INFORMACIÓN GEORREFERENCIADA.”, mismo que ha sido desarrollado para optar por el título de: INGENIERO DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada.

En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
VALENCIA SÁNCHEZ
JHON XAVIER
CI: 1719808618

Quito, mayo del 2018

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: DESARROLLO DE UNA APLICACIÓN WEB QUE PERMITA EL MONITOREO DE INCIDENTES A TRAVÉS DE UN GIS, UTILIZANDO UNA APLICACIÓN A SER DESARROLLADA PARA DISPOSITIVOS MÓVILES CON SISTEMA OPERATIVO ANDROID, QUE PERMITA REGISTRAR LA INFORMACIÓN GEORREFERENCIADA., realizado por Jhon Xavier Valencia Sánchez, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, mayo del 2018.



ROBINSON DIMITRI LLERENA PAZ
CI: 1707106942

Dedicatoria

El presente proyecto se los dedico a mis padres y mi hermana Camila Valencia ya que con su apoyo incondicional he podido vencer todos los obstáculos que se atravesaron a lo largo de mis estudios, obstáculos que me ayudaron a superar intelectualmente y socialmente, su ayuda me ha fortalecido en cada uno de mis aspectos tales son mis actitudes y mis virtudes. Las conversaciones por largas horas, me ayudan a seguir día a día pensando en lo mejor y tratando de aprovechar las oportunidades, así estas no sean fáciles con el objetivo de superarme. Por todo aquello les doy esta pequeña dedicatoria llena de mucho cariño y a la vez mucha emoción por este nuevo peldaño en mi vida.

Logrando así esta meta muy importante en mi vida además gracias a ellos ahora soy quien soy. Solo me queda decirles “Muchas gracias y un Dios les pague por tanto apoyo”.

Agradecimiento

Agradezco a la Universidad Politécnica Salesiana que ha contribuido en mi formación académica con un grado intelectual muy alto y por sus recursos excepcionales tales como sus agradables bibliotecas, laboratorios y otros que ha favorecido en mi formación profesional.

A mi tutor el Ingeniero Robinson Llerena por su ayuda en la planificación de este proyecto, ya que con su conocimiento se pudo lograr un objetivo de mucha ayuda para el público en general y el compartir de ideas nuevas.

ÍNDICE

Problema	2
Justificación.....	2
Objetivo Grupo	3
Objetivo general	4
Objetivos Específicos.....	4
Alcance del proyecto.....	5
Prototipos	6
Capítulo 1	7
1.1 Antecedentes	7
1.2 Metodología	10
1.3 Ingeniería de software	11
1.4 Procesos de software	13
1.5 Programación Extrema.....	14
1.5.1 Riesgos de la programación extrema	14
1.5.2 Actividades de la programación extrema.....	15
1.5.3 Valores de la programación extrema.....	16
1.5.4 Fases de una programación extrema	17
1.6 Lenguajes de programación	18
1.6.1 JavaScript.....	18
1.6.1.1 Angular JS.....	19
1.6.1.1.1 Plantillas.....	20
1.6.1.1.2 Estructura MVC	20
1.6.1.1.3 Estructura MVC en AngularJS.....	21
1.6.2 JSON	22
1.7 Herramientas	23
1.7.1 Firebase	23
1.7.2 Netbeans IDE	25
1.7.3 Android Studio	26
1.9 Sistema de Información gráfica	27
1.10 Prototipos	27
Capítulo 2.....	29
2.1 Detalle de requerimientos	29
2.2 Diagrama de actividades	29
2.2.1 Diagrama de actividades aplicación web	30
2.2.2 Diagrama de actividades aplicación móvil	32
2.2.3 Diagrama de actividades aplicación general.....	33

2.3 Casos de uso.....	35
2.3.1 Caso de uso aplicación web	35
2.3.2 Caso de uso aplicación móvil.....	36
2.4 Diagrama de clases.....	36
2.4.1 Diagrama de clases aplicación web.....	37
2.4.1 Diagrama de clases aplicación móvil.....	38
2.5 Diagrama de base de datos	38
2.6 Interfaces prototipos.....	40
2.6.1 Interfaces aplicación web	41
2.6.2 Interfaces aplicación móvil	43
Capítulo 3.....	48
3.1 Sistema web	48
3.1.1 Arquitectura.....	48
3.1.2 Código principal.....	49
3.2 Sistema Móvil	52
3.2.1 Arquitectura.....	52
3.3 Pruebas	53
3.3.1 Módulo pruebas aplicación móvil.....	53
3.3.2 Módulo de pruebas aplicación web.....	57
CONCLUSIONES	61
RECOMENDACIONES	62
LISTA DE REFERENCIAS	64

ÍNDICE DE FIGURAS

Figura 1. App Help Me interfaz inicial	8
Figura 2. App inicial de iRescue	10
Figura 3. Arquitectura de diseño de Angular JS.	21
Figura 4. Ejemplo de formato de un diseño JSON.....	22
Figura 5. Diagrama de actividades de la aplicación web.	30
Figura 6. Diagrama de actividades aplicación móvil.	32
Figura 7. Diagrama de actividades aplicación general.....	34
Figura 8. Caso de uso aplicación web.	35
Figura 9. Caso de uso aplicación móvil.	36
Figura 10. Caso de uso aplicación web.	37
Figura 11. Diagrama de clases aplicación móvil.	38
Figura 12. Diagrama de base de datos no SQL.....	39
Figura 13. Prototipo interfaz ingreso.	41
Figura 14. Prototipo interfaz visualización de datos.....	42
Figura 15. Prototipo interfaz reporte de datos.....	43
Figura 16. Interfaz de bienvenida de la aplicación móvil.	44
Figura 17. Interfaz de registro de la aplicación móvil.	44
Figura 18. Interfaz registro de tipo de incidente de la aplicación móvil.....	45
Figura 19. Interfaz para obtener la fotografía del incidente de la aplicación móvil. 46	
Figura 20. Interfaz para obtener la ubicación del incidente de la aplicación móvil.. 46	
Figura 21. Interfaz para finalizar el incidente de la aplicación móvil.....	47
Figura 22. Diagrama de arquitectura de la aplicación web	48
Figura 23. Librerías implementas en la aplicación web.....	50
Figura 24. Librerías Firebase implementas en la aplicación web.	50
Figura 25. Código principal para aplicación web.	51
Figura 26. Arquitectura aplicación móvil	52
Figura 27. Resultado prueba conexión Smartphone	54
Figura 28. Resultado usuario activos	55
Figura 29. Resultados interacción diario de los usuario con pantallas principales	55
Figura 30. Resultados de registros por países.	56
Figura 31. Resultados de versiones de la aplicación.....	58
Figura 32. Resultado de actividades.....	58
Figura 33. Resultado de sesiones activas aplicación web.	59

ÍNDICE DE TABLAS

Tabla 1. Grupo de beneficiarios	3
Tabla 2. Programación extrema frente a riesgos	14
Tabla 3. Fases de la metodología de programación extrema.	17
Tabla 4. Características de Smartphone	54
Tabla 5. Tabla de resultados de pruebas aplicación móvil.....	57
Tabla 6. Tabla de características de una computadora de ejecución del sistema web	57
Tabla 7. Tabla de resultados de las pruebas del sistema web.	59

Resumen

El desarrollo de este proyecto, se pretende aplicar todos los conocimientos adquiridos durante el periodo de estudio en la carrera de Ingeniería en Sistemas e implementar tecnologías de desarrollo de software que permita registrar información mediante una aplicación móvil y una aplicación web para la obtención de los registros almacenados en una base de datos noSQL en la nube, utilizando un sistema de información geográfica. Las tecnologías utilizadas en el desarrollo de software se acoplan para obtener una aplicación única y útil. Conjuntamente el presente proyecto pretende ayudar a toda la ciudadanía del Ecuador que se encuentra en un momento catastrófico, como puede ser un accidente de tránsito, un asalto o cualquier evento de esa índole, permitiendo obtener una ayuda oportuna, eficaz y rápida.

Por otro lado, facilite al personal administrador de la aplicación web visualizar los registros de forma amigable con una comunicación estable, en tiempo real y una información eficiente para realizar las respectivas intervenciones por parte del personal.

Abstract

This document contains information about methodologies, tools, and procedures for to build quality software and meet customer's requirements, trying to meet all the needs of a software project with nice interface, a safe development and custom software for the client.

The project has a many different ideas, but the principal is to register an incident and get the best information with a smartphone. The register captured with a smartphone has a many parts, for example the users need to get photography in this moment, information about this accident, get a exactly position and the level of the incidente. The information collected is processed in a type of plain text format called JSON, which allows obtaining records according to an adequate order that complies with this very light text format, the process ends with the registration of information is stored on a database not SQL. Firebase offers a database that supports the aforementioned format and adapting to certain rules that the end user requires.

This register send to tool of Firebase, and the data save in database in the real time, the project content two parts, the first is to programming a website and the other is programming a mobile application, the website represent many registers capture in those moment and the administrator processes the record according to the level of the tragedy and the mobile application records the incidents.

The development of these applications is intended to help the process of registering incidents that can be generated at any time, allowing the administrator of the website to take relevant actions to process these records, easily and very quickly.

INTRODUCCIÓN

Este presente documento contiene todas aquellas técnicas, métodos, herramientas, conocimientos y estructuras generales que ayudan a un buen planteamiento de proyecto de software, cuyo propósito el desarrollo de una aplicación móvil que permita el registro de información de un incidente para dispositivos inteligentes con sistema operativo Android y un aplicación web que accede a la visualización de los registros almacenados por la aplicación móvil, otorgando un proceso de ejecución estable. Es por ello que este proyecto surgió del problema, en la recopilación de la información de accidentes de cualquier índole ya sea por información incorrecta o por una mala ubicación obtenida, para la cual se aplicaran los conocimientos obtenidos durante la formación de un Ingeniero en Sistemas logrando una orientación profesional por parte del tutor. Obteniendo como resultado subsanar este problema de forma más sencilla.

En el capítulo inicial detalla todas las herramientas y metodologías para obtener un sistema exitoso y estructurado, el segundo capítulo definida por un conjunto de herramientas de modelado UML que permite a un sistema ser ordenado de forma muy detallada y establecida en la ingeniería de un software, como es la estructura del sistema y un correcto diseño de la base de datos. Dichas metodologías pretende a un desarrollador de software determinar un camino idóneo durante el proceso de desarrollo del sistema y ofrecer una idea del funcionamiento del sistema. Finalizando con el capítulo tres, que describe líneas de código que interactúa entre el sistema web, la aplicación móvil, la herramienta de Firebase para el registro y la obtención de registros almacenados en la base de datos. Este capítulo pretende ofrecer pruebas predeterminadas y obtener resultados agradables para los usuarios finales.

Problema

Durante el proceso de registro de un incidente es importante la agilidad, la rapidez y eficiencia por parte del personal (bomberos, paramédicos, etc.) de ayuda en estos desastres, que pueden ocasionarse a cualquier momento obteniendo pérdidas materiales o humanas.

La fase del registro de los incidentes pueden generar algunos inconvenientes como por ejemplo información recopilada errónea o a destiempo, ocasionando algún desajuste en el proceso anteriormente mencionado, es por ello que surge el desarrollo de una aplicación móvil y una aplicación web que permita mejorar el proceso de obtener la información del acontecimiento suscitado, cubriendo así la necesidad inicial y sobre todo mejorando el tiempo de respuesta por parte del personal de ayuda.

Justificación

Hoy en día los teléfonos inteligentes se han tornado en un elemento imprescindible en la vida del ser humano, estos elementos nos ayudan con actividades cotidianas mejorando el desempeño ya sea intelectual, físico o emocional, de forma ágil y sencilla.

Es por ello que el desarrollo de la aplicación móvil con sistema operativo Android, permitirá a los usuarios con la aplicación instalada en sus dispositivos móviles:

- Registrar un accidente de cualquier tipo, obteniendo la ubicación exacta, información del acontecimiento, el nivel del incidente y una fotografía generada por la aplicación que se encuentra instalada en el dispositivo móvil,

siendo ésta información almacenada en una base de datos noSQL y en un storage en la nube de Firebase en formato JSON.

Además permitirá a los administradores de la aplicación web:

- Tener un control de los registros almacenados por la aplicación móvil en la base de datos en tiempo real, y la visualización geográfica real de dichos datos.

Objetivo Grupo

El proyecto está dirigido al siguiente grupo de beneficiarios:

Tabla 1. Grupo de beneficiarios

Grupo	Características	Impacto
Administrador de la página web	Asegura que los registros almacenados, sean procesados de acuerdo a un estado. Ayuda a la eficaz toma de decisiones, lo que conlleva a salvar vidas y evitar pérdidas materiales.	Ayudar a la toma de decisiones muy rápidas y eficientes. La información será de ayuda en su desempeño, salvando vidas y evitar pérdidas materiales.
Personas que registran un incidente	Registra algún acontecimiento de interés público.	Los registros serán muy rápidos y de forma sencilla.

Nota: Tabla de grupo de beneficiarios
Elaborado por Valencia Jhon (2018)

Objetivo general

Desarrollar una aplicación móvil y una aplicación web, cuyo objetivo principal es informar la existencia de un incidente a un personal calificado de forma oportuna y rápida, cuyo registro conformado por una fotografía, la ubicación exacta del suceso, el nivel de acontecimiento y una descripción. Finalizando con el almacenamiento de la información recopilada por la aplicación móvil en una base de datos noSQL.

Objetivos Específicos.

Diseñar una aplicación web que sea capaz de dar a conocer la información almacenada en la base de datos de forma visual y agradable al usuario.

Diseñar una aplicación móvil que sea capaz de interactuar con el usuario de una forma ágil sencilla y sobre todo muy rápida, permitiendo el registro de la información capturada.

Analizar una estructura idónea de base de datos en la plataforma de Firebase y sus respectivas seguridades que las amerita.

Obtener un producto que cumpla con todos los requerimientos, además que le permita a los usuarios un buen funcionamiento y desempeño de dicha aplicación. Con tecnologías de alto rendimiento capaces de adaptarse a las metodologías que cumplan con este proyecto.

Alcance del proyecto

Desarrollar un aplicación móvil para registro de datos y una aplicación web para monitoreo de los mismo.

La aplicación móvil cuenta básicamente con un módulo:

Proceso de visualización de registros:

Es el encargado de mostrar a un usuario los puntos exactos donde se realizo un acontecimiento de tipo incidente, este módulo obtiene información de una base de datos almacenada en la nube, cuyos registros son guardados por diferentes usuarios en cualquier ubicación del mundo.

Además de obtener los datos generales de los sucesos, es decir una información exacta constituida por una fotografía, una ubicación puntual, un detalle del usuario que registró este acontecimiento y el nivel del incidente, realiza búsquedas de acuerdo al tipo de incidente ('Accidente de tránsito, incendio u otro'), el nivel de emergencia, por tipo de estado y la acción de cambio de estado.

La aplicación móvil es totalmente distinta a la aplicación web, ya que la app se utiliza de acuerdo a módulos, en los cuales son tratados con un modelo tipo cascada donde todos los módulos son administrados uno por uno.

El proceso de la aplicación móvil básicamente está estructurado de forma de una cascada, donde sus módulos son procesados uno a uno, esto permite un mejor resultado al momento de obtener los registros de datos, los cuales se los detalla a continuación:

El inicio de la app, es una pantalla principal y la información breve de la aplicación.

Tipo de acción del registro, la app permite realizar:

Obtención de un detalle, el tipo de acontecimiento y el nivel que permite registrar una información muy definida del suceso.

Obtener una fotografía, hoy en día todos los Smartphone permite la captura de una fotografía ya sea de cualquier tipo, es por ello que la app móvil puede conseguir una imagen ya sea una tomada con anterioridad o en ese momento, una vez capturado el elemento es procesado y almacenado en la nube, el storage de Firebase permite almacenar cualquier tipo de formato fotográfico.

Obtener la ubicación, así como el recurso de obtener una fotografía, que ofrecer los Smartphone, también permite obtener una ubicación exacta, por medio del GPS, permitiendo así obtener la información en un formato de latitud y longitud los cuales trabajan los elementos de ubicación y después procesado de igual manera a la nube y por último la pantalla de finalización del registro, básicamente informa al usuario que el proceso y la información se encuentra almacenada con éxito.

Prototipos

Los prototipos fueron desarrollados al inicio del proyecto, realizando las pruebas pertinentes y obteniendo cambios durante los pocos prototipos que se fueron desarrollando, a demás agregando nuevos cambios importantes que ayuden un poco más, al beneficio del personal que administra estos productos de software.

Capítulo 1

Marco Metodológico

1.1 Antecedentes

Durante la llegada de teléfonos inteligentes, conocidos como Smartphone, que ofrece muchos servicios a usuarios finales, que ayudan a realizar actividades cotidianas.

Es ahí donde las aplicaciones entran con mucha fuerza, cuyo objetivo principal es satisfacer a los usuarios, con aplicaciones de cualquier tipo. Hoy en día estas aplicaciones cubren la mayoría de campos de estudio, como son diversión, educación, servicio, emergencias, etc. Cuya diferencia de una aplicación a otra es el impacto que ofrece al usuario y a la vez se convierten en necesarias.

A continuación se detalla casos de éxito existentes a nivel mundial, de que cubren la necesidad de emergencias.

Help Me – SOS International (A© It´s posible, 2014)

El aplicativo móvil Help Me – SOS International desarrollado por una empresa española A©It´s posible, para dispositivos iOS y Android permitiendo el registro de accidentes en cualquier parte del mundo.

Objetivo de la aplicación: Desarrolló de una aplicación para dispositivos móviles con cualquier sistema operativo Android o iOS, que permite enviar una ubicación exacta de la persona que necesita una ayuda de tipo emergencia, este registro permite realizar una llamada a emergencias, policía o de cualquier índole, permitiendo a usuarios sean turistas que se encuentran en otro país, la app móvil cuenta con más de

cientos de números telefónico entre policías, bomberos y policiales. Cubriendo así doscientos países a nivel mundial.

Beneficios de Help Me – SOS International a usuarios:

- Permite una descarga desde su página principal.
- Envía directamente la dirección física y coordenada GPS de la ubicación a quien lo necesita.
- Servicios rápidos e inteligentes para salvar vidas.
- Permite el envío de correos electrónicos y SMS a instituciones de emergencias.
- Aplicación de forma muy intuitiva y fácil de uso, permitiendo a los usuarios instalar en cualquier tipo de idioma.

Interfaz inicial de la aplicación Help Me

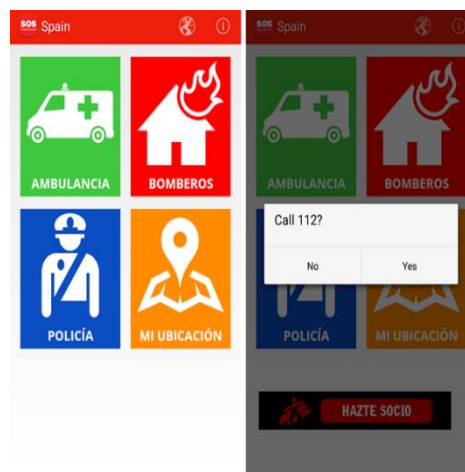


Figura 1. App Help Me interfaz inicial
Fuente: A©It´s posible, 2018.

iRescue(2016)

La aplicación móvil iRecure de forma gratuita para todos los usuario con dispositivos iOS con modelos de Iphone y Ipad permitiendo el envío de información del vehículo accidentado.

Objetivo de la aplicación

Desarrollado para ayudar a personas que se encuentran en los vehículos, sin alguna salida, es por ello la creación de la aplicación ya que permite a los servicios de emergencia enviar una información de dicho vehículo accidentado, con el beneficio de poder ayudar a la o las personas que se encuentran en el interior del vehículo.

Beneficios de iRescue:

- Permite una descarga desde su página principal.
- Disponible en dos idiomas Inglés y Español.
- Dispone con más de 40 marcas y 900 modelos de vehículos para un rápido desempeño en una emergencia.
- Envío de una información técnica de automóvil y ubicación GPS mediante correos electrónicos.
- Información y posición del dispositivo en un mapa en tiempo real.

Interfaz inicial de la aplicación iRescue

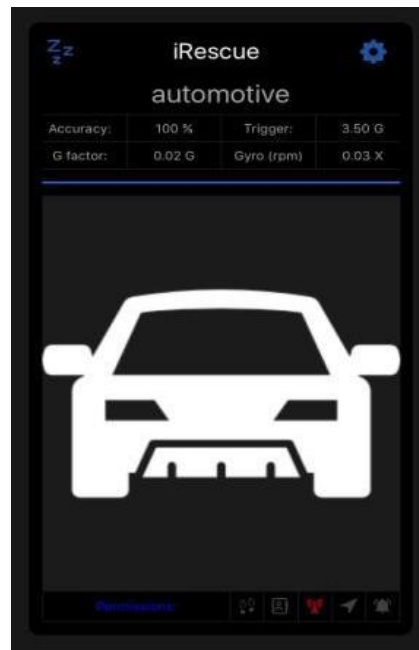


Figura 2. App inicial de iRescue
Fuente: iRescue, 2018.

Las aplicaciones móviles descritas, determinan accidente de cualquier índole, almacenadas en base de datos tradicionales, por ende no permiten realizar modificaciones dentro de ellas, esto puede ocasionar un problema durante un cambio que pueda presentarse a futuro. Es por ello que el presente proyecto pretende evitar estos pequeños problemas y mejorar el rendimiento en tiempo real ofreciendo tecnologías existentes que ayudan al cliente y como al desarrollador, cubriendo con las necesidades inicial de cada proyecto a ser desarrollado.

1.2 Metodología

Una metodología es una modelo de trabajo, se las utiliza como guías para satisfacer las actividades que llevan a cabo, utilizando una secuencia de pasos al camino del éxito del desarrollo de un software. Las metodologías son detalladas de forma ordenada y secuencial, obteniendo las tareas que deben cumplir, los artefactos que se

generara y las relaciones entre ellas. Por lo tanto, permite organizar detalladamente los procesos de un desarrollo de software con pautas y restricciones a seguir. Los procesos son detallados al inicio de cada proyecto.

Existen reglas y limitaciones que son definidas por las metodologías, por ende en el desarrollo de aplicaciones, las reglas pueden llegar a ser un impedimento a cumplir las necesidades de los clientes, es decir que no existe una metodología que pueda ser aplicada con total perfección.

“La Metodología se fundamenta en el uso de técnicas y herramientas de desarrollo y documentación, fundamentalmente gráficas, que facilitan el análisis, el diseño y la programación” (Alonso, 2002).

1.3 Ingeniería de software

“La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas principales de la especificación del sistemas, hasta el mantenimiento de este después de que se utiliza” (Mora, 2015).

Partiendo del concepto anterior, es importante la ingeniería de software, ya que permite una ayuda desde el inicio de un proyecto de desarrollo de software hasta el final, tal es el caso de mantenimientos que se realizan cada periodo de tiempo, para reconocer si el sistemas se encuentra en un estado obsoleto o realizar cambios extras que los pudiese realizar.

Los ingenieros de software, aplican un sin número de métodos, teorías, o herramientas que permitan solucionar los problemas que se encuentran durante el

desarrollo, llamado como “disciplina de la ingeniería”. Es importante que el desarrollo de software se encuentre en buen funcionamiento y debidamente estructurado, que permita a obtener un recurso de calidad y con un grado de impacto muy importante.

La ingeniería de software enfrenta tres retos fundamentales:

- El reto de la heterogeneidad, es importante que los nuevos productos cumplan con las necesidades del cliente pero a la vez también cubran su funcionamiento en sistemas operativos de diferentes tipos y distribuidos en todos los computadores que los requieren. El reto de la heterogeneidad es obtener métodos para desarrollar un software seguro, confiable, eficiente y con un grado de flexibilidad muy alto para cualquier ambiente de producción.
- El reto de la entrega, el tiempo es el gran enemigo de un desarrollo de software. Las técnicas, métodos y otros elementos que surgen, consumen demasiado tiempo si no son claros y eficaces. El reto de la entrega es optimizar los tiempos de entrega, pero que dichos sistemas sean capaces de realizar cambios de forma rápida y de oportuna.
- El reto de la confianza. La finalidad de un desarrollo de software es obtener un producto que cumpla con requisitos indispensables de una persona, empresa u otros, ayudando en las labores cotidianas de la vida diaria, permitiendo ser confiable a la hora de su uso. El reto de la confianza es obtener recursos, técnicas y objetivos que muestren a los usuarios finales pueden confiar plenamente en este nuevo producto.

En general, la ingeniería de software trata de cubrir la mayor parte de aspectos de la producción de software, contando con procesos técnicos, actividades de gestión de proyectos y herramientas que permiten a ingenieros obtener un camino práctico, sencillo y de gran utilidad, cumpliendo así los objetivos estructurados al inicio de cada desarrollo de software.

1.4 Procesos de software

Los procesos de software involucran un conjunto de actividades y pasos a seguir de forma estructurada y ordena a la vez, obteniendo como resultado final un software de calidad. Las actividades son planificadas y ejecutadas por los ingenieros de software, existen cuatro actividades muy importantes en el desarrollo del software, los cuales son:

- Especificación de software: la etapa define los requerimientos, funcionalidades y objetivos que debe cumplir un software.
- Desarrollo del software: Permite el desarrollo, cumpliendo los requisitos de la etapa inicial en un límite de tiempo definido.
- Validación del software: Comprende un análisis de software, permitiendo verificar si el desarrollo anterior sea satisfactorio para el cliente.
- Evolución del software: Es software debe ser adaptable para futuros cambios, cubriendo así nuevos requisitos funcionales por el usuario.

Estos procesos pueden ser utilizados de diferentes formas, dependiendo mucho de qué tipo de software se desarrollara. Sin embargo, el uso inapropiado de un proceso reducirá la calidad o el desempeño del producto, por la cual los recursos y costos

aumentaran.

1.5 Programación Extrema

“La programación extrema se basa en una serie de reglas y principios que se han ido gestando a lo largo de toda la historia de la ingeniería del software. Usadas conjuntamente proporcionan una nueva metodología de desarrollo software que se puede englobar dentro de las metodologías ligeras, que son aquéllas en la que se da prioridad a las tareas que dan resultados directos y que reducen la burocracia que hay alrededor tanto como sea posible” (Fowler, s.f.).

Partiendo de una definición concreta, la metodología única y compacta que permite a los programadores poder realizar cambios permitiendo así, obtener nuevas mejoras o versiones mejoradas del sistema, optando a que se ajuste al cliente y las necesidades detalladas al inicio de ejecución del software.

La característica general, es tratar de ofrece todos los recursos necesario para cubrir los objetivos del cliente y cuando éste los necesite, encontrando el mejor camino rápido, incluso si existen cambios al finalizar el período de programación. Estos cambios de gran ayuda durante la ejecución de programación, ya que permite una aproximación mejorada y más real del sistema, por ende ajustarse a los requisitos previamente definidos.

1.5.1 Riesgos de la programación extrema

Tabla 2. Programación extrema frente a riesgos

XP frente a riesgos en un desarrollo de software	
Retraso en la planificación	El software no se encuentra en la fecha de entrega

Sistemas deteriorados	El software es creado después de varios años, por la cual los costos y el mantenimiento son tan complicados que su única salida es no ser desarrollado.
Tasa de defectos	El software se encuentra listo en producción, pero contiene errores que es mejor no ser utilizado.
Requisitos mal comprendidos	El software no cubre las funcionalidades descritas desde el inicio.
Cambios de negocios	El problema planificado desde el inicio es modificado, por ende el software no se adapta a los requerimientos.
Falsa riqueza	El software contiene de grandiosas funcionalidades muy intuitivas y divertidas pero no solucionan el problema, llegando a la conclusión que no puede ganar beneficios monetarios para el usuario.
Cambios de personal	Durante la ejecución de programación por un par de años por parte de los desarrolladores, comienzan a odiar ya sea por cambios repetitivos o cambios diferentes no definidos, obteniendo como resultado el abandono del proyecto.

Nota: tabla que contiene riesgos frente a los desarrollos de software y como enfrenta la metodología de programación extrema.

Elaborado por Valencia Jhon (2018).

1.5.2 Actividades de la programación extrema

Existen cuatro actividades que son ejecutadas frente a un desarrollo de software:

Codificar: Es la actividad más importante y fuerte, que obtiene el código de acuerdo a un lenguaje de programación, cuya función es resolver funcionalidades durante del desarrollo de software, tratando de crear ideas a través del código.

Probar: la actividad permite medir si el sistema se encuentra en un buen funcionamiento y permite entender que lo codificado es una solución para cubrir un objetivo. Las pruebas deben ser sensatas y valientes con el fin de examinar a fondo del sistema. Es importante que las pruebas se las realicen con anterioridad y ajustable

al límite de tiempo de entrega, ya que es posible que salten a la luz fallas de gran complejidad.

Escuchar: Es importante plasmar en un hoja de papel o tener muy en claro los requerimientos de los usuarios, este punto es la parte esencial de todos los sistemas, el escuchar es el objetivo de entender detalladamente la lógica de negocio, y por ende optar con las mejores ideas cubrirán con las exigencias nuevas del sistema.

Diseñar: La interfaz de usuario es la parte importante entre la lógica de programación y el usuario, en ella está enfocada toda la programación de forma visual frente al usuario, el diseñar es mantener un estándar de fácil ayuda e intuitivo para el beneficiario. La interfaz de usuario es la carta de presentación del sistema.

1.5.3 Valores de la programación extrema

La programación extrema cuenta con cinco valores, que permite mejorar en el rendimiento de un desarrollo de software de forma intuitiva, ordenada y de mucha colaboración, las cuales son:

Comunicación: Este valor muy importante, cuyo objetivo primordial es dar a conocer al grupo de trabajo como se está formado o estructurando el sistema, con formas prácticas de comunicación.

Sencillez: El sistema contiene los componentes más sencillos y funcionalidades óptimas con las que se registraron en los requisitos iniciales.

Retroalimentación: Al final se realizan pruebas que nos ayudan con la fiabilidad del sistema.

Valentía: Este valor básicamente es enfocado al desarrollador, que consiste en ser

valientes ante los problemas que se puedan presentar y asegurar de forma rápida estos cambios.

Respeto: Este valor involucra en el ambiente de trabajo, donde el trabajo detallado por una persona es respetada por los demás y de la misma forma para todos los integrante. El equipo de trabajo debe trabajar como uno.

1.5.4 Fases de una programación extrema

Las pruebas son un paso muy importante en la ingeniería de software, que permite obtener errores de diversos tipos y así las etapas anteriores sean culminadas con éxito, es por ello que la programación extrema contiene fases o prácticas para un desempeño óptimo y garantizando un proyecto de calidad.

Tabla 3. Fases de la metodología de programación extrema.

Fases	
Equipo completo	El equipo de trabajo cuenta con todo el personal, es decir con el cliente y persona que se involucran en el desarrollo
Planificación	Se detalla el orden que se ejecutara las pequeñas versiones.
Pruebas del cliente	El cliente propone sus pruebas con los desarrolladores para validar las funcionalidades
Versiones pequeñas	Son partes pequeñas muy importantes que permiten al usuario final ver una acción útil generada en un corto tiempo.
Diseño simple	Detallar de forma simple y preciso, este fase puede ser una de las más difíciles ya que compromete al desarrollador obtener un resultado factible pero con un código sencillo y eficaz.
Programadores	El equipo de trabajo cuenta con parejas de programadores para obtener un rendimiento de ideas distintas y claras.
Desarrollo guiado por la pruebas automáticas	Se realiza pruebas automáticas contando con herramientas

	especializadas, y se las ejecutan con frecuencia.
Mejora del diseño	El código es la parte esencial del sistema, mientras el código sea depurado y limpio, mejora la calidad de software y ser utilizado para otras funcionalidades.
Integración continua	Durante el cambio de versiones en producción, es importante contar con una versión estable y compilada con éxito. Cuando se tenga la nueva versión se compila y verificar si cuenta con errores, es importante que las versiones nuevas contengan como mínimo dos o tres funcionalidades nuevas, permitiendo así ejecutarlas y encontrar errores puntuales si existiese.
Código “es de todos”	Cualquier persona involucrada en el desarrollo del software debe conocer, diseñar y modificar el código.
Normas de codificación	Deben existir ciertos estilos de codificación permitiendo así ser utilizado por cualquier persona del grupo sin la necesidad de preguntar que detalla cada parte del código.
Metáforas	Se detalla cada funcionalidad con una frase o nombre que permite identificar a que hace referencia esa parte de código.
Ritmo sostenible	En cada inicio de semana se detalla de forma ordena, las tareas a ser cumplidas, es importante que se divida en fases durante el desarrollo, permitiendo así no obtener días muertos donde no se sabe que realizar o ejecutar.

Nota: Tabla que contiene las fases que puede generar una programación extrema.
Elaborado por Valencia Jhon (2018).

1.6 Lenguajes de programación

1.6.1 JavaScript

Es un lenguaje de desarrollo de aplicaciones de tipo cliente/servidor, permite que las páginas web incorporen formas intuitivas y agradables para los usuarios finales. Este

lenguaje es ejecutado en los navegadores y su sintaxis se asemeja a la del lenguaje de java.

“JavaScript es un lenguaje de programación que permite el script de eventos, clases y acciones para el desarrollo de aplicaciones Internet entre el cliente y el usuario. JavaScript permite con nuevos elementos dinámicos ir más allá de clicar y esperar en una página Web.”(JavaScript, s.f.).

1.6.1.1 Angular JS

AngularJS es un framework Javascript que se ejecuta en el lado del cliente para aplicaciones web. Utiliza un patrón MVW más conocido como Modelo – Vista - Controlador.

Durante el paso de los años, los cambios y avances que ha obtenido el desarrollo de las aplicaciones con nuevos cambios, que ayuda a los desarrolladores de software.

Las mejoras ayudan a conseguir un software de calidad y sobre todo cumplir los requisitos que el usuario desea satisfacer y cuán increíbles son al momento de programar, en el caso de los desarrolladores obteniendo resultados factibles y rápidos a su vez.

A medida que el tiempo avanza, las aplicaciones cada vez son más complejas y con requerimientos extremadamente fuertes, por ende necesitan soluciones inteligentes para lograr un producto de calidad. Simultáneamente se desarrollaron herramientas que ayuden a cubrir todos los requisitos.

Una herramienta que ayuda en el desarrollo de aplicaciones con elementos simples de usar es el llamado AngularJS, este framework creado por Google.

Existen cientos de framework que cumplen con su trabajo al momento de ser utilizados en los desarrollos. Pero AngularJS es uno de los más populares por sus componentes que lo hace único y más rápido en transacciones. Este framework permite realizar aplicaciones de la manera más fácil, tan solo con entender sus componentes, se puede lograr un sin número de tareas, basta con codificar un pequeña sentencia de comandos en las líneas de código del proyecto.

1.6.1.1.1 Plantillas

Angular permite crear app de una única página, es decir podemos incorporar diferentes elementos sin la necesidad de ser recargada nuevamente en el navegador. Este mecanismo puede ser alimentado con una plantilla de contenidos dinámicos que permite una iteración con el usuario más intuitivo y en tiempo real.

Las plantillas ofrecen funciones que ayuda con una forma organizada y muy didácticas, además permite realizar tareas que evalúan condiciones que se muestran al usuario de forma visual. Las funciones son realmente inteligentes que permiten recopilar información solo con interactuar con un elemento llamado javascript y una página HTML con elementos de AngularJS incorporados.

1.6.1.1.2 Estructura MVC

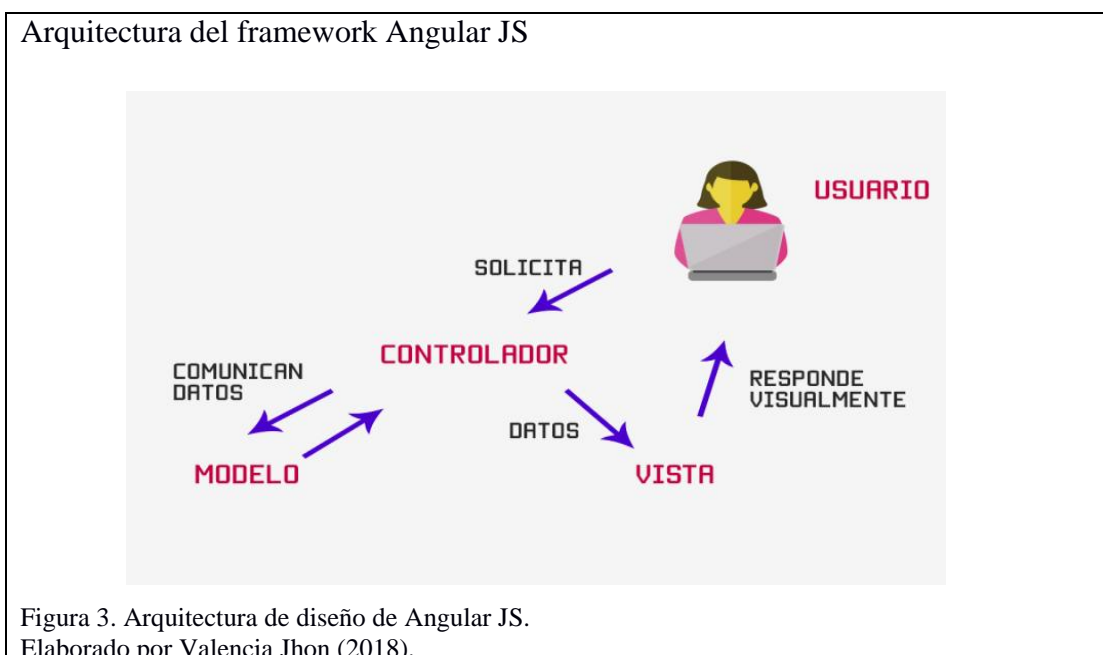
La estructura conocida como modelo, vista y controlador, cuya función principal es separar la lógica de programación, con elementos del lado del cliente como por ejemplo vistas, mientras que la lógica de programación se encarga de realizar operaciones de acuerdo a las necesidades de la aplicación.

Este proceso es muy reconocido en los framework con un algo nivel de complejidad, ya que estructura el código de la forma más simple y esencial para otras necesidades. Sin la estructura mencionada sería imposible mantener y general aplicaciones con requisitos de gran complejidad.

1.6.1.1.3 Estructura MVC en AngularJS

AngularJS permite la creación de controladores, estos nacen de las necesidades de los usuarios permitiendo realizar cualquier transacción, tal es el caso de la creación, modificación o eliminación de datos, el controlador es el encargado de procesar información al modelo, mientras que para la vista ofrece la información de forma agradable al usuario.

Las vistas en AngularJS básicamente son elementos que interactúan entre la computadora y el usuario final otorgando elementos visuales de calidad e intuitivos, ofreciendo así información de un modelo y proceso en un controlador.



1.6.2 JSON

“Es un formato textual estándar para representar datos estructurados cuya aceptación crece continuamente debido que es más simple y legible que XML” (JSON, s.f.).

Las estructuras de JSON no conforman a un esquema sino que introduce tantos datos como su estructura, como se puede ver en la figura siguiente:

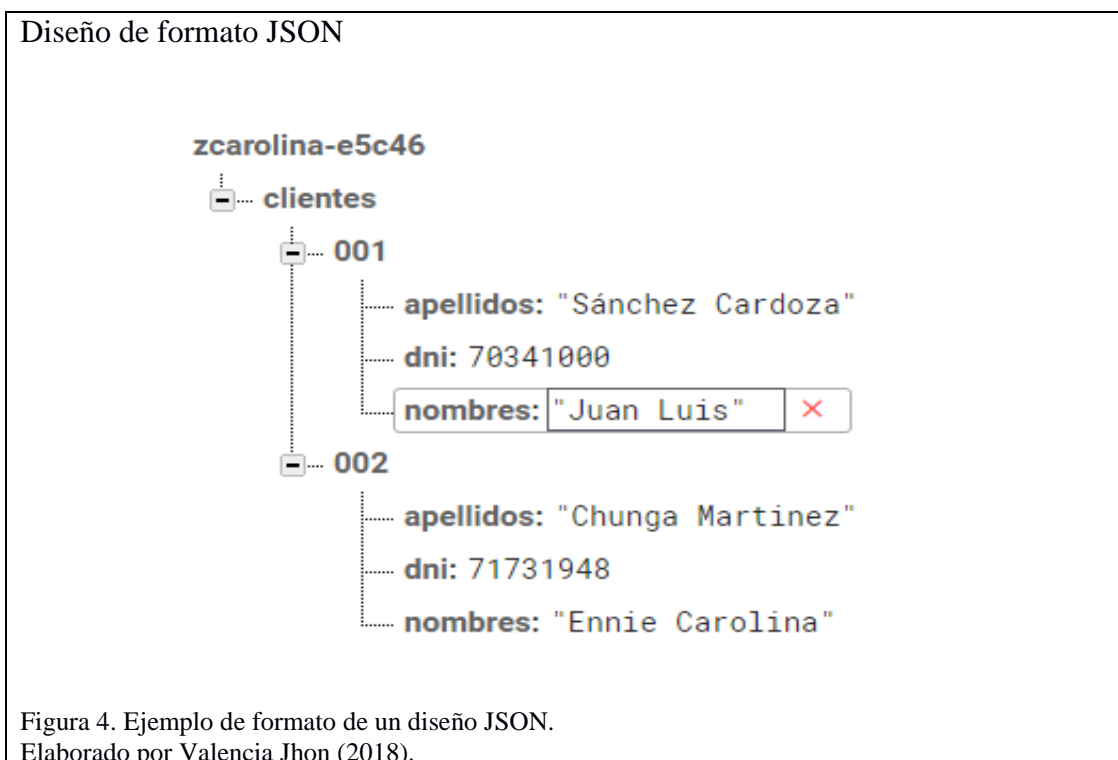


Figura 4. Ejemplo de formato de un diseño JSON.
Elaborado por Valencia Jhon (2018).

La mayoría de sistemas están optando esta estructura para representar la información almacenada y optar por soporte de exportación e importación de datos, sin obtener un archivo muy pesado que ofrece bases de datos tradicionales.

JSON permite obtener información de una forma fácil y sencilla permitiendo a los programadores a optimizar recursos. Permite acoplarse sin ningún inconveniente al sistema que lo requiere, es un texto de formato plano y puede ser administrado por

cualquier lenguaje de programación.

JSON está constituido por dos partes estructuras:

- Objetos JSON, es un conjunto de pares de nombres/valores, encerrados por un conjunto de llaves ({}), cada nombre es seguido por dos puntos (:) y sus respectivos valores separados por una coma (,).
- Lista JSON, es una lista de valores, determinados entre corchetes ([]) y separados por una coma (,).

1.7 Herramientas

El desarrollo de este proyecto, se determino un conjunto de herramientas para un mejor desempeño y mejorar la calidad de software. Herramientas que parten desde plasmar las ideas mediante un código de programación y tecnologías necesarias para aplicaciones.

1.7.1 Firebase

Es una plataforma creada por GOOGLE, su función primordial es facilitar la creación de apps con una elevada calidad, de forma sencilla y rápida. La plataforma se encuentra en la nube y se encuentra disponible para diferentes plataformas como iOS, Android y web.

Firebase ofrece diversas funciones para combinar y adaptar las aplicaciones a la plataforma a medida de las necesidades del cliente.

Puntos importantes que ofrece Firebase:

Base de datos en tiempo real: La base de datos en tiempo real es la primera característica de Firebase, ya que su estructura de datos en formato JSON, almacenados en la nube. Lo que hizo realmente única a esta base de datos fue la capacidad de sincronizar los datos entre los usuario que realizan peticiones a la base de datos, como por ejemplo inserciones o modificaciones sumamente al instante.

La base de datos es simplemente un objeto JSON de gran magnitud con un grado administrable leve, es decir solo es un árbol de valores de forma sencilla.

La estructura JSON permite modelar los datos de una forma sencilla, flexible y agradable. Las bases de datos en tiempo real no contienen esquemas, esto quiere decir que no adopta una estructura fija desde el inicio del modelado de la base de datos.

Sin embargo, si se preocupa de la validación de datos a ingresar, la base de datos en tiempo real incorpora un amplio lenguaje de reglas impuestas por el servidor, que permite validar la estructura de datos en cada acción de escritura.

Sistema de autenticación: La mayoría de aplicaciones de software necesitan la identidad de un usuario. El reconocimiento de la identidad de ese usuario, que permite a la aplicación almacenar datos de forma segura en la nube y brindar una experiencia personalizada en todos sus dispositivos.

La autenticación de Firebase ofrece servicio de BackEnd y SDK fácil de usar, son hechas para autenticar a los usuarios que desean acceder a la aplicación. La función admite autenticación con contraseñas, proveedores de entidades federales más populares como es el caso de Facebook, Google y Twitter.

Storage: El storage de Firebase es un almacenamiento limitado de 10 gigas, que permite subir a la nube y compartir entre usuario que utilizan el sistema.

La herramienta usa el servicio de Google Cloud Storage, que ayuda con una sincronización de forma rauda con nuestra aplicación. Existe un depósito de almacenamiento conocido como “Bucket” donde se almacena las imágenes del aplicativo de Android, esto ayuda que dicho contenido se sincronicen con las aplicaciones web directamente a una ruta que ofrece y almacena los datos de las imágenes.

Hosting: Firebase ofrece Hosting de contenido web de producción. Este servicio ayuda de forma rápida y fácil a las apps web, utilizando contenido estático en una red global de distribución de contenidos.

Funciones claves de este servicio:

- Conexión segura: El Hosting de Firebase se integra con una SSL para mayor seguridad en su servicio de apps web.
- Entrega rápida: Cada elemento almacenado en la nube en caché en SSD de CDN de todo el mundo, sin importar la ubicación del usuario el contenido se agrega de forma rápida.
- Implementación rápida: Las herramientas de línea de comando ayuda a la adición de implementación del la app en solo segundos.

1.7.2 Netbeans IDE

“Netbeans IDE es un entorno de desarrollo integrado (IDE), modular, de base estándar (normalizado), escrito en el lenguaje de programación Java. El proyecto Netbeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (framework) para

compilar cualquier tipo de aplicación.” (Netbeans, s.f.).

La herramienta más utilizada por los desarrolladores de software, este IDE muy potente y muy seguro al momento de compilar, cuenta con cientos de plugins que permiten desarrollar a un más rápido y optimizando recursos que no poseen otros IDE's, la herramienta soporta la mayoría de lenguajes de programación en la actualidad.

1.7.3 Android Studio

“Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJIDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android” (Android Studio, s.f.)

Este IDE permite gestionar de forma agradable a los desarrolladores, cuenta con un sin número de consolas permitiendo seleccionar un tipo de dispositivo móvil y utilizarlo para la compilación de la nueva app. La utilidad llega a un punto de obtener muchos recursos, por ende las máquinas que no contienen características de alto rendimiento, se vuelven lentas.

Una característica muy importante, es ofrecer a los desarrolladores, obtener un código de forma estático, y rápido ya que cuenta con una interfaz gráfica para realizar interfaces (Front-End) de mucho agrado, o crea un proyecto de acuerdo a una plantilla.

1.9 Sistema de Información gráfica

Es una integración de hardware, software y datos geográficos. Cuya función básica es analizar, manipular, almacenar y desplegar una información geográfica de forma visual agradable al usuario con el fin de resolver problemas complejos.

Un SIG funciona como un almacén de datos con información gráfica, que permite separar información en diferentes capas temáticas, con el fin de trabajar de manera fácil y rápida, facilitando al usuario la posibilidad de relacionar la información encontrada a través de la topología de los objetos, con el fin de obtener una información agradable de forma gráfica.

Las principales funciones que se llevan a cabo en un SIG son:

- Obtener información: la información es recopilado por algún elemento sea software o hardware y almacenados.
- Análisis: Este proceso es el encargado de ejecutar diversas funcionalidades y presentar al usuario un informe de los datos ya almacenados.

1.10 Prototipos

Durante la ejecución del proyecto la aplicación web y móvil, sufren cambios que permiten ajustarse a las necesidades del usuario, es por ello que existen prototipos finales. Estos prototipos ofrecen a los usuarios finales obtener una visión de que es lo que realiza la aplicación y se utilizan para las pruebas respectivas, arrojando resultados para los desarrolladores y para el cliente.

Es importante que estos prototipos cuenten con una versión final, estable y con un código limpio, por ende si sufre algún cambio, este sea posible ser corregido por el grupo de trabajo y de la misma forma ser entendido.

Capítulo 2

2.1 Detalle de requerimientos

Los requerimientos son la parte fundamental en todo desarrollo de software, es importante que el cliente describa de forma más detallada posible y así las ideas son plasmadas en el pensamiento de los desarrolladores.

A continuación se observa un detalle de los requerimientos más importantes para este desarrollo:

Registro de incidentes: Permite al usuario (público en general) registrar un incidente con la ayuda de un Smartphone tomando en cuenta la ubicación, una fotografía, la descripción general de lo sucedido y un nivel de emergencia.

Seguimiento del registro: Permite al administrador de la página web obtener el estado en el que se encuentra el incidente y proceder a un cambio respectivo.

Reportes: permite al administrador obtener información general de el incidente, en este caso una visualización mediante una imagen, la ubicación en latitud y longitud, una descripción almacenada por el usuario que registró dicho inconveniente y el nivel del incidente.

La información se encuentra almacenada en la nube de Firebase, imágenes e información del suceso. Además una aplicación móvil fácil de utilizar para el público en general.

2.2 Diagrama de actividades

El objetivo de un diagrama de actividades es describir las acciones, esto representa a las operaciones y transacciones que se realizan al momento de ejecutar el sistema, permitiendo así saber el funcionamiento y las utilizadas que pueden llegar a obtener.

El diagrama de actividades se centra en los objetivos que cada funcionalidad ofrece para poder seguir al siguiente proceso y preguntando si la idea plasmada es el correcto o necesita ser modificada.

2.2.1 Diagrama de actividades aplicación web

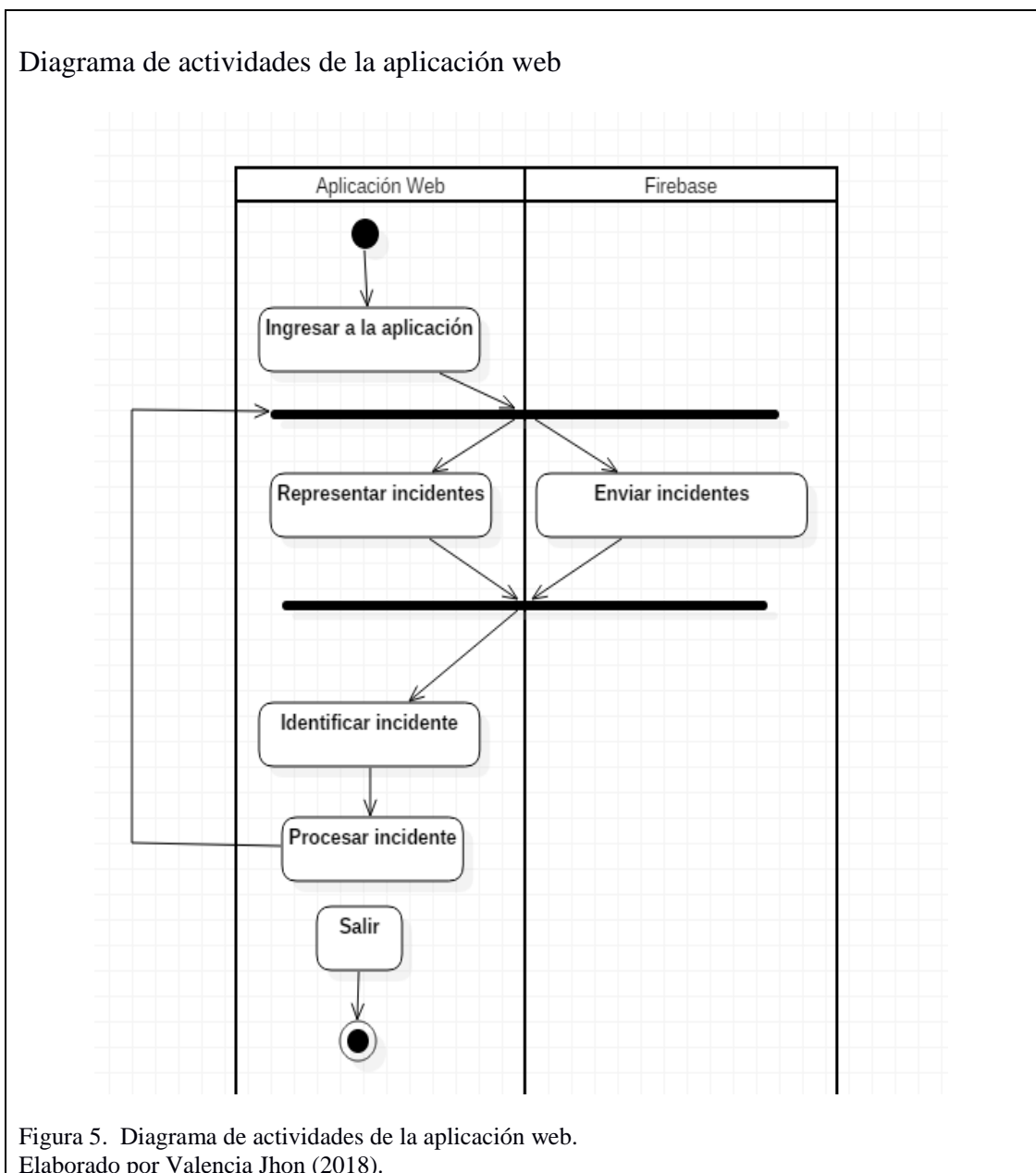


Figura 5. Diagrama de actividades de la aplicación web.
Elaborado por Valencia Jhon (2018).

El proceso de la aplicación web inicia desde el ingreso a la aplicación, establecido

mediante un login de inicio, donde el usuario ingresa su clave y contraseña definido con anterioridad, una vez que el ingreso sea con un estado de exitoso, el sistema web permitirá obtener y representar los incidentes registrados por los usuarios, este proceso básicamente realiza dos acciones el presentar y en enviar datos por parte de la herramienta de Firebase, la acción se encuentra activos, el sistema web escucha peticiones que la herramienta de Firebase ofrece, así obtiene los recursos a mostrar en pantalla.

Posteriormente completado el proceso anterior el administrador o usuario de la página web identifica el incidente, para luego ser procesado, mediante una llamada que realice y otra acción pertinente por parte del usuario. El administrador es el cargado de realizar el proceso marcando un estado al incidente.

El proceso finaliza cuando un administrador desea salir, es decir realizado una petición de un “Logout” o salida, mediante un botón encontrado en la página web.

La figura número 5 muestra un diagrama de actividades, que permite obtener una referencia que realiza la aplicación web, permitiendo identificar cada actividad y la forma en la cual se desempeña. El diagrama de actividades permite a futuro estructurar de forma eficiente al sistema tan solo verificando las tareas que comprende en cada una de ellas y poder mejorar si fuese el caso.

2.2.2 Diagrama de actividades aplicación móvil

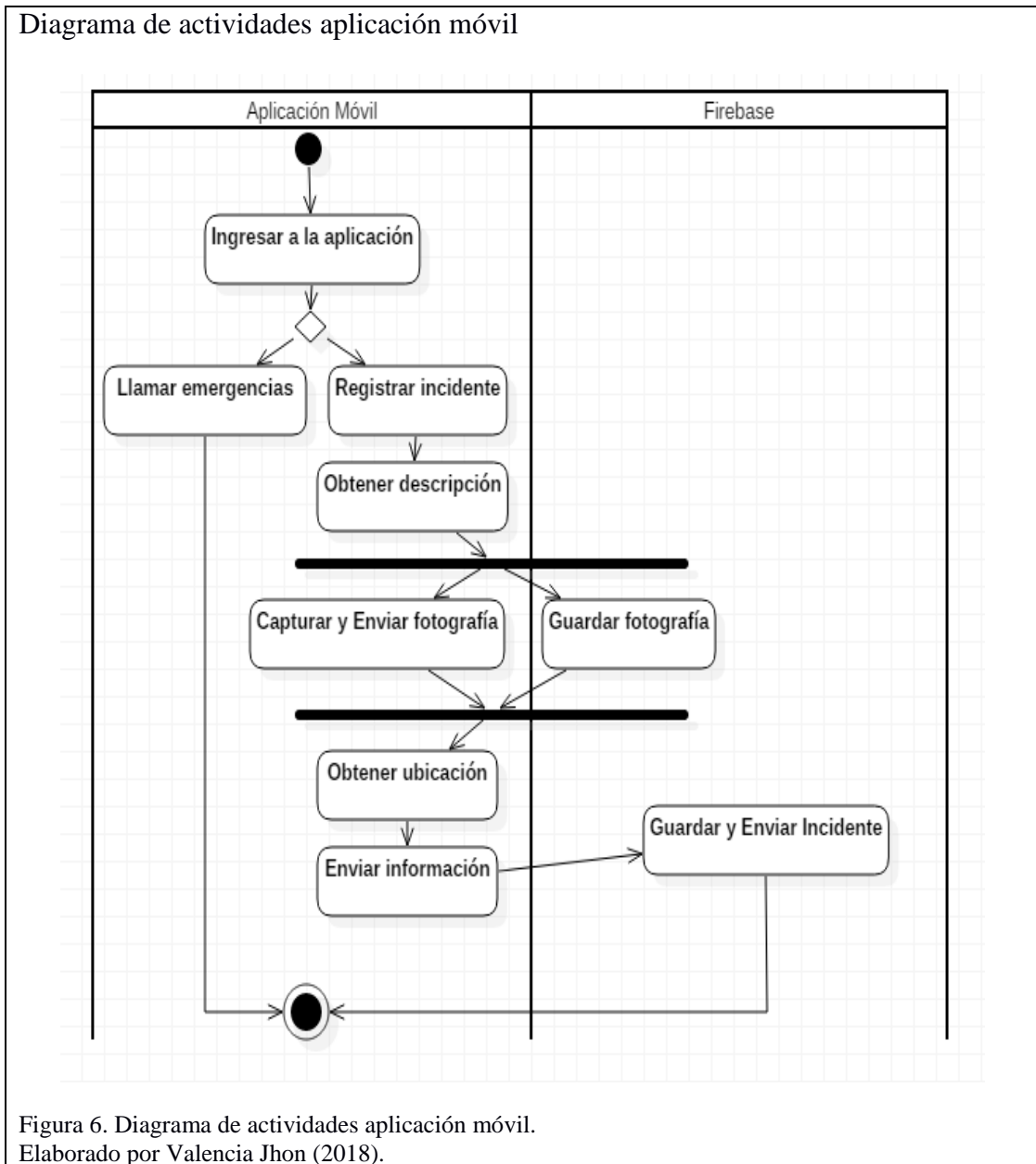


Figura 6. Diagrama de actividades aplicación móvil.
Elaborado por Valencia Jhon (2018).

El proceso de la aplicación móvil es muy diferente a la descrita en el punto 2.2.1, donde la app no permite una autenticación de login. Permite el envío de información capturado o tomada por el usuario, el proceso inicia con un ingreso rápido, continuando con una toma de decisión, la cual se divide en dos y son: llamar a emergencias, este permite realizar una llamada a un sitio de emergencia, mientras que la otra es un proceso de registros.

El proceso del registro incidente, inicia con la obtención de una breve descripción, donde se detalla una pequeña información de lo que está ocurriendo en ese momento. Continuando con este proceso, con la obtención de una fotografía, esta es procesada y guardada en la Storage de Firebase que se encuentra en la nube. Posteriormente el teléfono inteligente obtiene los datos de latitud y longitud es decir la ubicación exacta donde se encuentra.

El proceso finaliza con el envío de datos en formato JSON, donde es almacenado en tiempo real en una base de datos no SQL en la nube de Firebase.

La figura número 6 muestra un breve y completo proceso de las actividades que conlleva una aplicación móvil, permitiendo así una estructura muy fácil en el desarrollo y mejorando el proceso y obtención de recursos.

2.2.3 Diagrama de actividades aplicación general

Diagrama de actividades aplicación general

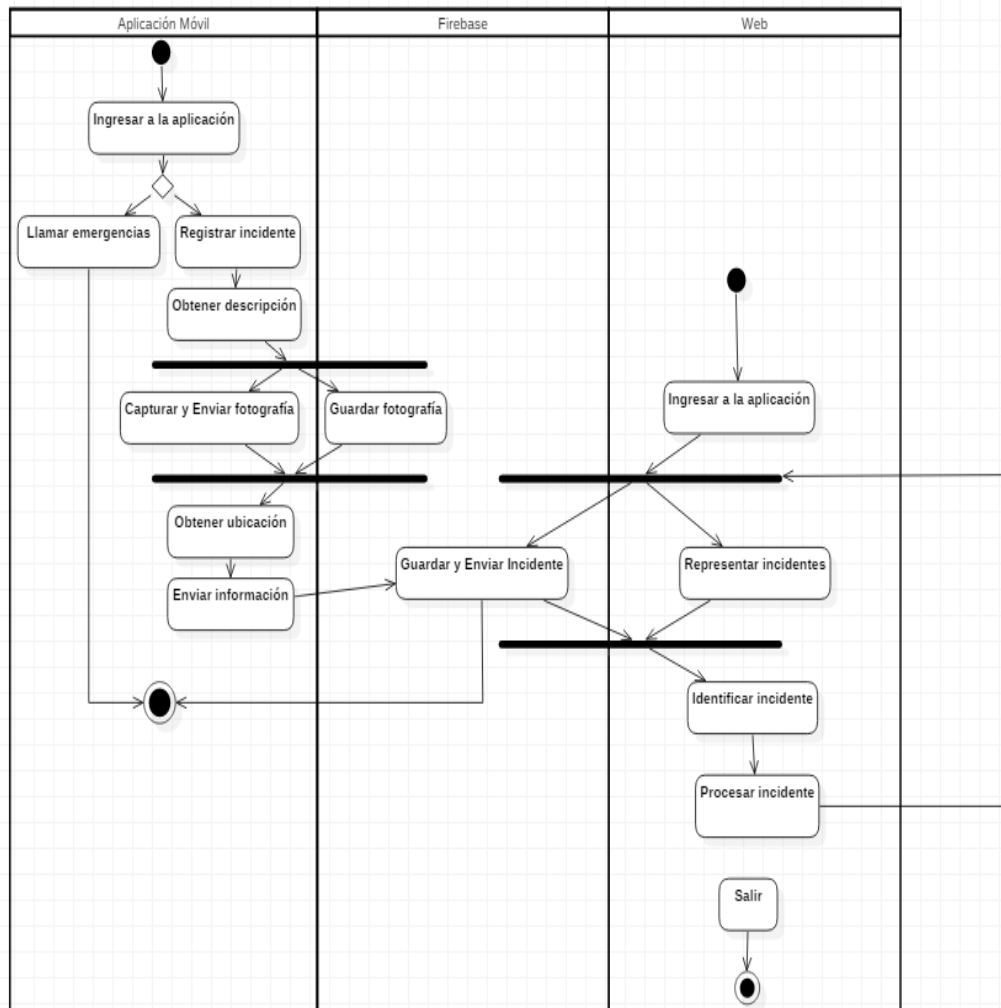


Figura 7. Diagrama de actividades aplicación general.
Elaborado por Valencia Jhon (2018).

La figura 7 muestra las actividades general, donde interviene las aplicaciones web como móvil y la herramienta Firebase. La importancia de este diseño es determinar cómo actúan las actividades en cada etapa, verificando las mejor estructuras de procesos.

Cabe recalcar que las actividades que se encuentran en la herramienta de Firebase, son importantes, la información necesaria que será repartida a la web y nueva información que será almacenada en la nube de Firebase, otorgada por la aplicación

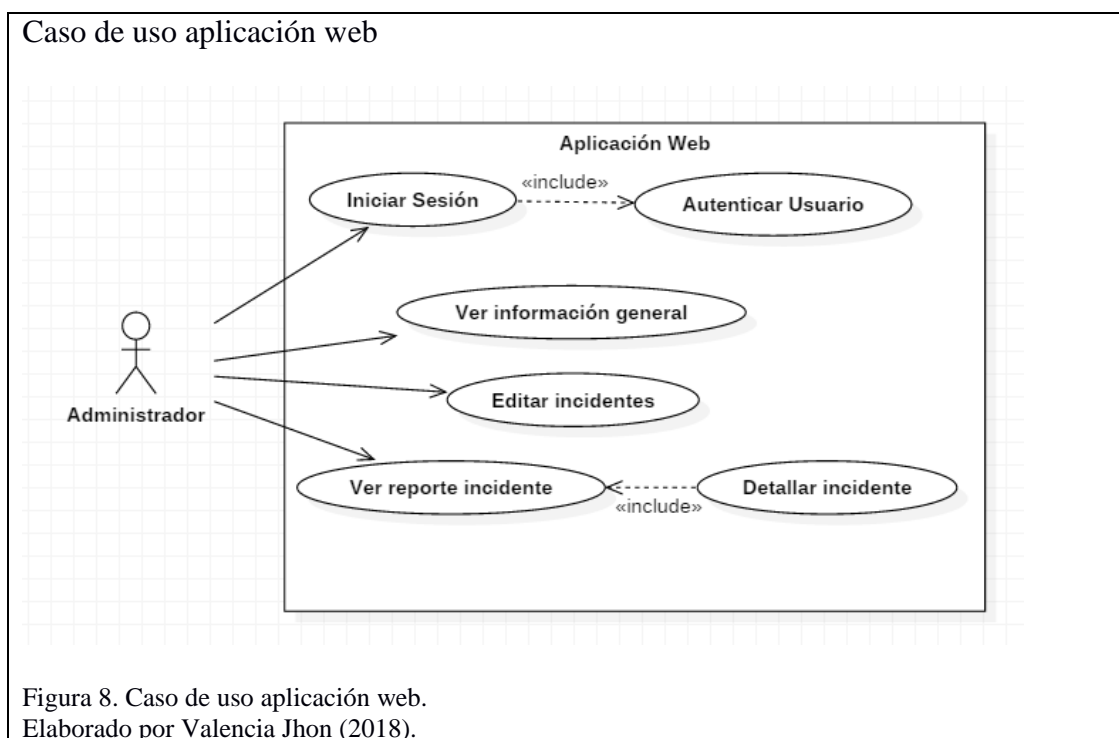
móvil, el proceso básicamente se sincroniza en las dos aplicaciones, para obtener un resultado rápido, en cuestión de milisegundo donde actualiza la información.

2.3 Casos de uso

Los casos de uso es un detalle de las actividades que deben realizarse dentro de un sistema de software donde involucran los actores o entidades que ejecutaran las actividades, verificando cual es proceso adecuado para realizar dicha tarea y recursos que pueden ser utilizados.

Es importante detallar de forma ordena y muy inteligente, por ende mejorara los beneficios que ofrecerá el software.

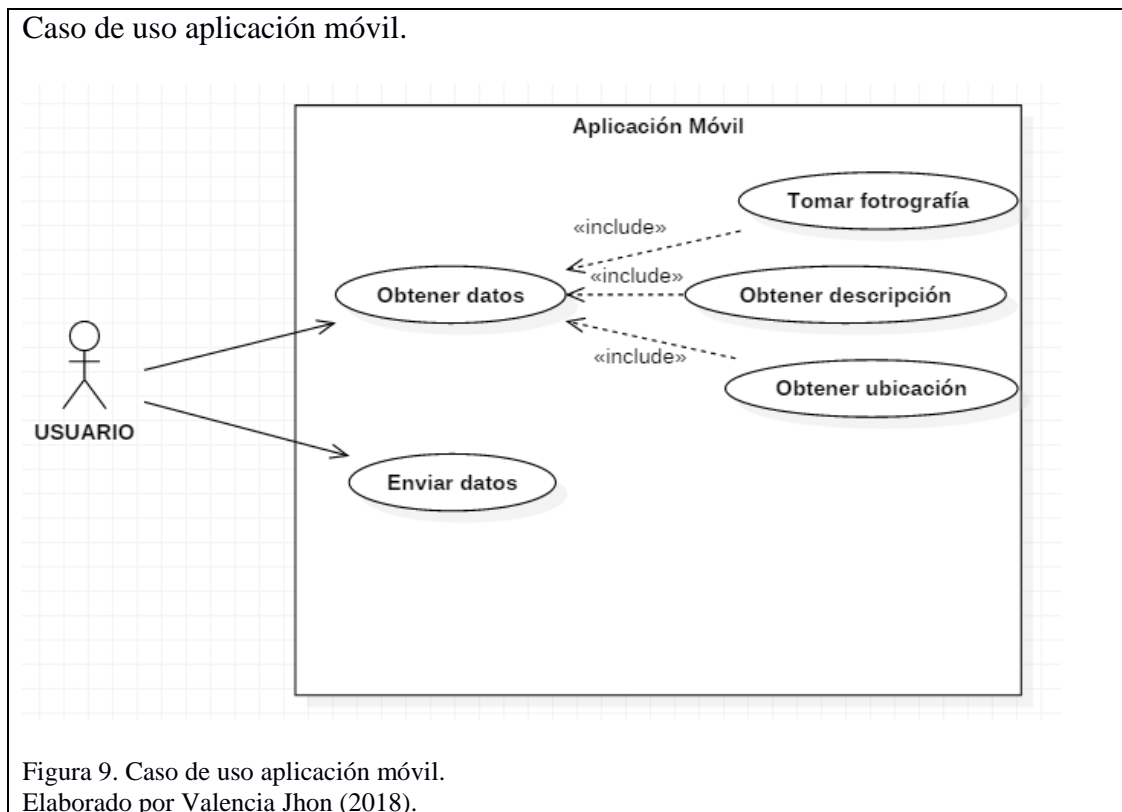
2.3.1 Caso de uso aplicación web



La figura 8 muestra como actor al administrador de la aplicación web, dentro de este caso de uso existe un solo involucrado, ya que es la única persona que puede

administrar la página web. El administrador realizar varias actividades muy necesarias en la aplicación, como es ingreso, ver información general del incidente, edición y verificar un reporte del incidente.

2.3.2 Caso de uso aplicación móvil



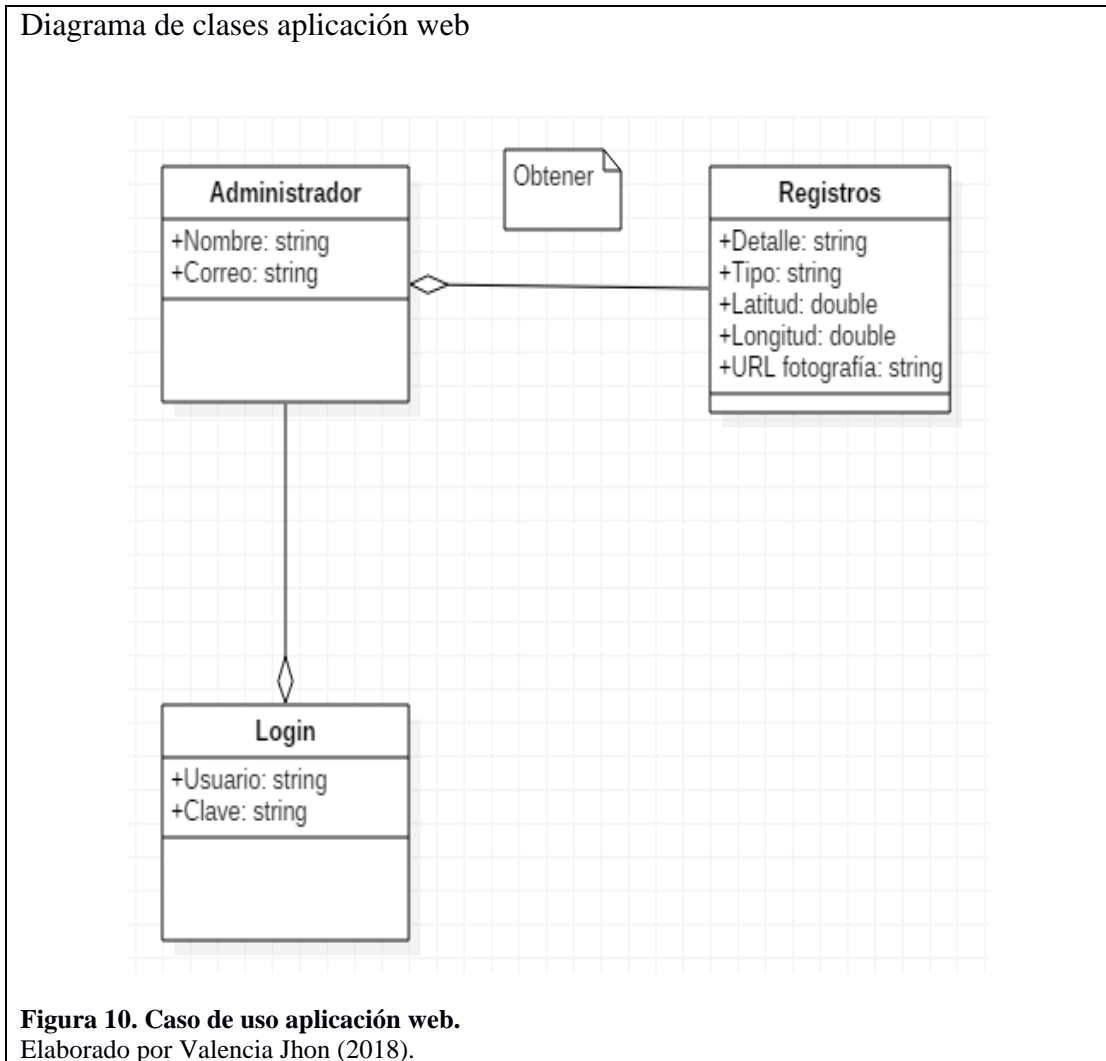
La figura 9 muestra un proceso de actividades que con lleva la aplicación móvil, este proceso no necesita de un registro o un acceso a la aplicación. Las actividades que puede realizar, es obtener datos y ser enviados. La aplicación puede ser utilizada por cualquier persona, tan solo con descargar e instalar.

2.4 Diagrama de clases

Los diagramas de clases son estructuras estáticas que describe la organización de un sistema, mostrando las clases, atributos, métodos y relaciones entre ellos, obteniendo

una organización de forma estructurada.

2.4.1 Diagrama de clases aplicación web



El diagrama de clases de la aplicación web que se muestra en la figura 10, pretende identificar los clases que ofrece el sistema, las acciones se la ejecuta de forma transparente para el usuario.

2.4.1 Diagrama de clases aplicación móvil

Diagrama de clases aplicación móvil.

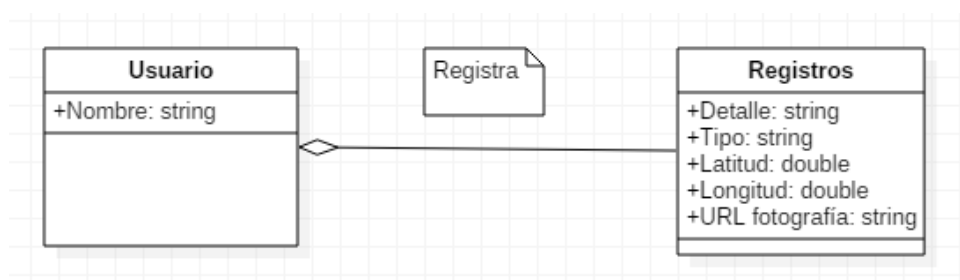


Figura 11. Diagrama de clases aplicación móvil.

Elaborado por Valencia Jhon(2018).

El desafío de realizar un diagrama de clases para una aplicación es conseguir que dos aplicaciones usen una base de datos no SQL, con alojamiento en la nube que pretende realizar transacciones en tiempo real, para obtener resultados rápidos. Es por ello que las clases dentro de la aplicación móvil es básicamente el almacenamiento de registros, la aplicación es ejecutada sin la necesidad de acceder por medio de un login u obtener alguna información extra por parte del cliente, en este caso el usuario que registra.

2.5 Diagrama de base de datos

Diagrama de base de datos no SQL

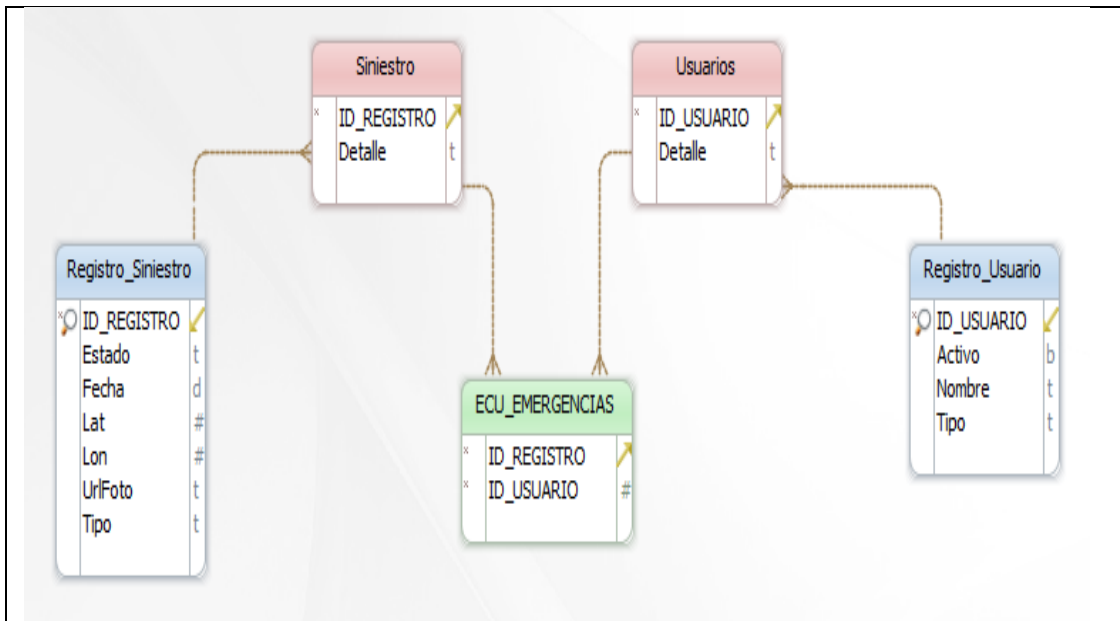


Figura 12. Diagrama de base de datos no SQL
Elaborado por Valencia Jhon (2018).

El diseño de la base de datos que se muestra en la figura 12, está orientada a documentos, la base de datos permite almacenar grandes cantidades de datos de forma muy ordenada y fácil para recuperar la información. La información es almacenada en una base de datos que soporten estos formatos, es decir la información es creada en archivos de formatos JSON, la cual permite obtener información muy pesada. Las bases de datos orientadas a documentos no almacenan tablas o relaciones como las bases de datos tradicionales, al contrario almacenan estructuras no estrictas lo que hace más eficientes al manejar grandes volúmenes de datos y fáciles de realizar algún cambio sin la necesidad de agregar campos o editar las tablas como ofrecen las bases de datos SQL.

El detalle de la figura 12, muestra una tabla o estructura ECU_EMERGENCIAS, que permite la obtención de dos registros; del usuario y emergencia, cabe recalcar que los id son generados en la base de datos de Firebase, es decir ofrece un código único en cada registro almacenado que es el caso para el registro de incidentes, ya que cada

registro es almacenado por usuario en una base de datos noSQL.

Este nuevo paradigma elimina todas las características de las bases de datos convencionales, es por ello que la nueva forma de diseño está surgiendo, permitiendo a los desarrolladores realizar contenido instintivo y cómodo de uso, obteniendo mejores resultados, un trabajo exitoso y un cliente satisfecho.

2.6 Interfaces prototipos

Los prototipos de interfaces son un conjunto de modelos de forma visual, que permite al usuario y a los desarrolladores, obtener una idea de cómo estará estructurada la pantalla visual para el usuario final, es decir una ubicación de los elementos que interactúa el usuario con la aplicación.

Las interfaces de usuario, son un prototipo y no es una versión final, durante la ejecución de varios prototipos, estos cambian de acuerdo a las necesidades del usuario, permitiendo obtener mejores resultados y un interfaz amigable para el usuario. Es importante que las interfaces de usuario sean de forma intuitiva y fácil de usar.

Los prototipos contiene un conjunto de elementos y estructuras que poco a poco son detalladas y estudiadas para un mejor rendimiento, los elementos son un conjunto de características HTML, esto quiere decir un conjunto de botones, ingresos de texto, obtener una fotografía, imágenes, colores, etc. Básicamente es un diseño estructurado y determinado, permitiendo al desarrollador plasmar de forma visual una idea por parte del usuario.

2.6.1 Interfaces aplicación web

Prototipo interfaz ingreso



Figura 13. Prototipo interfaz ingreso.
Elaborado por Valencia Jhon (2018)

La figura 13 muestra un prototipo inicial, que permite un ingreso de acuerdo al nombre de usuario y la contraseña codificada con anterioridad, el registro de usuarios se los realiza desde la consola de Firebase, permitiendo el ingreso de un correo electrónico y unos campos adicionales descritos en el ítem 2.5.

Prototipo interfaz visualización de datos



Figura 14. Prototipo interfaz visualización de datos.
Elaborado por Valencia Jhon (2018)

Este prototipo muestra al administrador, todos los registros almacenados por parte de la aplicación móvil de forma visual, permitiendo obtener la ubicación exacta. Los registros son detallados por un icono de acuerdo al tipo de incidente y ubicados en un mapa otorgado por Google Maps.

Prototipo interfaz reporte de datos.



Figura 15. Prototipo interfaz reporte de datos.
Elaborado por Valencia Jhon (2018)

Los reportes de emergencias permite identificar datos que expone para los administradores de la página web, permitiendo realizar alguna acción que permite evaluar el tipo de incidente y dar la prioridad respectiva, este prototipo indica de forma breve una información del registro con una imagen e información necesaria.

2.6.2 Interfaces aplicación móvil

Los prototipos móviles son elementos visuales que permite a un usuario identificar a la aplicación y permite a los desarrolladores obtener mejores resultados en la hora de desarrollar de forma más rápida con la metodología de programación extrema.

Interfaz de bienvenida de la aplicación móvil.



Figura 16. Interfaz de bienvenida de la aplicación móvil.
Elaborado por Valencia Jhon (2018).

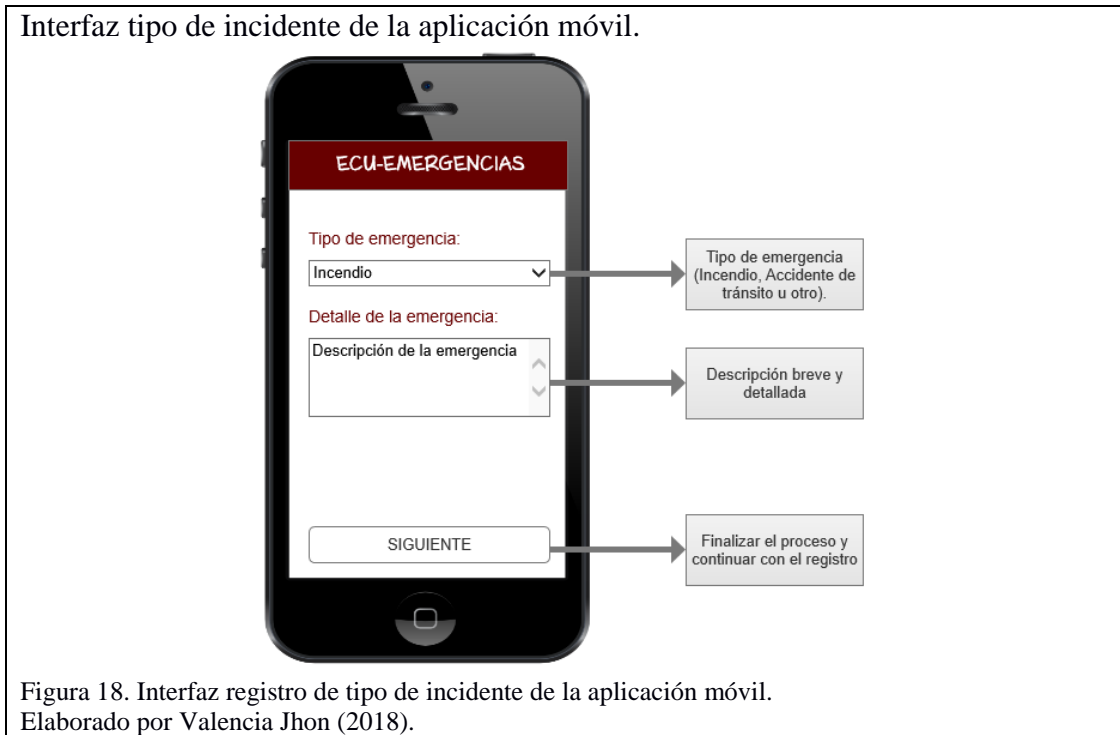
La pantalla inicial del prototipo cuenta con el nombre, la bienvenida y la versión de la aplicación, la pantalla finaliza después de 3 segundos ya que cuenta con un temporizador previamente programado.

Interfaz de registro de la aplicación móvil.



Figura 17. Interfaz de registro de la aplicación móvil.
Elaborador por Valencia Jhon (2018).

La figura 17 es muestra el diseño de la pantalla de registrar un incidente o llamar a un número determinado de emergencias. Permitiendo al usuario realizar la acción que crea conveniente para su caso.



Continuando con el registro de incidentes, el prototipo muestra una pantalla que permite al usuario registrar una información, como es el tipo y un detalle del registro. La información muy relevante para el personal de emergencia.

Interfaz para obtener la fotografía del incidente de la aplicación móvil.



Figura 19. Interfaz para obtener la fotografía del incidente de la aplicación móvil. Elaborado por Valencia Jhon (2018).

La figura 19 muestra un prototipo de selección de una fotografía en ese momento o elegir de la galería del teléfono inteligente, la acción permite enviar al Storage de Firebase, es decir la fotografía se almacena y genera una URL.

Interfaz para obtener la ubicación del incidente de la aplicación móvil.



Figura 20. Interfaz para obtener la ubicación del incidente de la aplicación móvil. Elaborado por Valencia Jhon (2018).

Interfaz para finalizar el incidente de la aplicación móvil.



Figura 21. Interfaz para finalizar el incidente de la aplicación móvil.
Elaborado por Valencia Jhon (2018).

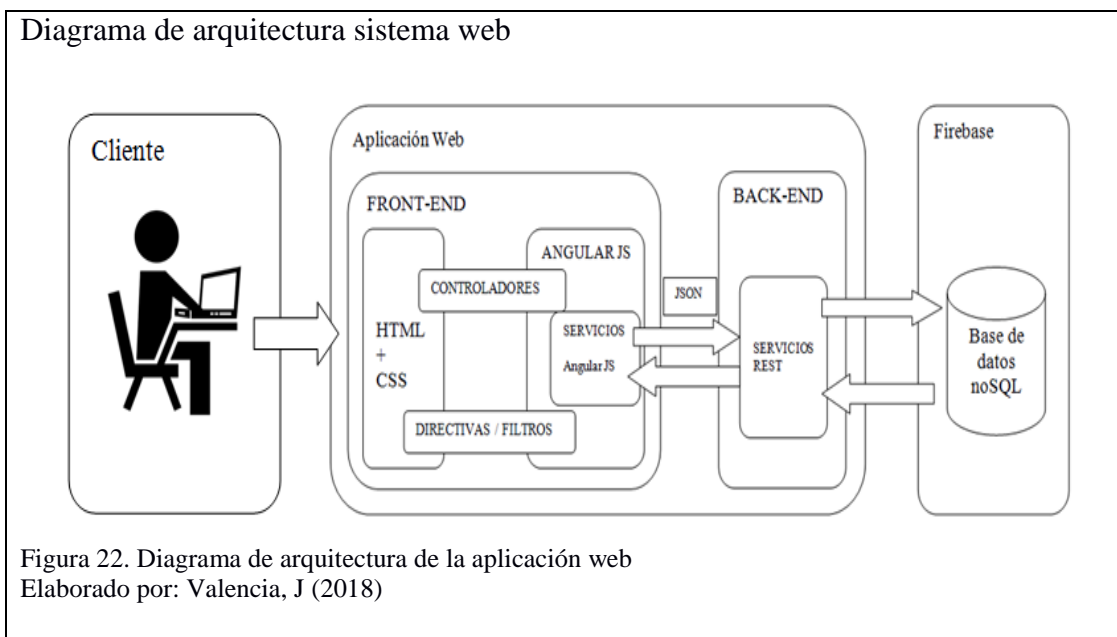
Finalizando con el registro que muestra en la figura 20 y 21, es obtener una ubicación exacta y posteriormente ser almacenada con la información anteriormente obtenida, y permitiendo al usuario obtener un nuevo registro o salir de la aplicación.

Capítulo 3

El capítulo tres contiene una breve información de las pruebas realizadas y el tiempo que demora las tareas, permitiendo así analizar y comprobar el resultado dado con el sistema.

3.1 Sistema web

3.1.1 Arquitectura



El diseño arquitectónico de un sistema web, consta de tres módulos o elementos, los cuales interactúan entre sí, para obtener una acción final. Estos elementos son: Cliente, la aplicación web y la herramienta de google Firebase. La arquitectura web funciona de la siguiente manera:

Inicial mente el usuario realiza una petición a los servicios que ofrece la página web, la cual es procesada de acuerdo a la petición realizada.

La aplicación web ofrece dos módulos importantes las cuales son en Front-End, cuya función inicial es presentar de forma visual al usuario e interactuar con el sistema,

ofreciendo formatos visuales fáciles y agradables para el cliente. El módulo Frond-End ofrece elementos de HTML y CSS para mejorar el ambiente visual, con etiquetas implementadas en el proyecto que son gratis y fácil uso, además el servicio que interactúan las etiquetas con un Framework llamado Angular JS que permite un ambiente más rápido en transacciones y peticiones por parte del cliente. Angular JS se ejecutan en el lado del navegador, sin la necesidad de obtener servicios en un servidor o características de redimiendo que otras aplicaciones necesitan. Y el módulo de Back-End podría llamarse el administrador del sistema, es decir el encargado de realizar las transacciones sin afectar el rendimiento del Front-End. El Back-End cumple su rol en el sistema operativo, permitiendo realizar acciones, peticiones o intercambio de datos entre una base de datos e interfaces del usuario.

Las acciones se la ejecutan por debajo del cliente, sin que se las pueda observar.

La herramienta de google Firebase ofrece una base de datos noSQL en tiempo real, permitiendo almacenar la información enviada por la aplicación web, la herramienta se encuentra a cada momento a la espera de peticiones nuevas, sin la necesidad de realizar una actualización de la página o realizar otras peticiones. El formato de la base de datos que se detallada en el ítem 2.5 de este documento, es de suma importancia y de preferencia tanto para la aplicación como para la base de datos.

3.1.2 Código principal

El proceso de programación de la aplicación, este sistema web contiene elementos de Frond-End en código HTML y AngularJS.

La implementación de AngularJS es básicamente incluir las librerías que ofrece este Framework, mediante unas sentencias de script como muestra en la figura:

Librerías implementas en la aplicación web.

```
<!-- Angular JS -->
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.5.5/angular.min.js"></script>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.5.5/angular-animate.min.js"></script>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.5.5/angular-aria.min.js"></script>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.5.5/angular-messages.min.js"></script>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular-route.js"></script>
<!-- Google MAP -->
<script src="https://maps.googleapis.com/maps/api/js?libraries=places&key=AIzaSyBtis7t13bSldrJpL twWkVGfSF 18SY0"></script>
<script src="https://rawgit.com/allenhwkim/angularjs-google-maps/master/build/scripts/ng-map.js"></script>
<!-- Componentes Extra -->
<script src="js/angular-cookies.js" type="text/javascript"></script>
<script src="js/jquery.min.js" type="text/javascript"></script>
<script src="bower_components/ngprogress/build/ngprogress.js" type="text/javascript"></script>
```

Figura 23. Librerías implementas en la aplicación web.

Elaborado por Valencia Jhon (2018).

La figura 23 contiene librerías incluidas en este proyecto. Sujeta una llamada a los elementos de Angular JS, para obtener las etiquetas que ofrece este framework, permitiendo ser de mucho apoyo en todo este proyecto, cabe recalcar que las llamadas se las realiza en las páginas principales del diseño. Igualmente contiene dos llamas importantes que ejecutara, las cuales son librerías de google Maps, que ofrece un mapa único para el proyecto, sumamente simple y sencillo, para este caso se implemento ng-map una librería que interactúa las herramientas de Angular JS con las herramientas de google Maps, para un mejor servicio y una estructura con mejor diseño.

Finalizando con librerías adicionales, que son importantes en un proyecto, el principal es jquery, este framework javascript sirve como una base para la ejecución de aplicaciones avanzadas, cuya aportación es obtener tareas habituales que los desarrolladores normalmente utilizan.

Librerías Firebase implementas en la aplicación web.

```
<!-- AngularFire -->
<script src="https://cdn.firebase.com/libs/angularfire/2.1.0/angularfire.min.js"></script>
<!-- Firebase -->
<script src="https://www.gstatic.com/firebasejs/3.3.0/firebase.js"></script>
```

Figura 24. Librerías Firebase implementas en la aplicación web.

Elaborado por Valencia Jhon (2018).

La figura 24 contiene dos etiquetas realmente importantes, deben establecerse a la par, ofrecen conexiones a la herramienta de google Firebase y comunicaciones entre Angular JS.

Las librerías descritas anteriormente se las puede descargar e implementarlas en la estructura jerárquica del proyecto. Los archivos finalizados en js son ubicados en la carpeta “public/js” con el fin de ofrecer una infraestructura de software de alto nivel.

En el modelo de Back-End se detalla la lógica del negocio ofreciendo funcionalidades para el sistema y el usuario.

Código principal para aplicación web.

```
var config = {
  apiKey: "AIzaSyASlq1XX5J28f_rHY5Ty10Edw1WYZhdJx8",
  authDomain: "pruebas-76b0b.firebaseio.com",
  databaseURL: "https://pruebas-76b0b.firebaseio.com",
  storageBucket: "pruebas-76b0b.appspot.com"
};

firebase.initializeApp(config);
angular.module('app', ['firebase', 'ngMaterial', 'ngRoute', 'ngMap', 'ngCookies', 'ngProgress'])
  /////////////// Deshabilitar ARIA WARNINGS ////////////////////////
  .config(function ($mdAriaProvider) {
    // Globally disables all ARIA warnings.
    $mdAriaProvider.disableWarnings();
  })
  .config(function ($mdThemingProvider) {
    $mdThemingProvider.theme('default')
      .primaryPalette('amber', {
        'default': '400',
        'hue-1': '700',
        'hue-2': '900'
      })
      .accentPalette('red');
  })
  .config(function ($routeProvider) {
    console.log(' -- >' + $routeProvider);
    $routeProvider
      .when('/mapa', {
        templateUrl: 'mapa/mapa.html'
      })
      .otherwise({
        redirectTo: '/'
      })
  })
  ;
})
```

Figura 25. Código principal para aplicación web.
Elaborado por Valencia Jhon (2018).

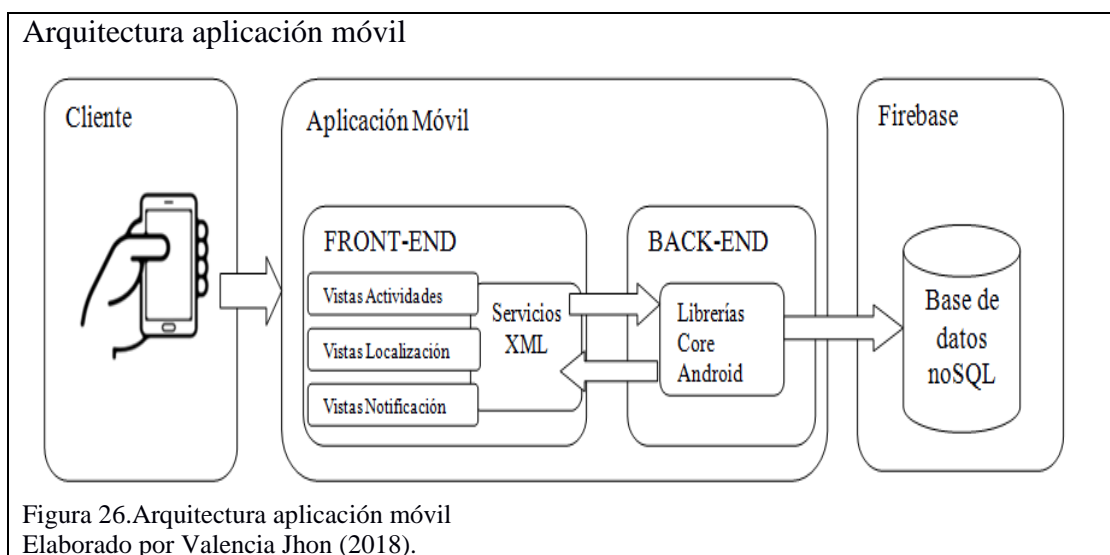
La parte inicial de la lógica de negocios, que muestra la figura 24 contiene la configuración de conexión entre la aplicación web y Firebase, en la cual se detalla una clave única, el dominio, el storage y la comunicación con la base de datos, las líneas de configuración se encuentran disponibles en la página oficial de Firebase.

Inicializa etiquetas que AngularJS ofrece determinadas en la línea de módulos, son llamadas de acuerdo a las necesidades del usuario, tan solo con describirlas en este línea destina a librerías, y capaces para ser utilizadas en todo el proyecto.

La primera parte del la lógica de negocios contiene a su vez elementos de rutas, donde solo son llamadas desde otras funcionalidades para ser utilizadas.

3.2 Sistema Móvil

3.2.1 Arquitectura



La figura 26 describe la arquitectura de una aplicación móvil, está conformado por tres componentes que son: el cliente, aplicación móvil y la herramienta de google Firebase.

El componente cliente, es aquella persona que interactuar con la aplicación móvil a través de un Smartphone permitiendo realizar una acción específica, tal es el caso de un registro de un incidente, donde la aplicación móvil ofrece dos componentes llamados Front-End y Back-End descritos de la misma forma que en la arquitectura de de la aplicación web, pero ocupan otro lenguaje.

El Front-End encargado de interactuar la app con el usuario permitiendo obtener vistas o actividades que ofrece Android. Las actividades ofrece un sin número de acciones que permite al usuario realizar con mayor facilidad como en este caso la obtención de una ubicación por medio de latitud y longitud, una fotografía y una descripción que posteriormente será enviada en un formato JSON, de forma que el Back-End ofrece mecanismos para obtener estos elementos descritos, mediante librerías importadas a la aplicación móvil.

De igual forma la herramienta de Firebase ofrece una base de datos en tiempo real para el registro de los incidentes.

3.3 Pruebas

Las pruebas se dividieron en dos módulos, el principal para la aplicación móvil, y el segundo para la aplicación web.

3.3.1 Módulo pruebas aplicación móvil.

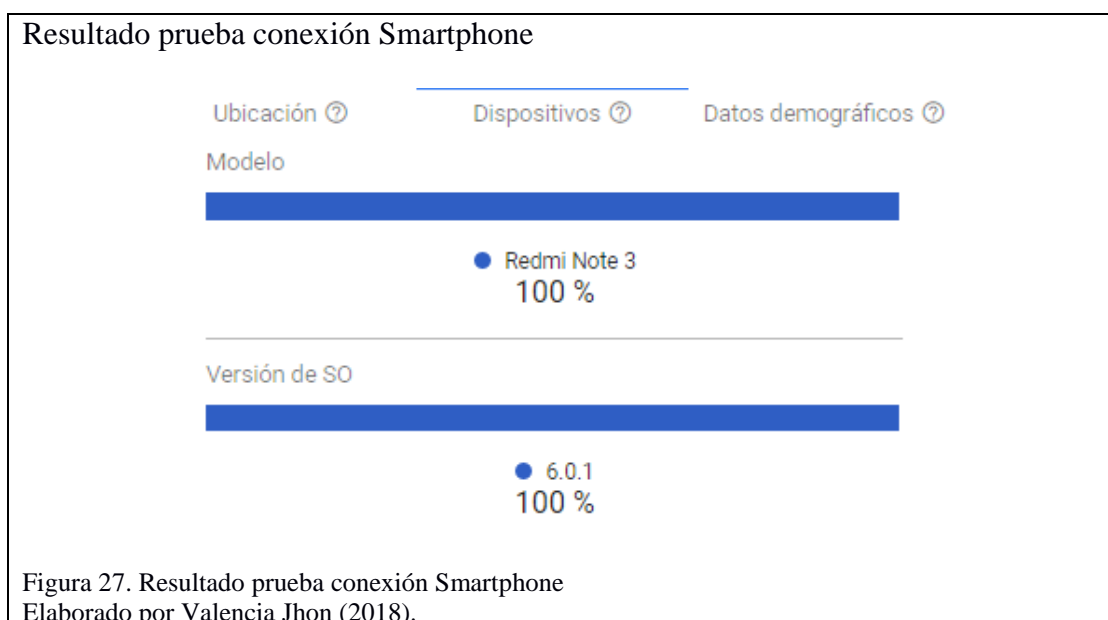
Este módulo utilizo un teléfono inteligente llamado Xiaomi Redmi Note 3 con las siguientes características:

Tabla 4. Características de Smartphone

Características	
Dimensiones	15 cm x 7,6 cm x 8,6 mm
Peso	164 gramos
Pantalla	5,5 pulgadas IPS con resolución FHD
Densidad de píxeles	403 ppi
Procesador	MediaTek Helio X10 y PowerVR G6200
RAM	3 GB
Sistema operativo	Android 6.0.1 Lollipop y MIUI 7
Almacenamiento	16 GB
Cámaras	Trasera de 13 mega píxeles y delantera de 5 mega píxeles
Batería	4.000 mAh
Otros	LTE, radio FM, cara rápida, lector de huellas

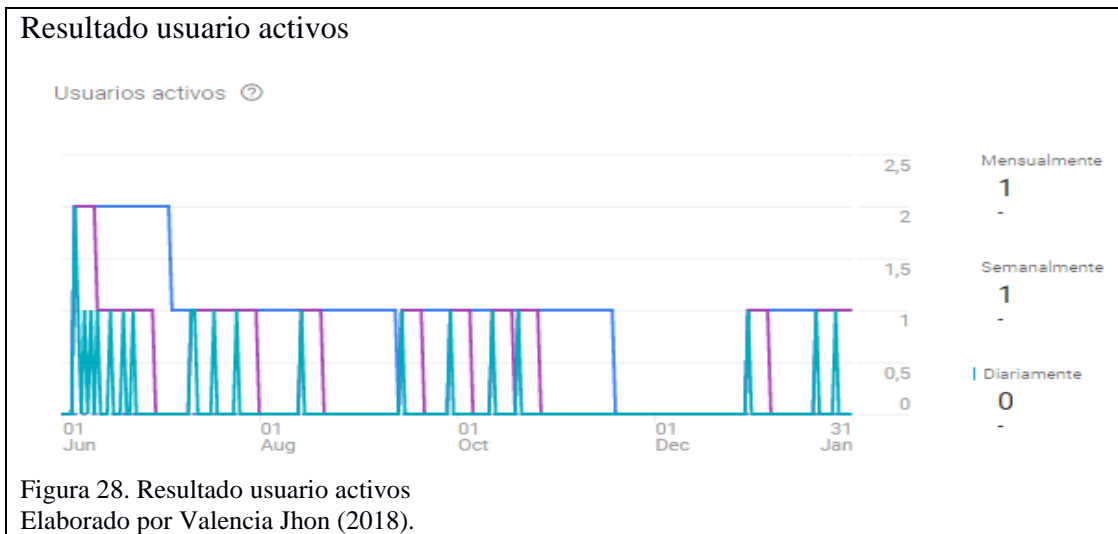
Nota: Tabla de características de Smartphone Xiaomi Redmi Note 3
Elaborado por Valencia Jhon (2018).

El primer paso fue la instalación breve de la aplicación, finalizando con total éxito. La aplicación se la utilizo para realizar pruebas en varias localidades de Quito, logrando a obtener un resultado de 44 registros obtenidos entre los meses de Julio y Diciembre del 2017.

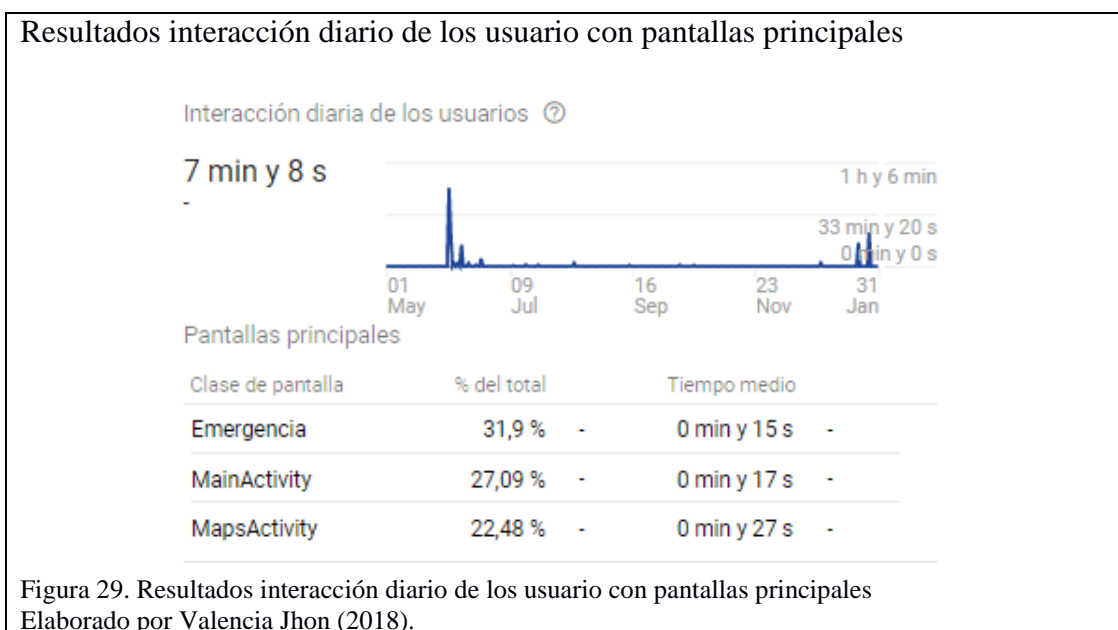


La figura 27 muestra que los dispositivos, que realizaron acciones fueron desde el teléfono inteligente descrito con anterioridad y verificamos que el Sistema Operativo

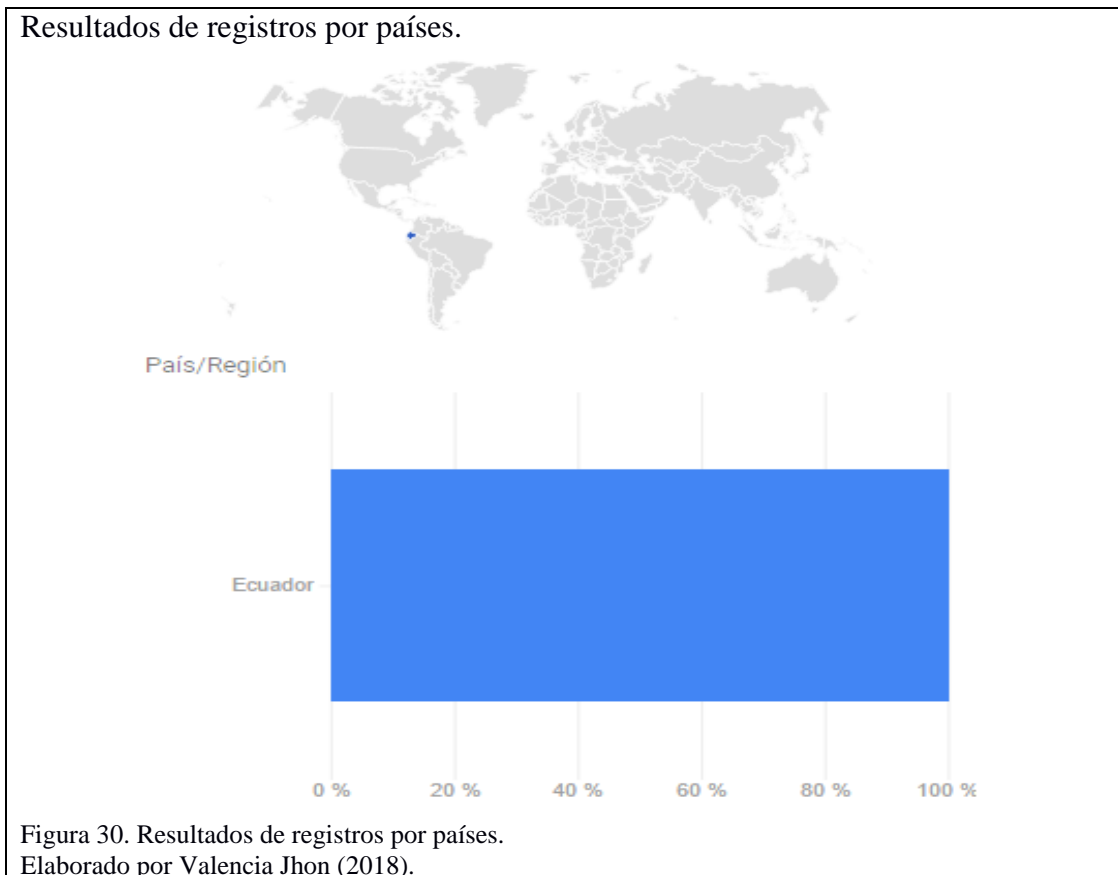
con el que trabaja es Android 6.0.1.



La figura 28 muestra, una cantidad de usuario activos entre los meses de Julio y Diciembre para respectivas pruebas de ingreso de datos. Obteniendo ingresos de usuario con un pico elevado en el mes de julio con dos registros de usuarios activos, y los meses restantes con solo un registro activo y con un mes de diciembre con cero usuario activos.



La figura de iteración diario de los usuarios nos ofrece el resultado de cada actividad de la aplicación móvil es decir la utilidad que ofrece cada pantalla en sí, con el registro de emergencia, las actividades main() y actividades con google Maps. El tiempo mínimo uso que nos ofrece cada pantalla se la encuentra en la parte derecha de la figura 29.



Todos los registros son creados en el Ecuador, verificando en la figura número 30, que ofrece una gráfica de los registros creados a nivel mundial.

Las figuras entre la 27 y 30, son datos obtenidos de análisis analíticos que ofrece la herramienta de Firebase, sin la posibilidad de ser modificadas, estos análisis se los realiza de acuerdo a la implementación de datos creados por los dispositivos Android.

Tabla de resultados de pruebas aplicación móvil.

Tabla 5. Tabla de resultados de pruebas aplicación móvil.

Prueba	Resultado
Dificultades con la versión de Android	Resultado con éxito, ninguna dificultad de versión
Instalación de la aplicación	Resultado con éxito, instalación sin ningún inconveniente.
Ejecución de la aplicación	Resultado con éxito, la aplicación se ejecuto con total normalidad.
Comunicación con Firebase	Resultado con éxito, comunicaciones en tiempo real.
Información <ul style="list-style-type: none"> ● Obtención de fotografía ● Obtención de tipo de registro y descripción ● Obtención de ubicación 	Resultados con éxito, para la obtención de estos tres elementos, mencionando que la fotografía se almacena en la nube y en la base de datos no SQL de Firebase.
Almacenamiento de registros	Registros almacenados con éxito.
Calidad de interfaces	Interfaces muy amigables nota, el color es muy llamativo.
Funcionalidades Aplicación en general	Resultado con éxito, las funcionalidades de la aplicación se ejecutan sin ningún error.

Nota: tabla de resultados de las pruebas ejecutadas en la aplicación móvil.
Elaborado por Valencia Jhon (2018).

3.3.2 Módulo de pruebas aplicación web

La aplicación web se ejecuto en una computadora con las siguientes características:

Tabla 6. Tabla de características de una computadora de ejecución del sistema web

Características	
Distribución	Linux Ubuntu 14.04 LTS
Arquitectura	AMD64
Memoria RAM	4GB
Procesador	Core i5
Disco Duro	1 Tera

Nota: Tabla de características de una computadora de ejecución del sistema web
Elaborado por Valencia Jhon (2018).

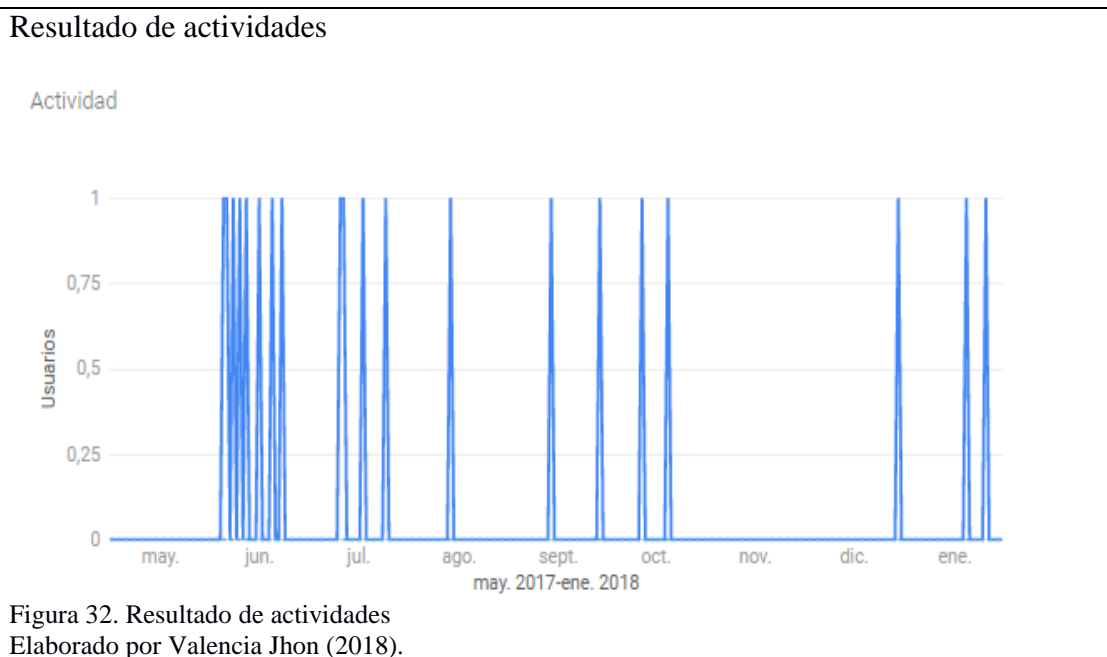
El primer procedimiento fue, instalación de un servidor de aplicaciones en este caso

Wamp Server para distribución de Linux, que permite ejecución de páginas web

diseñadas con cualquier framework.



El resultado de la figura 31, muestra las versiones obtenidas de la aplicación web, es decir la versión se mantiene estable desde el mes de Junio hasta el mes de Diciembre, sin recibir algún cambio muy importante. Llegando a la conclusión de que las interfaces son de muy agrado para los administradores de la página web.





Las figuras 32 y 33 muestran, las sesiones actividades desde la aplicación web, verificando que la aplicación web se encuentra con total normalidad y con un registro de actividades diariamente entre las fechas de Mayo del 2017 y Enero 2018. El cien por ciento de los registro de sesiones se inicial en el País Ecuador con un total de 26.

Las estadísticas de las figuras entre la 30 y 32, es obtenida desde la herramienta de Firebase de aplicaciones web, tratando de resolver alguna necesidad importante para el usuario, a la vez mostrar resultados importantes que fueron detallados en cada ítem.

Tabla de pruebas sistema web

Tabla 7. Tabla de resultados de las pruebas del sistema web.

Prueba	Resultado
Dificultades con la máquina de ejecución de la aplicación	Resultado con éxito, sin ninguna dificultad en la máquina de ejecución.
Instalación de la aplicación	Resultado con éxito, instalación sin ningún inconveniente.

Ejecución de la aplicación	Resultado con éxito, la aplicación se ejecuto con total normalidad.
Comunicación con Firebase	Resultado con éxito, comunicaciones en tiempo real.
Información <ul style="list-style-type: none"> ● Visualización de fotografía ● Visualización de tipo de registro y descripción ● Visualización de ubicación 	Resultados con éxito, para la visualización de estos tres elementos.
Reportes	Reportes obtenidos con éxito.
Calidad de interfaces	Interfaces muy amigables nota, el color es muy llamativo de igual manera que la aplicación móvil.
Funcionalidades de la aplicación en general	Resultado con éxito, las funcionalidades de la aplicación se ejecutan sin ningún error.

Nota: Tabla de resultados de pruebas ejecutas en la aplicación web.
Elaborado por Valencia Jhon (2018).

CONCLUSIONES

- El manejo de base de datos orientada a documentos, elimina los paradigmas de las bases de datos convencionales, las cuales generan funciones, tablas y otros elementos que las bases de datos noSQL no soportan.
- El formato de texto plano JSON, determinan un mejor desempeño de los datos, definiendo una secuencia estructurada de datos y accede a cambios futuros tan solo con editar o agregar elementos en un sector del texto sin afectar al resto.
- Los prototipos son una ayuda muy importante en los desarrollos de software, ya que permite descubrir la funcionalidad de la aplicación sea web o móvil, permitiendo plasmar las ideas que los usuario necesitan.
- El manejo de datos hoy en día, se los realiza a través de storage o bases de datos en la nube, permitiendo obtener los datos de forma rápida y oportuna, sin la necesidad de la obtención de elementos físicos, que con lleva a gastos muy fuertes.

RECOMENDACIONES

- Determinar la calidad de software que será generada, obteniendo todos los requisitos por parte del cliente y pruebas pertinentes, esto pretende aliviar molestias a futuro y cambios que por lo general se los realiza en tiempos finales de desarrollo.
- Ejecutar reglas de seguridad en base de datos y storage que la herramienta de google Firebase ofrece de manera gratuita, permitiendo el acceso a estos datos por el personal calificado y sofisticado.
- Obtener copias de seguridad en la base de datos de acuerdo a tiempos determinados, los tiempos son calculados de acuerdo al volumen de datos que se genera o el tipo de lógica de negocio de datos.
- Planificar de forma ordena el desarrollo de las aplicaciones, ejecutando al pie de la letra las funcionalidades a ser ejecutas de acuerdo a un límite de tiempo establecido, sin que sobre pase en tiempos a otros previamente detallados.
- Ejecutar pruebas detalladas y estructuradas con el fin de obtener beneficios de la aplicación o al contrario determinar estructuras con fallas internas, lo cual limita la calidad de software y usuario finales no satisfechos.
- Desempeñar las actividades de desarrollo de acuerdo a un cronograma de actividades, con el fin de obtener resultados favorables para las personas involucradas en el proyecto.

GLOSARIO DE TÉRMINOS

App: Es una aplicación de software que básicamente se pueden instalar en un dispositivos móviles, para ayuda al usuario en una acción específica.

Smartphone: Es un dispositivo físico, que su funcional primordial es interactuar entre un aplicación de software y el usuario.

Sistema de posicionamiento global: Conocido por sus siglas GPS, que nos ayuda a la navegación por satélite.

Sistema de información geográfica: Conocido por sus siglas GIS o SIG que significa de igual manera pero con diferente idioma, su función primordial es ayudar almacenar, presentar y la unión de datos espaciales sobre todo el mundo mediante un conjunto de herramientas.

Firestore: Es una plataforma de desarrollo de google en la nube, que soporta otras plataformas como por ejemplo (iOS, web, Android y servicios web). Firestore nos puede otorgar un almacenamiento de datos, bases de datos y aplicaciones necesarias como por ejemplo apis de google para implementar en proyectos de desarrollo de forma gratuita y de una forma fácil.

AngularJS: Es un framework de código abierto, realizado en javascript que se forma de un conjunto de librerías útiles para el desarrollo de aplicaciones web, de forma rápida en las transacciones que se ejecutan en el lado del cliente.

FrontEnd: Son aquellas tecnologías que se ejecutan en el lado del navegador, es decir la interpretación del sistema al usuario.

Backend: Son aquellas que se ejecutan en la parte del servidor como por ejemplo la conexión entre la base de datos y la aplicación.

JSON: Es un formato de texto plano muy ligero, que permite la iteración entre un sistema y la base de datos.

LISTA DE REFERENCIAS

- Amaro Soriano, J. E. (2011). *Android: Programación de dispositivos móviles a través de ejemplos*. Barcelona: Marcombo.
- Amo, F. A., Martínez, L., & Franci, N. (2005). *Introducción a la ingeniería del software*. México DF: Delta Publicaciones.
- Calero Muñoz, C., & PiattiniVelthuis, M. (2010). *Calidad del producto y proceso software*. s.f.: Editorial Ra-Ma.
- Davis, H., & Iwanow, D. (2009). *Google Advertising Tools: Cashing in with AdSense and AdWords*. California: O'Reilly Media, Inc.
- Dincer, A., & Uraz, B. (2013). *Google Maps JavaScript API Cookbook*. United Kingdom: Packt Publishing Ltd.
- Gibson, R., & Schuyler, E. (2006). *Google Maps Hacks. California, Estados Unidos*. California: O'Reilly Media, Inc.
- Gironés, J. T. (2012). *El Gran Libro de Android*. Barcelona: Marcombo.
- Green, B., & Seshadri, S. (2013). *AngularJS*. California: O'Reilly Media, Inc.
- Guochang, X. (2007). *GPS: Theory, Algorithms and Applications*. Berlín: Springer Science & Business Media.
- Letham, L. (2001). *GPS fácil. Uso del sistema de posicionamiento global. Argentina: Editorial Paidotribo*. Argentina: Paidotribo.
- Nixon, R. (2011). *HTML5 para iOS y Android*. Madrid: Anaya Multimedia.
- Pérochon, S. (2012). *Android: Guía de desarrollo de aplicaciones para Smartphones y Tablet*. Barcelona: Eni Ediciones.
- Ramos Salavert, I., & Lozano Pérez, M. D. (2010). *Ingeniería del software y bases de datos: tendencias actuales*. Cuenca: Univ de Castilla La Mancha.

Sommerville, I. (2005). *Ingeniería del software*. s.f.: Pearson Educación.

Worboys, M. (1995). *GIS: A ComputerSciencePerspective*. Florida: CRC Press.