

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



TESIS DE GRADO

**“MODELO WEB DE LOCALIZACIÓN Y REPORTE DE ACCIDENTES DE TRANSITO BASADO EN ANDROID”
CASO: Dirección Departamental de Transito de La Paz**

Mención: GESTIÓN Y PRODUCCIÓN

**Postulante: Gustavo Paco Machaca
Tutor Metodológico: Ing. Enrique Flores Baltazar
Tutor Especialista: Lic. Norma Mamani Quispe
Tutor Revisor: Lic. María Magdalena Aguilar Guanto**

EL ALTO – BOLIVIA

2020

Dedicatoria

Dedico este trabajo a Dios por darme la fortaleza y tenacidad para terminar este proyecto, a mi padre Ignacio y hermanos Silverio, Beatriz, Richard, Marina, Virginia y Verónica que con sus consejos han formado la persona que soy y en especial a mi madre Cecilia por su amor, dedicación y constante cooperación.

Agradecimientos

Agradezco a Dios por la fuerza y salud que me da para seguir adelante, agradezco a toda mi familia, por el apoyo que me brindo a lo largo de toda mi vida, e impulsarme a seguir y no rendirme.

A mis tutores por toda la paciencia y el apoyo que me brindaron en todo este proceso.

A todos mis amigos y amigas por brindarme la ayuda y apoyarme en los momentos más difíciles.

¡Muchas Gracias a todos!!

RESUMEN

Este trabajo se orientó a la presentación del marco teórico enfocado a mejoras futuras en relación con los accidentes de tránsito, que sirviera para desarrollar un modelo que posibilitara el desarrollo de algún producto útil para la mayor cantidad de personas, con el objetivo de mejorar la atención oportuna de las mismas. Por esta razón, se decidió investigar de qué manera se podría ayudar al común de la gente, con la tecnología actual, sin tener que incurrir en altos costos de implementación. es así, que el tema “Modelo web de localización y reportes de accidentes de tránsito basado en android” fue elegido para la misma.

En Bolivia, los automóviles no cuentan con sensores que permitan detectar accidentes de tránsito. Esta es la razón por la cual se buscará una arquitectura que permita hacer el uso de Teléfonos móviles, sin que sea necesario tener un automóvil de alta gama con sensor de choque integrado de fábrica.

Este trabajo consta de cinco capítulos los cuales se describen a continuación.

El capítulo I trata sobre la parte introductoria de presente tesis de grado identificando los problemas y objetivos a tratarse en la localización de accidentes de tránsito mediante celulares inteligentes.

En el capítulo II se muestra información teórica que ayudara a comprender las características que nos proporciona un Teléfono Inteligente, entre los cuales el uso de los sensores.

En el capítulo III se muestra el desarrollo del sistema que permitirá solventar los objetivos mencionados en el capítulo I.

En el capítulo IV se muestra pruebas y resultados.

En el capítulo V se realiza conclusión y resultados del modelo.

PALABRAS CLAVE: Móvil, Accidente, Sistema, Sensor, Transito

ABSTRACT

This work was oriented to the presentation of the theoretical framework focused on future improvements in relation to traffic accidents, which would serve to develop a model that would allow the development of some useful product for the largest number of people, with the aim of improving care timely of the same. For this reason, it was decided to investigate how the common people could be helped, with current technology, without having to incur high implementation costs. Thus, the topic “Android based location and reporting model for traffic accidents” was chosen for it.

In Bolivia, cars do not have sensors to detect traffic accidents. This is the reason why an architecture will be sought that allows the use of mobile phones, without having to have a high-end car with an integrated shock sensor from the factory.

This work consists of four chapters which are described below.

Chapter I deals with the introductory part of this degree thesis identifying the problems and objectives to be addressed in the location of traffic accidents using smart phones.

Chapter II shows theoretical information that will help us understand the characteristics provided by a mobile phone, among which the use of sensors.

Chapter III shows the development of the system that will allow solving the objectives mentioned in Chapter I.

Chapter IV Evidence and results are shown

Chapter V makes the conclusion and results of the model.

KEY WORDS: Mobile, Accident, System, Sensor, Traffic

INDICE

1.	MARCO PRELIMINAR.....	¡Error! Marcador no definido.
1.1.	INTRODUCCION.....	¡Error! Marcador no definido.
1.2.	ANTECEDENTES.....	2
1.3.	PLANTEAMIENTO DEL PROBLEMA	4
1.3.1.	Problema Principal.....	4
1.3.2.	Problemas Secundarios.....	5
1.4.	OBJETIVOS	5
1.4.1.	OBJETIVO GENERAL.....	5
1.4.2.	Objetivos Especificos	5
1.5.	HIPOTESIS	¡Error! Marcador no definido.
1.5.1.	Identificación De Variables	6
1.5.2.	Operacionalizacion de variable e indicadores	6
1.5.3.	Conceptualización de Variable.....	7
1.6.	JUSTIFICACION.....	¡Error! Marcador no definido.
1.6.1.	Justificación Científica	7
1.6.2.	Justificación Social	8
1.6.3.	Justificación Técnica.....	8
1.6.4.	Justificación Económica.....	9
1.7.	METODOLOGIA.....	¡Error! Marcador no definido.
1.7.1.	Metodología Científico	9
1.7.2.	Metodología Mobile-D.....	9
1.7.3.	Métricas de calidad de Software (Qsos)	10
1.7.4.	METRICAS DE COSTOS COCOMO II.....	11
1.7.5.	Métricas de Calidad de Software (ISO-9126).....	12
1.8.	HERRAMIENTAS	13
1.8.1.	AndroidDevelopment Tool ADT	13
1.8.2.	Php.....	13
1.8.3.	Sqlite	13
1.8.4.	Eclipse.....	13
1.8.5.	Xampp	14
1.8.6.	Base de Datos	14
1.8.7.	Css3	14
1.8.8.	Html.....	15
1.8.9.	Google Maps	15
1.9.	LIMITES Y ALCANCES	15
1.9.1.	LIMITES.....	15
1.9.2.	ALCANCES	16
1.9.3.	APORTES	17
2.	MARCO TEÓRICO	18
2.1.	INTRODUCCIÓN.....	18
2.2.	DESCRIPCIÓN DE TIPOS DE ACCIDENTES DE TRANSITO.....	18
2.2.1.	Clasificación según número de vehículos	18

2.2.1.1.	Son Accidentes Simples	19
2.2.1.2.	Accidentes Múltiples	19
2.2.1.3.	Accidentes Múltiples	21
2.3.	TELÉFONOS INTELIGENTES	21
2.4.	SISTEMA OPERATIVO ANDROID	22
2.4.2.	Características De Android	23
2.4.3.	Arquitectura De Android.....	23
2.5.	ESTRUCTURA DE UNA APLICACIÓN DE ANDRIOD	25
2.5.1.	Activity	25
2.5.2.	Intents.....	26
2.5.3.	Listener.....	27
2.5.4.	Servicios	27
2.5.5.	Content Provider	27
2.5.6.	Android Manifiest	27
2.6.	VERSIONES DEL SISTEMA OPERATIVO ANDROID.....	28
2.7.	SENSORES DE UN TELEFONO.....	30
2.8.	SISTEMA DE CORDENADAS DE ANDROID	32
2.9.	MENSAJES	33
2.10.	MODELO CON UN MINIMO DE DATOS UNIFORMES DEL ACCIDENTES	34
2.10.1.	Organización de los Elementos de Datos MMUCC.....	34
2.11.	LENGUAJE PARA EL INTERCAMBIO DE DATOS DE EMERGENCIA EDXL 36	
2.11.1.	Descripción.....	36
2.11.2.	Estrucctura del Elemento de Distribución EDXL	37
2.11.3.	Requerimientos para el diseño	38
2.12.	PROTOCOLO PARA ENVIO DE MENSAJES BACKEND FRONDEND 40	
2.12.1.	CONJUNTO DE DATOS DE URGENCIA	40
2.13.	DECIBEL	41
2.14.	MEDICION DE LA ACUSTICA.....	42
2.15.	GEOLOCALIZACIÓN.....	43
2.16.	MENSAJERIA EN LA NUBE	45
2.17.	INGENIERIA DE SOFTWARE	49
2.17.1.	CARACTERISTICAS DE INGENIERIA DE SOFTWARE	50
2.17.3.	OBJETIVOS DE INGENIERIA DE SOFTWARE	50
2.17.4.	ETAPAS DE INGENIERIA DE SOFTWARE	51
2.18.	METODOLOGIA MOBILE-D.....	52
2.18.1.	Fases.....	53
2.18.1.1.	Fase Exploración.....	53
2.18.1.2.	Fase de Inicialización	54
2.18.1.3.	Fase de Producción.....	54
2.18.1.4.	Fase de Estabilización.....	54
2.18.1.5.	Fase de Pruebas	55
2.19.	METRICAS DE CALIDAD DE SOFTWARE – QSOS.....	55

2.19.1.	Pasos De La Metodología Qsos	56
2.19.2.	METRICAS MAS USADAS POR QSOS	58
2.19.3.	Formulas y Evaluación de Criterios.....	60
2.20.	COSTOS COCOMO II	60
2.20.1.	MODELO DE DESARROLLO	62
2.20.2.	Modelo Básico	64
2.20.4.	Modelo Intermedio	65
2.20.5.	Modelo Detallado.....	68
2.21.	METRICAS DE CALIDAD DE SOFTWARE (ISO-9126).....	68
3.	MARCO APLICATIVO	72
3.1.	INTRODUCCIÓN.....	72
3.2.	METODOLOGIA MOBILE-D.....	72
3.2.1.	Fase 1: Exploración	73
3.2.1.1.	Establecimiento de las Partes.....	73
3.2.1.2.	Definición del Alcance	73
3.2.1.3.	Establecimiento del Proyecto	74
3.2.2.	Fase 2: Inicialización	75
3.2.2.1.	Configuración del Proyecto.....	75
3.2.2.2.	Planificación Inicial En 0 Iteraciones.....	75
3.2.2.3.	Jornada de Trabajo en o Iteración	79
3.2.2.4.	Caso de Uso.....	80
3.2.2.5.	Diagrama de Actividades.....	84
3.2.2.6.	Diagrama de Secuencia	84
3.2.2.7.	Diagrama de Clases	87
3.2.2.8.	Elaboración de prototipos mediana de fidelidad (Dia de publicación en iteración 0).....	87
3.2.3.	Fase 3: Producción.....	90
3.2.3.1.	Implementación	90
3.2.3.2.	Creando el Proyecto en Google Api	97
3.2.4.	Fase 4: Estabilización.....	100
3.2.5.	Fase 5: Pruebas	100
3.2.5.1.	Plan de Pruebas.....	100
3.2.5.2.	Ambiente de Pruebas	102
3.3.	MÉTRICAS DE CALIDAD (MODELO - QSOS).....	103
3.3.1.	Tarjeta de identificación de Software “Localización de Accidentes de Tránsito”	103
3.3.2.	Evaluaciones los criterios de valoración (Sistema: Detección de Accidentes de Tránsito)	103
3.4.	COSTO DE COCOMO.....	105
3.5.	METRICAS DE CALIDAD DE SOFTWARE (ISO - 9126).....	108
3.5.1.	Usabilidad.....	108
3.5.2.	Funcionalidad	110
3.5.3.	Confiabilidad.....	113
3.5.4.	Mantenibilidad	114
3.5.5.	Portabilidad.....	115

3.5.6.	Resultados.....	115
4.	PRUEBAS Y RESULTADOS	117
4.1.	PRUEBAS	117
4.1.1.	Capturas de Pantalla de la Aplicación Movil	117
4.1.2.	Capturas de Pantalla de la Plataforma Web	120
4.2.	RESULTADOS	124
4.3.	PRUEBA DE HIPOTESIS	128
4.3.1.	Definición De La Hipótesis	128
4.3.2.	Evaluación De Resultados	128
4.3.3.	Determinación de la Región Critica.....	129
4.3.4.	CALCULO ESTADISTICO DE LA PRUEBA	130
4.3.5.	TOMA DE DECISION	130
5.	CONCLUSIÓN Y RECOMENDACIONES	131
5.1.	CONCLUSIÓN.....	131
5.2.	CUMPLIMIENTO DE LOS OBJETIVOS	131
5.3.	RECOMENDACIONES.....	132

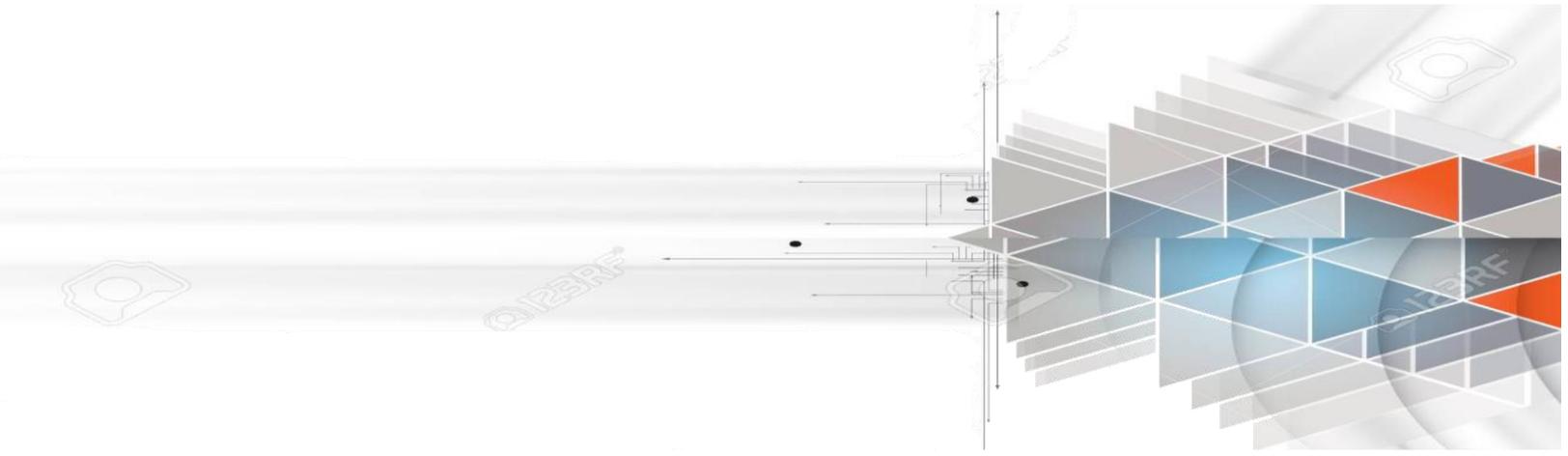
INDICE DE TABLAS

Tabla 1.1	Descripción de Operacionalización de variables.....	6
Tabla 2	Descripción de Operacionalización de variables.....	7
Tabla 2. 1	Lista de versiones de Android con su respectivo API	30
Tabla 2. 2	Conjunto de datos de urgencia	41
Tabla 2. 3	Nivel de intensidad del sonido	42
Tabla 2. 4	Evaluación de Criterios	60
Tabla 2. 5	Ecuaciones del Modelo Básico de COCOMO	65
Tabla 2. 6.	Ecuaciones del Modelo Intermedio de COCOMO	67
Tabla 3. 1	Requerimientos definidos (Gestión de Aplicación)	79
Tabla 3. 2	Requerimientos definidos (Georreferenciación)	80
Tabla 3. 3	Requerimientos definidos – Consultar Información (Datos de Conductor).....	80
Tabla 3. 4	Caso de uso de Usuarios	82
Tabla 3. 5	Caso de Uso Administrador	83
Tabla 3. 6	Caso de Uso de Registro de Usuario.....	83
Tabla 3. 7	Detalle del plan de pruebas	101
Tabla 3. 8	Cronograma estimado para la realización de las pruebas	102
Tabla 3. 9	Pantalla Principal Plataforma Web.....	103
Tabla 3. 10	Evaluaciones los criterios de valoración	104
Tabla 3. 11	Evaluaciones los criterios de valoración	105
Tabla 3. 12	Conversion de Punto de Fusion a KLDC	106
Tabla 3. 13	Coeficiente ab y cb y los exponentes bb y db	107
Tabla 3. 14	Tabla de Usabilidad del Sistema.....	109
Tabla 3. 15	Cálculo de Funcionalidad Según el Punto Función.	111
Tabla 3. 16	Tabla de Ajuste de Complejidad	111
Tabla 3. 17	Resultados	115
Tabla 4. 1	Datos de la Prueba usando la aplicación de la presente tesis.	126

INDICE DE FIGURAS

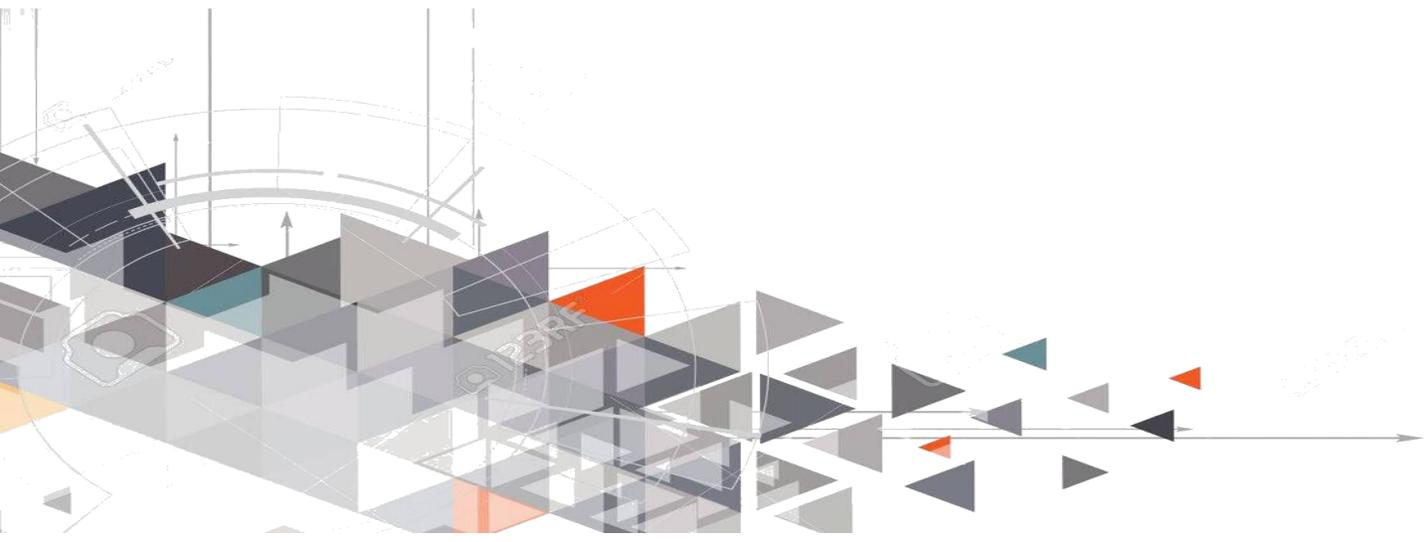
Figura 1. 1 Descripción de Operacionalización de variables	10
Figura 2. 1 Telefonos Moviles (www. mobiles.com, 2002)	21
Figura 2. 2 Arquitectura de Android[www.android.com,2002]	23
Figura 2. 3 Sistema de coordenadas en Android.[groupandroid, 2011].....	32
Figura 2. 4 Estructura de Google Cloud Messaging [developers.google.com, 2008)]	46
Figura 2. 5 Diagrama de Registro del dispositivo [android.com, 2007]	47
Figura 2. 6 Diagrama del envío de la notificación [android.com, 2007]	48
Figura 2. 7 Etapas de la Ingeniería de Software [Boem, 2014]	52
Figura 2. 8 Ciclo de desarrollo de la metodología MOBILE –D [Bonilla, 2014] ..	53
Figura 2. 9 Etapas de fase Exploración [Bonilla, 2014]	54
Figura 2. 10 Proceso fase de Inicialización [Bonilla, 2014].....	54
Figura 2. 11 Proceso fases de Producción [Bonilla, 2014]	54
Figura 2. 12 Proceso fase de Estabilización [Bonilla, 2014]	55
Figura 2. 13 Sistema de Calidad QSOS [www.org.com, 2006]	55
Figura 3. 1 Diagrama de despliegue del Sistema [Elaboración propia]	76
Figura 3. 2 Arquitectura del Sistema [Elaboración Propia]	78
Figura 3. 3 Diagrama de Caso de Uso General [Elaboración Propia]	81
Figura 3. 4 Diagrama de Caso de Uso de Usuarios [Elaboración Propia]	81
Figura 3. 5 Diagrama de Caso de Uso de Usuarios [Elaboración Propia]	82
Figura 3. 6 Diagrama de Caso de Uso de Registro de Socio [Elaboración Propia]	83
Figura 3. 7 Diagrama de Actividades [Diagrama de Actividades]	84
Figura 3. 8 Diagrama de Secuencia del Celular hasta que se detecta un choque [Elaboración Propia]	85
Figura 3. 9 Diagrama de Colaboración [Elaboración Propia].....	85
Figura 3. 10 Diagrama de secuencia del celular luego de detectarse un choque [Elaboración Propia]	86
Figura 3. 11 Diagrama de Colaboración del Administrador [Elaboración Propia]	86
Figura 3. 12 Diagrama de Clases Móviles [Elaboración Propia].....	87
Figura 3. 13 Diagrama de Presentación Login [Elaboración Propia]	88
Figura 3. 14 Diagrama de Presentación Entorno Home [Elaboración Propia] ...	88
Figura 3. 15 Diagrama de Presentación Inicio [Elaboración Propia].....	89
Figura 3. 16 Diagrama de Presentación Entorno Geolocalización [Elaboración Propia]	89
Figura 3. 17 Diagrama de Presentación, Registro de usuario y vehículo [Elaboración Propia]	90
Figura 3. 18 Diagrama de flujo de Implementación [Elaboración Propia]	91
Figura 3. 19 Pasos para la obtención de datos [Elaboración Propia]	92
Figura 3. 20 Fragmento de código, para el uso del sensor acelerómetro [Elaboración Propia]	93

Figura 3. 21 Fragmento de código para obtener la localización [Elaboración Propia]	93
Figura 3. 22 Fragmento de código para obtener la intensidad de sonido [Elaboración Propia]	94
Figura 3. 23 Fragmento de código para el envío de datos de Frontend a Backend [Elaboración Propia]	95
Figura 3. 24 Fragmento de código para notificaciones lado Frontend [Elaboración Propia]	96
Figura 3. 25 Fragmento de código para notificaciones lado Backend [Elaboración Propia]	96
Figura 3. 26 Creando Google Api [Elaboración Propia]	97
Figura 3. 27 Creando el proyecto [Elaboración Propia]	97
Figura 3. 28 Creando Credenciales [Elaboración Propia]	98
Figura 3. 29 Creando Api Key [Elaboración Propia]	98
Figura 3. 30 Obteniendo la clave de acceso al API [Elaboración Propia]	99
Figura 3. 31 Activando Google Maps API Key [Elaboración Propia]	99
Figura 3. 32 Habilitando Maps SDK for Android [Elaboración Propia]	99
Figura 3. 33 Esquema de la fase de estabilización [wpcentral.com, 2006]	100
Figura 4. 1 Aplicación Instalada [Elaboración]	117
Figura 4. 2 Menu Principal de la Aplicación Móvil [Elaboración Propia]	117
Figura 4. 3 autenticación del Usuario	118
Figura 4. 4 configuración de datos personales y del vehículo [Elaboración Propia]	118
Figura 4. 5 Visualización de la aplicación en ejecución [Elaboración Propia]	119
Figura 4. 6 Dialogo de información de un hospital [Elaboración Propia]	119
Figura 4. 7 Trazado de cómo llegar a un hospital determinado [Elaboración Propia]	120
Figura 4. 8 Pantalla Principal Plataforma Web [Elaboración Propia]	120
Figura 4. 9 Pantalla de Estadísticas de reportes de Accidente [Elaboración Propia]	121
Figura 4. 10 Pantalla de Registro de Reportes [Elaboración Propia]	122
Figura 4. 11 Pantalla registro de nuevo Administrador [Elaboración Propia]	122
Figura 4. 12 Pantalla de Registro de Accidentes [Elaboración Propia]	123
Figura 4. 13 Pantalla de Geolocalización de Hospitales [Elaboración Propia]	123
Figura 4. 14 Pantalla de Geolocalización del Accidente [Elaboración Propia]	124



CAPÍTULO I

MARCO PRELIMINAR



1. MARCO PRELIMINAR

1.2. INTRODUCCIÓN

En Bolivia a causa de muchos problemas sociales, políticos, económicos y culturales la sociedad se ve obligada a marchar o bloquear en las carreteras como medida de protesta, esto causa embotellamiento vehicular y una pérdida económica para los trabajadores de a pie especialmente para los choferes.

Con la incertidumbre de que puedan ocurrir paros, marchas, bloqueos y otras manifestaciones. Los choferes se pasan en luz roja, se insultan y hasta existen peleas con golpes entre conductores; esto se registra a diario en las vías de las ciudades bolivianas. Estar al mando de un volante da poder, sobre todo frente a los indefensos peatones que tienen que correr en las esquinas o andar por la calzada porque los vehículos ocupan la acera.

El perfil del conductor tipo en Latinoamérica se resume en: falta de cortesía, a menudo egoísta y egocéntrico, propenso a la intolerancia, permeable al ánimo revanchista que impera en el medio y obtuso para la autocrítica o la auto observación. Esa descripción se adecua al comportamiento de los choferes de La Paz que siempre están apurados, tanto que no pueden ni esperar unos segundos de luz roja. O cuando hay alguna marcha o desfile, los choferes dan vueltas prohibidas y circulan en contra ruta, sin cuidado. Esto causa accidentes de tránsito en algunos casos.

Los accidentes de tránsito en nuestro país han ocasionado en el periodo 2000 – 2017 36.512 víctimas mortales y 342.766 lesionados, incrementándose las cifras año tras año. (INE, 2017)

Una de las causas de accidentes de tránsito es cuando el conductor atiende una llamada mientras conduce, pero este móvil se podría utilizar de la mejor manera posible para este suceso.

El uso de teléfonos inteligentes se hace cada vez más útil y necesario. Actualmente, los teléfonos inteligentes, suelen venir con una cantidad de

sensores que va aumentando con cada nuevo modelo o versión. Ya es común encontrarse con acelerómetros en la mayoría de los celulares, poco a poco, están comenzando a encontrarse giróscopos (para medir la velocidad angular), sensores de proximidad (para bloquear la pantalla al hablar por teléfono, de manera de, por ejemplo, no se presione botones con la oreja sin querer), brújulas electrónicas, etc.

Android, a partir de la versión 2.3 (Gingerbread) fue un paso más allá e incluyó dos nuevos sensores de fusión, estos permiten obtener la aceleración lineal sobre los 3 ejes y una matriz de rotación. Usando todos los sensores de un celular con estas características, se podría llegar a conocer exactamente como se está moviendo un celular o, mejor aún, saber cómo se movió en el pasado. Una característica de este tipo, en un choque, podría significar tener un indicio (si bien a grandes rasgos), de cómo ocurrió un accidente.

Existen hardware sobre los sensores para automóviles como: sensores de colisión, sensores de temperatura, sensores de velocidad, sensores de cambio de dirección, y otros.

En el siguiente trabajo se propone un modelo web de detección de accidentes de tránsito que haga uso de aplicaciones Android y sensores de colisión para mejorar los resultados, en el tiempo de atención a los pacientes accidentados en siniestros de tránsito.

1.3. ANTECEDENTES

Actualmente en Bolivia no existe este modelo sobre la detección de accidentes de tránsito mediante plataforma Android.

Internacional

“AACN¹ AdvancedAutomaticCollision/CrashNotification” en EE. UU., por: la compañía TSPs en el año 1996, los vehículos cuentan con un dispositivo que puede recabar y enviar información a una central que pueda analizar el tipo de ayuda necesaria para el accidente en particular, como ser: cantidad de ambulancias necesarias, envió de una grúa para remover el/los vehículo/s o no, envió de helicópteros, etc., además de alertar a los servicios de emergencias y decidir donde convendría derivar a los afectados sobre la base de cantidad de camas disponibles, tipo de complejidad que el hospital está preparado para tratar, etc (**Computerworld Honors Case Study, 2005**).

“eCall” eCall Discussion Paper. Finnish eCall Experts 6.6., un sistema automático de servicio de emergencias dentro del vehículo. Un vehículo equipado con eCall cuenta con una terminal capaz de obtener el posicionamiento mediante satélites, comunicaciones inalámbricas y sensores para la detección de colisiones, vuelcos e incendios (**Desarrollado en la Unión Europea, 2002**).

Nacional

[Chuquimia Condori, Juan Carlos]; “GEOREFERENCIACIÓN DE CAJEROS AUTOMÁTICOS Y AGENCIAS BANCARIAS MEDIANTE APLICACIÓN ANDROID”, Se fundamenta en desarrollar un prototipo el cual será usado en su mayoría por personas que dependen de cuentas bancarias, los mismos que realizan transacciones asistiendo a una sucursal o mediante el uso de cajeros automáticos (ATM²), permitiéndoles geolocalizar estos centros en cualquier región y localidad de Bolivia. El prototipo será desarrollado para dispositivos móviles que cuentan con una plataforma Android, debido a que en su gran mayoría utiliza este sistema, el mismo permitirá un acercamiento entre las personas y estos centros de transacción, aprovechando la portabilidad y

¹ Notificación automática avanzada de colisión / choque

² **ATM** son las siglas en inglés de Automated Teller Machine, es decir, cajero automático

movilidad de las personas para usar esta aplicación en cualquier momento y lugar, dando la oportunidad de poder consultar sobre los Cajeros Automáticos y Agencias de Banco más próximos que se encuentran alrededor de los usuarios, cuando ya se encuentren en una zona específica. La aplicación se desarrollará de tal manera que sea utilizable tanto en Smartphones como en Tablets, manteniendo la usabilidad y la estética para las diferentes posibilidades de resoluciones y tamaños de pantallas, para lograr abarcar un mayor número de posibles usuarios. (**Universidad Mayor de San Andrés, 2015**).

[Guachalla Blanco, Fabio Rodolfo]; “Plataforma de geolocalización de personas mediante dispositivo móvil”, Las nuevas tecnologías y la sociedad de la información están creando nuevas vías de comunicación social, en nuestro medio se hace evidente el crecimiento del uso de terminales móviles facilitando el acceso a Internet, produciendo sociedades inteligentes. El presente trabajo tiene como objetivo transformar esa necesidad emergente en un marco teórico y aplicativo que convergen como una base para el desarrollo de un sistema cuyo proceso principal es la geolocalización de personas por medio del GPS de su dispositivo móvil. La construcción de esta aplicación fue posible con el uso de herramientas tales como: PhoneGap, para la creación de la interface, SOA para el modelado de los servicios usando RESTful para la integración de la aplicación con el SOA de ésta a partir de marcadores propios de la librería. (**Universidad Mayor de San Andrés, 2015**).

1.4. PLANTEAMIENTO DEL PROBLEMA

1.4.1. Problema Principal

Debido a muchos accidentes automovilísticos en Bolivia, en los cuales no se ha podido llegar a tiempo al lugar del mismo, dejando pasar la llamada “Golden Hour”, aquella hora posterior al accidente donde se tienen mayores posibilidades de llegar a la brevedad posible al lugar del hecho y ayudar a las víctimas de manera oportuna.

1.4.2. Problemas Secundarios

- ✓ El tiempo de respuesta con proveedores de atención pre-hospitalaria no es inmediata, en algunos casos causando muertes.
- ✓ Aumento de muertes, como consecuencia de accidentes de tránsito.
- ✓ La detección del accidente de tránsito dura demasiado tiempo, provocando que los afectados no sean socorridos a tiempo.
- ✓ Los conductores no son notificados a tiempo del lugar donde ocurrió el accidente, esto puede provocar más accidentes con los conductores que transitan en la ruta donde ocurrió el hecho.

¿La aplicación móvil basada en el modelo web de localización y reportes de accidentes de tránsito basado en android, será capaz de cooperar al usuario y al Dirección Departamental de Transito de La Paz en la atención rápida y eficaz de accidentes, utilizando Geolocalización?

1.5. OBJETIVOS

1.5.1. OBJETIVO GENERAL

Desarrollar un modelo para la localización automática de accidentes de tránsito que informe instantáneamente al Dirección Departamental de Transito de La Paz para que estos puedan llegar en el menor tiempo posible.

1.5.2. Objetivos Específicos

- ✓ Analizar la información y los requerimientos necesarios
- ✓ Enviar mensajes a Dirección Departamental de Transito de La Paz del lugar de hecho del accidente.
- ✓ Desarrollar una aplicación utilizando el sensor acelerómetro, para detectar la colisión.
- ✓ Aplicar una metodología de desarrollo agil adecuado para el proyecto.
- ✓ Visualizar en un mapa la localización precisa del incidente.
- ✓ Realizar el reporte de Accidentes de Tránsito.

1.6. HIPÓTESIS

El uso de los teléfonos inteligentes con sensor acelerómetro, permitirán el desarrollo de un sistema capaz de detectar un accidente de tránsito, para que así los afectados sean atendidos en el menor tiempo posible”.

1.6.1. Identificación De Variables

VARIABLE DEPENDIENTE

Diseño de aplicaciones móviles

VARIABLE INDEPENDIENTE

modelo web de localización y reportes de accidentes de tránsito

1.6.2. Operacionalización de variable e indicadores

OPERACIONALIZACION DE LA VARIABLE DEPENDIENTE Diseño de aplicaciones móviles

Tabla 1.1
Descripción de Operacionalización de variables

DIMENSIONES	INDICADORES	ITEMS	TECNICAS E INSTRUMENTOS
Atención	Tiempo de atención	¿Alguna aplicación para el modelo de localización de accidentes de tránsito?	Software
Capacidad	Hospitales	¿Es indispensable una aplicación móvil para modelo de localización de accidentes de tránsito?	Cuestionario
Costo	Transito	¿Mejorara la atención de accidentes una aplicación móvil?	Computadora
	Usuarios		Dispositivo Móvil
	Ambulancias		

Fuente: Elaboración propia

OPERACIONALIZACIÓN DE LA VARIABLE INDEPENDIENTE Modelo web de localización y reportes de accidentes de tránsito

Tabla 1. 2
Descripción de Operacionalización de variables

DIMENSIONES	INDICADORES	ITEMS	TECNICAS E INSTRUMENTOS
Marco teórico	Eclipse SQLite Java.	¿Qué tipo de aplicaciones se puede realizar en la plataforma Android? ¿Qué beneficios ofrece un dispositivo móvil con Android?	Software Computadora
Fundamentación investigativa	Metodología Mobile-D	¿Qué ventajas da utilizar una aplicación móvil? ¿Qué metodología es la adecuada para desarrollar este tipo de aplicaciones?	Diagramas

Fuente: Elaboración propia

1.6.3. Conceptualización de Variable

Localización del accidente

Localización: Localización es el acto el resultado de localizar: indicar o establecer la ubicación geográfica de algo o alguien. El termino alude al emplazamiento en el espacio, defino a partir de distintas referencias

Accidente: Es un suceso no planeado y no deseado que provoca un daño, lesión u otra incidencia negativa sobre un objeto o sujeto. Para tomar esta definición, se debe entender que los daños se dividen en accidentales e intencionales (o dolosos y culposos) (Robertson, 2015).

1.7. JUSTIFICACIÓN

1.7.1. Justificación Científica

Con el desarrollo de nuevas tecnologías y herramientas en las distintas ramas de la ciencia, hacen posibles aplicarlas en distintos campos, dando una solución más práctica y rápida, es así, que el uso de los teléfonos inteligentes que cuentan con tecnología Android, nos presenta otra posible solución para mejorar la atención a los que sufran accidentes de tránsito haciéndoles llegar los servicios de ayuda

en el menor tiempo posible, la aplicación en el dispositivo móvil enviara un mensaje de alarma y los datos posibles del accidente a la central (web service) del administrador, transito este enviara grúa si fuese necesario en el hecho.

1.7.2. Justificación Social

Con el desarrollo del sistema de detección de accidentes de tránsito se busca beneficiar a aquellos conductores y/o pasajeros que pueden sufrir un accidente de tránsito, ayudar a los afectados que sean atendidos oportunamente, a su vez beneficiara a instituciones que interceptan en este tipo de hechos como ser: (hospitales enviando ambulancias, bombero, etc....), llegando al lugar del hecho en el menor tiempo posible.

Así también el transito se beneficiará, para poder realizar cortes oportunas de vías en las cercanías del accidente, además con los datos del accidente que le proporcionara el sistema vera que si es necesario enviar una grúa o no.

1.7.3. Justificación Técnica

Se justifica tecnológicamente con la elección de la plataforma Android, por expandirse exponencialmente en los últimos años, siendo Android una de las plataformas más usadas por la gente, basándose en un lenguaje de programación Java, introduciendo el concepto de geolocalización que es muy importante en la actualidad, teniendo a nuestra disposición los mapas de google.

Se Propone un prototipo para el desarrollo de una aplicación móvil cuyo objetivo es servir como fuente para informar acerca de los accidentes de tránsito, mediante el uso de un teléfono inteligente con disponibilidad de internet y GPS. Asimismo, aplica la metodología de desarrollo Mobile-D para el desarrollo de la aplicación móvil y elabora un plan de pruebas para verificar que el prototipo cumpla con los requerimientos definidos.

1.7.4. Justificación Económica

En el desarrollo del software de detección de accidentes de tránsito se basará en software libre y código abierto, lo cual implica independencia total en cuestión de licencias y de desarrollo tecnológico, un costo nulo de adquisición. Implica a su vez un ahorro en dinero a los conductores en que sus automóviles tengan que equipar con la última tecnología para ser detectados en el accidente que puedan tener.

1.8. METODOLOGÍA

1.8.1. Metodología Científico

La metodología de investigación de la presente tesis se apoya en el Método Científico que sirve de guía en la organización de todo el proceso de investigación, el mismo que llegara a cubrir los requerimientos necesarios para que los objetivos planteados se lleguen a cumplir. La investigación científica es muy importante para poder resolver los problemas en el área del transporte público aplicando diferentes principios conceptos que clasifican la teoría a la práctica, entre estas tenemos la Observación, Experimentación y la Entrevista.

1.8.2. Metodología Mobile-D

La metodología a usar para el desarrollo Modelo web de localización y reportes de accidentes de tránsito basado en android, será Mobile-D se ha apoyado en muchas otras soluciones bien conocidas y consolidadas: eXtremeProgramming (XP³), Crystalmethodologies y RationalUnifiedProcess (RUP⁴). Los principios de programación extrema se han reutilizado en lo que se refiere a las prácticas de desarrollo, las metodologías Crystal proporcionan un input muy valioso en términos de la escalabilidad de los métodos y el RUP es la base para el diseño completo del ciclo de vida.

³ *eXtreme programming*, es una metodología ágil relacionada con el desarrollo de software

⁴ El **Proceso Unificado de Rational** o **RUP** es un proceso de desarrollo de software desarrollado por la empresa Rational Software

El ciclo de desarrollo de Mobile-D se divide en cinco fases: exploración, inicialización, productización, estabilización, y prueba del sistema.

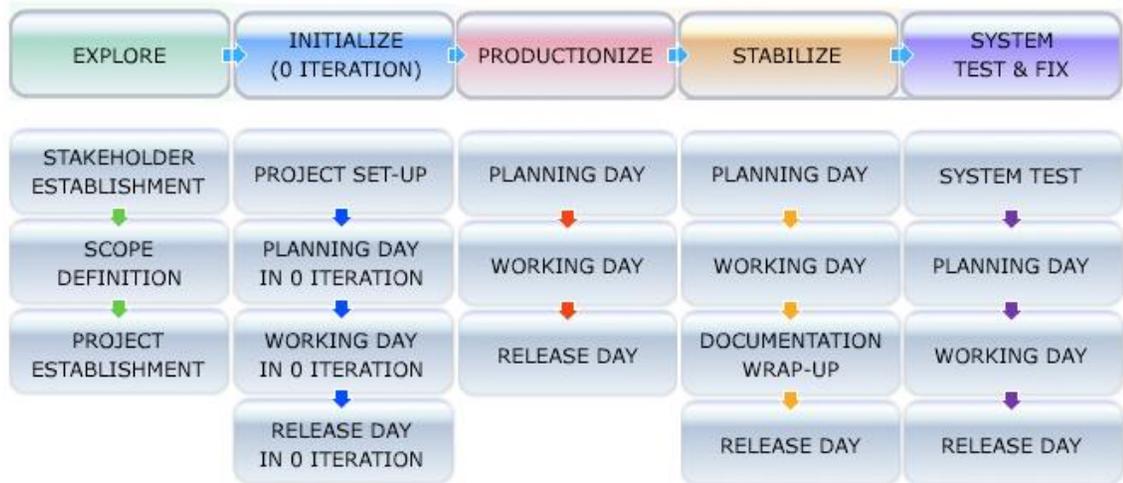


Figura 1. Descripción de Operacionalización de variables

1.8.3. Métricas de calidad de Software (Qsos)

El método QSOS⁵ describe un proceso formal para evaluar, comparar y seleccionar soluciones de código abierto. Este proceso consta de cuatro pasos interdependientes e iterativos:

⁵ Qualification end Selection of Open Source Software, Es uno de los modelos que permite la cuantificación y calificación de Software

✓ **Definición**

Constitución y enriquecimiento de los marcos de referencia que serán utilizados en los pasos siguientes.

✓ **Evaluación**

Evaluación del software hecho de acuerdo a 3 ejes de criterios: cobertura funcional, riesgos del usuario y riesgos del proveedor de servicios.

✓ **Calificación**

Carga de los criterios divididos en 3 ejes, modelando el contexto (requerimientos de usuario y/o estrategia escogida por el proveedor de servicios)

✓ **Selección**

Aplicación de filtro configurado en el paso anterior a los datos encontrados en los primeros dos pasos, de manera de realizar consultas, comparaciones y selección de productos.

1.8.4. MÉTRICAS DE COSTOS COCOMO II

Modelo diseñado por Barry W.Boehm para dar una estimación del número de meses que tomara desarrollar un producto software y tiene tres submodelos: Básico, Intermedio y detallado.

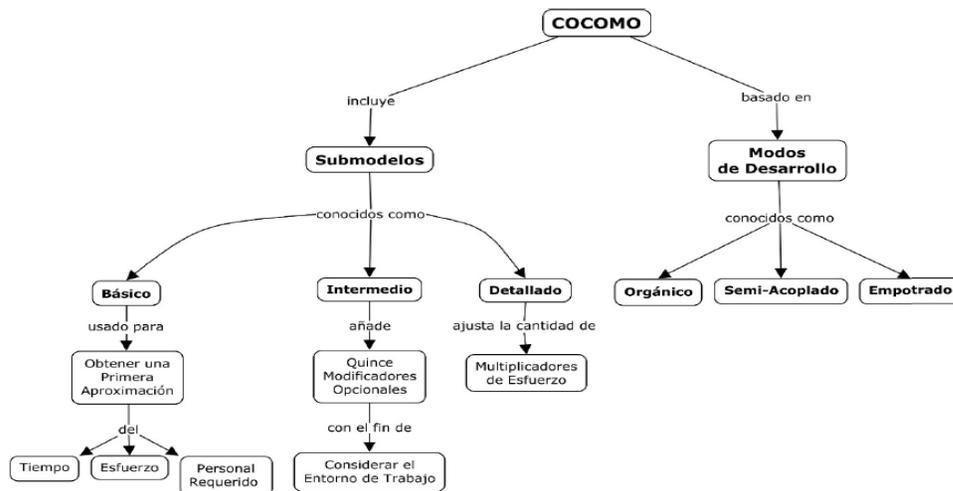


Figura 1. 2Conceptualización básica de COCOMO.

- ✓ **Modelo Básico** Estima el esfuerzo y el tiempo empleado en el desarrollo de un proyecto de software usando dos variables predictivas denominadas factores de costo (cost drivers): el tamaño del software y el modo de desarrollo.
- ✓ **Modelo Intermedio** Comparado con el modelo anterior, éste provee un nivel de detalle y precisión superior, por lo cual es más apropiado para la estimación de costos en etapas de mayor especificación. COCOMO Intermedio incorpora un conjunto de quince variables de predicción que toman en cuenta las variaciones de costos no consideradas por COCOMO Básico.
- ✓ **Modelo Detallado** Provee los medios para generar estimaciones con mayor grado de precisión y detalle. Difiere del Modelo Intermedio en dos aspectos principales que ayudan a superar las limitaciones.

1.8.5. Métricas de Calidad de Software (ISO-9126)

Métrica de Calidad para Evaluar Software ISO 9126. La norma ISO 9126 presenta dos partes, la primera es el modelo de calidad para tratar la calidad externa e interna, y la segunda es el modelo de calidad uso para tratar la calidad en uso. Para la evaluación de la calidad la ISO ha formulado entre otros los estándares ISO/IEC 9126, ISO/IEC 14598 e ISO/IEC 25000. El estándar ISO 9126 fue formulado inicialmente en 1991 estableciendo un modelo de calidad y su uso como marco para la evaluación de software. En esta norma se distingue entre calidad interna y calidad externa, y se introduce también el concepto de calidad en uso; esta norma es una de las normas ISO que goza de más reconocimiento dentro de la comunidad y tiene como fundamento modelos de calidad aportados por diversas investigaciones realizadas en los últimos 30 años para la caracterización de la calidad del producto software, que estará disponible en: (www.003).

1.9. HERRAMIENTAS

1.9.1. AndroidDevelopment Tool ADT

Herramientas de desarrollo de Android (ADT) es un plugin para el IDE⁶ de Eclipse que está diseñado para darle un potente entorno, integrado en el que la construcción de aplicaciones de Android.

ADT amplía las capacidades de Eclipse para que pueda configurar rápidamente nuevos proyectos de Android, crear una interfaz de usuario de la aplicación, agregar paquetes basados en la API de Android Framework, depurar sus aplicaciones utilizando las herramientas del SDK de Android, e incluso exportar firmado (o sin firmar) archivos .apk con el fin de distribuir su aplicación.

El desarrollo en Eclipse con ADT es muy recomendable y es la manera más rápida de comenzar. Con la configuración guiada proyecto que ofrece, así como la integración de herramientas, editores de XML personalizados, y el panel de salida de depuración, ADT le da un impulso increíble en el desarrollo de aplicaciones Android.

1.9.2. Php

PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

1.9.3. Sqlite

Es una herramienta que permite almacenar la información a la memoria interna de teléfonos móviles.

1.9.4. Eclipse

Eclipse es un entorno de desarrollo integrado, de Código abierto y Multiplataforma. Mayoritariamente se utiliza para desarrollar lo que se conoce como **Aplicaciones de Cliente Enriquecido, entorno de desarrollo integrado,**

⁶ Entorno Integrado de Desarrollo

opuesto a las aplicaciones **Cliente-liviano, entorno de desarrollo integrado** basadas en navegadores. Es una potente y completa plataforma de Programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java. No es más que un entorno de desarrollo integrado (IDE) en el que encontrarás todas las herramientas y funciones necesarias para tu trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar.

1.9.5. Xampp

Es un entorno de desarrollo web para Windows con el que podemos crear aplicaciones web con Apache PHP y base de datos en MYSQL, esta herramienta incluye además con un administrador de base de datos PhpMyAdmin con el cual podemos crear una nueva base de datos, que está disponible en:(www.006).

1.9.6. Base de Datos

- ✓ **Mysql:** Es un sistema de gestión de base de datos relacional o SGBD. Este gestor de base de datos en multihilo y multiusuario, lo que le permite ser utilizado por varias personas al mismo tiempo, e incluso, realizar varias consultas a la vez, lo que lo hace sumamente versátil.

Nació como una iniciativa de **Software Libre** y aún sigue ofreciéndose como tal, para usuarios particulares. Pero si se desea utilizarlo para promover datos en una empresa, se puede comprar una licencia, como un software propietario, que es autoría de la empresa Oracle Corporation, que estará disponible en: (www.007).

1.9.7. Css3

CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas, que está disponible en: (www.008).

1.9.8. Html

Es un lenguaje de programación que se utiliza para el desarrollo de páginas de internet. Se trata de la sigla que corresponde a de HyperText Markup Language, es decir, lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto, que estará disponible en: (www.009).

1.9.9. Google Maps

Google Maps es un tipo de servidor de aplicaciones que proporciona mapas en internet y que pertenece a Alphabet Inc. (la empresa matriz de Google) y que ofrece mapas geo localizados e información adicional de diferentes tipos que puedes encontrar en Google Maps como, por ejemplo: fotografías por satélite, información de tráfico e incluso la ruta entre diferentes ubicaciones en el mapa, que estará disponible en: (www.0010).

1.10. LIMITES Y ALCANCES

1.10.1. LIMITES

- ✓ Queda fuera de alcance en obtención de datos del Dirección Departamental de Transito de La Paz
- ✓ Se limitará en la presentación de dos sistemas de software: la aplicación móvil y un modelo web.
- ✓ Queda fuera del alcance de este trabajo obtener un coeficiente sobre la gravedad de las lesiones tal como puede hacerlo URGENCY. Este índice sirve principalmente porque en la actualidad, los autos están hechos para recibir todo el impacto y que no se lesionen los ocupantes, sin embargo, debido a esto los especialistas en emergencias podrían pensar que alguien no está herido cuando en realidad presenta heridas internas, entonces, de acuerdo a este índice, se los deriva a un centro de trauma o no.

- ✓ Queda fuera del alcance de la tesis también los estudios médicos asociados. Para este se hará uso de informes y estudios ya realizados por otros entes.
- ✓ Queda fuera del alcance de la tesis un análisis exhaustivo de los posibles métodos de detección de choques, en todo caso queda abierta la oportunidad para extender este trabajo y posibilitar que alguien con estudios físicos y/o matemáticos puedan realizar este trabajo.
- ✓ Queda fuera del alcance de la tesis la obtención de la posición dentro de túneles y demás lugares donde el GPS no pueda ser utilizado por falta de visualización directa a los satélites.

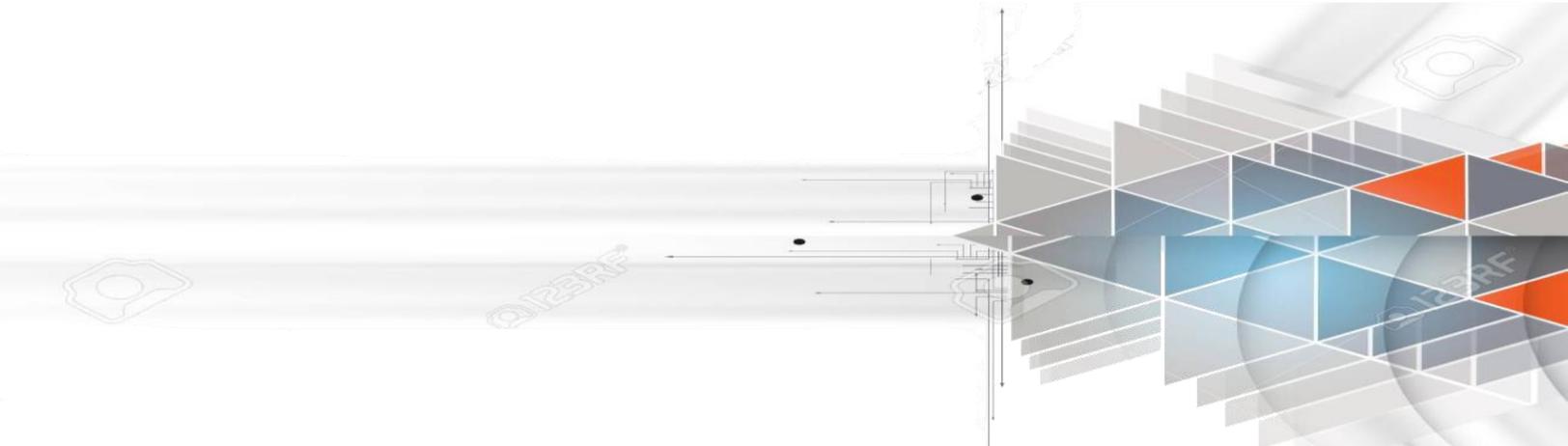
1.10.2. ALCANCES

- ✓ La central (web service) recibirá la información de todos los accidentes de tránsito y luego analizará y enviará la información a quien corresponda. Dependiendo de cuan informatizados estén los entes de emergencias en Bolivia.
- ✓ Se pretende con esta aplicación web y Android que todos los accidentes sean localizador de manera rápida y poder enviar ambulancia al lugar del hecho.
- ✓ La aplicación móvil tendrá un módulo donde se pueda verificar el lugar más cercano de un hospital.
- ✓ Poner en algún sistema de cartografía como por ejemplo Google Maps o el sistema de Microsoft la información de accidente de tránsito, etc., de manera que los demás movildades pudieran evitar el recorrido sobre el cual se encuentra el siniestro.
- ✓ Este tipo de software permitirá también que en el futuro se puedan generar reportes de los accidentes de tránsito para encontrar cuales son los puntos que pueden ser mejorados con objetos de contar con carreteras y en general, caminos más seguros.

- ✓ Será extensible de forma que, en principio, se informará al servicio de emergencias, pero luego, si algún ente de emergencia necesite recibir esa información, pueda entonces agregarse fácilmente.

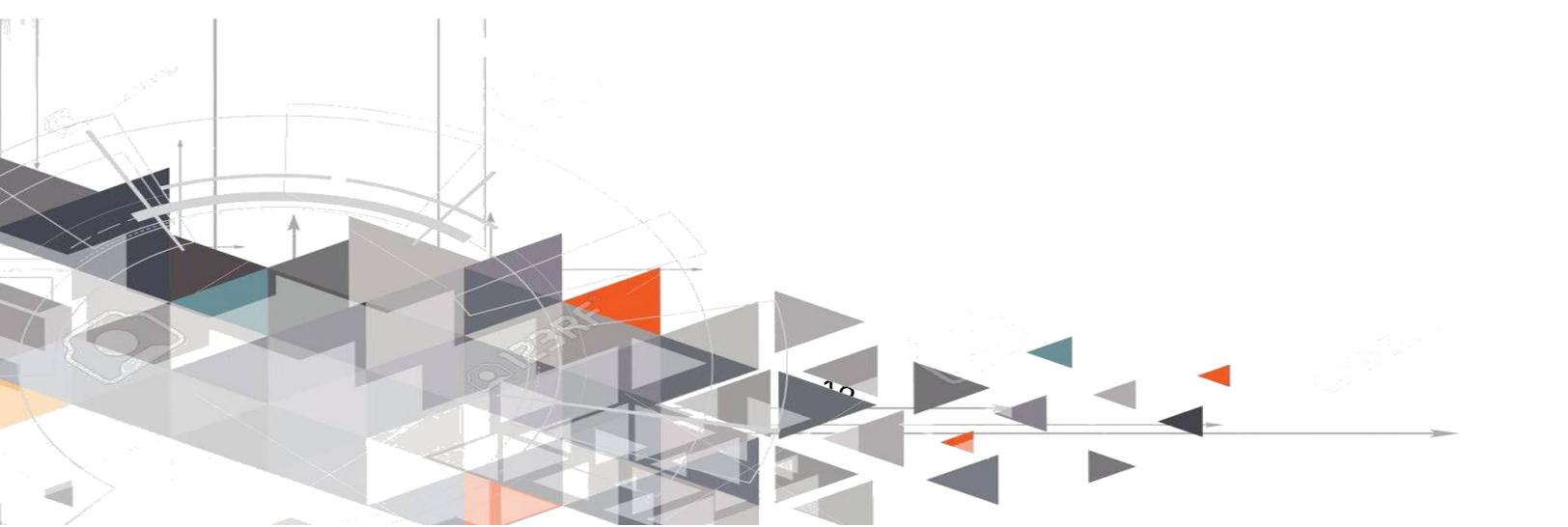
1.10.3. APORTES

Con el presente trabajo de modelo se brindará a los conductores y/o pasajeros una aplicación de detección de accidentes de tránsito para teléfonos inteligentes que cuentan con: la tecnología Android y sensores, para que este envíe un mensaje cuando tengan un accidente de tránsito a la central (web service). El web service enviara mensajes a los entes correspondientes, para que las víctimas sean atendidas a tiempo, dejando pasar la llamada "Golden Hour", aquella hora posterior al accidente donde se tienen las mayores posibilidades de ayudar a las víctimas.



CAPÍTULO II

MARCO TEÓRICO



2. MARCO TEÓRICO

2.1. INTRODUCCIÓN

En la mayoría de los accidentes que ocurre en Bolivia por diferentes razones, los heridos no son socorridos a tiempo. En este capítulo detallaremos los recursos que se utilizará para desarrollar el sistema propuesto en la presente modelo, para poder cumplir los objetivos ya mencionados en el anterior capítulo.

Primero identificaremos los tipos de accidentes que existen, luego se hará un pequeño resumen y sus herramientas que nos ofrece un celular, seguidamente se mencionará los modelos y protocolos que existen para el envío de mensajes desde un móvil a un Web Service, también se analizará la interoperabilidad del sistema. Finalmente, se verá el funcionamiento de la metodología Mobile-D.

2.2. DESCRIPCIÓN DE TIPOS DE ACCIDENTES DE TRANSITO

Existen varias formas de clasificar los accidentes y definir los distintos tipos que las clasificaciones contienen.

Las ordenaciones más socorridas y conocidas separan los diferentes tipos a partir del número de vehículos que intervienen en el accidente, de sus características, de su significación estadística, o por la gravedad que el caso reviste para las personas. Esta última no tiene una real importancia desde el punto de vista técnico porque la gravedad es solo una consecuencia, un efecto causa que lo origine.

En el siguiente apartado se detallará los diferentes tipos de accidente de tránsito que existen de acuerdo a su clasificación y en los cuales se podrá utilizar la aplicación que se propone en la presente tesis, y además mencionaremos otros tipos de accidentes de tránsito en los cuales no podrá intervenir la aplicación.

2.2.1. Clasificación según número de vehículos

Para este modus operandi el accidente debe clasificarse en razón de resultado final, es decir, del accidente realmente ocurrido. En este sentido se diferencian

de los accidentes simples, en que sólo interviene un vehículo, de los accidentes múltiples, en que interviene dos o más vehículos o peatones.

2.2.1.1. Son Accidentes Simples

a. Túnel: Es la vuelta de costado que se produce cuando el vehículo se apoya sobre las ruedas de un lado para girar en el sentido transversal al de marcha. Esto, también se conoce como volcamiento o volcadura transversal.

La posición final del tonel o volcadura transversal se indica en cuartos a la derecha o izquierda según sea el giro y se dice que ha quedado en 1/4 se queda sobre el costado inmediato a la posición normal de rodaje; 2/4 si ha quedado sobre el techo; 3/4 si es sobre el costado contrario al del inicio del giro; 4/4 si dada la vuelta completa, queda otra vez en la posición normal de rodaje. Sucesivamente se puede seguir indicando cuartos, Según sean las vueltas y posiciones.

b. Vuelta de Campana: Que es la volcadura en sentido longitudinal del vehículo, en que la posición final se manifiesta También de la manera descrita anteriormente.

c. Volteo: Es la precipitación a un plano inferior en que el vehículo, sin apoyo, gira en su sentido longitudinal cayendo sobre el techo

d. Choque: Es el embestimiento de un vehículo contra un obstáculo inmóvil de la vía cercano a ella, que puede ser incluso otro vehículo con la condición que no se encuentra en movimiento.

e. Accidentes Simples Combinados: Que es la producción sucesiva o simultánea de varios accidentes simples

2.2.1.2. Accidentes Múltiples

Los accidentes múltiples pueden subdividirse en dos grandes grupos; los que ocurren, entre vehículos y peatones, y las colisiones que suponen el embestimiento de un vehículo a otro, estando ambos en movimiento.

- a. **Volteo:** Este tipo de accidente se diferencia del atropello en que no existe una caída hacia delante del peatón, considerando el sentido de la dirección del móvil, sino que, por efecto de la velocidad, acciones evasivas u otras circunstancias, el peatón es levantado por el impacto cayendo sobre el capot, parabrisas, techo o al suelo por la parte de atrás del vehículo; también se distinguen fases en su producción.
- b. **Colisión:** Designase con tal expresión a los accidentes que se producen entre dos vehículos en movimiento cuando sus trayectorias se encuentran. En esta familia de accidentes la condición suficiente y necesaria es el movimiento en que deben encontrarse los vehículos.
- c. **Colisión Frontal:** Que es aquella en embestimiento o impacto se da y recibe con las partes frontales delanteras de los móviles, estando estos en movimiento. Ellas pueden ser centrales, cuando los ejes longitudinales de los vehículos coinciden, o excéntricas, cuando los ejes longitudinales no coinciden en una recta.
- d. **Alcance:** Que es aquella en que el embestimiento o impacto se da con la y recibe con las partes frontales delanteras contra la parte frontal posterior de otro y se produce cuando un vehículo que transita a mayor velocidad que otro que le precede, le da alcance el igual que la colisión frontal, puede ser central o excéntrico.
- e. **Colisión Lateral:** Que es aquella en que el embestimiento o impacto se da con las partes frontales de un vehículo contra el forro lateral de la carrocería o contra el chasis de otro. Las colisiones laterales pueden ser perpendiculares u oblicuas o diagonales, según sea la posición de los ejes longitudinales de los vehículos en el momento inmediatamente anterior al impacto.
- f. **Colisiones Mixtas:** En muchas ocasiones las diversas modalidades de colisiones se suceden denominándose a la serle de ellas, colisiones mixtas.

2.2.1.3. Accidentes Múltiples

Permite incluir como accidente de tránsito todo aquel que por sus características concuerde con la definición principal, haciendo abstracción del requisito de lugar.

Se adopta esta nomenclatura partiendo de la base que se debe definir cada accidente, en lo posible con una sola palabra, de manera de que no se creen confusiones y separe convenientemente los conceptos de choque, colisión, etc.

Los siguientes tipos de accidentes de tránsito no se tomará encuentra en el desarrollo del sistema propuesto en la presente tesis, los cuales son accidentes causados por:

(Despiste, salto, caída, raspado, atropello, impacto o embestimiento, acercamiento, comprensión o aplastamiento, arrastre, volteo propiamente tal, proyección, aplastamiento).

2.3. TELÉFONOS INTELIGENTES

Un **teléfono inteligente** (*Smartphone* en inglés) es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades semejantes a una minicomputadora y conectividad que un teléfono móvil convencional. El término inteligente hace referencia a la capacidad de usarse como un ordenador de bolsillo, llegando incluso a reemplazar a un ordenador personal en algunos casos. La característica más importante (una de ellas) de casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad. (Juan Félix Basterretche, 2007)



Figura 2. 1 Teléfonos Móviles ([www. mobiles.com](http://www.mobiles.com), 2002)

Entre las características mencionadas se destacan su excelente acceso y conectividad a Internet, su soporte de clientes de correo electrónico, la eficaz administración de nuestros datos y contactos, entre otras. (Arturo Baz Alonso, 2008)

En resumen su manejo suele ser sencillo, ya sea a través de la escritura sobre una pantalla (sensible electrónicamente) mediante un "stick", los teléfonos inteligentes tienen incorporados los sms, correo, y otras funcionalidades que son fáciles de manejar.

2.4. SISTEMA OPERATIVO ANDROID

Existen varios sistemas operativos para móviles entre ellos están: iPhone, Android⁷, Windows Mobile, Blackberry OS, Symbian, iOS.

Sin embargo, en el siguiente apartado se detallará el Sistema Operativo Android, ya que la aplicación propuesta en el presente modelo, se desarrollará sobre esta plataforma, su funcionalidad será desde la versión 2.3. (José Enrique Amaro Soriano - 2019)

Android es una plataforma de software y un sistema operativo para dispositivos móviles basada en un kernel ⁸Linux, desarrollada por Google y más tarde por la Open Handset Alliance. Esta plataforma permite a los desarrolladores escribir código en Java que se ejecute en móviles mediante las librerías Java desarrolladas por Google. (Ribas Lequerica, Joan - 2018)

También se pueden escribir aplicaciones en otros lenguajes, como por ejemplo C, para posteriormente ser compiladas en código nativo ARM y ejecutarlas, aunque este proceso de desarrollo no está soportado oficialmente por Google. La mayor parte de la plataforma de Android está disponible bajo licencia de software libre de Apache y otras licencias de código abierto.

⁷ **Android** es el nombre de un sistema operativo que se emplea en dispositivos móviles, por lo general con pantalla táctil

⁸ El kernel es software reemplazable de más bajo nivel que interactúa con tu ordenador.

2.4.2. Características De Android

- ✓ Amplia variedad de diseños (VGA, librerías de gráficos 2D y 3D)
- ✓ Almacenamiento de datos en BBDD SQLite
- ✓ Conectividad (GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth y Wi-Fi)
- ✓ Mensajería (SMS y MMS)
- ✓ Navegador Web
- ✓ Máquina virtual de Java
- ✓ Las aplicaciones escritas en Java pueden ser compiladas y ejecutadas en la máquina virtual de Dalvik, la cual es una especializada máquina virtual diseñada para uso en dispositivos móviles.
- ✓ Soporte de formatos (MPEG-4, H.264, MP3, AAC, OGG, AMR, JPEG, PNG, GIF)
- ✓ Soporte para hardware adicional (cámaras de video, pantallas táctiles, GPS, acelerómetros...)
- ✓ Entorno de desarrollo (emulador, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para Eclipse IDE).

2.4.3. Arquitectura De Android

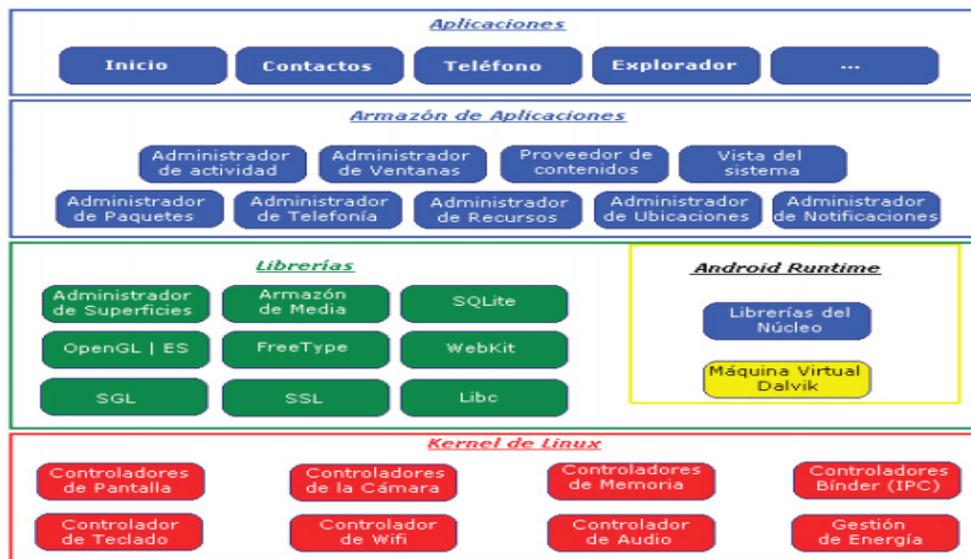


Figura 2. 2 Arquitectura de Android[www.android.com,2002]

A continuación, se explica los diferentes componentes de la arquitectura de Android:

- ✓ Aplicaciones: ejemplos de aplicaciones base son un cliente de email, programa de SMS, calendario, mapas, navegador, contactos, y otros. Todas las aplicaciones están escritas en el lenguaje de programación Java.
- ✓ Framework de aplicaciones: los desarrolladores tienen acceso completo a las APIs del framework usado por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquiera aplicación puede publicar sus capacidades y cualquiera otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario. Este framework está formado por un extenso conjunto de vistas tales como listas, cajas de texto, botones...
- ✓ Content Providers, permiten a las aplicaciones acceder a información de otras aplicaciones o compartir su propia información.
- ✓ Resource Manager, proporciona acceso a recursos que no son código como pueden ser gráficos, cadenas de texto...
- ✓ Notification Manager, permite a las aplicaciones mostrar alarmas personalizadas en la barra de estado.
- ✓ Activity Manager, gestiona el ciclo de vida de las aplicaciones.
- ✓ Librerías: Android incluye un set de librerías C/C++ usadas por varios componentes del sistema. Estas capacidades se exponen a los desarrolladores a través del framework ⁹de aplicaciones de Android, el cual interactúa con las librerías mediante JNI (Java Native Interface). Algunas son: System C Library (implementación librería C estándar), librerías de medios, librerías de gráficos, 3D, SQLite¹⁰, entre otras.

⁹ Un Framework es esencial para el desarrollo de diferentes programas

¹⁰ SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kiB)

- ✓ Runtime de Android: Android incluye un set de librerías base que proveen la mayor parte de las funcionalidades disponibles en las librerías base del lenguaje de programación Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima.
- ✓ Nucleo Linux: Android depende de un Linux versión 2.6 para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red, y modelo de drivers. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila.

Android SDK¹¹ es un kit de desarrollo de software (Software Development Kit o SDK) incluye un conjunto de herramientas de desarrollo, tales como un debugger, librerías, un emulador (basado en QEMU), documentación, código de tutoriales. Está soportado en S.O. Windows, Linux y Mac. El entorno de desarrollo (Integrated Development Environment o IDE) oficialmente soportado es Eclipse conjuntamente con el plugin ADT proporcionado por Google.

2.5. ESTRUCTURA DE UNA APLICACIÓN DE ANDROID

Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software. (Ribas Lequerica, Joan - 2017)

2.5.1. Activity

Las Activities (o Actividades) son el elemento constituyente de Android más común. Para implementarlas se utiliza una clase por cada Actividad que extiende de la clase base Activity. Cada clase mostrará una interfaz de usuario, compuesta por Views (o Vistas). (Ribas Lequerica, Joan - 2017)

¹¹ **SDK** es el acrónimo de Software Development Kit (Kit de desarrollo de software)

Cada vez que se cambie de Vista, se cambiara de Actividad, como por ejemplo en una aplicación de mensajería que se tiene una Vista que muestra la lista de contactos y otra Vista para escribir los mensajes. Cuando cambiamos de Vista, la anterior queda pausada y puesta dentro de una pila de historial para poder retornar en caso necesario. También se pueden eliminar las Vistas del historial en caso de que no se necesiten más. Para pasar de vista en vista, Android utiliza una clase especial llamada Intent. (Ribas Lequerica, Joan - 2017)

2.5.2. Intents

Un Intent ¹²es un objeto mensaje y que, en general, describe que quiere hacer una aplicación. Las dos partes más importantes de un Intent son la acción que se quiere realizar y la información necesaria que se proporciona para poder realizarla, la cual se expresa en formato URI. Un ejemplo serio ver la información de contacto de una persona, la cual mediante un Intent con la acción ver y la URI que representa a esa persona se podría obtener. (Ribas Lequerica, Joan - 2017)

Relacionado con los Intents hay una clase llamada IntentFilter que es una descripción de que Intents puede una actividad gestionar. Mediante los IntentFilters, el sistema puede resolver Intents, buscando cuales posee cada actividad y escogiendo aquel que mejor se ajuste a sus necesidades. El proceso de resolver Intents se realiza en tiempo real, lo cual ofrece dos beneficios: (Ribas Lequerica, Joan - 2017)

- ✓ Las actividades pueden reutilizar funcionalidades de otros componentes simplemente haciendo peticiones mediante un Intent.

Las actividades pueden ser remplazadas por nuevas actividades con IntentFilters equivalentes.

¹²Intent es un objeto mensaje y que, en general, describe que quiere hacer una aplicación

2.5.3. Listener

Los Listeners se utilizan para reaccionar a eventos externos (por ejemplo, una llamada). Los Listeners no tienen UI pero pueden utilizar el servicio Notification Manager para avisar al usuario. (Ribas Lequerica, Joan - 2017)

Para lanzar un aviso no hace falta que la aplicación se esté ejecutando, en caso necesario, Android la iniciará si se activa el Listener por algún evento.

2.5.4. Servicios

Un Servicio es básicamente un código que se ejecuta durante largo tiempo y sin necesidad de UI¹³, como puede ser un gestor de descarga en el cual se indican los contenidos a descargar y posteriormente el usuario puede acceder a una nueva Vista sin que el gestor se interrumpa. (Ribas Lequerica, Joan - 2017)

En caso de que haya múltiples servicios a la vez, se les puede indicar diferentes prioridades según las necesidades.

2.5.5. Content Provider

En Android, las aplicaciones pueden guardar su información en ficheros, BBDD SQLite. Pero en caso de que lo que se quiera sea compartir dicha información con otras aplicaciones, lo necesario es un Content Provider. Un Content Provider es una clase que implementa un conjunto estándar de métodos que permite a otras aplicaciones guardar y obtener la información que maneja dicho Content Provider. (Ribas Lequerica, Joan - 2017)

2.5.6. Android Manifest

En Android existe un archivo XML llamado AndroidManifest que, aunque no forme parte del código principal de la aplicación, es necesario para su correcto funcionamiento. Este archivo es el fichero de control que le dice al sistema que tiene que hacer con todos los componentes anteriormente mencionados en este apartado que pertenecen a una aplicación en concreto. (Ribas Lequerica, Joan - 2017)

¹³ UI se refiere a la creación de la interfaz, ya sea gráfica o desarrollada con diversas tecnologías web como CSS, jQuery, jQuery UI, EXT JS, YUI, etc.

2.6. VERSIONES DEL SISTEMA OPERATIVO ANDROID

El sistema operativo Android, al igual que los propios teléfonos móviles, ha evolucionado rápidamente, acumulando una gran cantidad de versiones, desde la 1.0 para el QWERTY HTC G1, hasta la 5.0 que acaba de salir al mercado. (Jorge Santiago Nolasco - 2013)

Cupcake: Android Versión 1.5 Características: Widgets, teclado QWERTY virtual, copy & paste, captura de videos y poder subirlos a Youtube directamente.

Donut: Android Versión 1.6 Características: Añade a la anterior la mejoría de la interfaz de la cámara, búsqueda por voz, y navegación en Google Maps.

Eclair: Android Versión 2.0/2.1 Características: Mejoras en Google Maps, salvapantallas animado, incluye zoom digital para la cámara, y un nuevo navegador de internet.

Froyo: Android Versión 2.2 Características: Incluye hotspot Wi-Fi, mejora de la memoria, más veloz, Microsoft Exchange y video-llamada.

GingerBread: Android Versión 2.3 Características: Mejoras del consumo de batería, el soporte de video online y el teclado virtual, e incluye soporte para pagos mediante NFC. **Honey Comb: Android Versión 3.0/3.4**

Ice Cream Sandwich: Android Versión 4.0 Características: Multiplataforma (tablets, teléfonos móviles y netbooks), barras de estado, pantalla principal con soporte para 3D, widgets redimensionables, soporte usb para teclados, reconocimiento facial y controles para PS3.

Jelly Bean: Android Versión 4.1 Características: Multiplataforma (tablets, teléfonos móviles y netbooks), barras de estado, pantalla principal con soporte para 3D, widgets redimensionables, soporte USB para teclados, reconocimiento facial y controles para PS3.

Android 4.4 KitKat Android KitKat es una de las versiones de Android más emblemáticas, y no solo por lo pegadizo de su nombre (la primera vez que Google

se aliaba con otra empresa para su mascota de Android). Hasta hace no mucho, seguía siendo una de las versiones más usadas, e incluso hoy en día, cuatro años más tarde, mantiene una honorable cuota del 13,4% en la distribución de versiones.

Android 5.0 Lollipop El diseño de Google Now se expande, documenta y aplica a todo Android: llega **Material Design**, un soplo de aire fresco que ya se iba haciendo necesario tras seis años improvisando sin unas reglas claras. Material Design llegaba en varias aplicaciones de Google, pero ahora la pelota estaba en el tejado de los desarrolladores para que adaptaran sus aplicaciones a este nuevo diseño

Android 6.0 Marshmallow Con Marshmallow llega **Direct Share**, la forma más rápida de enviar contenido a un contacto específico y **Now On Tap**, ese botón mágico que busca qué hay en tu pantalla para ofrecerte información relacionada

Android 7.0 Nougat En cuanto al rendimiento, Nougat **mejora el Doze** de Marshmallow, haciéndolo efectivo incluso cuando el teléfono está en movimiento. Además, el **nuevo compilador JIT** reduce en un 75% la instalación de una aplicación y requiere de menos almacenamiento

Android 8.0 Oreo Project Treble es la estrella de Android Oreo. Una nueva arquitectura modular del sistema para facilitar el proceso de actualizar un terminal y, teóricamente, lograr que lleven menos trabajo y, por tanto, te lleguen antes. Eso sí, probablemente no veremos su impacto hasta dentro de unos años

Android 9.0 Pie El **bienestar digital** es otra de las grandes novedades, una serie de herramientas con las que puedes controlar el uso que haces del móvil, aunque su uso aún no se ha extendido a todos los móviles con Pie. Por último, llegaba **la navegación por gestos** de forma oficial, la cual cambiaba para siempre la barra de navegación, simplificándola con un único botón en la mayoría de las ocasiones

En la siguiente tabla se muestra en resumen de todas las versiones:

Tabla 2. 1*Lista de versiones de Android con su respectivo API*

Nombre código	Número de versión	Nivel de API
Apple Pie ¹	1.0	1
Banana Bread ¹	1.1	2
Cupcake	1.5	3
Donut	1.6	4
Eclair	2.0 – 2.1	5 – 7
Froyo	2.2 – 2.2.3	8
Gingerbread	2.3 – 2.3.7	9 – 10
Honeycomb ²	3.0 – 3.2.6	11 – 13
Ice Cream Sandwich	4.0 – 4.0.5	14 – 15
Jelly Bean	4.1 – 4.3.1	16 – 18
KitKat	4.4 – 4.4.4	19 – 20
Lollipop	5.0 – 5.1.1	21 – 22
Marshmallow	6.0 – 6.0.1	23
Nougat	7.0 – 7.1.2	24 – 25
Oreo	8.0 – 8.1	26 – 27
Pie	9.0	28
Android 10 ³	10.0	29
Android 11 ⁴	11.0	30

Nota: groupandroid.com, 2020

2.7. SENSORES DE UN TELEFONO

Actualmente, los teléfonos inteligentes o teléfonos móviles, suelen venir con una cantidad de sensores que va aumentando con cada nuevo modelo o versión. Ya es común encontrarse con acelerómetros en la mayoría de los celulares, y poco a poco, están comenzando a encontrarse giróscopos (para medir la velocidad angular), sensores de proximidad (para bloquear la pantalla al hablar por teléfono, de manera de, por ejemplo, no se presionen botones con la oreja sin querer), brújulas electrónicas, etc. Android, a partir de la versión 2.3 (Gingerbread) fue un paso más allá e incluyó dos nuevos sensores de fusión, estos permiten obtener la aceleración lineal sobre los 3 ejes y una matriz de rotación. Usando todos los sensores de un celular con estas características, se podría llegar a conocer

exactamente cómo se está moviendo un celular o, mejor aún, saber cómo se movió en el pasado. Una característica de este tipo, en un choque, podría significar (Ing. Juana Vallegos, 2009)

Cuáles son los sensores que existen en los teléfonos:

- ✓ **Acelerómetro:** Devuelve la aceleración en los 3 ejes, menos la gravedad en ese eje. Los valores son devueltos en la unidad (m/s²). Si el celular se encuentra en caída libre, los valores devueltos serán 0 para los 3 ejes.
- ✓ **Campo Magnético:** Todos los valores son devueltos en la unidad micro-Tesla y se mide el campo magnético en el ambiente en los 3 ejes.
- ✓ **Giróscopo:** Devuelve la velocidad angular en cada uno de los 3 ejes. Todos los valores devueltos están en la unidad radianes/segundo.
- ✓ **Sensor de Luz:** Devuelve el nivel de luz en unidades Lux SI
- ✓ **Presión:** Devuelve la presión atmosférica en hPa (milibar)
- ✓ **Proximidad:** Devuelve la distancia medida en centímetros, aunque algunos sensores sólo soportan un valor binario como salida, un valor alto al estar lejos y un valor bajo al estar cerca (un objeto del sensor).
- ✓ **Gravedad:** Devuelve la dirección y magnitud de la gravedad. Todos los valores devueltos están en la unidad m/s². Estando en reposo, los valores deberían ser los mismos que el acelerómetro.
- ✓ **Aceleración Lineal:** Describe la aceleración lineal aplicada sobre el celular sin incluir la gravedad. Si se ve el código fuente de Android, también se encuentra que aplica un filtro paso bajo. Esto es para eliminar valores pequeños no deseados que no son reales, sino que son producto de que el sensor está hecho electrónicamente. Todos los valores devueltos están en la unidad m/s². La salida de los sensores Acelerómetro, Gravedad y Aceleración Lineal deben obedecer la ecuación: $\text{aceleración} = \text{gravedad} + \text{aceleración lineal}$
- ✓ **Vector de Rotación:** Representa la orientación del dispositivo como una combinación de ángulo y eje, en el cual el celular ha girado a través de un

ángulo θ alrededor de un eje $\langle x, y, z \rangle$. Los tres elementos del vector de rotación son $\langle x \cdot \sin(\theta/2), y \cdot \sin(\theta/2), z \cdot \sin(\theta/2) \rangle$, tal que la magnitud del vector de rotación es igual al $\sin(\theta/2)$, y la dirección del vector de rotación es igual a la dirección del eje de rotación. Los tres elementos del eje de rotación son iguales a las tres últimas componentes de una unidad cuaternión (quaternion en inglés) $\langle \cos(\theta/2), x \cdot \sin(\theta/2), y \cdot \sin(\theta/2), z \cdot \sin(\theta/2) \rangle$. Los elementos del vector de rotación no tienen unidades. (Ing. Juana Vallegos, 2009)

2.8. SISTEMA DE CORDENADAS DE ANDROID

El sistema de coordenadas en Android está definido relativo a la pantalla del celular en su orientación más común. Los ejes no son cambiados cuando se cambia la orientación de la pantalla del celular.

El eje X es horizontal y apunta a la derecha, el eje Y es vertical y apunta arriba y el eje Z apunta hacia el exterior de la cara frontal de la pantalla. En este sistema, las coordenadas detrás de la pantalla tienen valores de Z negativos.

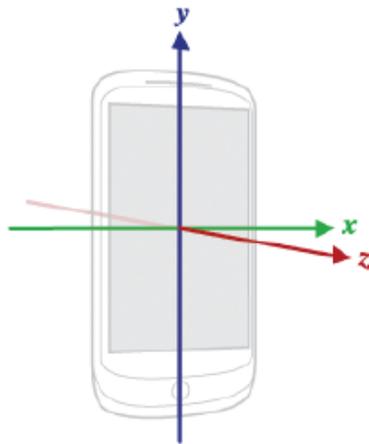


Figura 2. 3 Sistema de coordenadas en Android.[groupandroid, 2011]

2.9. MENSAJES

Existen varios tipos de mensajes para el intercambio de datos de accidentes de tránsito entre los cuales están: Conjunto Mínimo de Datos (MSD), y el Conjunto de Varios Datos (FSDE). Estos mensajes se deben estandarizar a datos XML para poder ser enviados en nuestro caso al web service. En el siguiente apartado detallaremos. (Michael Nielsen, 2006)

La interoperabilidad del sistema es muy importante, ya que existen muchos celulares distintos, y el despliegue del sistema completo debe tener la menor complejidad posible, así como la mayor posibilidad de uso de la infraestructura existente. La interoperabilidad es más que comunicación por voz, tanto las organizaciones como los sistemas no pueden compartir fácilmente información de incidentes si no hay interoperabilidad clara entre todas las partes involucradas. (Michael Nielsen, 2006)

Para la interacción entre los distintos entes, es vital tener un idioma común entre las distintas aplicaciones, y de esta forma poder crear reportes históricos con información distribuida entre los distintos sistemas. Esto hace necesaria la creación de protocolos y estándares comunes a todas las aplicaciones.

Aquí se muestra una propuesta de mensajes a ser enviados desde el celular al servidor, de forma que ayuden a disminuir y/o prevenir los heridos en accidentes viales, obtener estadísticas, etc. Luego se hace un resumen de los protocolos y estándares existentes en distintas partes del mundo en relación a incidentes de tránsito y alertas en general.

✓ **Model Minimum Uniform Crash Criteria (MMUCC)**

- Reporte que deben llenar los oficiales en la escena del choque para su posterior análisis.
- Se actualiza cada 5 años.

✓ **Vehicular Emergency Data Set (VEDS)**

- Creado por el ComCARE Alliance ACN Data Set Working Group (911, EMS, OnStar/ATX).

- Conjunto de datos que especifican información del accidente para poder ser enviada de forma electrónica.
- ✓ **Emergency Data Exchange Language (EDXL)**
 - Iniciativa del DHS Disaster Management E-Gov.
 - Promueve el compartir la información de incidentes (interoperabilidad de datos) para todos los peligros a través de la comunidad de respuestas a emergencias.
 - EDXL – HAVE (Hospital Availability Exchange o Intercambio de Disponibilidad de Hospitales).
 - EDXL – SitRep (Situation Reporting o Reporte de Situación).
 - EDXL – TEP (Tracking of Emergency Patients o Seguimiento de Pacientes de Emergencias)

2.10. MODELO CON UN MÍNIMO DE DATOS UNIFORMES DEL ACCIDENTES

Modelo con un Mínimo de Datos Uniformes del Accidente MMUCC “*Model Minimum Uniform Crash Criteria – MMUCC*” es una guía que presenta un modelo con un mínimo conjunto de variables o elementos de datos uniformes para la descripción de un accidente de tráfico de un vehículo motorizado. El propósito del MMUCC es proporcionar un conjunto de datos para describir los accidentes de vehículos motorizados en una carretera que van a generar la información. (Jose Luis Pedradosa, 2009)

MMUCC recomienda la aplicación **voluntaria** de un “conjunto mínimo” de elementos de datos normalizados para promover la comparación de los datos dentro de la comunidad de la seguridad vial. **Sirve como base para los sistemas de datos de accidentes a nivel estatal.**

2.10.1. Organización de los Elementos de Datos MMUCC

Cada elemento de dato MMUCC incluye:

- ✓ Una definición
- ✓ Un conjunto de atributos específico y
- ✓ Una razón por la cuál es necesario.

Los elementos de datos se dividen en cuatro grupos principales que describen distintos aspectos de un accidente:

- ✓ Relacionados con los choques,
- ✓ Relacionados con los vehículos,
- ✓ Relacionados con las personas y
- ✓ Relacionados con las carreteras.

No especifica el formato digital para el intercambio de los datos, no dice que se hará por XML, podría ser un CSV por ejemplo o un JSON. Se suele usar XML porque es más estándar y sería más fácil para los distintos equipos ya existentes poder comunicarse con él.

Para el intercambio de datos con Frontend y Backend para el desarrollo del sistema propuesto en la presente tesis se utilizará MMUCC, sin embargo, cabe mencionar que existen otros modelos como:

- a. El Conjunto de Datos para la Emergencia Vehicular “*Vehicular Emergency Data Set – VEDS*” es un estándar basado en XML¹⁴ para reportar datos útiles y críticos, tanto médicos como de las colisiones en sí. El estándar fue desarrollado por ComCARE Alliance, y está destinado a la transmisión de información crítica para facilitar la respuesta a la emergencia de forma eficaz. (comCare Alliance, 2014)

Este conjunto de datos puede ser transmitido automáticamente a un centro de respuesta, el que puede luego transferirlo a un proveedor de servicios de emergencia. (VEDS - Vehicular Emergency Data Set, 2004)

Es sólo un formato de intercambio de datos recomendado. No es un estándar/protocolo de **transmisión de datos**.

- b. El Protocolo Comun para la Alerta - CAP es un formato de datos basado en XML para intercambiar alertas y emergencias públicas entre distintas

¹⁴ **XML** es una adaptación del SGML (Standard Generalized Markup Language), un lenguaje que permite la organización y el etiquetado de documentos.

tecnologías de alertas. CAP permite que un mensaje de alerta sea consistentemente diseminado simultáneamente entre muchos sistemas de alertas y muchas aplicaciones. (open aosis, 2010)

CAP es un formato simple pero general para intercambiar todo tipo de alertas de emergencias de amenazas y alertas públicas sobre todo tipo de redes.

A diferencia de MMUCC, CAP es un protocolo que usa XML y está destinado a **todo tipo de emergencias**, no sólo vehiculares

2.11. LENGUAJE PARA EL INTERCAMBIO DE DATOS DE EMERGENCIA EDXL

2.11.1. Descripción

[El *Emergency Data Exchange Language* (EDXL) o Lenguaje para el intercambio de Datos de Emergencias es una familia de estándares XML relacionados el intercambio de mensajes e información en relación a accidentes, emergencias y desastres. (IEEE Incident Management Working Group, 2000)

Los estándares EDXL se pueden clasificar en tres categorías:

1. Un estándar que especifica el enrutamiento de los datos para la información de emergencia, incluyendo:
 - a. **EDXL Distribution Element (EDXL-DE).**
2. Un estándar que especifica el contenido, que incluye:
 - a. **EDXL Resource Messaging (EDXL-RM)**
 - b. **EDXL Hospital AVailability Exchange (EDXL-HAVE)**
 - c. **EDXL Situation Reporting (EDXL-SitRep)**
 - d. **EDXL Tracking of Emergency Patients/Victims (EDXL-TEP/TEV)**
3. Un estándar que incluyen ambos enrutamiento y contenido, incluyendo:
 - a. **EDXL Common Alerting Protocol (EDXL-CAP).**

La especificación del Elemento de Distribución describe un framework para la distribución de un mensaje estándar para compartir datos entre sistemas de información de emergencias usando EDXL (Emergency Data Exchange Language) basado en XML. Este formato puede ser usado sobre cualquier sistema de transmisión de datos, incluyendo, pero no limitándose al binding SOAP HTTP. (IEEE Incident Management Working Group, 2000)

El objetivo principal del Elemento de Distribución es facilitar el enrutamiento de cualquier mensaje de emergencia XML con un formato correcto a los destinatarios.

El Elemento de Distribución puede ser pensado como un “contenedor”. Provee la información para enviar conjuntos de mensajes particulares (como alertas o Mensajes de Recursos), incluyendo información clave de enrutamiento como: (IEEE Incident Management Working Group, 2000)

- ✓ Tipo de Distribución,
- ✓ Geografía,
- ✓ Incidente, y

Identificaciones del emisor/destinatario

2.11.2. Estructura del Elemento de Distribución EDXL

El Elemento de Distribución de EDXL (DE por sus siglas en inglés) comprende un element <EDXLDistribution> como está descrito luego, elementos opcionales <targetArea> describiendo un área objetivo geoespacial o política para la entrega del mensaje, y un conjunto de elementos <contentObject> cada uno conteniendo información específica en relación a un ítem de contenido particular. El contenido incluido puede ser cualquier XML u otro tipo de contenido o una URI para acceder al contenido. El bloque <EDXLDistribution> puede ser usado sin contenido para formar el cuerpo de una query de enrutamiento para, o en respuesta de, un servicio de directorio. (OASIS, 2006)

<EDXLDistribution>

El elemento <EDXLDistribution> sostiene la intención del autor en cuanto a la difusión del mensaje o del conjunto de mensajes.

El uso del elemento <EDXLDistribution> no garantiza que todos los enlaces y nodos de las redes van a implementar la política de difusión afirmada o que no ocurrirá una divulgación involuntaria. Donde hay información sensible que se está transmitiendo por redes no seguras, se debe usar encriptación en concordancia con el estándar Web Services (OASIS Web Services Security (WSS) TC http://www.oasisopen.org/committees/tc_home.php?wg_abbrev=wss)

Security (WSS) con cualquier actualización y fé de erratas publicada por el OASIS Web Services Security Technical Committee, o algún otro esquema de encriptación que sirva.

<targetArea> El <targetArea> es un elemento contenedor de una zona geoespacial o política, objetivo del destinatario del contenido del mensaje. Contiene datos necesarios de la intención del autor, basados tanto en la ubicación del objetivo, como en la difusión de ese mensaje particular o conjunto de mensajes.

<contentObject> El <contentObject> es un elemento contenedor para mensajes específicos. El elemento <contentObject> DEBE o bien contener un contenedor de contenido <xmlContent> o bien un contenedor de contenido <nonXMLContent>.

2.11.3. Requerimientos para el diseño

La especificación del Elemento de Distribución debe: (OASIS - 2004)

1. Definir una sola estructura XML (a una estructura equivalente si es transcodificada en algún otro formato) incluyendo los elementos requeridos y opcionales definidos debajo.
2. Especificar un área geográfica de entrega, expresada en coordenadas geoespaciales o códigos políticos/administrativos.
3. Permitir la habilidad de encapsular el payload (payload serían headers o metadata por ejemplo) o conjunto de payloads.

4. Adoptar un enfoque modular para la enumeración de los valores de los elementos que puede evolucionar con el tiempo, por ejemplo, haciendo referencia a un esquema diferente para esas enumeraciones.
5. Especificar identificadores únicos de emisor y distribución.
6. Especificar la fecha y hora cuando la distribución fue realizada.
7. Especificar la acción que puede tener el mensaje de distribución (por ejemplo, mundo real, prueba, ejercicio).
8. Especificar el tipo funcional del mensaje de distribución (por ejemplo reporte, request, actualización, cancelación, etc.)
9. Especificar que los siguientes elementos pueden estar presentes en un payload válido:
10. Una especificación del formato del mensaje de distribución (por ejemplo la URI de un esquema XML para el mensaje).
 - a. El rol funcional y/o el tipo de emisor del mensaje de la distribución
 - b. Uno o más roles funcionales y/o el tipo de destinatarios deseados del mensaje de distribución
 - c. Una referencia a uno o más mensajes de distribución previos
 - d. Uno o más tipos de respuesta de las actividades involucradas
 - e. Una referencia al tipo de incidente
 - f. Una o más caracterizaciones de la etiología del sujeto del evento o incidente (por ejemplo terrorismo, natural, bajo investigación, etc.)
 - g. El nombre u otra identificación del incidente de uno o más eventos o incidentes
 - h. Una referencia a uno o más tipos de respuesta
 - i. Una o más direcciones de destinatarios específicos (como una URI)
 - j. Especifica una afirmación del nivel de confidencialidad de los payloads combinados.
11. Además, el elemento Content Object contenido en el Elemento de Distribución (DE) DEBE:

- a. Permitir la encapsulación de uno o más payloads en cada uno de los elementos Content Object.
- b. Especificar el rol funcional y/o el tipo de emisor de cada payload
- c. Especificar uno o más roles funcionales y/o los tipos de destinatarios deseados de cada payload
- d. Especificar una medida del nivel de confidencialidad de cada payload.

12. Proporcionar o hacer referencia a las listas específicas (enumeraciones) de los valores y sus definiciones para:

- a. Tipos de incidentes
- b. Tipos de riesgos y/o eventos
- c. Tipos de agencias
- d. Tipos de actividades de respuesta
- e. El rol funcional y/o tipo de emisor
- f. Los roles funcionales y/o tipos de destinatarios deseados

El nombre u otra identificación del incidente de uno o más eventos o incidentes.

2.12. PROTOCOLO PARA ENVIÓ DE MENSAJES BACKEND FRONDEND

Existen protocolos de emergencia ya establecidos para el envío de mensajes a Backend Frontend que son los siguientes: Conjunto de Datos de Urgencia (UDS), y Datos Extendidos para la Emergencia (EDS). (ComCARE Alliance ACN – 2004)

El protocolo que emplearemos para el desarrollo del sistema propuesto será UDS, ya que este protocolo contiene solamente datos que son para emergencias de tipo accidentes de tránsito. Mientras EDS contiene datos para todo tipo de emergencia como: sismos, incendios, y otros.

2.12.1. CONJUNTO DE DATOS DE URGENCIA

Conjunto de Datos de Urgencia “*Urgency Data Set - UDS*” es un protocolo para el envío de mensajes que utiliza solamente la información necesaria para él envió de alertas de accidentes de tránsito, este protocolo es corto y preciso en la

recopilación de los datos que serán enviados a diferentes centros de emergencia, según este protocolo los datos son los siguientes: (ComCARE Alliance ACN – 2004)

Tabla 2. 2
Conjunto de datos de urgencia

Nombre	Tipo de datos	Descripción
CrashDate	datetime	Fecha UTC de cuando ocurrió el accidente. Viene desde el dispositivo móvil.
UdsVersion	varchar	Versión del mensaje EMS. Necesario por si en el futuro cambia o una máquina debe leerlo en el medio del transporte.
CrashId	bigint	Foreign key to the Crash record.
Speed	Int	En km/h
Bearing	Float	En grados * 255 / 360 (redondeado al entero más cercano).
Latitude	Float	Latitud WGS84 en grados (decimales) *2^16 signado -90 90)
Longitude	Float	Longitud WGS84 en grados (decimales) *2^15 (signado -180 180)
Accuracy	Float	Precisión obtenida en la ubicación del choque
Rollover	Char	True si se detectó un vuelco
AutomaticTrigger	Char	True si la activación no fue manual
AirbagDeployed	Char	True si se detectó que el airbag fue activado
CountryId	Int	Id del país desde donde fue enviado el mensaje UDS.
MobileNumber	nchar	Número de teléfono desde el cual se envió el mensaje UDS.
SubscriberId	varchar	El id único de suscripción, por ejemplo, el IMSI para un dispositivo móvil GSM.
RetriesNumber	Int	En cuántos reintentos se puede enviar. 0 sería que se envió al primer intento.

Notas: [eCall.com, 1999]

2.13. DECIBEL

El decibel, es la unidad de medida relativa empleada en acústica (sonido), electricidad, telecomunicaciones y otras especialidades, su símbolo es **dB**.

El decibel es una unidad logarítmica, adimensional y matemáticamente escalar. Es la décima parte de un **belio** (símbolo **B**), que es el logaritmo de la relación entre la magnitud estudiada y la de referencia, pero no se utiliza por ser

demasiado grande en la práctica, y por eso se utiliza el decibel. El belio recibió este nombre en honor de Alexander Graham Bell. (Federico Miyara, 2002)

Un belio equivale a 10 decibeles y representa un aumento de potencia de 10 veces sobre la magnitud de referencia.

Tabla 2. 3

Nivel de intensidad del sonido

200 Db	Bomba atómica similar a Hiroshima y Nagasaki. de distancia). Cohete en despegue.
180 dB	Explosión del Volcán Krakatoa (a 160 km
140 dB	Umbral del dolor. Auto de Fórmula 1.
137.5 dB	Record Guinness de ruido en un estadio.
130 dB	Avión en despegue.
120 dB	Motor de avión en marcha.
110 dB	Concierto / acto cívico.
100 dB	Perforadora eléctrica.
90 dB	Trafico / Pelea de dos personas.
80 dB	Tren.
70 dB	Aspiradora.
50/60 dB	Aglomeración de gente / Lavaplatos.
40 dB	Conversación.
20 dB	Biblioteca.
10 dB	Respiración tranquila.
0 dB	Umbral de audición.

Nota: [Ing. Francisco Rufa, 2003]

2.14. MEDICIÓN DE LA ACÚSTICA

Para medir el sonido se utiliza una escala logarítmica porque la sensibilidad que presenta el oído humano a las variaciones de intensidad sonora sigue una escala aproximadamente logarítmica, no lineal. Por ello el belio **B** y su submúltiplo el decibel **dB**, resultan adecuados para valorar la percepción de los sonidos por un oyente. (Federico Miyara, 2002)

Como el decibel es una unidad relativa, para las mediciones acústicas se asigna el valor de 0 dB al umbral de audición del ser humano, que por convención se

estima que equivale a un sonido común a presión de 20 micropascales, algo así como un cambio de la presión atmosférica normal de 1/5 000 000 000. (Federico Miyara, 2002)

Para el cálculo de la sensación recibida por un oyente, a partir de las unidades físicas medibles de una fuente sonora, se define el nivel de potencia, **L_w**, en decibeles, y para ello se relaciona la potencia de la fuente del sonido a estudiar con la potencia de otra fuente cuyo sonido este en el umbral de audición, por la formula siguiente: (Federico Miyara, 2002)

$$L_w = 10 * \text{Log}_{10} \frac{W_1}{W_0} (dB) = 10 * \text{Log}_{10} \frac{W_1}{10^{-12}} (dB)$$

Fórmula 1. La intensidad del sonido.

En donde **W₁** es la potencia a estudiar, en vatios (variable), **W₀** es el valor de referencia, igual a **10-12 vatios/m²** y **Log₁₀** es el logaritmo en base 10 de la relación entre estas dos potencias. Este valor de referencia se aproxima al umbral de audición en el aire. Si **W₁** es mayor que la potencia de referencia **W₀** el valor en decibeles es positivo. Y si **W₁** es menor que la referencia **W₀** el resultado es negativo. Un aumento en un factor 10 (10 veces) en la potencia **W₁** con respecto a la referencia significa un aumento de 10 unidades (10 dB) aditivas en la escala logarítmica (intensidad subjetiva). Y que al aumentar al doble (factor 2) la potencia **W₁** con respecto a **W₀** significa un aumento aditivo de 3 dB en la escala logarítmica (**Log₁₀2 = 0,301 B = 3,01 dB**). (Federico Miyara, 2002)

El análisis sobre la medición del sonido detallado anteriormente, permitirá obtener la intensidad de sonido con lo que ocurrió el choque (accidente automovilístico), en la aplicación móvil propuesta en la presente tesis.

2.15. GEOLOCALIZACIÓN

También denominada georreferenciación, es la determinación de la ubicación geográfica por medio de dispositivo móvil o computadora. Para determinar la ubicación mediante Smartphone, existe tres maneras para hacerlo, los cuales

son mediante: GPS, red de datos o WiFi, y triangulación de torres de telefonía móvil GSM, los cuales se detalla a continuación: (Gerson Beltrán López, 2012)

- ✓ Un sistema GPS (Global Positioning System) o Sistema de Posicionamiento

Global es un sistema compuesto por un lado por una red de 30 satélites denominada NAVSTAR, situados en una órbita a unos 20.000 km. de la Tierra, y por otro lado por unos receptores GPS, que permiten determinar nuestra posición en cualquier lugar del planeta, bajo cualquier condición meteorológica. El GPS convencional presenta dificultades a la hora de proporcionar posiciones precisas en condiciones de baja señal. Por ejemplo, cuando el aparato está rodeado de edificios altos (como consecuencia de la recepción de múltiples señales rebotadas) o cuando la señal del satélite se ve atenuada por encontrarnos con obstáculos, dentro de edificios o debajo de túneles. (Gerson Beltrán López, 2012)

- ✓ Es posible saber tu localización sin tener GPS, calculándola gracias a la red WiFi a la que estás conectado.

Todas las redes WIFI están transmitiendo su identificador, **la dirección MAC**, y ésta puede leerse a una distancia de hasta 500 metros. En este radio, en una ciudad se puede acceder a varias WIFIs. Si sabemos en qué posición está cada una, podemos triangular nuestra posición a partir de las matrículas de las redes (las direcciones MAC) (Federico Miyara, 2002)

Sólo queda por lo tanto, crear una **base de datos donde para cada MAC se guarde su posición GPS**. Para ello se envían vehículos convenientemente equipados por nuestras calles para ir registrando estos datos. SkyHook es una de las empresas que se dedica a esto, proporcionando la posición en base a las redes WIFI. Otra por supuesto, es Google con su Google Street View. (Federico Miyara, 2002)

- ✓ Por otra parte, desde hace años, las operadoras (GSM) ofrecen **el servicio de localización de móviles**.

Cualquier móvil sirve para esto ya que no se basan en GPS y en la recepción de señal WIFI. Es más sencillo que esto.

Por su diseño, un móvil está en contacto con una o varias torres de telefonía.

Mediante un cálculo de triangulación es posible saber la posición del móvil con un pequeño margen de error. (www.ehu.es, 2010)

Este margen dependerá del número de torres que hay alrededor del terminal, que en ciudad es más alto, permitiendo una precisión de hasta 200 metros.

Hoy en día, incluso se usa para reconstruir la posición de un acusado en base a la posición de su móvil en un día determinado y de los ficheros que guarda su operadora.

Aunque puede sonar un poco paranoico, si tienes el móvil encendido, se te puede localizar.

2.16. MENSAJERÍA EN LA NUBE

La Mensajería en la Nube Mediante Google “*Google Cloud Messaging – GCM*” este servicio permite enviar los datos mediante la nube.

Este servicio de Google recibió en sus comienzos las siglas **C2DM** (*Cloud to Device Messaging*), pero después de su salida de fase beta modifico su nombre a **GCM** (*Google Cloud Messaging*). En resumen, es la manera de recibir notificaciones push en nuestras aplicaciones móviles. (developers.google.com, 2018)

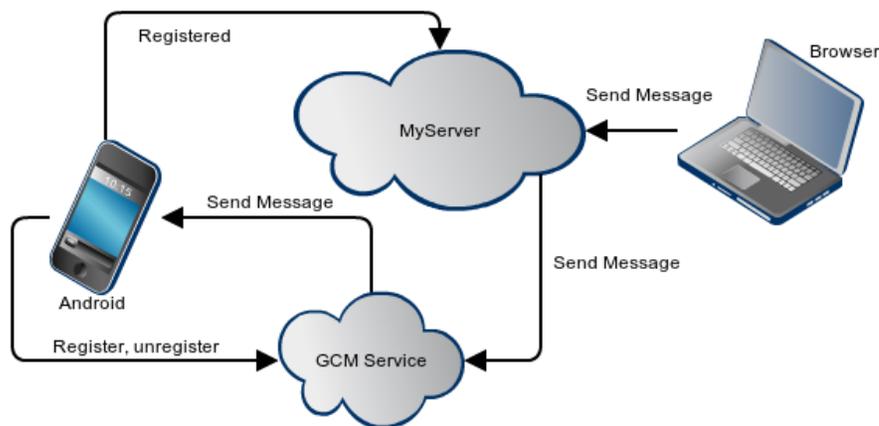


Figura 2. 4 Estructura de Google Cloud Messaging [developers.google.com, 2008]

En términos de tecnología de movilidad, las Notificaciones Push son aquellos mensajes que recibimos en el dispositivo y que han sido emitidos desde cualquier punto de un sistema. Tenemos un ejemplo muy claro en el popular WhatsApp, donde son los usuarios los que envían mensajes a los dispositivos de otros usuarios. Otro ejemplo con notificaciones enviadas de manera automática (y no manual como WhatsApp) podría ser una aplicación de cliente de correo, cuando el servidor detecta un mensaje entrante envía una notificación al dispositivo del usuario. (developers.google.com, 2018)

Las Notificaciones Push permiten el envío de mensajes desde cualquier parte de un sistema a una aplicación móvil tanto si la aplicación está siendo utilizada por el usuario, si está corriendo en un segundo plano, si todavía no ha sido arrancada o, incluso, si el dispositivo está en reposo. (developers.google.com, 2018)

Las notificaciones Push en Android mediante Google Cloud Messaging (GCM) se ejecutan en un escenario que está compuesto por, al menos, tres actores (Ver Figura 2.4.):

- ✓ **Google Cloud Messaging:** El servicio de Google habilitado para el envío de Notificaciones Push a dispositivos Android.
- ✓ **Servidor:** con un servicio (REST, SOAP, aplicación web, etc...) que será el encargado de gestionar los identificadores de registro de dispositivos a

los que podemos enviar las notificaciones y de comunicarse con GCM solicitando el envío de notificaciones al dispositivo (o dispositivos) deseado.

- ✓ **Dispositivo Android:** que recibirá las notificaciones. La notificación push utilizando GCM tiene dos fases:

Fase 1: Registro del dispositivo (Teléfonos Inteligentes)

Lo primero que debemos hacer para que nuestra aplicación Android pueda recibir Notificaciones Push desde GCM será registrarla en dicho servicio. Es una forma de decirle a GCM: "soy un dispositivo que quiere recibir notificaciones de una aplicación". La forma que tenemos de decirle la aplicación de la que queremos recibir notificaciones es indicándole un **número de proyecto** (lo veremos en el siguiente apartado). Esto sería lo equivalente al **paso 1** del diagrama que viene a continuación (Ver Figura 2.5.).

Si todo está correcto, GCM nos responderá con un identificador de registro, también denominado **token device**. Será de la forma:

```
APA33bH6YRxig6cFFUE_utY2aEaEhVThPDkh5xQ6pFbjf
GKkSLZceGkCy8wjhx8QxdfgpMbFSZnAqPcJ5hmDhgkKOULj0UQd759gifSXiN
nsNn4UcMvj KKyk9dsl6n_WJYEfAc6DDy934I1cgxBzs1iNCIdq2VnQ
```

Esto se correspondería con el **paso 2** del diagrama (Ver Figura 2.7.).



Figura 2. 5 Diagrama de Registro del dispositivo [android.com, 2007]

¿Y qué hacemos con ese identificador de registro? Pues ahí es donde entra en juego nuestro servidor, en concreto el servicio que tenemos desplegado en él.

Lo que haremos será **enviar ese identificador a nuestro servicio** (vía HTTP, por ejemplo...) de forma que éste lo pueda almacenar ya que lo necesitará cuando se comunique con GCM para indicarle que debe enviar una notificación (lo veremos en el siguiente punto). Por supuesto, esa petición de envío del identificador de registro **la podemos complementar con cualquier otra información adicional** como el usuario del dispositivo, características o lo que sea ya que el servicio es nuestro. Esto sería lo relativo al **paso 3** del diagrama (Ver Figura 2.5.).

Fase 2: Envío de la notificación

Podemos enviar una Notificación Push a cualquiera de los dispositivos Android que tengamos registrados en nuestro servicio desde **cualquier parte del sistema**.

Para ello, nuestro servicio tendrá una operación donde recibirá la **información relativa al mensaje** que queremos enviar en forma de notificación y **el destinatario** o destinatarios. La información relativa al destinatario puede ser directamente el identificador del registro u otra información que el servicio sepa relacionar con dicho identificador de registro. Esto sería lo equivalente al **paso 1** del siguiente diagrama (Ver Figura 2.6.)

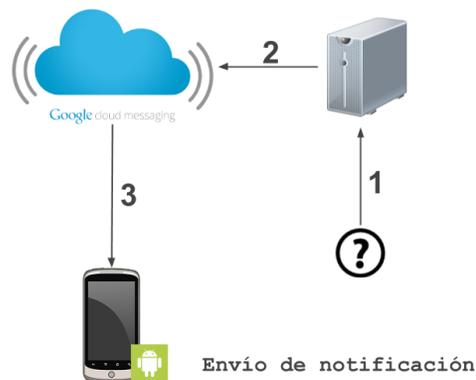


Figura 2. 6 Diagrama del envío de la notificación [android.com, 2007]

A continuación, con la información de la petición de envío de notificación que recibió nuestro servicio, éste envía una nueva petición a GCM (puede hacerse de manera síncrona o asíncrona: HTTP o XMPP) para que mande la notificación al dispositivo. El destinatario de la notificación se indica mediante el **identificador de registro** que obtuvimos en el punto anterior. Necesitaremos adjuntar unas **credenciales de servidor** a nuestra petición.

En el punto siguiente veremos cómo obtenerlas. Esto sería lo equivalente al **punto 2** del diagrama (Ver Figura 2.6.).

Una vez que hemos enviado la petición a Google Cloud Messaging con **nuestras credenciales del servidor**, la información relativa a la **notificación** y su **destinatario**, GCM enviará dicha Notificación Push. **Paso 3** del diagrama (Ver Figura 2.6.).

2.17. INGENIERÍA DE SOFTWARE

El establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea fiable y funcione eficientemente sobre máquinas reales. (F. Bauer Conferencia de la OTAN, 1969)

La Ingeniería del Software incluye la aplicación práctica del conocimiento científico en el diseño y construcción de los programas y la documentación requerida para su desarrollo, operación y mantenimiento. (B. Boehm 1976)

La especificación, desarrollo, gestión y evolución de sistemas software. No está limitado por materiales sujetos a leyes físicas o procesos de fabricación manual. Teorías, métodos y herramientas necesarios para desarrollar software. (Ian Sommerville Software Engineering, 5ª edición 1996)

Ingeniería de software en otras palabras es el estudio dedicado a la creación de software de buena calidad, barato y fácil de desarrollar y mantener. Es la aplicación de la ingeniería al software.

2.17.1. CARACTERÍSTICAS de INGENIERÍA DE SOFTWARE

Es una especialidad de la ingeniería que tiene como objetivo principal, el desarrollo costeable de sistemas de software confiables que funcionen de modo eficiente y comprende todos los aspectos de la producción del software.

El software presenta 3 elementos que lo caracterizan:

- ✓ Los programas y/o algoritmos.
- ✓ Las estructuras de datos.
- ✓ Los documentos.

2.17.2. OBJETIVOS DE INGENIERÍA DE SOFTWARE

Los principales objetivos de la ingeniería de software son:

- ✓ Diseñar programas informáticos que se adecúen a las exigencias de la sociedad objetivos de la Ingeniería en Software
- ✓ Liderar y acoplar el desarrollo de programaciones complicadas.
- ✓ Actuar en todas las fases del ciclo de vida de un producto.
- ✓ Computar los costos de un proyecto y evaluar los tiempos de desarrollo.
- ✓ Realizar el seguimiento de costes y plazos.
- ✓ Liderar equipos de trabajo de desarrollo software.
- ✓ Estructurar la elaboración de evidencias que comprueben el perfecto funcionamiento de los programas y que se adaptan a los requerimientos de análisis y diseño.
- ✓ Diseñar, construir y administrar bases de datos.
- ✓ Liderar y orientar a los programadores durante el desarrollo de aplicaciones.
- ✓ Incluir procesos de calidad en los sistemas, calculando métricas e indicadores y chequeando la calidad del software producido.

- ✓ Liderar y orientar a los programadores durante el desarrollo de aplicaciones.
- ✓ Incluir procesos de calidad en los sistemas, calculando métricas e indicadores y chequeando la calidad del software producido.

Estructurar e inspeccionar el trabajo de su equipo de los técnicos de mantenimiento y los ingenieros de sistemas y redes.

2.17.3. ETAPAS DE INGENIERÍA DE SOFTWARE

La ingeniería de Software consta de siete etapas. A continuación, cada una de ellas brevemente definidas:

- a. Etapa de análisis:** Es el procedimiento de investigación de un problema al que se desea encontrar la solución. Se define con claridad el Problema que hay que resolver o el programa que se desea inventar, identificando los elementos principales que conformarán el producto.
- b. Etapa de Diseño:** Es el procedimiento que emplea la información acumulada en la etapa de análisis al diseño del producto. La labor principal de la etapa de diseño es crear un modelo o las características precisas para el producto o Componentes del Sistema.
- c. Etapa de Desarrollo:** Consiste en el empleo de los diseños creados durante la etapa de diseño para elaborar los elementos a utilizarse en el sistema.

d. Etapa de Pruebas o Verificación Prueba:



Figura 2. 7 Etapas de la Ingeniería de Software [Boem, 2014]

- e. Etapa de Implementación o Entrega Implantación:** Consiste en la distribución del producto y hacerlo llegar a manos del cliente.
- f. Etapa de Mantenimiento:** Consiste en aplicar las soluciones apropiadas a cualquier problema del producto y re- liberar el producto mejorado, dándole una nueva versión.
- g. Etapa final EOL (End-of-Life)** Consiste en ejecutar todas las labores que garanticen que tanto los clientes como los empleados tiene la certeza de que el producto ya no estará más a la disposición, por lo que no se venderá más.

2.18. METODOLOGÍA MOBILE-D

Metodología Mobile-D Primero que todo, para describir esta metodología de desarrollo es conveniente describir un poco su inicio; Se desarrolló como parte de un proyecto finlandés, ICAROS, alrededor de los años 2004 - 2005. Inicialmente, fue creada mediante un proyecto de cooperación muy estrecha con la industria. El grueso del trabajo fue realizado por los investigadores del VTT. Aun así, la metodología de diseño se elaboró con una participación importante de las empresas de TI finlandesas. Tal como se puede ver en los experimentos que se han documentado esto consiguió que la investigación llevada a cabo no se alejara demasiado de las reglas de desarrollo de las aplicaciones comerciales. (Jhoan Sebastian Gomez Medina 2016).

Este método está diseñado para un ciclo de desarrollo rápido y para equipos pequeños (menos de diez programadores), en el mismo espacio de trabajo, el objetivo de Mobile-D es que el ciclo de desarrollo propuesto se finalice en menos tiempo.

El ciclo del proyecto se divide en cinco fases: exploración, inicialización, productización, estabilización y prueba del sistema. En general, todas las fases (con la excepción de la primera fase exploratoria) contienen tres días de desarrollo distintos: planificación, trabajo y liberación. Se añadirán días para acciones adicionales en casos particulares (se necesitarán días para la preparación del proyecto en la fase de inicialización, por ejemplo).



Figura 2. 8 Ciclo de desarrollo de la metodología MOBILE –D [Bonilla, 2014]

2.18.1. Fases

La metodología cuenta con cinco fases por las cuales pasa el producto a realizarse, la línea de producción empieza con la fase de exploración, después a la fase de Iniciación, luego a la fase del producto, posteriormente a la fase de estabilización y finalmente la fase de pruebas.

2.18.1.1. Fase Exploración

En esta fase se realiza la planificación y los conceptos básicos del proyecto. Se realizan los alcances del proyecto y su establecimiento con las funcionalidades donde se va a llegar. En esta fase se presta especial atención a la participación de los clientes.



Figura 2. 9 Etapas de fase Exploración [Bonilla, 2014]

2.18.1.2. Fase de Inicialización

En esta fase se configura el proyecto; se preparan y verifican todos los recursos necesarios; los tecnológicos y de comunicación. Se establece el entorno técnico del proyecto. Esta es la fase en la que asegura el éxito o el fracaso de las demás fases del proyecto. (Balaguera, 2013)

Según los creadores de Mobile-D afirman que esta fase es la fundamental para el desarrollo ágil. Esta fase se divide en cuatro etapas.



Figura 2. 10 Proceso fase de Inicialización [Bonilla, 2014]

2.18.1.3. Fase de Producción

En esta fase se realiza la programación de los tres días (Figura 11) interactivamente hasta efectuar todas las funcionalidades necesarias de todos los módulos, se realiza las pruebas del sistema para verificar el correcto funcionamiento del desarrollo. (Balaguera, 2013).



Figura 2. 11 Proceso fases de Producción [Bonilla, 2014]

2.18.1.4. Fase de Estabilización

En esta fase se realiza la integración de todos los módulos donde se verifica el completo funcionamiento de sistema, en esta fase se asegura el éxito y la calidad

de la implementación del proyecto. Adicionalmente, en esta fase se puede incluir la producción de documentación. (Balaguera, 2013)



Figura 2. 12 Proceso fase de Estabilización [Bonilla, 2014]

2.18.1.5. Fase de Pruebas

La fase de pruebas y reparación del sistema es la última fase de la metodología Mobile-D, se entrega un producto terminado según lo determinado por el cliente, se realiza diferentes pruebas con los requisitos entregados por el cliente, se hace las correcciones y reparación de posibles errores. Una vez terminado todas las fases anteriores se contaría con una aplicación publicable y entregable al cliente. (Balaguera, 2013)

2.19. METRICAS DE CALIDAD DE SOFTWARE – QSOS

El método QSOS describe un proceso formal para evaluar, comparar y seleccionar soluciones de código abierto. Este proceso consta de cuatro pasos interdependientes e iterativos:



Figura 2. 13 Sistema de Calidad QSOS [www.org.com, 2006]

Pasos de la Metodología

- ✓ **Definición** Constitución y enriquecimiento de los marcos de referencia que serán utilizados en los pasos siguientes.
- ✓ **Evaluación** Software hecho de acuerdo a 3 ejes de criterios: cobertura funcional, riesgos del usuario y riesgos del proveedor de servicios.
- ✓ **Calificación** Carga de los criterios divididos en 3 ejes, modelando el contexto (requerimientos de usuario y/o estrategia escogida por el proveedor de servicios)
- ✓ **Selección** Aplicación de filtro configurado en el paso anterior a los datos encontrados en los primeros dos pasos, de manera de realizar consultas, comparaciones y selección de productos.

2.19.1. Pasos De La Metodología Qsos

Paso 1 Definición: Construcción y enriquecimiento de los marcos de referencia que serán utilizados en los pasos siguientes.

El objetivo de este paso es definir varios elementos de la tipología a ser utilizada por los 3 pasos que siguen. Los marcos de referencia son:

- ✓ **Familia de Software:** Este aspecto responde la pregunta “¿Qué tipo de software estamos analizando?”.
- ✓ **Tipos de Licencia:** Clasificación de las licencias más comunes de Software Libre y de código abierto.
- ✓ **Tipos de comunidades:** Clasificación de las comunidades que pueden desarrollar Software Libre.

Paso 2 Evaluación: Software hecho de acuerdo a 3 ejes de criterios: cobertura funcional, riesgos del usuario y riesgos del proveedor de servicios.

Este paso tiene como objetivo la colección de información por parte de las comunidades de código abierto. Esta evaluación comprende la elaboración de la

tarjeta de identificación del software, así como la elaboración de la hoja de evaluación del software.

- ✓ **Tarjeta de identificación del software:** Contiene datos y hechos acerca del software, es utilizada como base para el proceso de evaluación. Contiene elementos como nombre, fechas de creación, tipo de software autores, descripción general, los servicios que presenta, aspectos técnicos y funcionales, entre otros, se identifica como:
- ✓ **Hoja de evaluación del software:** Contempla la identificación, descripción y análisis en detalle de cada versión que se presenta del software.

Paso 3 Calificación: Carga de los criterios divididos en ejes, modelando el contexto (requerimientos de usuario y/o estrategia escogida por el proveedor de servicios).

El objetivo de este paso es definir los filtros que traduzcan las necesidades y restricciones relacionadas con la selección del software de código abierto en un contexto específico. Para ello se definen niveles de filtros sobre el software en base.

Tenemos cuatro tipos de filtros:

- ✓ Filtros sobre la tarjeta de identificación.
- ✓ Filtros sobre las funcionalidades.
- ✓ Filtros sobre los riesgos desde la perspectiva del usuario.
- ✓ Filtros sobre los riesgos desde la perspectiva del proveedor de servicios

Paso 4 Selección: Aplicación del filtro configurado en el paso anterior a los datos encontrados en los dos primeros pasos, de manera de realizar consultas, comparaciones y selección de productos.

Este paso tiene como objetivo identificar el software que contenga y satisfaga los requerimientos de usuario, o de manera más general permita la comparación de

software de una misma familia. Puede ser de dos modos: un modo estricto (selección estricta), y otro un poco más holgado (selección holgada).

- ✓ **La selección estricta** se basa en la eliminación del software tan pronto como el software no cumpla con lo formulado en el paso de Calificación. Este método es muy restrictivo y puede no seleccionar software alguno.
- ✓ **La selección holgada** se basa en darle puntuación nuevamente al software dependiendo de lo obtenido en el paso de Evaluación. Al final se escoge el software con más (o menos) puntos. (Oliver Pilot,2006)

2.19.2. MÉTRICAS MAS USADAS POR QSOS

Métricas generales: que se aplican a todo tipo de Software Libre u Open Source.

Métricas específicas: que se aplican a una familia determinada de software.

- ✓ Las **métricas generales** se describen en la “*Generic Section*” de la hoja de evaluación y se encuentran justo debajo de la tarjeta de identificación. Este tipo de métrica comprende aspectos como madurez, actividad en el desarrollo, portabilidad, entre otras.
- ✓ Las **métricas específicas** se describen justo después de la “*Generic Section*”. Comprenden aspectos inherentes a las características del tipo de software. Por ejemplo, para la familia de software de RDBMS se contempla el soporte de SQL, el soporte de *constraints* sobre las tablas, entre otros.

Durabilidad intrínseca (sustentabilidad)

- ✓ Madurez: Edad, Estabilidad, Historia, problemas conocidos, Probabilidad de forks, fuentes del forking
- ✓ Adopción: Popularidad (relacionada con: público en general, expertos Referencias (si se emplea en alguna solución conocida), Comunidad de contribuyentes (nivel de actividad), Libros disponibles
- ✓ Liderazgo de desarrollo: Equipo de desarrollo (tamaño), Estilo de gerencia (“dictatorial”, “un poco déspota”, “consejo de arquitectos”)

- ✓ Actividad: Desarrolladores (número total de desarrolladores, cargos bien/mal definidos e identificados), Actividad en solución de problemas, Actividad en el desarrollo de funcionalidades, Actividad en nuevos lanzamientos

Solución industrializada

- ✓ Independencia del desarrollo (si el software es desarrollado por una única compañía)
- ✓ Servicios: Entrenamiento, Soporte, Consultoría
- ✓ Documentación (no disponible, disponible/actualizada, disponible/no actualizada)

Aseguramiento de la calidad

- ✓ Aseguramiento de la calidad (utilizando algún método o modelo reconocido)

Explotabilidad

- ✓ Facilidad de uso, ergonomía (si requiere de conocimientos técnicos: bajo, medio o alto)
- ✓ Administración / Monitoreo (si proporciona herramientas de administración/monitoreo)

Adaptabilidad técnica (inherente al código fuente)

- ✓ Modularidad (software: monolítico, modularidad de primer nivel, completamente modular)
- ✓ Modificación del código (compilación: difícil y a mano, posible y a man)
- ✓ Extensión del código (requiere re-compilación, uso y manejo de plugins)

Estrategia

- ✓ Licencia
- ✓ Permisividad (sólo si el usuario quiere hacerse dueño del código)
- ✓ Protección respecto a forks propietarios
- ✓ Propietario de los copyrights (si es un individual, una comunidad o una empresa)

- ✓ Modificación del código fuente (imposible, uso de repositorios, ...)

2.19.3. Formulas y Evaluación de Criterios

Tabla 2. 4
Evaluación de Criterios

Dimensión	Criterios	Características	Escala 0,1 y 2			Promedio	
Calidad del sistema	Fiabilidad	Madurez	2	2	2	2	
		Aprendizaje	2	2	2	2	
		Operabilidad	2	2	2	2	
	Usabilidad	Accesibilidad	2	1	2	1.66	
		Interfaz de usuario estética	2	2	2	2	
	Eficiencia en el desempeño	Comportamiento de tiempo	2	2	2	2	
	Funcionalidad	Integridad funcional	1	2	2	1.66	
		Adecuación funcional	2	2	2	2	
	Calidad de servicio	Tangible	Documentación	2	2	2	2
		Empatía	Comunicación	2	2	2	2

2.20. COSTOS COCOMO II

COCOMO está compuesto por tres modelos que corresponden a distintos niveles de detalle y precisión. Mencionados en orden creciente son: Modelo Básico, Intermedio y Detallado. La estimación es más precisa a medida que se toman en cuenta mayor cantidad de factores que influyen en el desarrollo de un producto de software. (Adriana Gómez, 2015)

COCOMO permite estimar cómo se distribuye el esfuerzo y el tiempo en las distintas fases del desarrollo de un proyecto y dentro de cada fase, en las actividades principales. Las fases consideradas por COCOMO son:

- ✓ **Diseño del Producto (PD).** Se define la arquitectura del hardware, software y las estructuras de datos y control. También se desarrolla un bosquejo del manual del usuario y los planes de aceptación y testeo.
- ✓ **Diseño Detallado (DD)**
- ✓ **Codificación y Testeo de Unidades (CT)** En estas dos fases el diseño global de la fase anterior es implementado, creando las componentes de software, que son testeadas y evaluadas individualmente.
- ✓ **Integración y Testeo (IT).** Se fusionan todas las componentes de software desarrolladas con el fin de lograr que el producto de software funcione correctamente. Los requerimientos definidos son usados para controlar las aptitudes del producto liberado. (María del C.López, 2015)

Los costos y tiempos de las fases excluidas (Requerimientos y Mantenimiento) deben ser estimados en forma separada empleando otros modelos.

Se distinguen las siguientes actividades principales:

- ✓ **Análisis de Requerimientos.** Determinación, especificación, revisión y actualización de la funcionalidad, performance e interface del software
- ✓ **Diseño del producto.** Determinación, especificación, revisión y actualización de la arquitectura de hardware y software, diseño del programa y diseño de la base de datos.
- ✓ **Programación (Diseño detallado + Unit Test).** Comprende el diseño detallado, codificación, testeos unitarios e integración de las componentes individuales de software. Incluye planificación del personal de programación, adquisición de herramientas, desarrollo de la base de datos, y documentación al nivel de componente.
- ✓ **Planificación del Testeo.** Especificación, revisión y actualización de los planes de testeo del producto y de aceptación del mismo. Adquisición de herramientas de testeo y lotes de datos de prueba.
- ✓ **Verificación y Validación.** Ejecución de la validación y verificación de requerimientos, diseño, testeos del producto y testeos de aceptación.

Adquisición de herramientas de V&V. Se hacen las preguntas ¿estaremos construyendo el producto correcto? y ¿estaremos construyendo correctamente el producto?

- ✓ **Actividades de oficina.** Tareas referidas al gerenciamiento del proyecto. Incluye la administración de contratos y subcontratos, relaciones con los clientes, etc.
- ✓ **Administración de la Configuración y Aseguramiento de la Calidad (CM/QA):** La Administración de la Configuración incluye la identificación del producto, control de cambios, administración de las librerías de soporte, etc. Aseguramiento de la Calidad abarca el desarrollo y monitoreo de estándares, auditorías técnicas del producto de software y el proceso de desarrollo.

Manuales: Desarrollo y actualización de los manuales de usuario, operador y mantenimiento.

2.20.1. MODELO DE DESARROLLO

En el modelo COCOMO' 81 uno de los factores más importantes que influye en la duración y el costo de un proyecto de software es el Modo de Desarrollo. Todo proyecto corresponde a uno de los siguientes tres modos: (Adriana Gómez, 2015)

- ✓ **Modo Orgánico (Organic):** En esta clasificación se encuentran proyectos desarrollados en un ambiente familiar y estable. El producto a elaborar es relativamente pequeño y requiere pocas innovaciones tecnológicas en lo que refiere a algoritmos, estructuras de datos e integración de hardware. La mayoría de las personas conectadas con el proyecto tienen gran experiencia en sistemas relacionados dentro de la organización, y un entendimiento acabado de cómo el sistema contribuirá a los objetivos de la organización. Esto significa que todo el equipo de desarrollo podrá contribuir en las etapas iniciales del proyecto sin generar confusión en las comunicaciones debido a que todos conocen que tarea deben realizar.

Estas características permiten decir que los proyectos que se encuentran en este modo tienen una gran productividad y una pequeña deseconomía² de escala. Ejemplos de software que se encuentran bajo esta clasificación son:

- Modelos de negocios
- Modelos científicos
- Sistemas operativos de pequeña escala

✓ **Modo Semiacoplado (Semidetached):** Es un modelo para productos de software de tamaño y complejidad media. Las características de los proyectos se consideran intermedias a las de los modos Orgánico y Empotrado.

Esto implica: Que el equipo de desarrollo:

- Tiene un nivel intermedio de experiencia y conocimiento del sistema en desarrollo.
- Está conformado por algunas personas con vasta experiencia y otras inexpertas en el campo de aplicación.
- Está constituido por personas con amplios conocimientos sólo en algunos aspectos.

Con respecto al cumplimiento de especificaciones de interfase y funcionalidad:

- Son sistemas que presentan niveles variados de exigencia, algunas interfases rigurosas (auditadas por el gobierno) y otras interfases muy flexibles (mensajes de display al operador).

Los productos tienen un tamaño que llega a 300 KSLOC.

Ejemplos de software que se encuentran en esta clasificación son:

- Sistemas de control de producción
- Sistemas de procesamiento de transacciones
- Administradores de Bases de Datos

2.20.2. Modelo Básico

Modelo Básico El Modelo Básico de COCOMO estima el esfuerzo y el tiempo empleado en el desarrollo de un proyecto de software usando dos variables predictivas denominadas factores de costo (cost drivers): el tamaño del software y el modo de desarrollo. Las ecuaciones básicas son: (Adriana Gómez, 2015)

Esfuerzo:

$$PM = A \times (KSLOC)^B$$

Donde:

- ✓ **PM** es el esfuerzo estimado. Representa los meses-persona³ necesarios para ejecutar el proyecto
- ✓ **KSLOC** es el tamaño del software a desarrollar en miles de líneas de código
- ✓ **A y B** son coeficientes que varían según el Modo de Desarrollo (Orgánico, Semiacoplado, Empotrado)

Cronograma:

$$TDEV = C \times (PM)^D$$

Donde:

- **TDEV** representa los meses de trabajo que se necesitan para ejecutar el proyecto
- **C y D** son coeficientes que varían según el Modo de Desarrollo (Orgánico, Semiacoplado, Empotrado)

La Tabla 1 muestra la variación de la fórmula de estimación de esfuerzo y cronograma para los tres Modos de Desarrollo:

Tabla 2. 5
Ecuaciones del Modelo Básico de COCOMO

Modo de Desarrollo	Esfuerzo	Cronograma
Orgánico	$PM=2.4 \times (KSLOC)^{1.05}$	$TDEV=2.5 \times (PM)^{0.38}$
Semiacoplado	$PM=3.0 \times (KSLOC)^{1.12}$	$TDEV=2.5 \times (PM)^{0.35}$
Empotrado	$PM=3.6 \times (KSLOC)^{1.20}$	$TDEV=2.5 \times (PM)^{0.32}$

Notas: [Boehm 1981]

Este modelo es adecuado para una estimación rápida y temprana, pero su precisión es muy limitada debido a que no contempla factores que tienen significativa influencia en los costos, como por ejemplo, restricciones de hardware, experiencia y calidad del equipo de trabajo, y uso de técnicas y herramientas modernas.

2.20.4. Modelo Intermedio

Comparado con el modelo anterior, éste provee un nivel de detalle y precisión superior, por lo cual es más apropiado para la estimación de costos en etapas de mayor especificación. COCOMO Intermedio incorpora un conjunto de quince variables de predicción que toman en cuenta las variaciones de costos no consideradas por COCOMO Básico. (Adriana Gómez, 2015)

Existen diversos factores a considerar en el desarrollo de un buen modelo de estimación de costos de un proyecto de software. Para reducir el número a una cantidad relativamente manejable se utilizaron fundamentalmente dos principios:

Eliminar aquellos factores que son significativos solamente en una fracción relativamente pequeña o en situaciones especiales.

- ✓ Eliminar los factores que están altamente correlacionados con el tamaño y comprimir aquellos factores correlacionados entre sí

Los factores seleccionados se agrupan en cuatro categorías:

Atributos del producto de software

- ✓ RELY Confiabilidad Requerida

- ✓ DATA Tamaño de la Base de Datos
- ✓ CPLX Complejidad del Producto

Atributos del hardware

- ✓ TIME Restricción del Tiempo de Ejecución
- ✓ STOR Restricción del Almacenamiento Principal
- ✓ VIRT Volatilidad de la Máquina Virtual*
- ✓ TURN Tiempo de Respuesta de la computadora expresado en horas

Atributos del personal involucrado en el proyecto

- ✓ ACAP Capacidad del Analista
- ✓ AEXP Experiencia en Aplicaciones Similares
- ✓ PCAP Capacidad del Programador
- ✓ VEXP Experiencia en la máquina virtual
- ✓ LEXP Experiencia en el Lenguaje de Programación
- ✓ Atributos propios del proyecto
- ✓ MODP Prácticas Modernas de Programación
- ✓ TOOL Uso de Herramientas de Software
- ✓ SCED Cronograma de Desarrollo Requerido

El proceso de estimación del esfuerzo puede sintetizarse en los siguientes pasos:

- ✓ Se calcula el esfuerzo nominal $PM_{Nominal}$, al igual que en el modelo Básico, donde los únicos factores de costo son el tamaño y el modo de desarrollo.
- ✓ Se determina el Factor de Ajuste del Esfuerzo (EAF, Effort Adjustment Factor) según la fórmula:

$$EAF = \prod_{i=1}^{15} EM_i$$

Donde cada EM, llamado factor multiplicador de esfuerzo, es el valor que corresponde a cada atributo de acuerdo al grado de influencia (Muy Bajo, Bajo, Nominal, Alto, Muy Alto, Extra Alto) en el esfuerzo del desarrollo del software. La tabla 8-3 de [Boehm 1981] muestra las características que

determinan en que nivel se puede clasificar cada factor de costo, y la tabla 8-2 de [Boehm 1981] establece el valor de cada factor para cada nivel.

- ✓ Finalmente, se ajusta el esfuerzo nominal aplicando el EAF.

$$PM = A \times EAF \times (KSLOC)^B$$

La Tabla 3 muestra la variación de la ecuación de estimación de esfuerzo y cronograma según los tres modos de desarrollo.

Tabla 2. 6.
Ecuaciones del Modelo Intermedio de COCOMO

Modo de Desarrollo	Esfuerzo Nominal	Esfuerzo Ajustado	Cronograma
Orgánico	$PM_{nominal} = 3.2 \times (KSLOC)^{1.05}$	$PM = 3.2 \times EAF \times (KSLOC)^{1.05}$	$TDEV = 2.5 \times (PM)^{0.38}$
Semiacoplado	$PM_{nominal} = 3.0 \times (KSLOC)^{1.12}$	$PM = 3.0 \times EAF \times (KSLOC)^{1.12}$	$TDEV = 2.5 \times (PM)^{0.35}$
Empotrado	$PM_{nominal} = 2.8 \times (KSLOC)^{1.20}$	$PM = 2.8 \times EAF \times (KSLOC)^{1.20}$	$TDEV = 2.5 \times (PM)^{0.32}$

Notas: [Boehm, 1981]

Los porcentajes de distribución de esfuerzo y cronograma por fase se obtienen en función del modo y tamaño del producto, según Tabla 2.5

En lo referente a la distribución del esfuerzo por fase y por actividades, también se tiene en cuenta el tamaño del software y el modo de desarrollo. Las tablas correspondientes se encuentran en [Boehm 1981].

Es importante destacar que este modelo tiene dos limitaciones importantes a la hora de estimar grandes proyectos de software:

- ✓ La estimación de la distribución del esfuerzo para cada fase resulta imprecisa.
- ✓ No es muy práctico si el producto de software tiene un gran número de componentes.

2.20.5. Modelo Detallado

El Modelo Detallado provee los medios para generar estimaciones con mayor grado de precisión y detalle. Difiere del Modelo Intermedio en dos aspectos principales que ayudan a superar las limitaciones mencionadas

- ✓ Jerarquía de niveles del producto. En el Modelo Intermedio se pueden calcular valores diferentes de los factores de costo para cada componente de software. Este proceso puede resultar muy tedioso e innecesariamente repetitivo si las componentes están agrupadas en subsistemas de características generales similares.
- ✓ Multiplicadores de Esfuerzo (EM Effort Multipliers) sensitivos a las fases. El modelo Detallado provee un conjunto de multiplicadores diferentes para cada factor de costo, según la fase del ciclo de desarrollo que se considere

Las ecuaciones fundamentales de este modelo son similares a las del modelo COCOMO Intermedio, la única diferencia reside en el cálculo del Factor de Ajuste del Esfuerzo (EAF). El procedimiento incluye el cálculo de un Factor de Ajuste del Esfuerzo al nivel de módulo (EAFM) y otro al nivel de subsistema (EAFS).

2.21. METRICAS DE CALIDAD DE SOFTWARE (ISO-9126)

A continuación, se detalla cada una de las características que establece el estándar ISO-9126.

- a. **Funcionalidad.** En este grupo es conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada, el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado. Para este propósito se establecen los siguientes atributos.
- b. **Adecuación.** Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.
- c. **Exactitud.** Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.

- d. **Interoperabilidad.** Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.
- e. **Conformidad.** Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.
- f. **Seguridad.** Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.
- g. **Confiabilidad.** Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.

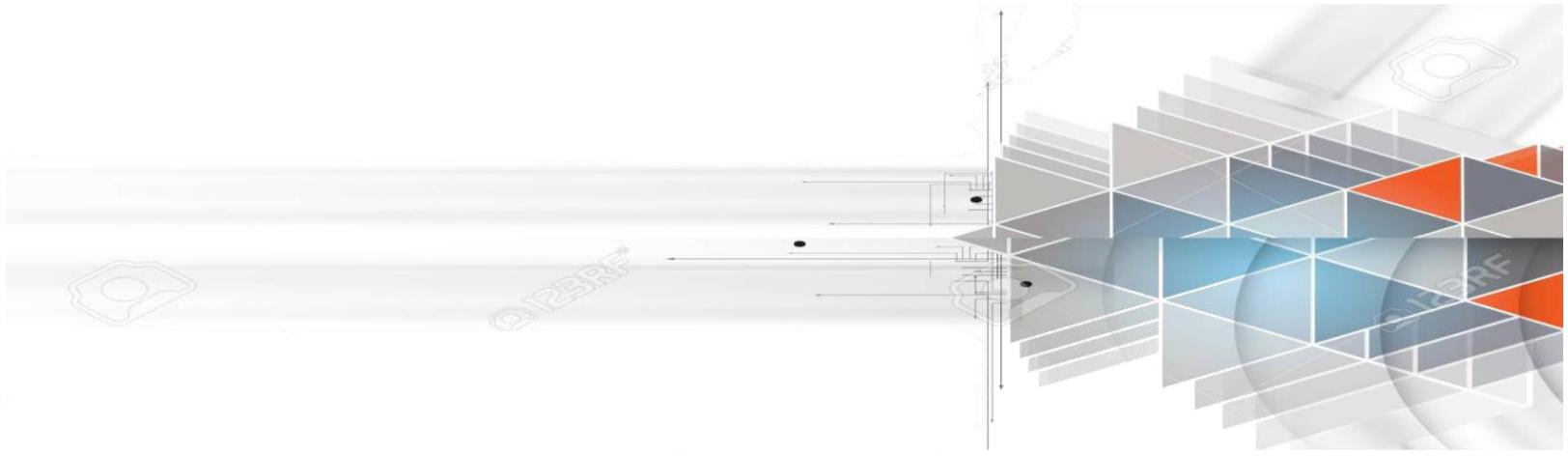
Las sub características que el estándar sugiere son:

- ✓ **Nivel de Madurez.** Permite medir la frecuencia de falla por errores en el software. Tolerancia a fallas. Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
- ✓ **Recuperación.** Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.
- ✓ **Usabilidad.** Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema. **Comprensibilidad.** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
- ✓ **Facilidad de Aprender.** Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.
- ✓ **Operatividad.** Agrupa los conceptos que evalúan la operación y el control del sistema.
- ✓ **Eficiencia.** Esta característica permite evaluar la relación entre el nivel de, funcionamiento del software y la cantidad de recursos

usados. Los aspectos a evaluar son. Comportamiento con respecto al Tiempo. Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos. Comportamiento con respecto a Recursos. Atributos del software relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.

- ✓ **Mantenibilidad.** Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad. En este caso, se tienen los siguientes factores:
- ✓ **Capacidad de análisis.** Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.
- ✓ **Capacidad de modificación.** Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente.
- ✓ **Estabilidad.** Permite evaluar los riesgos de efectos inesperados debidos a las modificaciones realizadas al software.
- ✓ **Facilidad de Prueba.** Se refiere al esfuerzo necesario para validar el software una vez que fue modificado.
- ✓ **Portabilidad.** En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos.
- ✓ **Adaptabilidad.** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- ✓ **Facilidad de Instalación.** Es el esfuerzo necesario para instalar el software en un ambiente determinado.
- ✓ **Conformidad.** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.

- ✓ **Capacidad de reemplazo.** Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.



CAPÍTULO III

MARCO APLICATIVO



3. MARCO APLICATIVO

3.1. INTRODUCCIÓN

En el capítulo anterior nos sirvió para exponer todo el marco teórico que ahora tomaremos como base para el desarrollo del modelo web de localización y reportes de accidentes de tránsito basado en Android para así lograr los objetivos propuestos en el primer capítulo.

Como se trata de un sistema donde hace el uso de teléfonos inteligentes surge la necesidad de utilizar una metodología orientada a móviles en este caso se eligió Mobile-D.

- ✓ Exploración
- ✓ Inicialización
- ✓ Productización
- ✓ Estabilización
- ✓ prueba del sistema

3.2. METODOLOGIA MOBILE-D

De acuerdo a la combinación de las metodologías se tomará como base de desarrollo la metodología Móvil-D, considerando las demás metodologías como etapas que reforzaran el desarrollo del Modelo web de Localización y Reportes de accidentes de tránsito basado en Android:

- ✓ **Fase de exploración** y los pasos a seguir son: el análisis de características de la población, problema o necesidades a entender, funcionalidades pedagógicas y didácticas.
- ✓ **Fase de inicialización** la cual está compuesta de un análisis de la enseñanza como un proceso de software móvil, y la especificación de módulos
- ✓ **Fase de producción** que es el encargado de la planificación del módulo, implementación del módulo.

- ✓ **Fase de estabilización** continuando con el desarrollo del prototipo, la integración de los módulos
- ✓ **Fase de prueba** y sus pasos a seguir son pruebas finales, reparación de errores

3.2.1. Fase 1: Exploración

Se necesita primero desarrollar una aplicación para móviles inteligentes que sea capaz de detectar cambios bruscos de velocidad de un automóvil luego este que enviara las informaciones al Web Service, los datos de envío ya mencionadas en el capítulo anterior.

3.2.1.1. Establecimiento de las Partes

Donde el usuario Conductor/Pasajero:

- ✓ Requiere la aplicación instalada en su móvil.
- ✓ La aplicación no necesita ninguna manipulación extra a parte de la instalación y de la ejecución de dicha aplicación.
- ✓ Que la aplicación cuente con una interfaz amigable.

Usuario Servicio de Emergencia: Hospital, Transito

- ✓ La Web Service les proporcionara información necesaria a los servicios de emergencias.

Solo los de servicios de emergencia tendrán acceso a los reportes detallados de los accidentes proporcionados en la Web Service.

3.2.1.2. Definición del Alcance

La Web Service recibirá la información de todos los accidentes de tránsito y luego analizará y enviará la información a quien corresponda. Dependiendo de cuan informatizados esté los entes de emergencias en Bolivia.

Se podrá saber:

- ✓ A qué centro de emergencia se podrá derivar a los heridos; dependiendo al lugar del accidente.

- ✓ Enviar la notificación del accidente a todos los Smartphone que cuentan con esta aplicación, de manera que los demás conductores puedan evitar el recorrido sobre el cual se encuentra el siniestro, esto también ayudara en el control de vías al tránsito.

Requisitos Previos

- ✓ Información del registro de hospitales del Departamento de La Paz.
- ✓ Celulares con Sistema operativo Android en versión 3.0 o superior

Objetivos:

- ✓ La ubicación del Accidente a reportar del choque
- ✓ Consultar la placa del vehículo en la Base de Datos.

Alcance:

Prototipo funcional de una App Android que consulte el registro de los usuarios con sus respectivas descripciones de la movilidad en una Base de Datos intermedia.

3.2.1.3. Establecimiento del Proyecto

En esta etapa se definió el entorno técnico y físico del proyecto.

Documento de requerimientos iniciales: Documento de Análisis Inicial y Diseño de Arquitectura base

- ✓ **Tecnología:** Android
- ✓ **Lenguaje de Programación:** Java
- ✓ **Librerías Java:** jdk 7.0
- ✓ **IDE:** Eclipse Juno
- ✓ **Sistema Operativo:** Android versión 3.0 o superior
- ✓ **Equipos:** 1 Gestor de Base de Datos Xampp, 1 Laptop con procesador 4 núcleos a más, 4 GB de RAM y con espacio mínimo disponible en Disco de 30 GB
- ✓ **Metodología de Desarrollo:** Mobile-D

3.2.2. Fase 2: Inicialización

3.2.2.1. Configuración del Proyecto

Preparación del ambiente:

- ✓ Instalación del jdk 7.0, instalación del Eclipse, instalación del Servidor de Base de Datos Xampp MySql, instalación del Apache.

Capacitaciones:

- ✓ Se hará una capacitación a los usuarios como choferes para el correcto uso de la aplicación móvil.
- ✓ Capacitación técnica al equipo de desarrollo sobre la tecnología de desarrollo móvil con Eclipse.

Plan de comunicación:

- ✓ Se solicitó la lista de correos, teléfonos y como medio de comunicación con los usuarios.

3.2.2.2. Planificación Inicial En 0 Iteraciones

Lo que se pretende hacer en el presente proyecto es un sistema que ayude a mejorar en el tiempo de las atenciones a los heridos en los accidentes de tránsito mediante él envió de alarma al servidor y este que pueda alertar a los administradores del tránsito.

a. Exposición del plan del proyecto y la arquitectura de la aplicación

La arquitectura de la solución está orientada a servicios, el proyecto consta de 3 partes:

- Aplicación Móvil.
- Servicio Web.
- Servidor de Base de datos.

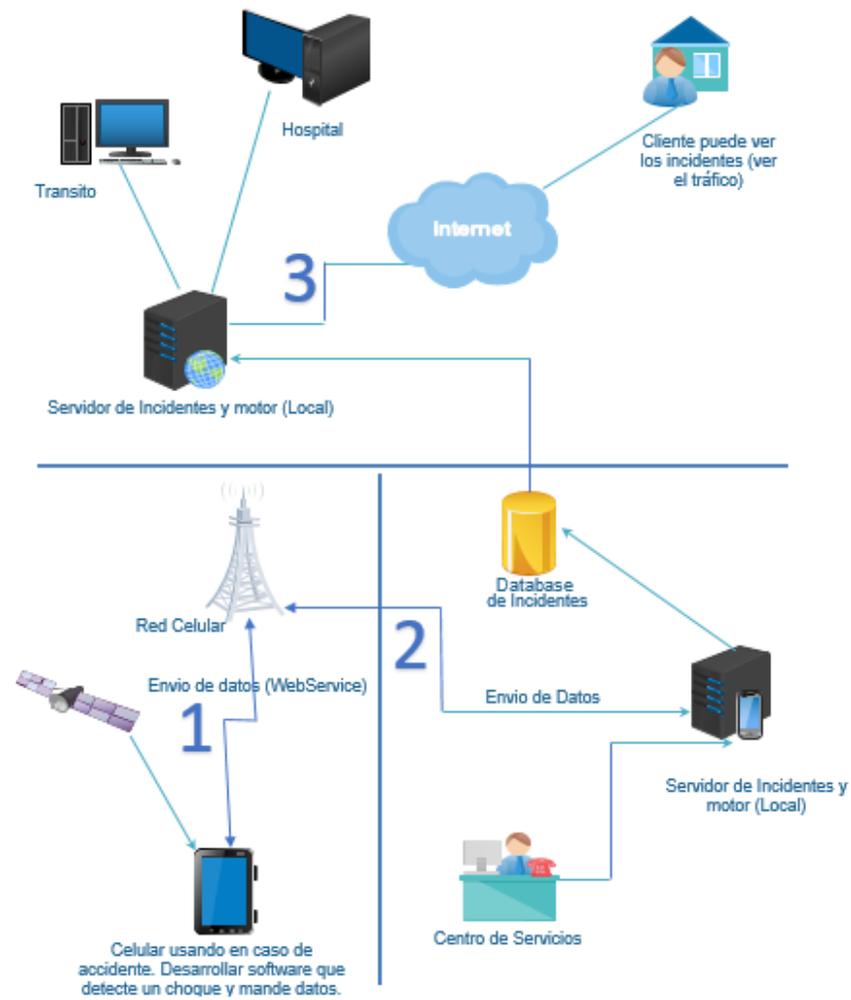


Figura 3. 1 Diagrama de despliegue del Sistema [Elaboración propia]

Como se podrá observar la Figura 3.1 se divide la arquitectura en 3 secciones, la primera, el software que está en el celular y la obtención de la posición por parte de este, la segunda, el motor que procesa todos los incidentes y tiene la información necesaria para saber a dónde conviene mandar los heridos y qué tipo de ayuda conviene enviar y como tercera y última sección, la obtención por parte de los usuarios comunes y las entidades de emergencias de la información acerca de las emergencias.

✓ **ETAPA 1: Obtención De La Posición/Celular**

El celular tiene corriendo un programa que detecta cambios bruscos en la velocidad y movimiento del celular, basado en el cambio y cantidad de fuerza G, los cuales se pueden obtener a partir de las medidas detectadas por el acelerómetro embebido dentro del celular. De esta manera, si el celular está en un bolsillo y se cae, o el celular está fijo en un holder y hay una frenada brusca pero no un choque, no se emite una alerta; sin embargo, sí en cambio la fuerza G es mayor a 4G, entonces en ese caso, sí se emite una alerta de emergencia, con las siguientes acciones:

- Se comienza una conexión a la Web Service seguro, la cual transferirá la información de urgencia más necesaria (UDS) recolectada por el celular, como ser posición, velocidad al momento del choque, etc.
- Luego que se confirma que el UDS se recibió, se comienza la transmisión del siguiente mensaje (EDS), el cual contiene más datos y es el que potencialmente cambiaría a futuro, de ser necesario, para agregar información, por ejemplo, una foto o una grabación de voz de unos segundos.
- Los datos a ser enviados al servidor están en formato JSON.
- Opcionalmente, luego de la llamada se generará algún tipo sonido fuerte para que la ubicación pueda ser establecida más fácilmente cuando el personal se encuentre en el lugar del accidente. (Esto va a depender de cuán rápido se agote la batería del celular).

El celular obtendrá la posición por medio del receptor A-GPS incorporado y/o la red celular y/o WiFi.

✓ **ETAPA 2: Servidor De Incidentes, Motor Y Persistencia**

El servidor de incidentes recibirá todos los incidentes y los procesará, de ser posible, teniendo en cuenta la información de hospitales, policía y enviará la información personalizada al tipo de accidente ocurrido.

Actualmente, en Bolivia solamente existen dos o tres establecimientos particulares o de convenio que cuentan con sistema informatizado de la historia clínica, entre ellos está el hospital clínico Los Pinos, La Paz.

✓ **ETAPA 3: Intranet Entre Policía**

Esta sección se encarga de la definición de la intranet que permitirá a los distintos servicios de emergencias ver las alertas para poder proceder de la manera más rápida y eficiente posible para la asistencia a los posibles heridos. Como se dijo anteriormente, cierta información podrá ser consumida por el público general y otra estará restringida.

b. Arquitectura del Diagrama de Despliegue

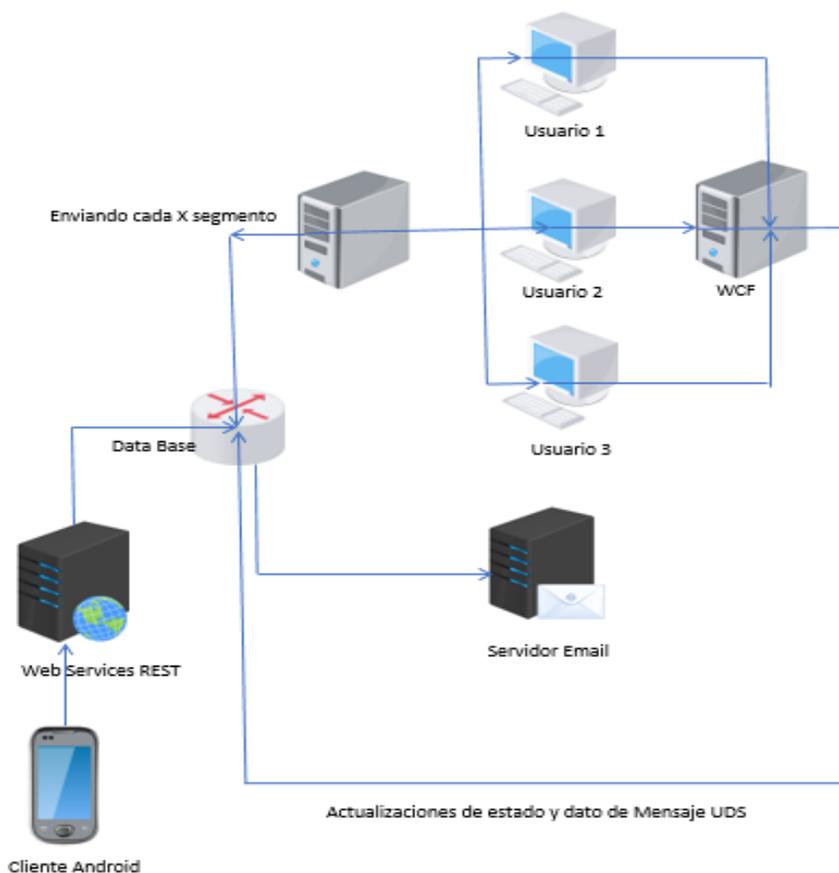


Figura 3. 2 Arquitectura del Sistema [Elaboración Propia]

Al detectar un choque, la aplicación en el celular se conecta con el servidor mediante la Web Service basados en REST, luego estos Web Service se encargan de persistir la información recibida en la base de datos central. Luego, un servicio de Windows consumirá la información de la base de datos, cada X segundos. Mediante la técnica de Comet, el servidor envía mediante “push”, los datos de los accidentes a los distintos usuarios en las centrales y luego mediante la Web Service desarrollados con WCF (Windows Communication Foundation), se podrán consumir los distintos datos almacenados en el sistema de acuerdo al perfil de cada usuario.

Tan pronto como llega un mensaje UDS a la Web Service asociado, se guarda la información en una base de datos de SQL Server 2008. Luego, un SQL se encarga de leer registros ingresados en la tabla y enviarlos vía email a las distintas partes interesadas, por ejemplo entidades de emergencia. Se decidió no enviar un email desde un trigger a que se llame a la Web Service que lo haga, ya que se busca la menor sobrecarga posible sobre la tabla en el servidor de base de datos.

3.2.2.3. Jornada de Trabajo en 0 Iteración

- ✓ **Explicación del producto a desarrollar en base a los requerimientos definidos.**

Requerimientos de usuario:

Tabla 3. 1
Requerimientos definidos (Gestión de Aplicación)

Identificador	F01	Nombre	Ingresar a la aplicación para la verificación
Tipo	Funcional	Prioridad	Alta
Necesidad	Si	Verificable	Si
Descripción	El usuario desde la aplicación debe poder ingresar para poder ver con que velocidad está recorriendo la movilidad		

Notas. Elaboración Propia

Tabla 3. 2
Requerimientos definidos (Georreferenciación)

Identificador	F02	Nombre	Georreferenciar la ruta del hospital más cercano desde la posición 0 del usuario
Tipo	Funcional	Prioridad	Alta
Necesidad	Si	Verificable	Si
Descripción	El usuario luego de tener el incidente debe poder visualizar el mapa en tiempo real donde estará trazado la ruta del hospital más cercano esto con el punto de partida del usuario (posición 0).		

Notas: Elaboración Propia

Tabla 3. 3
Requerimientos definidos – Consultar Información (Datos de Conductor)

Identificador	F03	Nombre	Consultar en la BD el número de placa e información necesaria de la movilidad
Tipo	Funcional	Prioridad	Media
Necesidad	Si	Verificable	Si
Descripción	El usuario además de verificar en tiempo real de la ruta, también debe poder visualizar datos concretos del autotransporte.		

Notas. Elaboración Propia

3.2.2.4. Caso de Uso

a. Caso de Uso General

El siguiente diagrama de caso de uso general describe las actividades que realizan, también muestra a los actores pertenecientes a la institución.

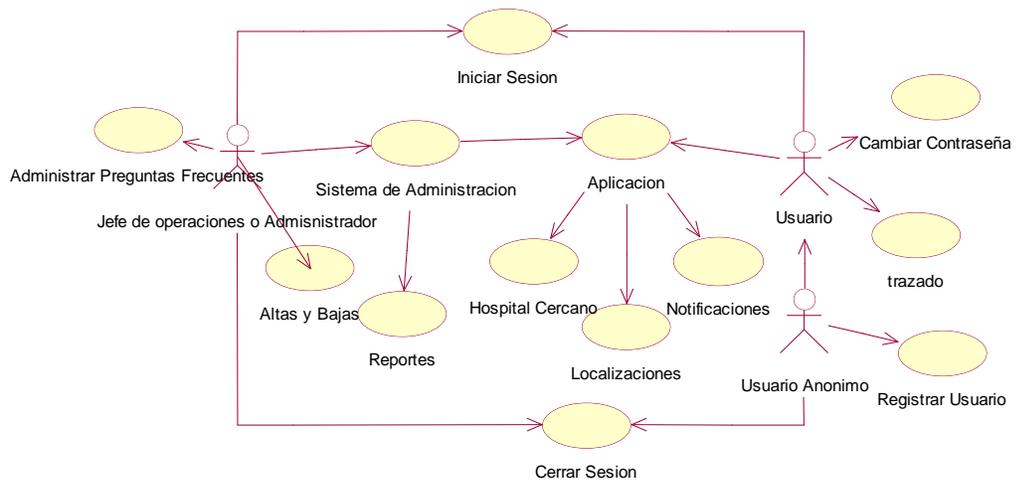


Figura 3. 3 Diagrama de Caso de Uso General [Elaboración Propia]

b. Caso de Uso - Usuarios

En el caso de uso se podrá ver todo lo que pueden hacer los usuarios después de haber iniciado sesión.

APLICACIÓN MOVIL

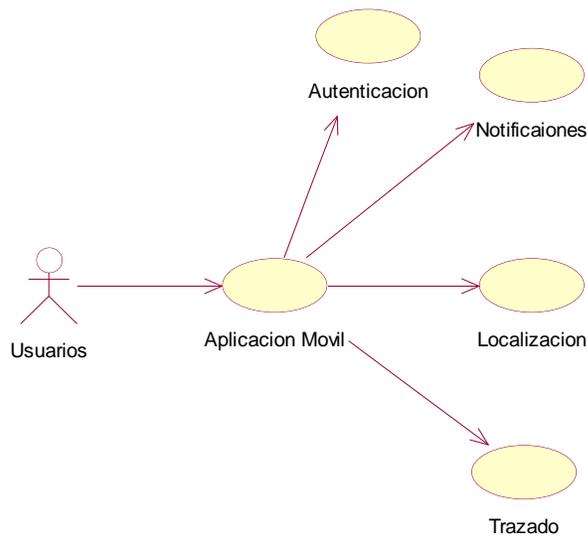


Figura 3. 4 Diagrama de Caso de Uso de Usuarios [Elaboración Propia]

Tabla 3. 4
Caso de uso de Usuarios

Caso de uso de Usuarios	
Actor	Descripción
Usuarios	El usuario debe instalar el apk de la aplicación en el dispositivo móvil, luego buscar el icono de Detección de Accidentes de Tránsito en el móvil e iniciar con la aplicación.

Notas: (Elaboración propia).

c. Caso del Uso - Jefe de Operaciones o Administrador

En el caso de uso se podrá ver todo lo que puede hacer el Administrador después de haber iniciado sesión en la red social o en el sistema de administración.

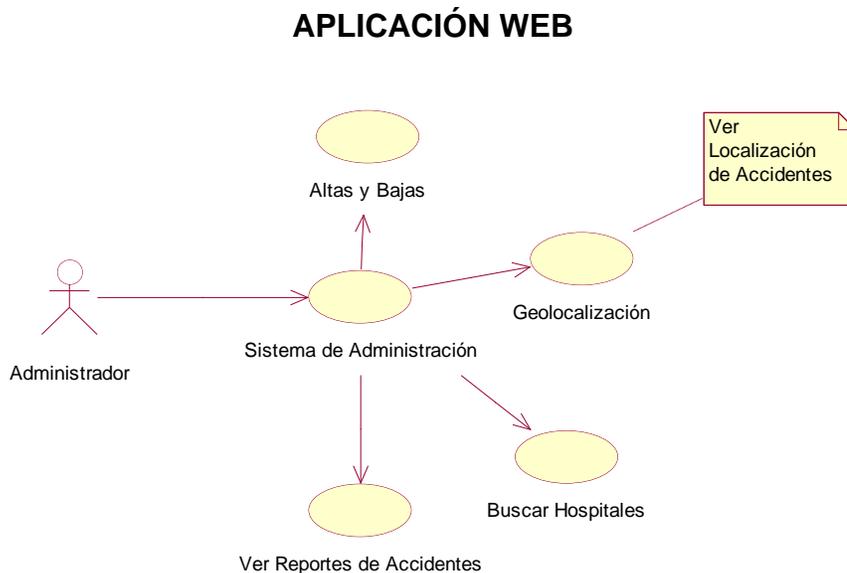


Figura 3. 5 Diagrama de Caso de Uso de Usuarios [Elaboración Propia]

Tabla 3. 5
Caso de Uso Administrador

Caso de uso del Administrador	
Actor	Descripción
Jefe de Operaciones o administrador	El administrador debe registrar a nuevos admin que no figuren en la base de datos Debe verificar los reportes mensuales de los accidentes

Notas: [Elaboración Propia]

d. Caso de Uso - Registro de Usuario

En el caso de uso se podrá ver los pasos para registrar a un socio, el administrador deberá iniciar sesión para entrar al sistema administración.

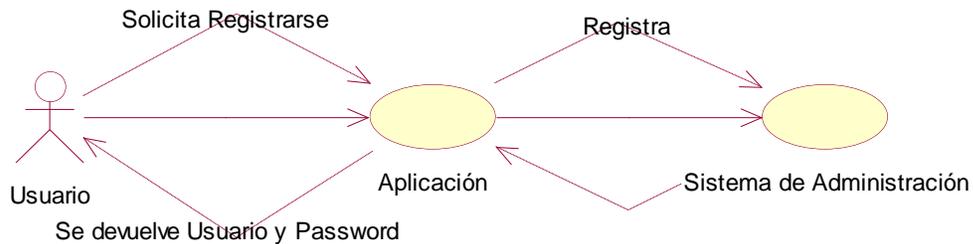


Figura 3. 6 Diagrama de Caso de Uso de Registro de Socio [Elaboración Propia]

Tabla 3. 6
Caso de Uso de Registro de Usuario

Actores	Descripción
Administrador	El Jefe de Operaciones o administrador y el usuario es la única persona que puede entrar al sistema de administración en cual puede registrar la descripción de vehículos, el cual se le asigna un usuario y password para que entre a la app de la institución.

Notas: [Elaboración Propia]

3.2.2.5. Diagrama de Actividades

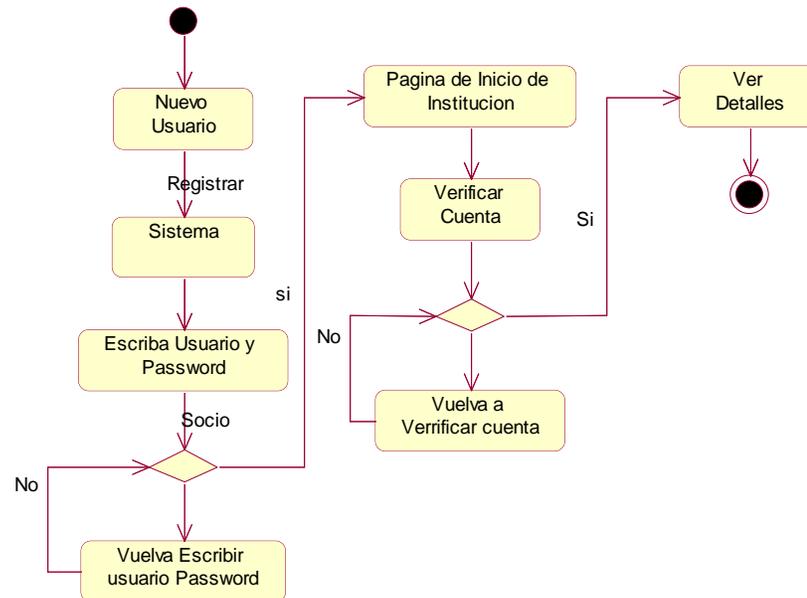


Figura 3. 7 Diagrama de Actividades [Diagrama de Actividades]

El diagrama representa la forma en que un usuario puede entrar a App de la institución y los pasos que debe seguir el usuario si es nuevo, debe ser registrado por el administrador o encargado si no el mismo usuario, y registrarse toda la descripción de sus vehículos.

3.2.2.6. Diagrama de Secuencia

a. Diagrama de secuencia del celular hasta que se detecta un choque

El diagrama nos refleja de cómo funciona internamente la aplicación hasta que se detecta un accidente.

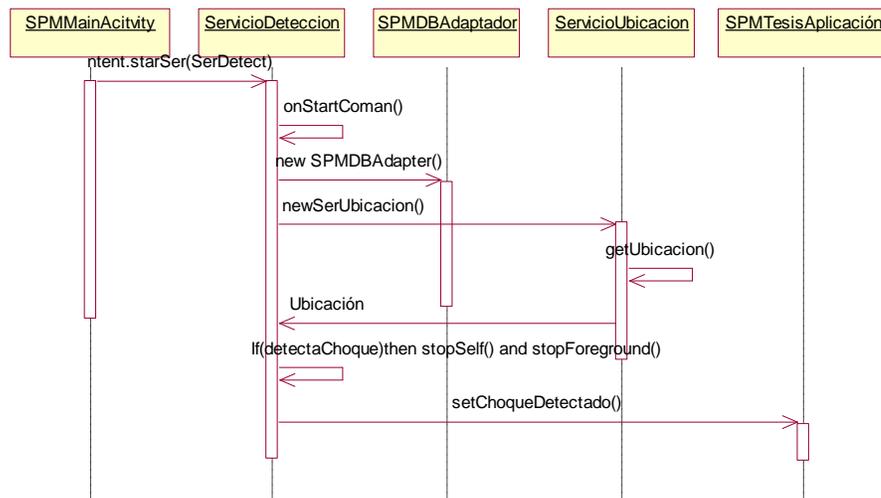


Figura 3. 8 Diagrama de Secuencia del Celular hasta que se detecta un choque [Elaboración Propia]

✓ **Diagrama de colaboración del celular hasta que se detecta un choque**

El diagrama de colaboración es para ver el funcionamiento del sistema, el cual los socios deben verificar su cuenta e enviar su localización.

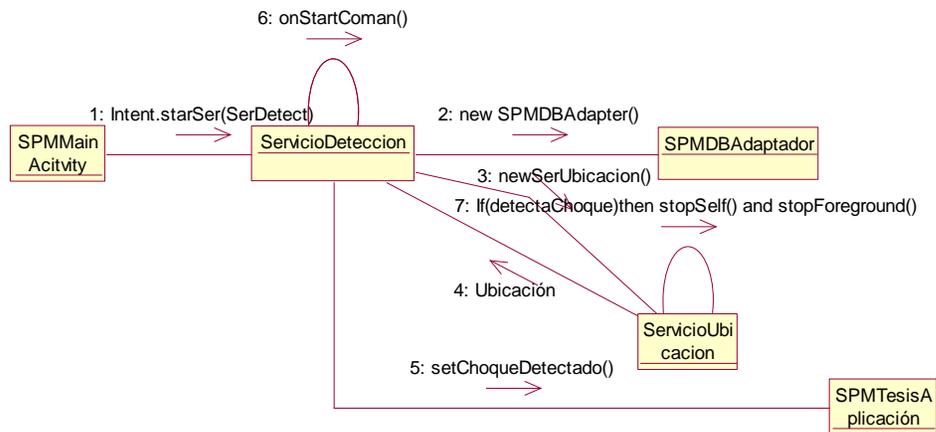


Figura 3. 9 Diagrama de Colaboración [Elaboración Propia]

b. Diagrama de secuencia del celular luego de detectar un choque

El diagrama nos refleja de cómo funciona internamente la aplicación después de haber ocurrido el accidente.

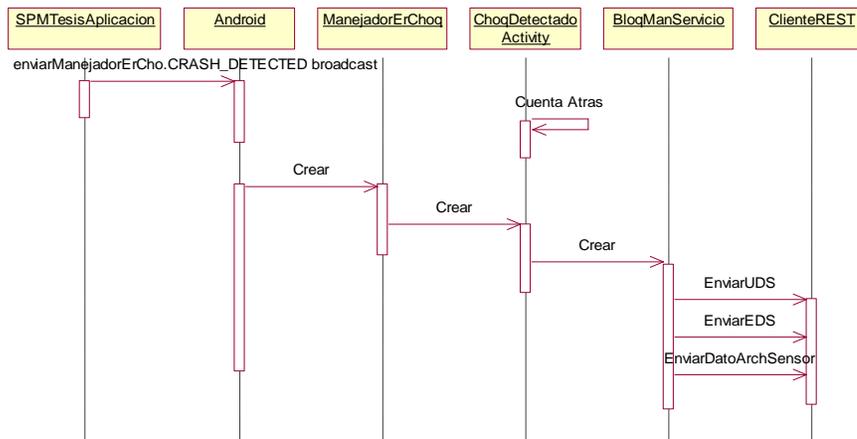


Figura 3. 10 Diagrama de secuencia del celular luego de detectarse un choque [Elaboración Propia]

✓ **Diagrama de colaboración del celular luego de detectar un choque**

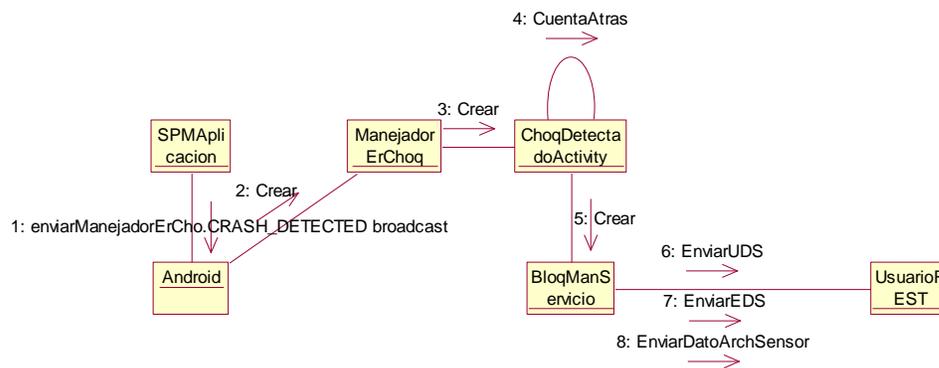


Figura 3. 11 Diagrama de Colaboración del Administrador [Elaboración Propia]

3.2.2.7. Diagrama de Clases

a. Diagrama de Clases para Móviles

El siguiente diagrama de clase será utilizado en los móviles ya que estos tendrán otra base de dato en su propio gestor que es SQLite.

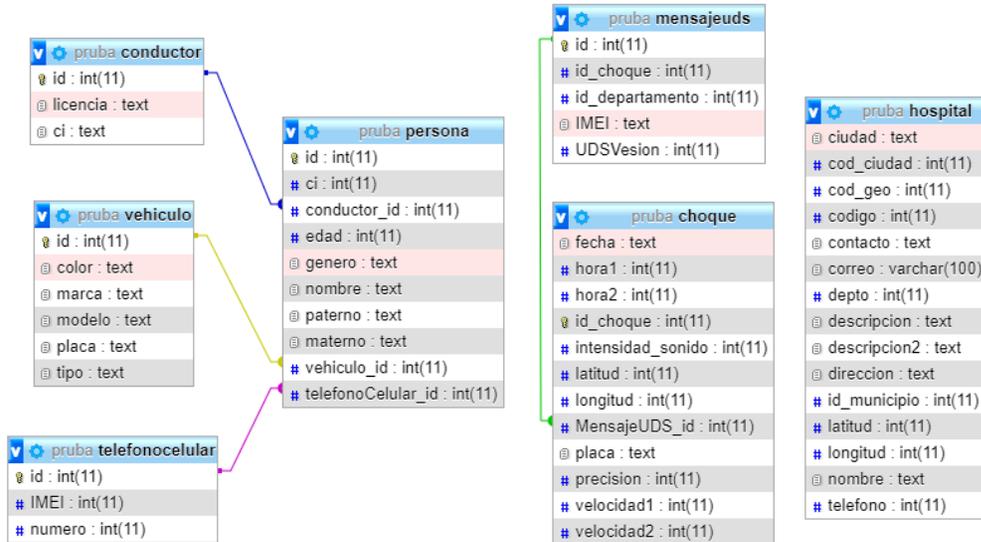


Figura 3. 12 Diagrama de Clases Móviles [Elaboración Propia]

3.2.2.8. Elaboración de prototipos mediana de fidelidad (Dia de publicación en iteración 0)

a. Diagrama de presentación de página web

Los diagramas de presentación, que se describen a continuación muestran cómo está estructurada el sistema. A continuación, se observa el diagrama de inicio del sistema.

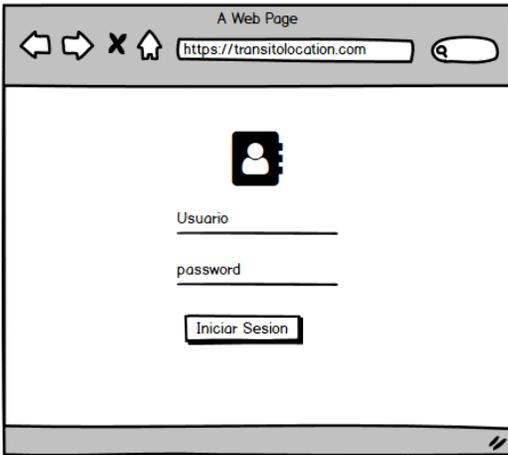


Figura 3. 13 Diagrama de Presentación Login [Elaboración Propia]

b. Diagrama de presentación entorno home

El diagrama de presentación de la red social, permite visualizar las interfaces que tienen la red social para los socios.

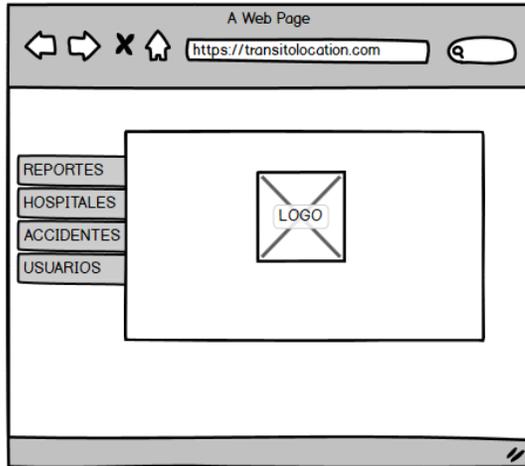


Figura 3. 14 Diagrama de Presentación Entorno Home [Elaboración Propia]

c. Gestión Reportes (Administrador)

El diagrama de presentación del sistema de administración de usuarios, permite visualizar las interfaces que implica ver los reportes de los accidentes y las direcciones de los hospitales.

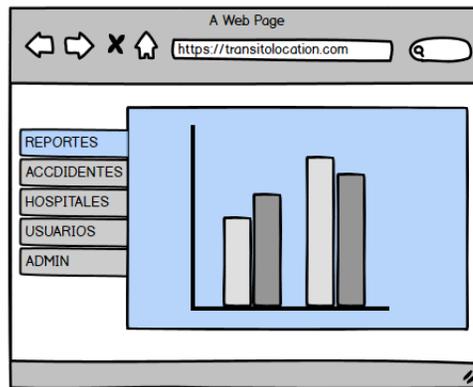


Figura 3. 15 Diagrama de Presentación Inicio [Elaboración Propia]

d. Diagrama de presentación de la Aplicación Entorno Ubicación del Accidentes

El diagrama de presentación de aplicación entorno ubicación del accidente, permite visualizar el recorrido, ubicación y ver hospitales cercanos.

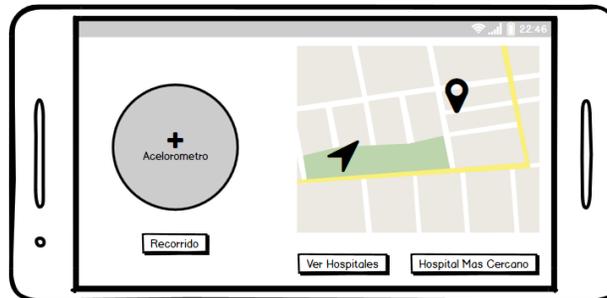


Figura 3. 16 Diagrama de Presentación Entorno Geocalización [Elaboración Propia]

e. Diagrama de Presentación Entorno Registros de usuarios y detalles del Vehículo

Se podrá registrar el usuario con sus datos personales y también registrará todos los detalles de sus vehículos

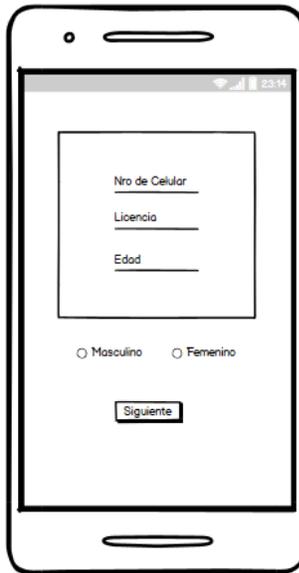


Figura 3. 17 Diagrama de Presentación, Registro de usuario y vehículo [Elaboración Propia]

3.2.3. Fase 3: Producción

3.2.3.1. Implementación

En esta parte hablaremos de: Implementar la aplicación Cliente/Servidor, Implementar la aplicación para el almacenamiento y envío de mensajes de alarma desde el Smartphone al servidor, también el envío de notificación de un accidente a todos los Smartphone que cuentan la aplicación Servidor/Cliente



Figura 3. 18 Diagrama de flujo de Implementación [Elaboración Propia]

En el desarrollo de la aplicación móvil, se obtendrán los datos en el siguiente orden: primero se calculará la variación de la velocidad (aceleración), y seguidamente se calculará la intensidad del sonido con lo que se produjo el choque; luego analizamos si estos dos valores cumplen con las condiciones para un accidente: (Ver Figura 3.8)

- ✓ Si cumple enviamos la alerta al servidor local, luego notificamos sobre el accidente a todos los usuarios.
- ✓ No cumple, no hace nada.

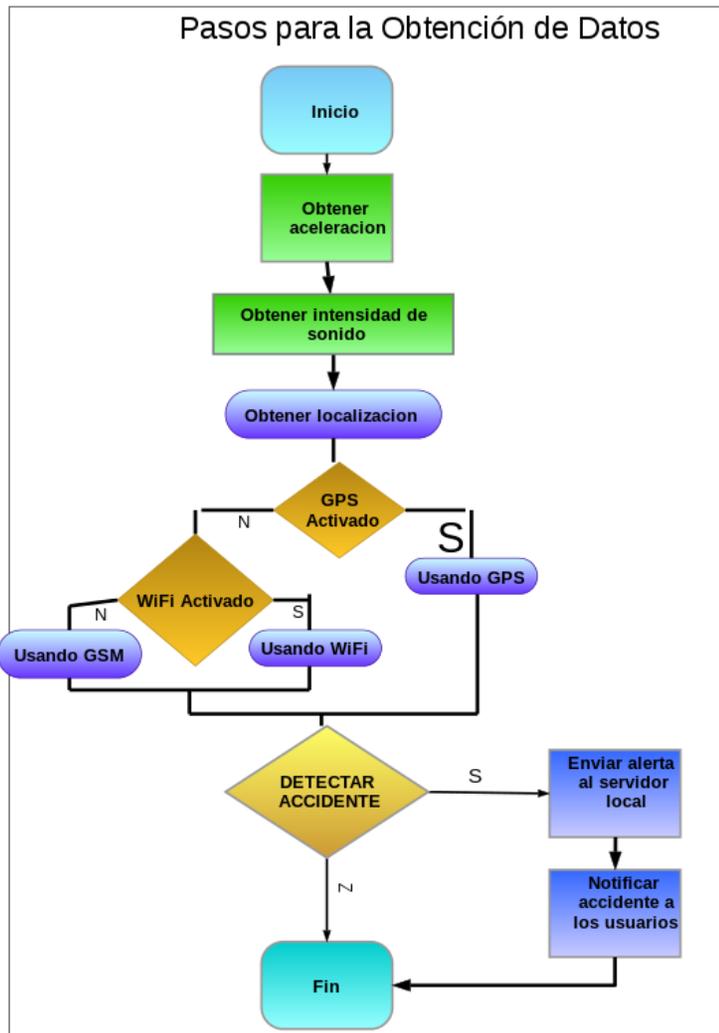


Figura 3. 19 Pasos para la obtención de datos [Elaboración Propia]

En el siguiente apartado se detalla cada uno de los pasos ya mencionados anteriormente:

Para obtener la variación de la velocidad (aceleración), primero activamos los permisos necesarios en AndroidManifest para el sensor acelerómetro, y almacenamos en la memoria interna la variable mSensor (la variación de la velocidad). (Ver Figura 3.20)

```

17         if (mSensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY) != null){
18             List<Sensor> gravSensors = mSensorManager.getSensorList(Sensor.TYPE_GRAVITY);
19             for(int i=0; i<gravSensors.size(); i++) {
20                 if ((gravSensors.get(i).getVendor().contains("Google Inc. ")) &&
21                     (gravSensors.get(i).getVersion() == 3)){
22                     Sensor = gravSensors.get(i);
23                 }
24             }

```

Figura 3. 20 Fragmento de código, para el uso del sensor acelerómetro [Elaboración Propia]

Para ver el código completo sobre el uso del acelerómetro se desglosa.

- ✓ La obtención de la localización de los teléfonos inteligentes, primero pregunta si esta activa GPS, si este no está entonces lo segundo que pregunta es si tiene acceso a Internet o red de datos, pero si no está disponible ninguno de los anteriores, entonces la obtención de la localización será utilizando GSM (Triangulación de Torres de las Telefonías Móviles). (Ver Figura 3.21).

```

64         if (isNetworkEnabled) {
65             locationManager.requestLocationUpdates(
66                 locationManager.NETWORK_PROVIDER,
67                 MIN_TIME_BW_UPDATES,
68                 MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
69             Log.d("Network", "Network");
70             if (locationManager != null) {
71                 location = locationManager
72                     .getLastKnownLocation(locationManager.NETWORK_PROVIDER);
73                 if (location != null) {
74                     latitude = location.getLatitude();
75                     longitude = location.getLongitude();
76                 }

```

Figura 3. 21 Fragmento de código para obtener la localización [Elaboración Propia]

Detalle sobre la obtención de localización.

- ✓ Para obtener la intensidad del sonido con que se produjo el accidente, primero activamos el micrófono del teléfono inteligente, luego utilizamos los recursos de Android que nos permite la obtención de la intensidad del sonido en decibeles (dB). Además, para obtener el sonido en dB utilizamos la Formula 1, el cual fue obtenido en el Capítulo II. (Ver Figura 3.22).

```

784 public void startRecorder(){
785     if(mRecorder == null){
786         mRecorder = new MediaRecorder();
787         mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
788         mRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
789         mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
790         mRecorder.setOutputFile("/dev/null");
791         try{
792             mRecorder.prepare();
793         }

```

Figura 3. 22 Fragmento de código para obtener la intensidad de sonido [Elaboración Propia]

- ✓ Visualización del código completo sobre la obtención de la intensidad del sonido
- ✓ Un factor fundamental que indica que hubo un choque es un cambio brusco en la aceleración y/o desaceleración, durante un breve período de tiempo. Así, podemos obtener (a grandes rasgos), la fuerza G aplicada en una frenada, mediante la siguiente fórmula:

$$a = \frac{V_f^2 - V_o^2}{2 * (X_f - X_o)}$$

Fórmula 2. Movimiento rectilíneo

De la Fórmula 2, podemos llegar a:

$$d = \frac{V^2}{2 * c}$$

Fórmula 3: La desaceleración.

Siendo:

d = Desaceleración producida

v = velocidad

c = la distancia de frenada recorrida

Se puede ver que la desaceleración producida es proporcional al cuadrado de la velocidad (v) (es la diferencia de la velocidad, pero en este caso la final es 0) e

inversamente proporcional al doble de la distancia de frenada recorrida (c). (Ver Fórmula 3)

Y finalmente para obtener la fuerza G con la que produjo el choque, dividimos la Formula 3 entre el valor de la gravedad 9.81 m/s².

$$Fuerza = \frac{Formula\ 3}{9.81 \frac{m}{s^2}}$$

Fórmula 4. El impacto con lo que se produjo el choque.

Enviando al servidor local el mensaje UDS, después de la detección del accidente, los datos a ser enviados están en el formato JSON, este método es una tarea asíncrona. El archivo Accidentes.php realiza una simple inserción de registros, en este caso de accidentes.

```
@Override
protected Boolean doInBackground(Void... params) {
    // TODO Auto-generated method stub

    ArrayList<NameValuePair> postParameters = new ArrayList<NameValuePair>();
    postParameters.add(new BasicNameValuePair("fechaAccidente", Utilitarios.dateToday() + " " + Utilitarios.currentlyTime()));
    postParameters.add(new BasicNameValuePair("velocidad", velocidad));
    postParameters.add(new BasicNameValuePair("latitud", latitud));
    postParameters.add(new BasicNameValuePair("longitud", longitud));
    postParameters.add(new BasicNameValuePair("presicion", presicion));
    postParameters.add(new BasicNameValuePair("numeroTelefono", infoT.getNumero()));
    postParameters.add(new BasicNameValuePair("ci", infoCli.getCi()));
    postParameters.add(new BasicNameValuePair("intensidadSonido", intensidadSonido));

    try{
        CustomHttpClient.executeHttpPost("http://192.168.43.119/server/Accidentes.php", postParameters);
    }
    catch(Exception e){
        Log.e("Fallo coneccion", "No se pudo enviar al servidor el mensaje de alarma " + e.getMessage());
    }

    return true;
}
```

Figura 3. 23 Fragmento de código para el envío de datos de Frontend a Backend [Elaboración Propia]

- ✓ Para alertar a todos los usuarios que cuentan con la aplicación móvil, los notificamos. En el siguiente apartado desglosaremos la creación del proyecto y la activación de los servicios de Google (Crear el proyecto en: <https://code.google.com/apis/console>).

```

79 notificationIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP
80 | Intent.FLAG_ACTIVITY_SINGLE_TOP);
81 PendingIntent intent = PendingIntent.getActivity(context, 0,
82 notificationIntent, 0);
83 notification.setLatestEventInfo(context, title, message, intent);
84 notification.flags |= Notification.FLAG_AUTO_CANCEL;
85 notificationManager.notify(0, notification);
86 notification.number |= ++count;
87 notification.defaults |= Notification.DEFAULT_LIGHTS;
88 notification.defaults |= Notification.DEFAULT_ALL;
89 }

```

Figura 3. 24 Fragmento de código para notificaciones lado Frontend [Elaboración Propia]

La notificación enviara a todos los usuarios registrados en el servidor de incidentes local, enviara los datos de la persona que sufrió el accidente junto con la localización. Para hacer el uso de los servicio de Google Cloud Messaging, generamos la clave del servidor

Para ver el código para notificaciones en el lado Frontend.

```

2  define("GOOGLE_API_KEY", "AIzaSyBYPYVdPW4yH0vaASiCvTmuN4UL648a12w");
3  define("GOOGLE_GCM_URL", "https://android.googleapis.com/gcm/send");
4
5  function send_gcm_notify($reg_id, $message) {
6
7      $fields = array(
8          'registration_ids' => array( $reg_id ),
9          'data'              => array( "message" => $message ),
10     );
11
12     $headers = array(
13         'Authorization: key=' . GOOGLE_API_KEY,
14         'Content-Type: application/json'
15     );
16
17     $ch = curl_init();
18     curl_setopt($ch, CURLOPT_URL, GOOGLE_GCM_URL);
19     curl_setopt($ch, CURLOPT_POST, true);
20     curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
21     curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
22     curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
23     curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($fields));
24

```

Figura 3. 25 Fragmento de código para notificaciones lado Backend [Elaboración Propia]

3.2.3.2. Creando el Proyecto en Google Api

- a. **Crear Nuevo Proyecto:** Lo primero que haremos será crear un proyecto en Google API, para ello accedemos a la consola de desarrolladores de Google con nuestro usuario.



Figura 3. 26 Creando Google Api [Elaboración Propia]

En la sección "Projects" pulsamos sobre el botón "Create Project", le asignamos un nombre y ya tendremos nuestro proyecto creado. Nos aparecerá una pantalla donde podremos ver información relevante a dicho proyecto.

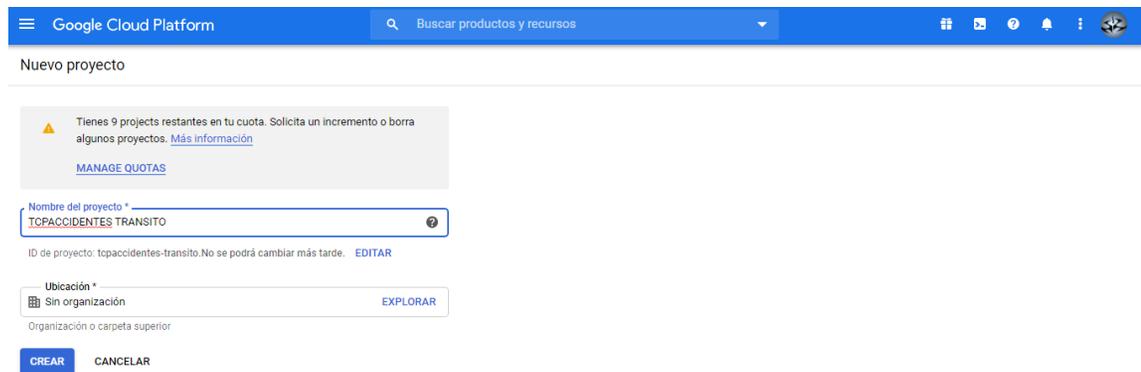


Figura 3. 27 Creando el proyecto [Elaboración Propia]

b. Crear Credenciales

Para acceder a las APIs se requiere credenciales. Aquí es donde las crearás. Simplemente haz click en el botón azul titulado «Crear credenciales».

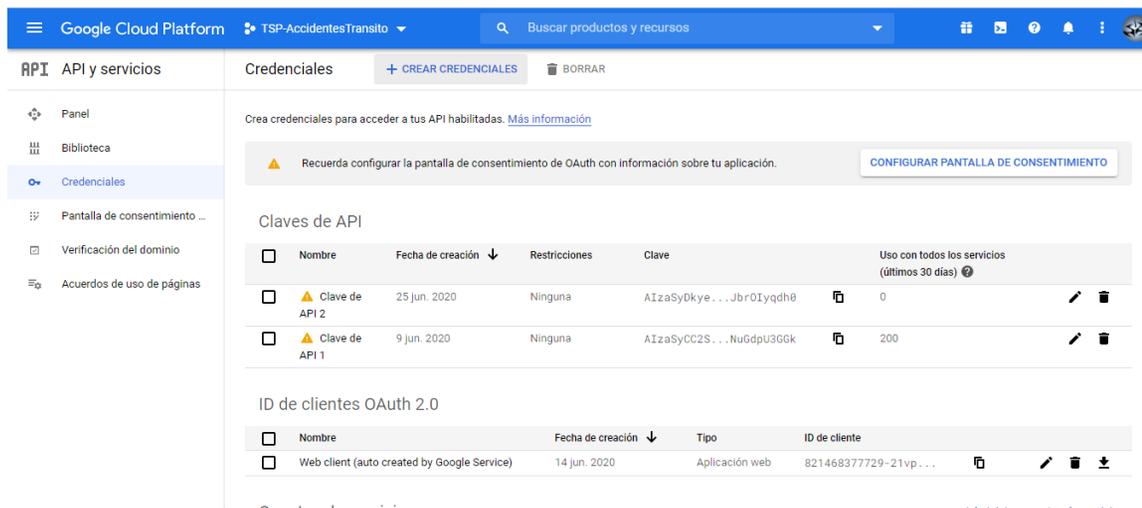


Figura 3. 28 Creando Credenciales [Elaboración Propia]

Luego selecciona la primera opción, nombrada: «Clave de API».

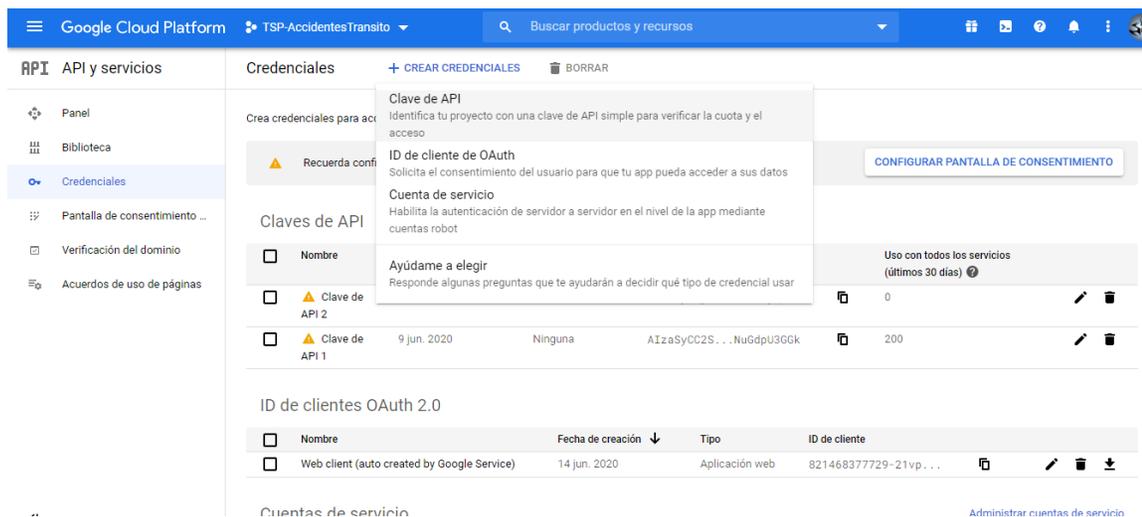


Figura 3. 29 Creando Api Key [Elaboración Propia]

c. Obteniendo la clave de acceso API

Por último, necesitaremos tener una clave de acceso a nuestro API para que podamos enviar peticiones a GCM y que éste las sirva las notificaciones al dispositivo o dispositivos destino. Esta clave de acceso será utilizada por nuestro servicio.



Figura 3. 30 Obteniendo la clave de acceso al API [Elaboración Propia]

d. Activar Google Maps API Key

Ahora que se tiene las credenciales, el siguiente paso para obtener una clave de API de Google Maps es seleccionar la API que deseas utilizar. Para hacer esto, simplemente accede a la Biblioteca desde el menú al lado izquierdo de la página. Para que esto sea activado en Maps Android SDK

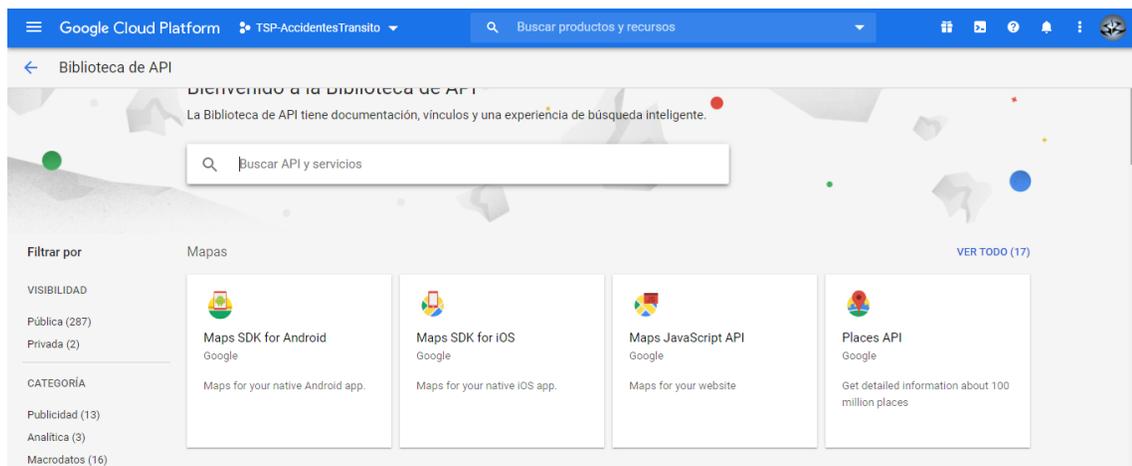


Figura 3. 31 Activando Google Maps API Key [Elaboracion Propia]

Seguidamente en el campo de búsqueda para la API que deseas, procedemos a verificar Maps SDK for Android y luego presionar el botón habilitar.

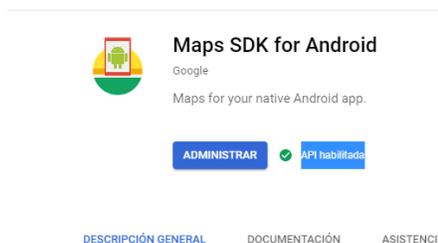


Figura 3. 32 Habilitando Maps SDK for Android [Elaboración Propia]

3.2.4. Fase 4: Estabilización

Para esta fase la tarea principal fue comprobar la sincronización entre el servidor y la aplicación buscando solucionar errores de carga y diferencia de datos, además se realizaron cambios entre pantallas de la aplicación para un mejor funcionamiento de librerías actuales como son material design y sus librerías de support.

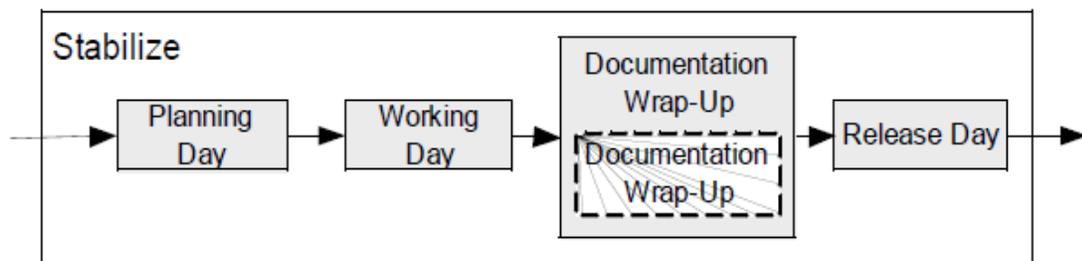


Figura 3. 33 Esquema de la fase de estabilización [wpcentral.com, 2006]

3.2.5. Fase 5: Pruebas

En este apartado se detalla el plan de pruebas a seguir para la evaluación final del aplicativo móvil, se identifican los tipos de pruebas realizados con su respectiva explicación, se detalla los resultados de dichas pruebas y se indican los pasos realizados para la publicación e instalación de la aplicación móvil.

3.2.5.1. Plan de Pruebas

Este documento sirve para detallar que elementos son los que se van a probar, como realizar las pruebas y confirmar que los métodos utilizados cumplan con las soluciones a cada uno de los requerimientos solicitados por el cliente. Se incluirán los tipos de pruebas a realizar, que herramientas se ocuparon durante el desarrollo de las pruebas y su ambiente, conjuntamente con un cronograma.

Tabla 3. 7
Detalle del plan de pruebas

TIPO DE PRUEBA	DESCRIPCIÓN DE PRUEBA	HERRAMIENTA UTILIZADA
Pruebas Funcionales	<p>Las pruebas funcionales son un tipo de prueba de caja negra que basa sus casos de prueba en las especificaciones del componente de software bajo prueba. Las pruebas funcionales generalmente describen lo que hace el sistema.</p> <p>Proceso a ejecutar Elaborar los casos de prueba Ejecutar los casos de prueba Registrar los errores hallados según los casos de prueba</p>	<p>Para la realización de estas pruebas no se requirió ninguna herramienta específica dado que la ejecución manual de este tipo de prueba sobre cada uno de los casos de uso fue suficiente.</p>
Pruebas no Funcionales	<p>Las pruebas no funcionales se concentran en los aspectos del software que no se encuentran relacionados a una función específica o usuario, se encarga de proveer una buena experiencia de usuario, algunos aspectos medidos podrían ser el rendimiento, compatibilidad, seguridad, fiabilidad, usabilidad, conformidad.</p> <p>Proceso a ejecutar Elaborar los casos de Prueba Ejecutar los casos de prueba Registrar los errores hallados según los casos de prueba</p>	<p>Para la realización de estas pruebas se hicieron uso de las herramientas propias del IDE (Integrated Development Environment, Entorno de Desarrollo Integrado) Android Studio como Android Virtual Device (AVD Manager) para la emulación de casos que requieran características específicas.</p>

Notas: Byron Jiménez [2016]

Tabla 3. 8

Cronograma estimado para la realización de las pruebas

TIPO DE PRUEBAS	DURACIÓN ESTIMADA (HORAS)
Funcionales	20
No Funcionales	20

Notas: Byron Jiménez

3.2.5.2. Ambiente de Pruebas

Para el momento de la realización de las pruebas no funcionales y funcionales se optó por el uso de los siguientes dispositivos móviles y equipo de computadora como ambiente de pruebas:

Dispositivos Celulares

Zony Xperia Z2

- **Pantalla** IPS LCD de 5,2 pulgadas
- **Resolución** 1920 x 1080 píxeles (424 ppp)
- **Procesador** Qualcomm Snapdragon 801 quad-core a 2.3 GHz
- **Procesador grafico** Adreno 330
- **Ram** 3 GB
- **Momoria** 16 GB, con microSD
- **Versión software** Android 4.4.2
- **Conectividad** LTE, NFC

Equipo de Computadora

Laptop Lenovo

- Microprocesador i5 Ghz
- Memoria Ram 4Gb
- Disco Duro 500 Gb
- Software Windows 8.1

3.3. MÉTRICAS DE CALIDAD (MODELO - QSOS)

Se describe el flujo de trabajo así para evaluar los prototipos además de sus componentes, la valoración se llevará con dos tipos de usuario: Administrador (afiliado del área administrativa)

3.3.1. Tarjeta de identificación de Software “Localización de Accidentes de Tránsito”

Tabla 3. 9

Pantalla Principal Plataforma Web

Información General			
Nombre del Software:	Sistema Detección de Accidentes de Tránsito	Fecha de creación:	23/05/2020
Tipo de Software:	Aplicación de escritorio	Autor:	Gustavo Paco Machaca
Fecha de elaboración de la tarjeta:	12/06/2020		
Servicios existentes			
Documentación:	✓		
Manual nivel Admin.	✓		
Aspectos técnicos y funcionales			
Funcionalidades	<ul style="list-style-type: none">• Registro de Nuevo Usuario• Geolocalización de hospitales• Detección de Accidentes• Notificación de Accidentes		
Tecnologías de implementación	<ul style="list-style-type: none">• Desarrollado en IDE Eclipse• Librerías Barcode.jar		

Notas: [Elaboración Propia]

3.3.2. Evaluaciones los criterios de valoración (Sistema: Detección de Accidentes de Tránsito)

Las métricas capturadas se clasificaron según la gravedad de la métrica y luego se agrupan en tres dimensiones principales (calidad del sistema, calidad de la información y calidad del servicio).

Tabla 3. 10*Evaluaciones los criterios de valoración*

Dimensión	Criterios	Características	Descripciones (pregunta relacionada)
Calidad del sistema	Fiabilidad	Madurez	¿El software es nuevo en la Institución Transito de La Paz?
	Usabilidad	Aprendizaje	¿Qué tan fácil de aprender o entender el software sin usar el manual del usuario?
		Operabilidad	¿El software es fácil de operar?
		Accesibilidad	¿Es fácil acceder a este software sin otro software o complemento de terceros?
		Interfaz de usuario estética	¿La interfaz de usuario es adecuada con su funcionalidad de software?
	Eficiencia en el desempeño	Comportamiento de tiempo	¿Es este software fácil de instalar / configurar y operar en poco tiempo?
	Funcionalidad	Integridad funcional	¿El software cumple con las expectativas y requisitos del usuario?
Adecuación funcional		¿Funciona el software adecuadamente?	
Calidad de servicio	Tangible	Documentación	¿Manual técnico y de usuario?
	Empatía	Comunicación	¿La comunidad reconoce sus problemas y ayuda a resolverlos?

Notas: [Elaboración Propia]

- ✓ **Valoración:** - Se realizó las pruebas con el responsable de soporte técnica de área de sistemas en el departamento de transito de La Paz – El Alto los cuales realizaron las verificaciones:

0 = no importante,

1 = importante,

2 = muy importante

Tabla 3. 11
Evaluaciones los criterios de valoración

Dimensión	Criterios	Características	Escala			Promedio
			0	1	2	
Calidad del sistema	Fiabilidad	Madurez	2	2	2	2
	Usabilidad	Aprendizaje	2	2	2	2
		Operabilidad	2	2	2	2
		Accesibilidad	2	1	2	1.66
		Interfaz de usuario estética	2	2	2	2
	Eficiencia en el desempeño	Comportamiento de tiempo	2	2	2	2
	Funcionalidad	Integridad funcional	1	2	2	1.66
		Adecuación funcional	2	2	2	2
Calidad de servicio	Tangible	Documentación	2	2	2	2
	Empatía	Comunicación	2	2	2	2

Notas: [Elaboración Propia]

✓ . **Análisis de datos y resultados.**

Los datos obtenidos se analizaron para investigar la puntuación de importancia de cada característica identificada.

✓ **Calidad de servicio y otra dimensión.**

Se incluyen una característica para el criterio de dimensión de calidad del servicio, en base al resultado se demuestra que el 100% de la valorización es aceptada por el encargado.

3.4. COSTO DE COCOMO

El factor de conversión a KLDC se presenta en la siguiente figura que muestra el nivel al que pertenece la herramienta PHP utilizada en el sistema.

Tabla 3. 12
Conversion de Punto de Fusion a KLDC

LENGUAJE	NIVEL FACTOR	LDC/PF
HTML	2.5	128
JAVA	6	53
PHP	11.00	29

Notas: (T-Chambi, 2007).

LDC= PD obtenido* FactorLDC/PF

LDC=308*29

LDC=8932

KLDC= **8932** / 1000

Miles de líneas de código → KLDC=8.93

Las fórmulas básicas de esfuerzo, tiempo y personal que se requiere, son las siguientes:

$$E = a_b(KLDC)^{bb} \left[\frac{\text{Personas}}{\text{mes}} \right]$$

$$D = c_b E^{db} [\text{meses}]$$

Donde:

E: Esfuerzo aplicado en personas por mes.

D: tiempo de desarrollo en meses cronológico.

KLDC: Número estimado de líneas de código distribuidas (en miles).

Tabla 3. 13*Coefficiente a_b y c_b y los exponentes b_b y d_b*

PROYECTO DE SOFTWARE	a_b	b_b	c_b	d_b
<i>Orgánico</i>	2.4	1.05	2.5	0.38
<i>Semi – acoplado</i>	3.0	1.12	2.5	0.35
<i>Empotrado</i>	3.6	1.20	2.5	0.30

Notas: (PRESSMAN, 2005).

Los proyectos de software son semi-acoplados ya que son proyectos intermedios en tamaño y complejidad, esto hace que satisfagan requisitos poco o medio rígido, tal es el caso del software desarrollado.

- **Cálculo de esfuerzo:**

$$E = 3.0 * (8.93)^{1.12}$$

$$E = 34.84 \text{ [personas/mes]}$$

- **Cálculo de tiempo:**

$$D = 2.5 * (34.84)^{0.35}$$

$$D = 8.66 \text{ [mes]}$$

Tiempo → 8 meses

- **Cálculo de personal:**

Numero de programadores

$$E/D = 34.84 / 8.66$$

Personas 4.02

Personal → 4 personas

- **Productividad:**

$$PR = LDC/E$$

$$PR = 8932/34.84$$

$$PR = 256.37 \text{ [LDC/persona mes]}$$

- **Costo total del proyecto:**

$$\text{Pago por mes por persona} = 2792 \text{ Bs o } 400 \$$$

$$\text{Costo Mes} = 4 * 2792 \text{ Bs}$$

$$\text{Costo Mes} \rightarrow 11168 \text{ Bs}$$

- **Costo total del proyecto**

$$\text{Costo Total} = 11168 * 8 = 89344 \text{ bolivianos}$$

Por tanto, se concluye que se requieren 4 desarrolladores y un trabajo aproximado de 8 meses y el costo total del proyecto será de 90680 bolivianos.

3.5. METRICAS DE CALIDAD DE SOFTWARE (ISO - 9126)

El desarrollar un software de calidad es el objetivo de todo desarrollador, por tanto, se le dedica muchos esfuerzos, pero también cabe mencionar que no se logra la perfección en el producto de software, pero se debe tomar en cuenta que todo software debe cumplir y/o superar las expectativas del cliente. Si cumple esta característica tendrá la dominación de un software de alta calidad.

La norma que se aplicó en la Aplicación Web y Android para institución Dirección Departamental de Transito de La Paz, fue la norma ISO 9126, la cual permitió determinar la calidad del sistema a través de la aplicación de métricas de calidad para medir las características del software, como ser:

3.5.1. Usabilidad

La usabilidad consiste de un conjunto de atributos que permite evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema, es decir realizar

una serie de preguntas que permiten ver cuán sencillo, fácil de aprender y manejar es para los usuarios. En la siguiente tabla se observa estos criterios en niveles de porcentajes a los que llegó el sistema en cuanto a su comprensibilidad, para el usuario, y posteriormente se da el porcentaje final de usabilidad del sistema.

En La siguiente tabla se halla el nivel de usabilidad del sistema:

Tabla 3. 14

Tabla de Usabilidad del Sistema

N°	PREGUNTA	ESTAS		PORCENTAJE %
		SI	NO	
1.	¿El sistema es comprensible?	9	0	1.00
2.	¿El sistema es agradable a la vista?	8	1	0.91
3.	¿El sistema hace lo que dice que hace?	8	0	1.00
4.	¿Las Respuestas del sistema son satisfactorias?	8	1	0.91
5.	¿Es fácil aprender a manejar el sistema?	7	1	0.91
6.	¿El sistema satisface las necesidades que usted requiere?	7	1	0.88

Notas: (S. Pressman, 2007).

$$\text{USUBILIDAD} = \frac{1.00+0.91+1.00+0.91+0.91+0.88}{6}$$

$$\text{USUBILIDAD} = 0.93$$

$$\text{USUBILIDAD} = 93\%$$

De acuerdo a los datos obtenidos en la tabla de usabilidad, se concluye que el sistema tiene una usabilidad del 0.93, es decir del 93%. Entonces nos indican que el sistema es fácil de manejar y comprensible.

3.5.2. Funcionalidad

La funcionalidad examina si el sistema satisface los requisitos funcionales esperados. El objetivo es revelar problemas y errores en lo que concierne a la funcionalidad del sistema y su conformidad al comportamiento, expresado o deseado por el usuario.

En la siguiente tabla se calcula el punto función, los cuales miden el software desde una perspectiva del usuario, dejando de lado los detalles de codificación.

- a. **Técnica Punto función:** Esta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación y la metodología utilizada.

Para el cálculo de Punto Función se toma en cuenta 5 características de dominio de información.

- **Número de entradas de usuario:** Se refiere a cada entrada que proporciona datos al sistema.
- **Número de salidas de usuario:** Se refiere a cada salida que proporciona el sistema al usuario, entre estos están: informes, pantallas, mensajes de errores, etc.
- **Número de peticiones de usuario:** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta de software en forma de salidas interactivas.
- **Número de Archivos:** Se cuenta archivos maestro lógico, estos pueden ser: grupo lógico de datos, o un archivo independiente.
- **Número de interfaces externas:** se cuenta las interfaces legibles por la máquina que se utilizan para transmitir información a otro sistema.

En la siguiente tabla se realiza el cálculo punto función hallando la suma de estas características y el factor de complejidad, el resultado es el punto función no ajustado.

Tabla 3. 15*Cálculo de Funcionalidad Según el Punto Función.*

PARÁMETROS DE ENTRADA	CUENTA	FACTOR DE COMPLEJIDAD	TOTALES
Número de entradas de usuario	12	* 4	48
Número de salidas de usuario	14	* 5	70
Número de peticiones de usuario	14	* 4	56
Número de archivos	12	* 10	120
Número de interfaces externas	2	* 7	14
Cuenta Total			308

Notas: Pressman, 2007).

Cálculo del punto función ajustada. Los valores de Funcionalidad, se obtiene de los resultados de la siguiente tabla, bajo las ponderaciones descritos en la escala.

Tabla 3. 16*Tabla de Ajuste de Complejidad*

IMPORTANCIA		1	2	3	4	5
ESCALAS	Sin importancia	Incremental	Moderado	Medio	Significativo	Esencial
1.- ¿Requiere el sistema copias de seguridad y de recuperación fiable?						X
2.- ¿Se requiere comunicación de datos?						X
3. ¿Existe funciones de procesos distribuidos?				X		
4. ¿Es crítico el rendimiento?				X		

5.- ¿El sistema web será ejecutado el SO? Actual?					X	
6.- ¿Se requiere una entrada interactiva para el sistema?					X	
7.- ¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?						X
8.- ¿Se actualiza los archivos de forma interactiva?						X
9.- ¿Son complejas las entradas, salidas, los archivos o las peticiones?				X		
10.- ¿Es complejo el procesamiento interno del sistema?					X	
11.- ¿se ha diseñado el código para ser reutilizado?						X
12.- ¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?						X
Cuenta total	$\sum(fi) = 51$					

Nota: (Elaboración en base a criterios: Pressman, 2007).

- ✓ Para el ajuste se utiliza la ecuación:

$$PF = \text{cuenta total} * (\text{grado de confiabilidad} + \text{Tasas de Error} * \sum(fi))$$

$$PF = 308 * (0.65 + 0.01 * 51)$$

$$PF_{\text{obtenida}} = 357.28$$

- ✓ Para el ajuste se utiliza la ecuación para hallar el punto función ideal al

100% de los factores que sería 60:

$$PF = 308 * (0.65 + 0.01 * 60)$$

$$PF_{\text{ideal}} = 385$$

- ✓ Calculando del % de funcionalidad real:

$$PF_{\text{real}} = PF_{\text{obtenida}} / PF_{\text{ideal}}$$

$$\text{FUNCIONALIDAD} = \frac{357.28}{385} * 100\%$$

$$\text{FUNCIONALIDAD} = 92.8\%$$

Interpretando, el sistema tiene una funcionalidad o utilidad del 92.8% para la empresa, lo que indica que el sistema cumple con los requisitos funcionales de forma satisfactoria.

3.5.3. Confiabilidad

La confiabilidad permite evaluar la relación entre el nivel de funcionalidad y la cantidad de recursos usados, es decir, representa el tiempo que el software está disponible para su uso, la misma se calcula utilizando la probabilidad de que un sistema presente fallas:

- ✓ **Comportamiento con respecto al tiempo:** Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos.
- ✓ **Comportamiento con respecto a Recursos:** Atributos software relativo a la cantidad de recursos usados y la duración de su uso en la realización de funciones.

La función a continuación muestra el nivel de confiabilidad del sistema:

$$F(t) = (\text{FUNCIONALIDAD}) * e^{-\lambda t}$$

Se observa el trabajo hasta que se observa un fallo en un instante t, la función es la siguiente.

Probabilidad de hallar una falla: $P(T \leq t) = F(t)$

Probabilidad de hallar una falla: $P(T > t) = 1 - F(t)$

FUNCIONALIDAD = 92.8 %

$\lambda = 0.01$ (es decir 1 error en cada 6 ejecuciones) $t = 12$ meses. Hallamos la confiabilidad del sistema:

$$F(12) = 0.928 * e^{-\frac{1}{6} * 12}$$

$$F(12) = 0.124$$

Con este resultado podemos decir que la probabilidad que el sistema no presente fallas es de 0.88.

CONFIABILIDAD = 88%

Para concluir decimos que el sistema tiene un grado de confiabilidad del 88% durante los próximos 12 meses, es decir, que de cada 100 ejecuciones existen 13 fallas de elección del sistema.

3.5.4. Mantenibilidad

La Mantenibilidad se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad.

Para hallar mantenibilidad del sistema se utiliza el índice de madurez de software (IMS), que proporciona una indicación de la estabilidad de un producto de software.

Se determina la siguiente función (IMS):

$$IMS = \frac{Mt - (Fc + Fa + FE)}{Mt}$$

Mt: Número de módulos total de la versión actual

Fc: Número de módulos de la versión actual que se cambiaron.

Fa: Número de módulos de la versión actual que se añadieron.

FE: Número de módulos de la versión anterior que se eliminaron en la versión actual.

$$IMS = \frac{15 - (1 + 0 + 0)}{15}$$

$$IMS = 0.93$$

La interpretación a este resultado establece un **93%**, lo que indica que no requiere de mantenimiento inmediatamente.

3.5.5. Portabilidad

En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos:

- ✓ **Adaptabilidad:** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- ✓ **Facilidad de Instalación:** Es el esfuerzo necesario para instalar el software en un ambiente determinado.
- ✓ **Conformidad:** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.
- ✓ **Capacidad de reemplazo:** Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.

El sistema fue desarrollado en PHP y Android, y de la base de datos MySQL, se ejecuta en todos los servidores web. También se comprobó que, en los distintos navegadores más usados en nuestra área, se le da una calificación del 95% de portabilidad. El resultado del 95% indica que el desenvolvimiento del sistema es correcto en los distintos navegadores.

3.5.6. Resultados

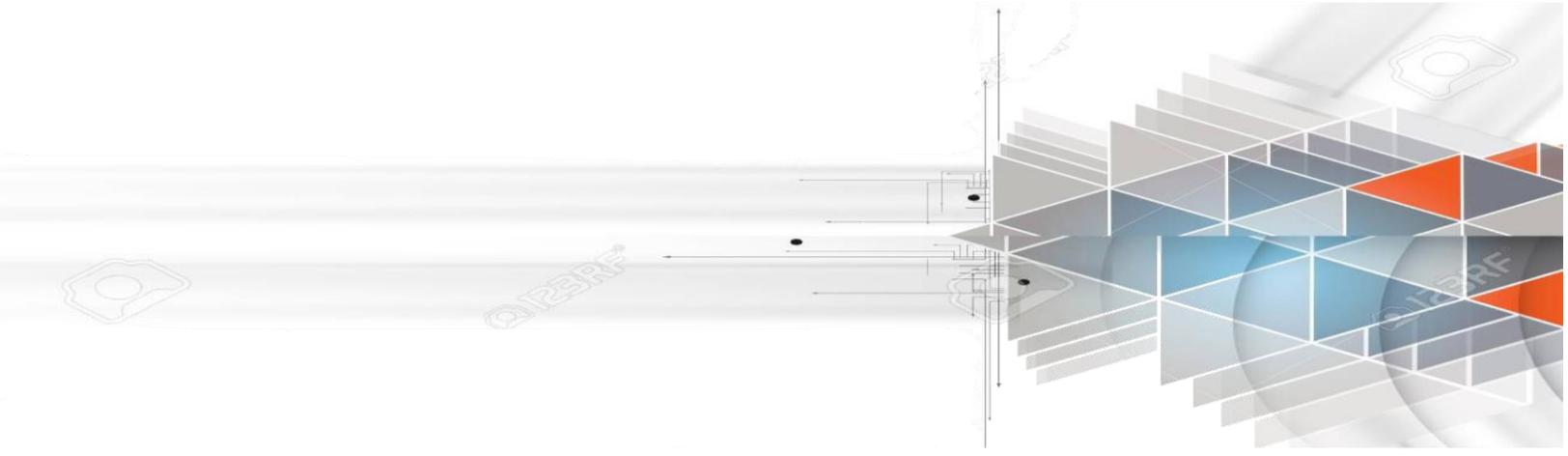
El factor de calidad total está directamente relacionado con el grado de satisfacción con el usuario que ingresa al sistema.

Tabla 3. 17
Resultados

CARACTERÍSTICAS	RESULTADO
Usabilidad	92%
Funcionalidad	92.8%
Mantenibilidad	92%
Portabilidad	95%
Confiabilidad	88%
Evaluación de calidad total	91.96%

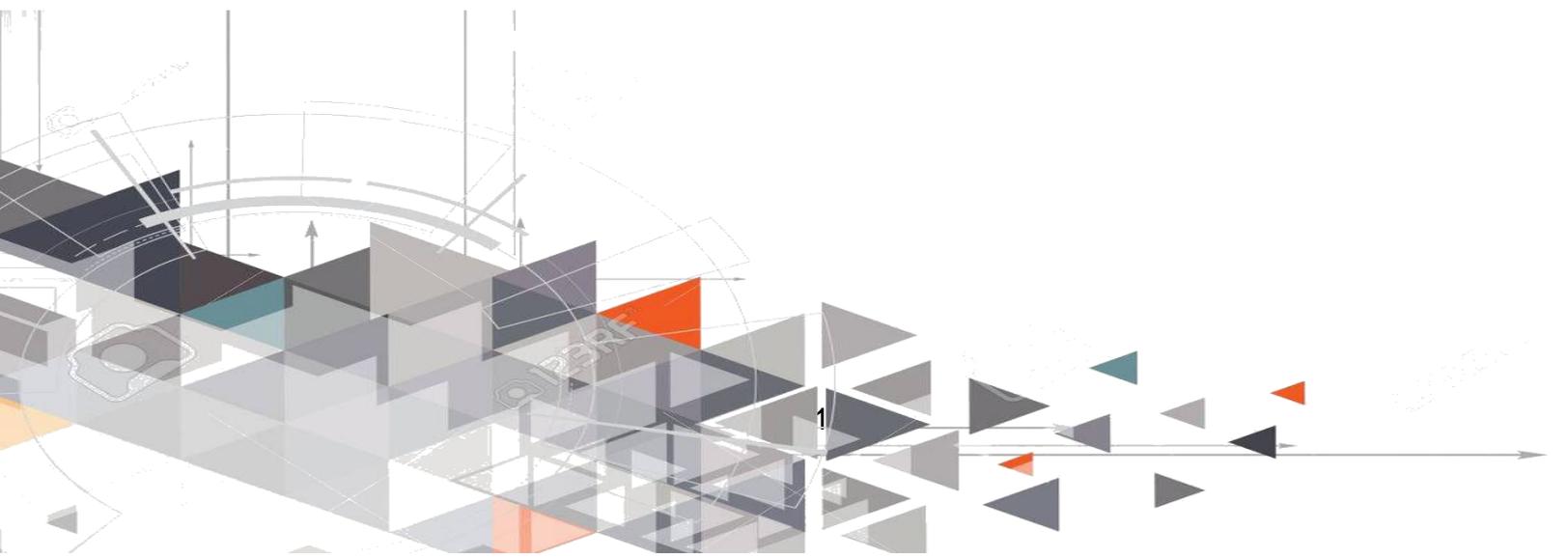
Notas: (Elaboración propia).

El nivel de aceptación satisfactorio, indica que los valores de preferencia se encuentran en el rango de 60-100 (según Pressman). La calidad del sistema corresponde al 91.96%, lo que se interpreta como la satisfacción que tiene un usuario al interactuar con el sistema.



CAPÍTULO IV

PRUEBAS Y RESULTADOS



4. PRUEBAS Y RESULTADOS

4.1. PRUEBAS

4.1.1. Capturas de Pantalla de la Aplicación Móvil

a. Aplicación instalada

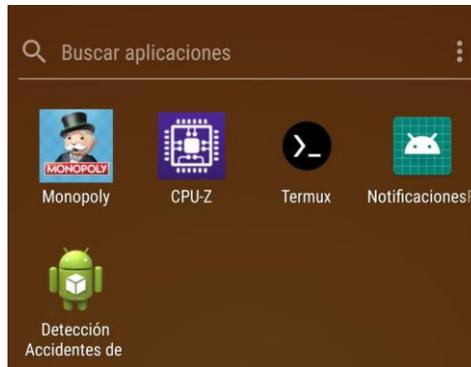


Figura 4. 1 Aplicación Instalada [Elaboración]

b. Menú Principal

Menú Principal: La pantalla principal de la aplicación cuenta con cuatro botones, dos en cada extremo. Los botones del extremo izquierdo son para encender y apagar la aplicación. Los botones del extremo derecho son para configuración e información sobre la aplicación.



Figura 4. 2 Menu Principal de la Aplicación Móvil [Elaboración Propia]

c. Botón Configuración

Al presionar este botón le pedirá autenticarse, donde la contraseña es el C.I. del usuario, luego podrá configurar los siguientes datos: el número de celular, los datos del usuario y la información de vehículo del usuario, como se ve en la Figura 4.3 y Figura 4.4.



Figura 4. 3 autenticación del Usuario



Figura 4. 4 configuración de datos personales y del vehículo [Elaboración Propia]

d. Botón Encender la Aplicación

Este botón permite visualizar una pantalla con dos divisiones: el lado izquierdo tiene un velocímetro que permite mostrar la velocidad con lo que va el vehículo y un botón que permite simular el accidente. En el lado derecho se visualiza la ubicación del vehículo en un mapa. Dentro del

mapa existen dos botones: Visualizar u Ocultar Hospitales y una opción para ir al Hospital más cercano desde la ubicación del vehículo.



Figura 4. 5 Visualización de la aplicación en ejecución [Elaboración Propia]

La visualización de hospitales en el mapa es mediante pines, cada pin contiene información a cerca del hospital y dos botones: hacer llamada y como llegar al hospital. (Ver Figura 4.6 y Figura 4.7)



Figura 4. 6 Dialogo de información de un hospital [Elaboración Propia]



Figura 4. 7 Trazado de cómo llegar a un hospital determinado [Elaboración Propia]

4.1.2. Capturas de Pantalla de la Plataforma Web

a. Pantalla Principal

En la pantalla principal de la plataforma web, se cuenta un menú 4 opciones: Home, Sobre Nosotros, Contactos e Iniciar Sesión.



Figura 4. 8 Pantalla Principal Plataforma Web [Elaboración Propia]

b. Pantalla Principal del Administrador

En la pantalla principal de la plataforma web, que cuenta con 2 secciones de Menús: Menú Lateral: Tiene 4 opciones de menú Gráficos, Reportes y Registro.

Menú Superior: Accidentes, Hospitales y Geo localización de Hospitales.

c. Pantalla de Sección GRÁFICOS

Tenemos en la parte central que nos muestra las Estadísticas de los accidentes que se registran o se envían a base de datos.



Figura 4. 9 Pantalla de Estadísticas de reportes de Accidente [Elaboración Propia]

d. Pantalla de Sección REPORTE

En esta parte de la pantalla nos muestra el registro de los Accidentes y reportes a realizar.

Administrador DIRECCION DEPARTAMENTAL DE TRANSITO DE LA PAZ - 2020

Accidentes Geolocalización Accidente Hospitales Geolocalización Hospitales

Gustavo Paco

Reportes del Accidente [Generar Reporte](#)

Nombre del Conductor:

Responsable del Reporte:

Licencia de Conducir:

Descripción del Accidente:

Foto del Accidente: Ningún archivo seleccionado

Código del Reporte:

[Generar](#) [Imprimir](#)

[Guardar](#) [Cancelar](#)

Figura 4. 10 Pantalla de Registro de Reportes [Elaboración Propia]

e. Pantalla sección de REGISTRAR

En esta sección de pantalla nos visualiza el registro de nuevo administrador y los privilegios que tiene para administrar.

Registrar Administrador

Nombre(*):

Tipo Documento(*):

Numero de Documento(*):

Dirección:

Telefono:

Email:

Cargo:

Login(*):

Clave(*):

Privilegios:

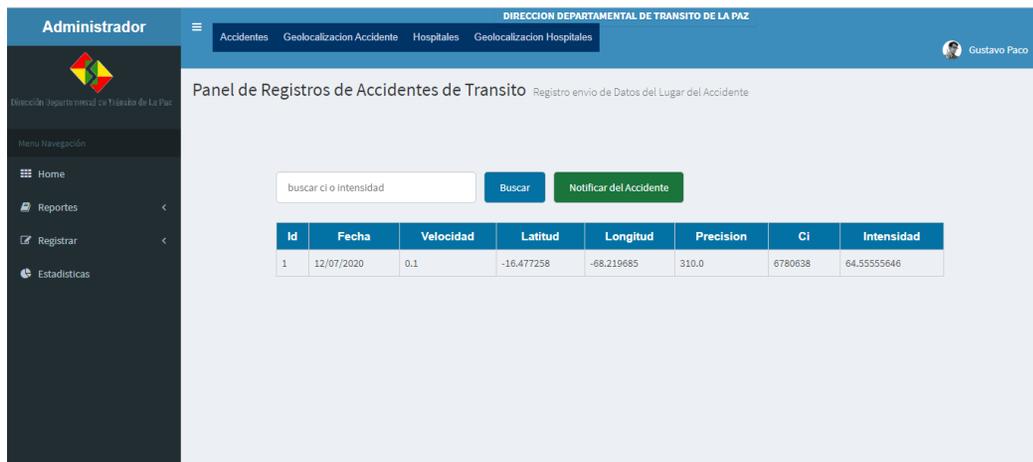
- Registrar
- Reportes
- Geolocalización
- Listar Hospitales
- Editar
- Eliminar
- Actualizar

Imagen: Ningún archivo seleccionado

Figura 4. 11 Pantalla registro de nuevo Administrador [Elaboración Propia]

f. Pantalla de sección de Accidentes

En esta sección de pantalla, visualizara la lista de todos los accidentes que fue enviado mediante celulares, además en la parte derecha tenemos los datos del último accidente, y finalmente en la parte inferior tenemos un botón que nos permite notificar el último accidente a todos los celulares que cuentan con la aplicación “DetectarAccidente TSPM”.



Id	Fecha	Velocidad	Latitud	Longitud	Precision	CI	Intensidad
1	12/07/2020	0.1	-16.477258	-68.219685	310.0	6780638	64.55555646

Figura 4. 12 Pantalla de Registro de Accidentes [Elaboración Propia]

g. Geolocalización de los hospitales

En esta sección de la pantalla se mostrará todos los puntos de los hospitales con sus respectivos datos del mapa.

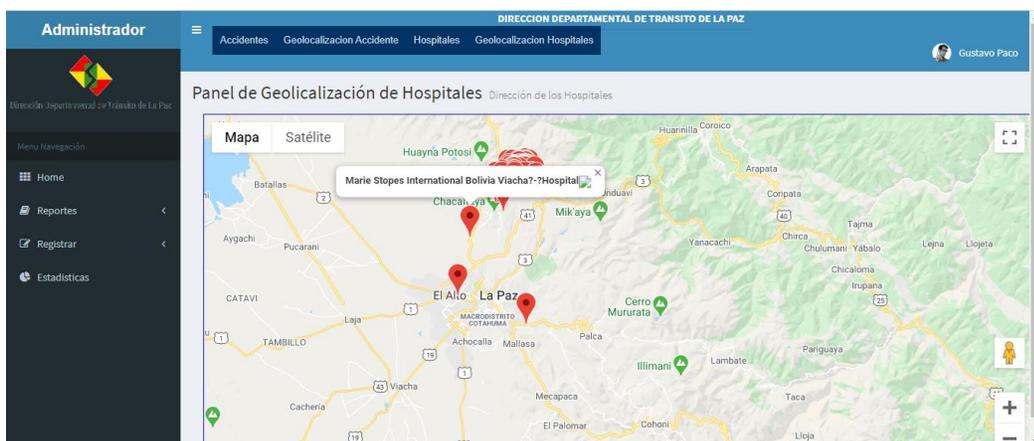


Figura 4. 13 Pantalla de Geolocalización de Hospitales [Elaboración Propia]

h. Geolocalización del Accidente

En esta sección de la pantalla se mostrará el punto del lugar del accidente dentro del mapa

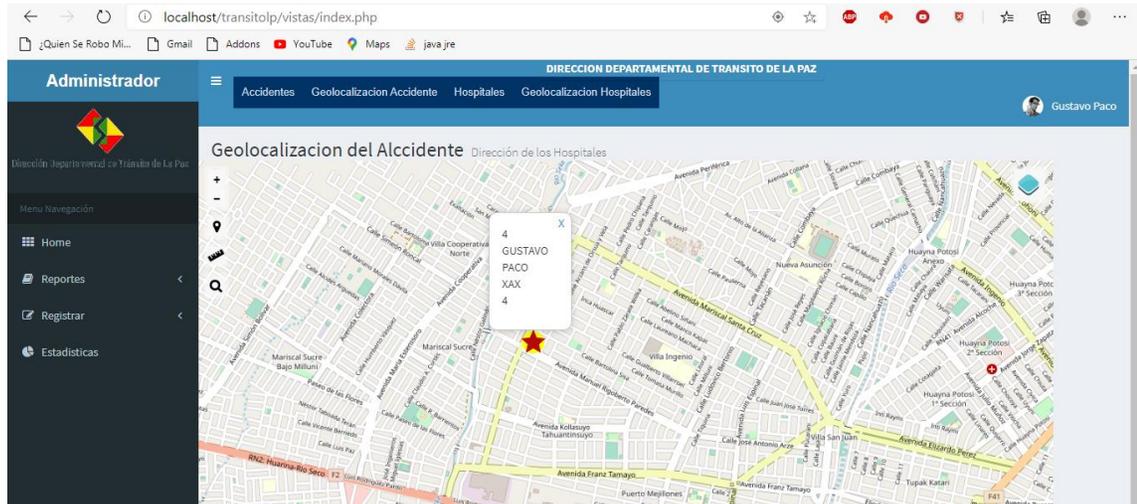


Figura 4. 14 Pantalla de Geolocalización del Accidente [Elaboración Propia]

4.2. RESULTADOS

La tabla que representa contiene los datos generados de las 40 pruebas realizadas para este prototipo, las cuales fueron obtenidos cada 5 segundos.

De los datos obtenidos en la Tabla de resultado (Ver Tabla 3.1) fueron tomadas en consideración las siguientes consideraciones: Para realizar las respectivas pruebas del prototipo, tanto de la aplicación para móviles (Teléfonos inteligentes) como de la plataforma web se utilizó un auto de juguete a control, para evidenciar que efectivamente detecta el accidente y envía al servidor los datos ya mencionados en el Capítulo II. Para obtener la ubicación del accidente en la aplicación para móviles se utiliza: GPS, WiFi o red de datos, GSM más conocido como triangulación de telefonías móviles. También se puede evidenciar que utilizando GPS se puede obtener la localización precisa y hasta exacta el error es casi 0 m, pero si nos encontramos dentro de un túnel o en medio de edificios no es tan precisa la ubicación. Sin embargo, utilizando WiFi o red de datos la obtención de la localización puede llegar a tener un error de 200 a 250 metros, también mucho depende de cuan bueno es la señal del WiFi o la señal de red de

datos (Ver Tabla 3.1). En Bolivia hablando específicamente en La Paz si quisiéramos utilizar WiFi es muy complicado utilizar libremente ya que piden autenticación.

Finalmente, utilizando GSM o triangulación de torres de las telefonías móviles la obtención de la localización es aún más imprecisa que el anterior ya que el error llega a 500 hasta 1000 metros, esto considerando que la radio enlaces de la telefonía móvil se encuentran de 0 a 2000 km (sin utilizar fibra óptica), pero si las radio enlaces se encuentran a más de 2000 km entonces la señal de la telefonía móvil será muy baja por consecuente la obtención de la localización tendrá un error mayor que 1 km, esto ocurre en las áreas rurales.

Sobre la velocidad (Ver Tabla 3.1), realizando las pruebas se pudo evidenciar que el velocímetro que cuenta el prototipo de la aplicación móvil varía entre 0 a 1km/h con respecto al velocímetro que tienen los autos (minibuses, buses o taxis.), considerando autos antiguos y nuevos.

Sobre la intensidad del sonido, el sonido que emite el auto a control llega a 62 dB, pero cuando choca contra un concreto la intensidad del sonido es mayor que 63 dB. Con estos datos llegamos a la conclusión de que ocurrió accidente. La última columna de la Tabla 3.1 nos detalla sobre la detección del accidente (celdas marcadas con rojo), de los 10 choques contra un concreto uno no envió a tiempo al servidor, esto sucedió porque no se pudo conectar correctamente con el servidor “señal de Internet baja”.

Por consecuente se pudo obtener los siguientes promedios: (Ver Tabla 3.1)

- ✓ Promedio de Acierto en Detectar Accidente: 90%
- ✓ Promedio de Acierto en geo localización:
 - Utilizando GPS: 90%
 - Utilizando WiFi / Red de datos: 78.6%
 - Utilizando GSM: 36%

Tabla 4. 1*Datos de la Prueba usando la aplicación de la presente tesis.*

Nro.	Latitud	Longitud	Velocidad (m/h)	Intensidad de Sonido (dB)	GPS %	WIFE %	GSM %	Ocurrió Accidentes
1	-16,479	- 69,220	0	43,0251	90	79	35	NO
2	-16,479	- 69,220	10	43,0251	85	80	45	NO
3	- 16,47921	- 69,041	25,1	43,0251	91	75	38	NO
4	- 16,47921	- 69,441	25,1	44,1341	99	74	40	NO
5	- 16,4731	69,241	25,1	44,3351	89	74	43	NO
6	- 16,4731	69,28541	25,1	46,0251	90	74	43	SI
7	- 16,4731	69,2541	30	43,0251	95	78	43	NO
8	- 16,492	- 68,137	34	41,3251	98	74	46	NO
9	- 16,492	- 68,137	35	43,6245	99	76	30	NO
10	- 16,492	- 68,137	33	48,3231	89	72	36	NO
11	- 16,492	- 68,137	35	43,0251	95	72	39	NO
12	- 16,49	- 68,137	37	48,3254	98	75	39	SI
13	- 16,49	- 68,137	35	40,3251	95	74	39	NO
14	- 16,489	- 68,137	33	43,0254	94	80	35	SI
15	- 16,489	- 68,137	35	44,1344	98	72	36	NO
16	- 16,32	- 68,137	40	43,0259	96	74	40	NO
17	- 16,489	- 68,137	40	47,5655	97	75	40	SI
18	- 16,487	- 68,137	34	49,2254	97	73	38	SI
19	- 16,487	- 68,137	36		99	73	38	SI
20	- 16,487	- 68,137	34	43,0251	96	75	38	SI
21	-	-	34	43,0251	97	74	39	NO

	16,487	68,137						
22	-	-	35	43,0251	97	79	40	SI
	16,487	68,153						
23	-	-	38	43,0251	98	80	36	SI
	16,485	68,153						
24	-	-	40	43,0251	90	73	37	NO
	16,485	68,153						
25	-	-	41	43,0251	93	75	37	SI
	16,483	68,153						
26	-	-	43	43,0251	95	75	37	SI
	16,483	68,153						
27	-	-	42	43,0251	95	75	36	SI
	16,476	68,153						
28	-	-	45	43,0251	98	72	35	SI
	16,476	68,153						
29	-	-	45	43,0251	96	74	36	NO
	16,476	68,153						
30	-	-	47	43,0251	99	75	37	NO
	16,475	68,153						
31	-	-	48	43,0251	95	75	39	NO
	16,475	68,153						
32	-	-	48	43,0251	97	77	40	SI
	16,476	68,153						
33	-	-	42	43,0251	96	78	36	NO
	16,476	68,153						
35	-	-	43	43,0251	94	79	37	SI
	16,476	68,153						
36	-	-	43	43,0251	93	80	35	SI
	16,487	68,153						
37	-	-	44	43,0251	99	74	36	SI
	16,487	68,153						
38	-	-	47	43,0251	95	75	39	NO
	16,487	68,153						
39	-	-	48	43,0251	94	75	36	NO
	16,496	68,153						
40	-	-	34	43,0251	99	74	37	NO
	16,501	68,153						
	Promedio de Acierto en geo localización				90	78,6	36	
								%
	Promedio de Acierto en Detectar Accidente						90	%
								%

Nota: Elaboración Propia

4.3. PRUEBA DE HIPÓTESIS

Una hipótesis es una suposición que se establece como base de una investigación que puede confirmar o negar su validez, su función principal es demarcar el problema que se va a investigar considerando componentes tales como lugar, características de los sujetos, tiempo y otros.

4.3.1. Definición De La Hipótesis

H₀: “La plataforma de geolocalización como proceso de un sistema de información geográfica permite recibir información con un 90% de confiabilidad sobre la ubicación del conductor mediante el GPS de su dispositivo móvil”.

H₁ : Se Rechaza H₀

En este caso se espera que el porcentaje de éxitos sea igual o mayor a 90%, tomando un nivel de significancia del 5%

H₀ : $p_0 \geq 0.9$

H₁ : $p_0 < 0.9$

4.3.2. Evaluación De Resultados

Para determinar si el porcentaje de éxitos obtenido en las pruebas puede ser considerado cercano al 90% de nivel de confianza esperado, se hará uso de una Prueba de Hipótesis para Proporciones.

Las variables usadas en dicha prueba serán las mismas mencionadas en la evaluación de casos de prueba:

$p_0 = 0.9$

$X = 39$

$n = 47.5$

$\alpha = 0$

4.3.3. Determinación de la Región Crítica

La región crítica para la hipótesis planteada, es la siguiente:

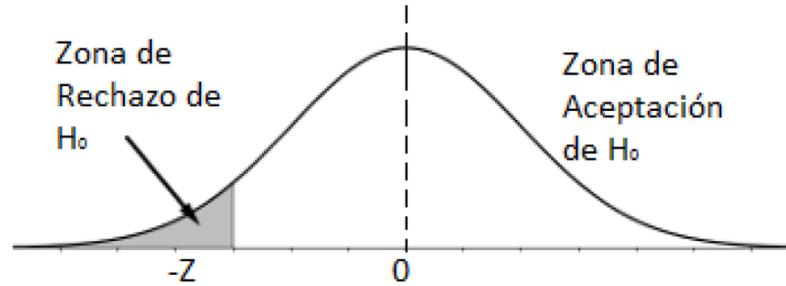


Figura 4.1. Región crítica para la hipótesis Fuente: Elaboración propia

Como n se refiere en este caso al número de pruebas, en este caso 47, el punto crítico a usar es Z_0 y se determina mediante:

$$-Z_0 = -Z_{1-\alpha} = -Z_{1-0.05}$$

Este valor se halla de la tabla de la función de distribución normal, la cual se encuentra en el anexo A. Para obtener el valor de z se elige de la tabla mencionada el valor más cercano a 0.95; el cual está ubicado en la fila 1.6 y columna 0.04

Z	...	0.04
...		
1.6		0.94950

Tabla 4.1. Resultado tabla de la función de distribución normal Fuente: Elaboración propia

El valor de z se obtiene sumando ambos valores:

$$-Z_{0.95} = -1.6 + 0.04 = -1.64$$

4.3.4. CÁLCULO ESTADÍSTICO DE LA PRUEBA

Como se conoce el número total de individuos en el espacio muestral, el valor estadístico de la prueba se obtiene mediante la fórmula:

$$Z = \frac{X - n * p_0}{\sqrt{n * p_0(1 - p_0)}}$$

Reemplazando los valores y haciendo los cálculos correspondientes obtenemos:

$$Z = \frac{39 - 47 * 0.9}{\sqrt{47 * 0.9(1 - 0.9)}} = -0.78$$

4.3.5. TOMA DE DECISIÓN

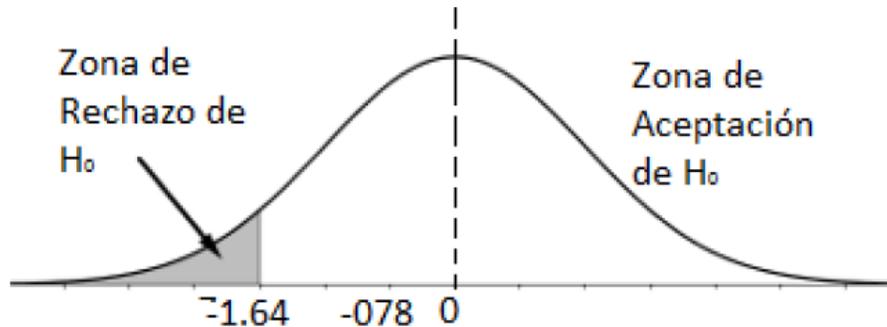
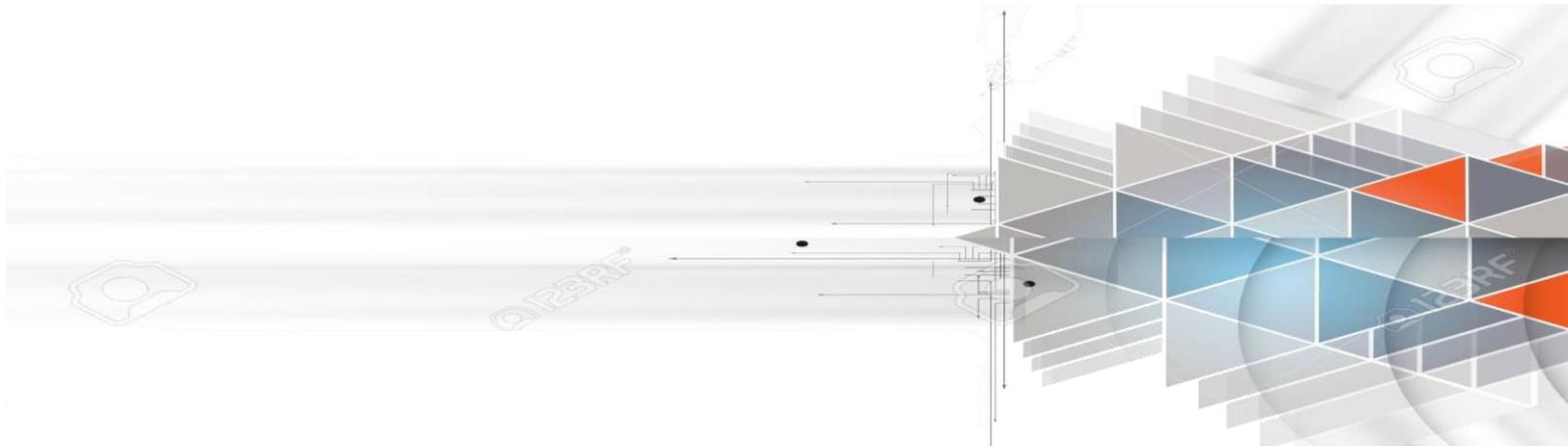


Figura 4.2. Distribución de Z en el gráfico para la toma de decisión

Fuente: Elaboración propia

El promedio de éxito del prototipo al momento de reconocer **las muestras se acerca al 90%**. Por tanto, como se acepta H_0 se podría concluir y afinar la hipótesis:

H_0 : “La plataforma de geolocalización como proceso de un sistema de información geográfica permite recibir información con un 90% de confiabilidad sobre la ubicación del conductor mediante el GPS de su dispositivo móvil”



CAPÍTULO V

CONCLUSIÓN Y RECOMENDACIONES



5. CONCLUSIÓN Y RECOMENDACIONES

5.1. CONCLUSIÓN

Se pudo concluir satisfactoriamente el desarrollo de: “Modelo Web de localización y reportes de accidentes de Tránsito basado en Android”, utilizando una metodología que es netamente para el desarrollo de aplicaciones móviles Mobile – D y su utilización para cubrir las necesidades del Sistema detección de Accidentes, colaborando así para que el transito acuda al lugar del hecho a la brevedad posible

5.2. CUMPLIMIENTO DE LOS OBJETIVOS

Con todo lo expuesto en los anteriores capítulos se ha llegado a las siguientes conclusiones:

- ✓ Respecto al objetivo general, se ha podido realizar una aplicación que sea capaz de detectar el accidente utilizando los recursos de teléfonos inteligentes como: el sensor acelerómetro y el micrófono este último para obtener la intensidad del sonido. La aplicación es capaz de interactuar con el servidor de incidentes local.
- ✓ Para los objetivos específicos:
 - Se implementó el servidor de incidentes local que sea capaz de alertar al administrador del sistema sobre el accidente.
 - Se realizó el módulo que sea capaz de enviar información necesaria a los del Tránsito.
 - Se desarrolló la aplicación móvil para teléfonos inteligentes que sea capaz de detectar el accidente utilizando sus recursos como: el sensor acelerómetro y el micrófono, este último para obtener la intensidad de sonido.
 - Se implementó la visualización en el mapa, la ubicación donde ocurrió el accidente con un pin.
 - Se desarrolló una opción en el mapa que muestra el hospital más cercano desde tu ubicación.

- Se realizó el módulo de notificación del accidente a los Teléfonos móviles que cuentan con la aplicación.

5.3. RECOMENDACIONES

Las tareas que se plantean a posterior para mejorar la detección de accidentes de tránsito son:

- ✓ Estudiar las causas del accidente; con los datos de los sensores que tiene el dispositivo móvil, así para poder buscar a los posibles culpables.
- ✓ Investigar acerca de las redes MANET (Mobile Ad-Hoc Network), las cuales son redes de dispositivos conectados de forma inalámbrica y que poseen propiedades de auto-configuración, además de poseer cierta movilidad (es decir se encuentran montados en plataformas móviles) y el de las redes VANET (Vehicular Ad-Hoc Network), cuyo objetivo es desarrollar plataformas de comunicación entre vehículos en movimientos y entre estos y la infraestructura vial. Ambas redes tienen sus propios campos de investigación, protocolos, etc. Mediante la conjunción de este tipo de redes se podría llegar a mitigar las congestiones vehiculares en gran medida advirtiendo donde hay problemas y todo esto se haría de forma dinámica ya que “las redes Ad-Hoc se caracterizan por no tener una infraestructura, sino que los mismos nodos se encargan de organizarse para formar una topología de comunicación”, de esta manera, todos los dispositivos móviles podrían comunicarse entre sí para poder recibir al instante una información de choque y así poder evitar lugares congestionados. Sería interesante luego poder extender esto a otros tipos de alertas.
- ✓ Por último una investigación profunda sobre como reproducir el choque a partir de los datos de los sensores como acelerómetro, giróscopo, etc. sería muy útil e importante para poder brindar dispositivos que sean más seguros a partir del análisis de los datos y beneficiar a las compañías de seguros para que sepan cómo fueron los sucesos.

BIBLIOGRAFÍA

[Alejandro Mauricio Copa Rojas. 2011] Monitoreo Cardíaco con Dispositivos Móviles Para la Prevención de Situaciones Críticas de Salud.

[PRESSMAN, 2002]. Pressman Roger S, 2002 INGENIERÍA DE SOFTWARE UN ENFOQUE PRACTICA. 5ta Edición Madrid España.

[Cabrera, A., Figueroa, R., y Solís, C., 2008] “Metodologías Tradicionales Vs. Metodologías Ágiles”. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.

[Fernández, Y., & Díaz, 2012]. "Patrón Modelo-Vista-Controlador".

[Hanckes, H., 2012] "Tutorial 10 - Modelo Vista Controlador". Pontificia Universidad Católica de Chile. Escuela de Ingeniería. Departamento de Ciencia de la Computación.

[Pressman, R., 2002]. "Ingeniería del software un enfoque práctico". Quinta edición. México. McGraw-Hill Interamericana Editores S.A de C.V. 80.

[Pressman, R., 2010]. “**Ingeniería del software: un enfoque práctico**”. Séptima edición. México. McGraw-Hill Interamericana Editores S.A de C.V.

[Sommerville, Ian., 2005]. “**Ingeniería del software. Séptima edición. Madrid**”

Telematics Today and Tomorrow – Bob Lange – General Motors.

El Sistema de Llamada de Emergencia (E-CALL) Vol.7, Fundación Instituto Tecnológico para la Seguridad del Automóvil (FITSA)

Advanced Automatic Crash Notification (AACN), 2005 Computerworld Honors

Case Study

[Fernández, Y., & Díaz, 2012]. eCall Discussion Paper. Finnish eCall Experts 6.6.2005 eCall Communications Test Bench – Structure and Content of MSD, FDS and MDS Messages, http://www.ecall.fi/eCall_msd_en_052009.pdf

[Ricardo Moya, 2018] Manual básico Android Studio

[Tomas, Jesús, 2018] El gran libro de Android 2ª edición

[Jaime Aranaz Tudela., 2009] Desarrollo De Aplicaciones Para Dispositivos Móviles Sobre La plataforma Android

[Ana Maria Garcia Sanchez, 2014] Evaluación de métricas de calidad del software sobre un programa Java.

[Adriana Gomez, Maria del C. Lopez, Migani, 2010] Un modelo de estimación de proyectos de software COCOMO

[Anna Schirokoff, Juha Luoma, 2016] Impacts of an automatic emergency call system on accident consequences, VTT Technical Research Centre of Finland

[Michael Nielsen, 2006]. Recommendations for the introduction of the pan-European eCall. On behalf of DG Co-Chair – DG eCall. Plenary Meeting of the eSafety Forum

REFERENCIAS WEB

Método Mobile-d.

(www.001): <http://www.genbetadev.com/desarrollo-aplicaciones-moviles/metodos-aplicaciones-para-el-desarrollo-de-aplicaciones-moviles>.

Métrica de Calidad para Evaluar Software ISO 9126.

[www.002]: <https://vanevargas.jimdo.com/m%C3%B3dulos/modelos/modelo-iso-9126/>.

Php.

(www.003): <http://php.net/manual/es/intro.whatis.php>.

Xampp.

(www.004): <http://codegeando.blogspot.com/2013/13/03/php.xampp-definicion-instalacion-y.html>.

Mysql.

(www.005): <http://culturacion.com/que-es-y-para-que-sirve-mysql/>.

Css3.

(www.006): <http://tc2proyectodegradoingdesistemas.blogspot.com/2012/05/que-es-y-para-que-sirve-css3.html>.

Html.

(www.007): <http://definicion.de/html/>.

Google Maps.

(www.008): <https://iiemd.com/google-maps/que-es-google-maps>.

Aplicación.

(www.009): <https://www.mastermagazine.info/termino/3874.php>.

Diagrama de Relación de Aplicaciones a Usuario Final y otros Programas Informáticos.

(www.010): <http://johnnjc.blogspot.com/2012/04/que-es-una-aplicacion-informatica.html>.

Aplicación Web.

(www.011): (<https://wiboomeia.com/que-son-las-aplicaciones-web-ventajas-y-tipos-de-desarrollo-web/>).

Niveles de una Aplicación Web.

(www.012): <http://desaplwebjnh.blogspot.com/2013/02/unidad-1-plataforma-web.html>.

Aplicación Móvil.

(www.013): http://www.alegsa.com.ar/Dic/aplicacion_movil.php.

Transporte Pesado.

(www.014): <https://transportedecargadepits.wordpress.com/2011/09/29/definicion-del-servicio-de-transportede>

Ventas Globales de Teléfonos Inteligentes por Fabricantes.

(www.015): <http://www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm>.

Ventas Globales de Teléfonos Inteligentes por Sistemas Operativos.

(www.016): <http://www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm>.

Faces de la Metodología Mobile-D.

(www.017): <http://pegasus.javeriana.edu.co/~PA133-05-PMovVidaAutomotor/Metodologia.html>.

Apache.

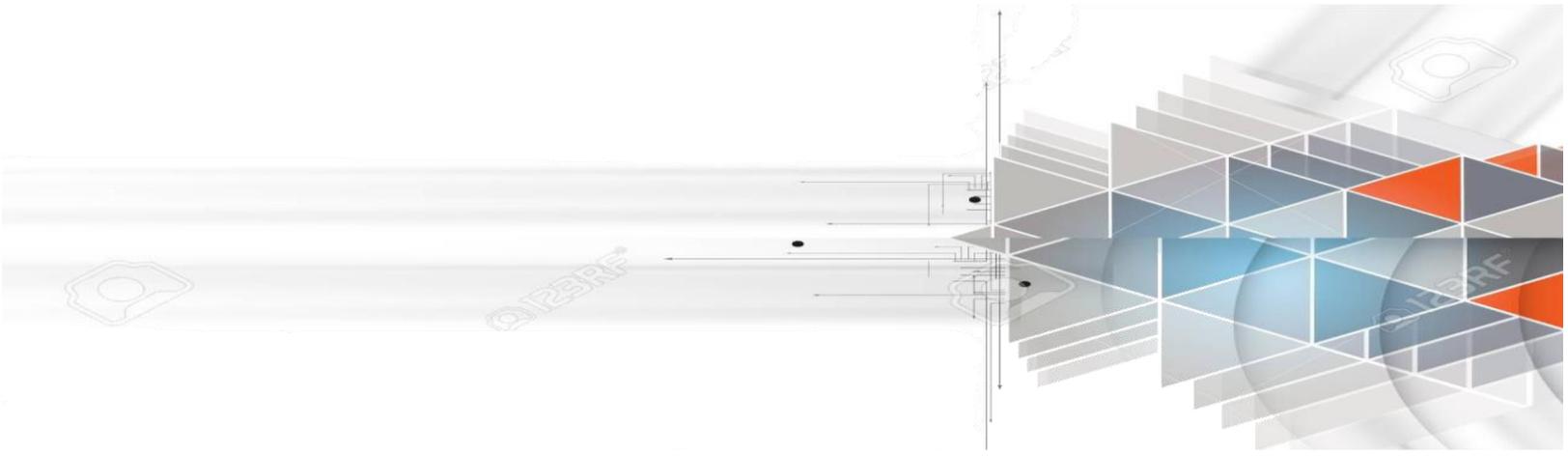
(www.018): http://culturacion.com/que-es-apache/.03_pfl/ch_02.pdf
http://www.kowoma.de/en/gps/data_composition.htm

Gps

(www.019): Transmitted GPS Signals. <http://www.kowoma.de/en/gps/signals.htm>
The GPS System. http://www.kowoma.de/en/gps/waas_egnos.htm

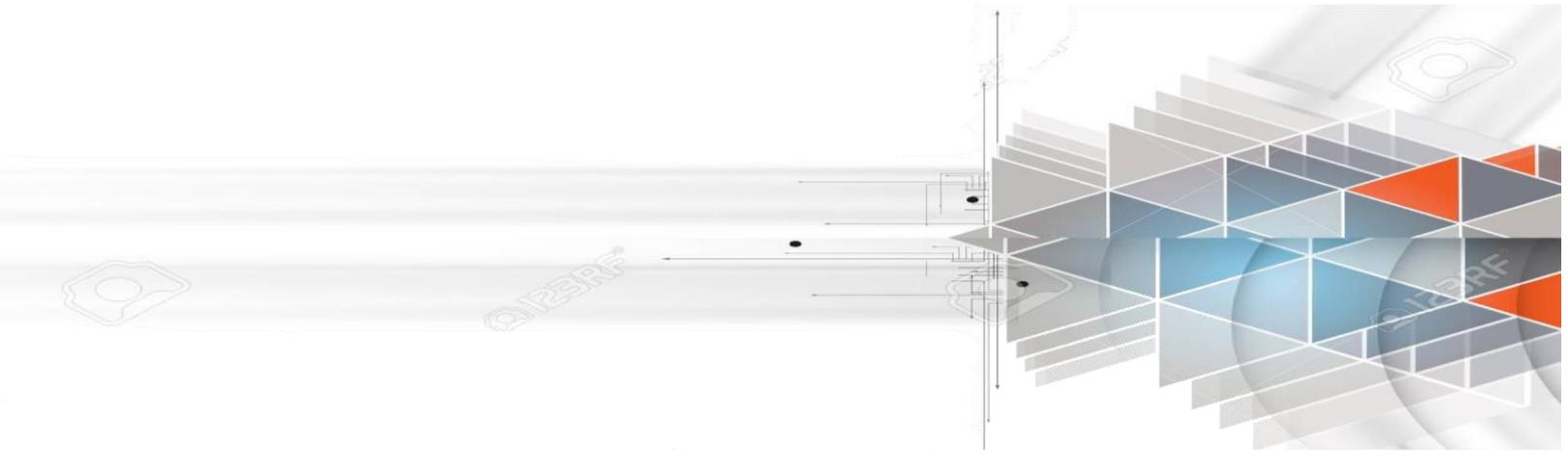
Sensores

(www.020): How Location Services Work on Mobile Devices.
<http://anders.com/cms/389>



ANEXOS



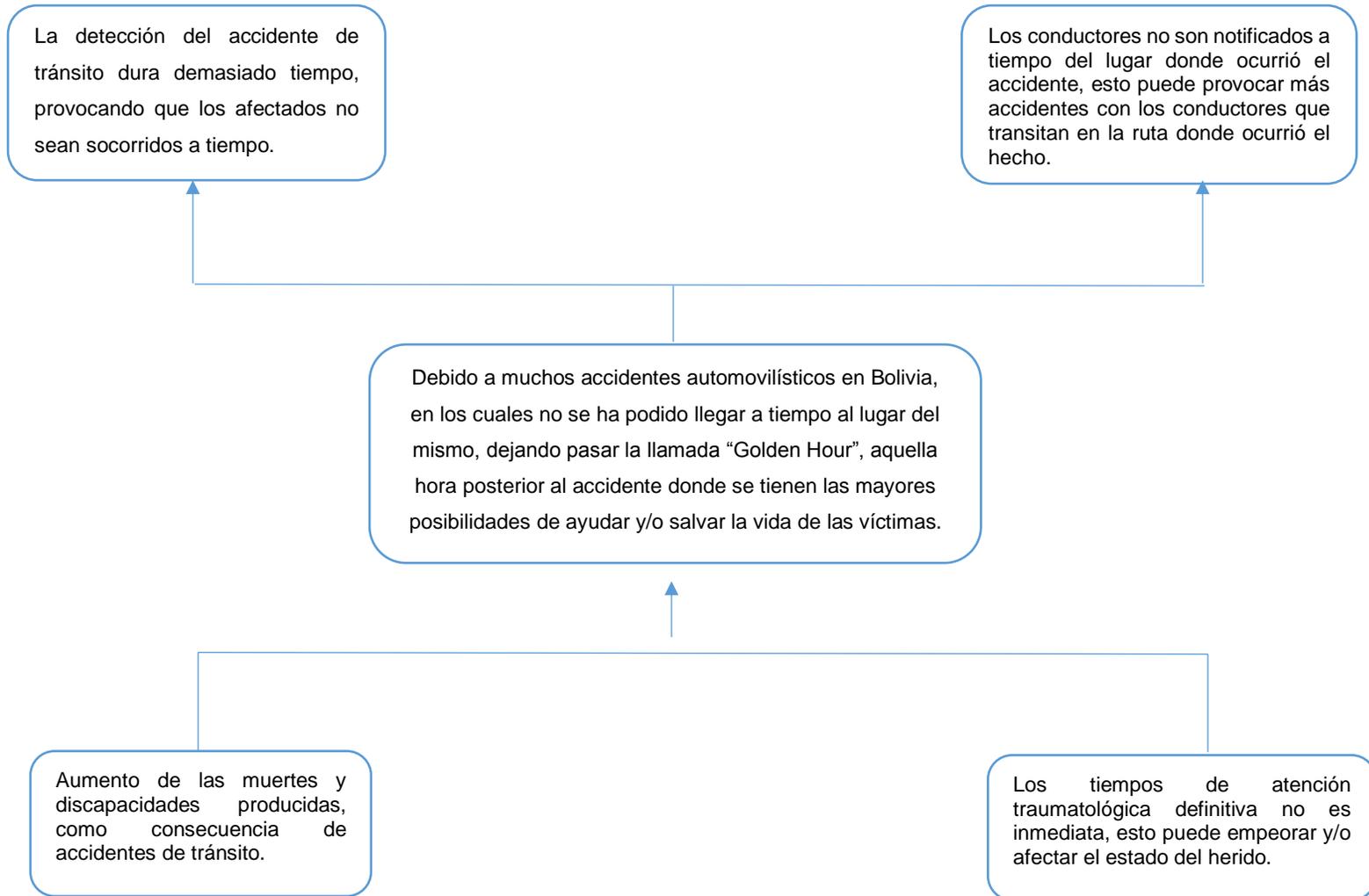


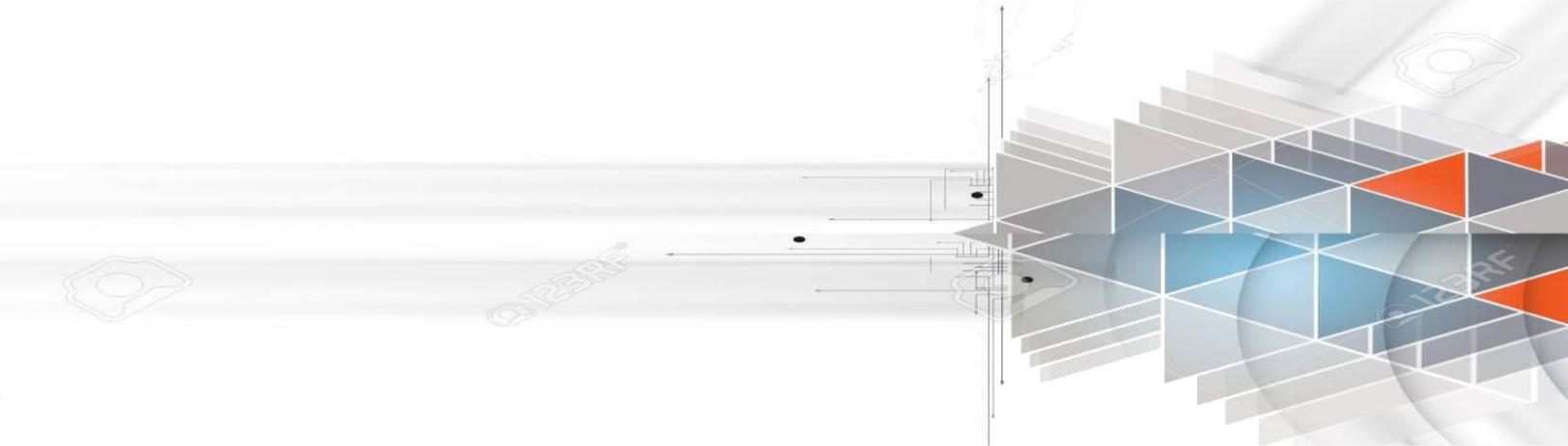
ANEXO A

ÁRBOL DE PROBLEMAS



ARBOL DE PROBLEMAS



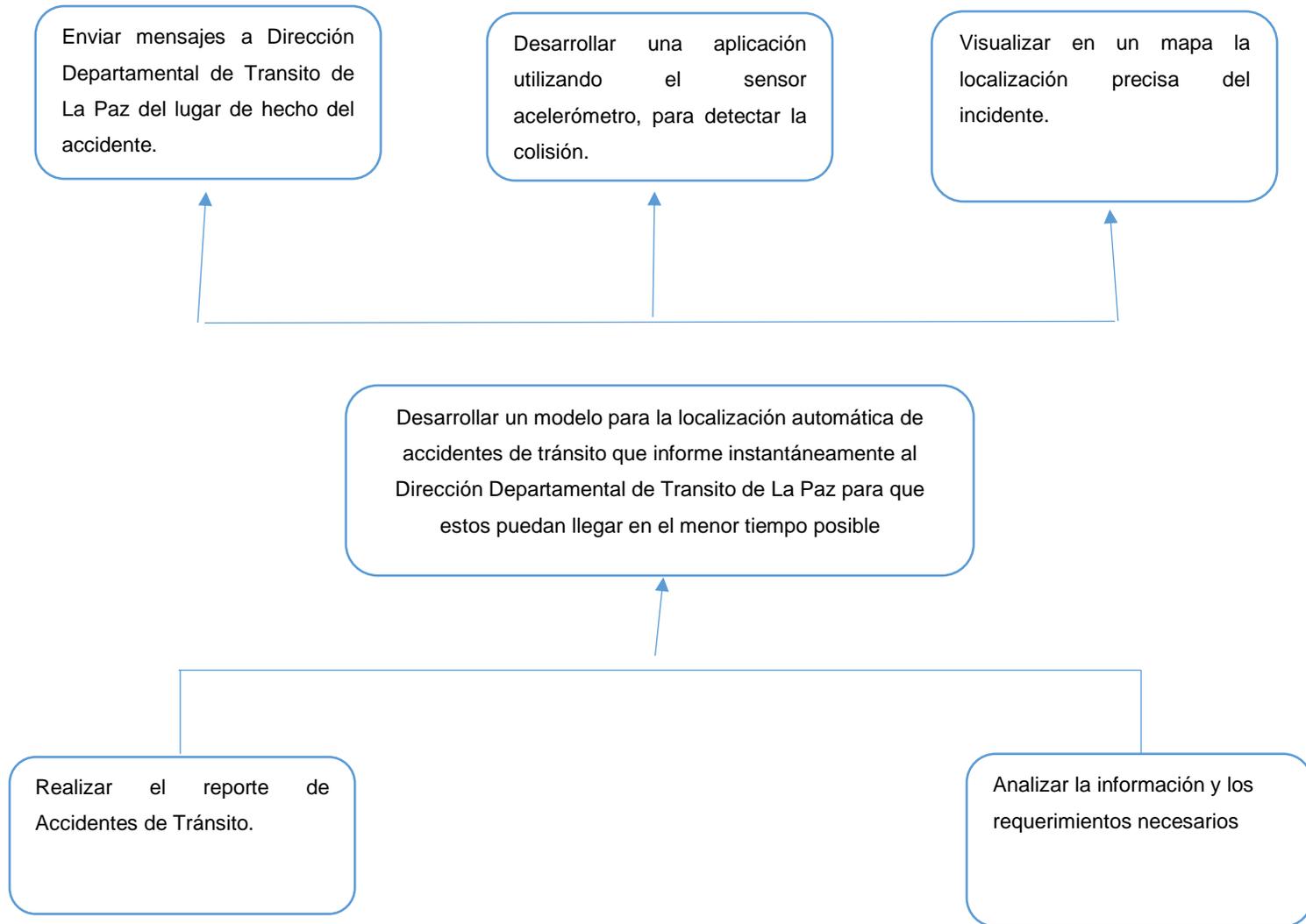


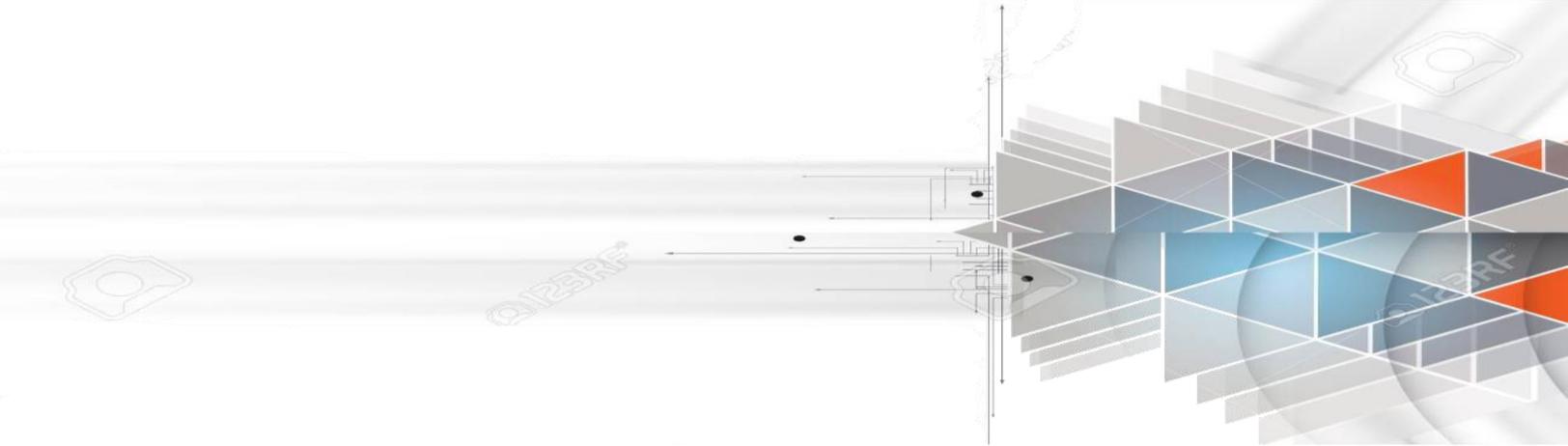
ANEXO B

ÁRBOL DE OBJETIVOS



ÁRBOL DE OBJETIVOS





ANEXO C

MANUAL DE

ADMINISTRADOR



INTRODUCCION

El sistema de administración web para localización de accidente de tránsito y reportes, fue desarrollado para cumplir la función de localizar los accidentes y realizar reportes por medio mapas de google Maps.

El administrador es la única persona que puede ingresar al sistema de administración y adicionar, modificar, buscar, localizar y eliminar a un nuevo administrador.

DEMOSTRACION DEL SISTEMA

✓ Módulo De Inicio



The screenshot shows a login form with the title "SoloAdministrador". Below the title is the instruction "Ingresa tus datos de Acceso". There are two input fields: "Usuario" with a user icon and "Password" with a search icon. A blue "Ingresar" button is located below the password field. A blue circle with the letter "a" and an arrow points to the "Ingresar" button.

a. **Botón de iniciar sesión:** El acceso al sistema es con una cuenta asignada por el administrador.

✓ Pantalla de Inicio



Menú Lateral Izquierda principal del sistema

Reportes

- Registrar Accidentes
- Adicionar responsable del Reporte

Registros

- Nuevo administrador.
- Listar Privilegios del Administrador

Graficos

Pantalla de inicio del Reportes (Registrar Accidente)

The screenshot shows the 'Reportes del Accidente' page in a web application. The interface includes a top navigation bar with 'Accidentes', 'Geolocalizacion Accidente', 'Hospitales', and 'Geolocalizacion Hospitales'. A left sidebar contains 'Administrador' and a navigation menu with 'Home', 'Reportes', 'Registrar', and 'Graficos'. The main content area features a table of accident reports with columns for 'Opciones', 'Nombre Conductor', 'Responsable del Reporte', 'Codigo de Reporte', 'Licencia de Conducir', 'Foto Accidente', 'Descripcion del Accidente', and 'Estado'. Annotations 'a' through 'f' point to specific elements: 'a' points to the 'Generar Reporte' button, 'b' points to the 'Agregar' button, 'c' points to the 'Opciones' dropdown, 'd' points to the 'Responsable del Reporte' column, 'e' points to the search bar, and 'f' points to the 'Estado' column.

Opciones	Nombre Conductor	Responsable del Reporte	Codigo de Reporte	Licencia de Conducir	Foto Accidente	Descripcion del Accidente	Estado
	Lopez Obrador	Sargento Olivares	01	737374		Localidad Carabuco	Activado
	luis	Sargento Olivares	9090	5432323		el alto	Activado
	Juan Gutierrez	Sargento Olivares	433	32312313		Choque de frente, minibus color rojo localidad caranavi	Activado
	Gustavo Paco Machaca	Sargento Olivares	00012	3424		lu el alto	Activado
	paco	Sargento Olivares	666	89877		6780638	Activado

- a) En el botón generar reporte al hacer click en el icono se generará otra pantalla para generar la siguiente pantalla

Lista General de Accidentes

Nombre del Conductor	Responsable Reporte	Código Reporte	LIC	Descripción del Accidente
Lopez Obrador	Sargento Olivares	01	737374	Localidad Carabuco
luis	paco	9090	5432323	el alto
Juan Gutierrez	Sargento Olivares	433	32312313	Choque de frente, minibus color rojo localidad caranavi
Gustavo Paco Machaca	Sargento Olivares	00012	3424	lu el alto
paco	Sargento Olivares	666	89877	6780638

- b. En el botón agregar al hacer click en el icono se generará otra pantalla para el registro de los accidentes.

- ❖ (a) Todos los campos de texto se registrarán los datos que pide el indicador de ayuda
- ❖ (b). En el botón guardar al hacer click registrara el guardado de los datos con un mensaje “Datos guardados satisfactoriamente”.
- ❖ (c). Botón cancelar, al hacer click regresara a la pantalla de inicio de repostes.
- ❖ (d). En el botón imprimir enviara a la pantalla de impresora instalada en su computadora, esto para imprimir el reporte del accidente.
- ❖ (e). En el botón seleccionar archivo al hacer click esto deberá enviar a la carpeta de selección de fotos del accidente en el ordenador

- ❖ (f). este botón generara el código del reporte
- ❖ (g). Este es un botón de selección que podrá escoger quien es el responsable del reporte.

c. En el botón x rojo realizara la acción de eliminar el registro de accidente

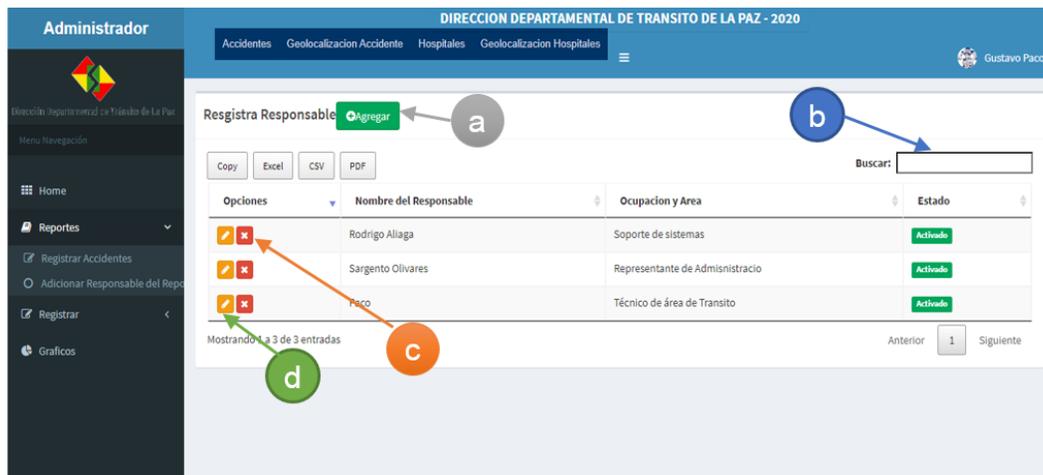
d. En el amarillo de modificar, enviara a otra pantalla para modificar y actualizar el registro.

- ❖ (a). En el campo se realizará la modificación de los datos ya registrados
- ❖ (b). El botón de guardar realizara la acción de modificar de los nuevos datos.
- ❖ (c). En el Botón seleccionar archivo podrá actualizar el nuevo archivo
- ❖ (d). El botón generar adicionara otro código del reporte

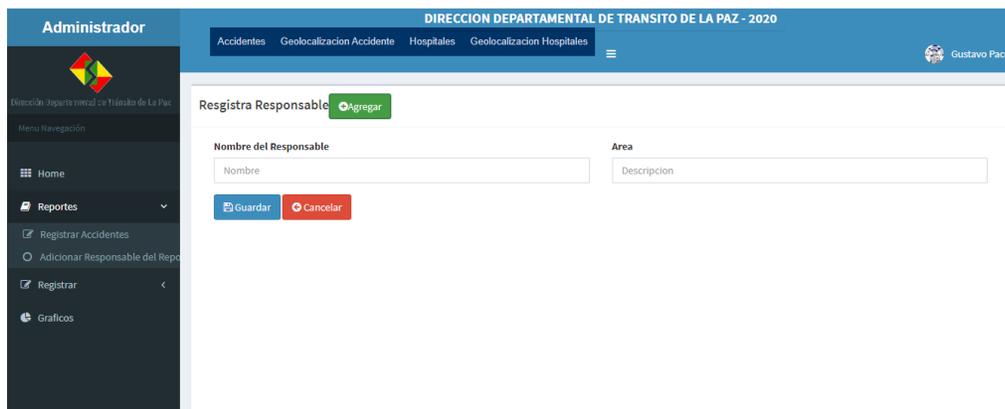
e. En el buscador deberás de ingresar el nombre o licencia del conductor y hacer click en el icono de buscar.

f. En el botón verde de activo se podrá ver si el reporte del accidente esta activo o inactivo.

Pantalla de inicio de reporte (Adicionar el responsable del reporte)



- ❖ (a). El botón agregar enviara a la pantalla de agregar responsable del reporte



- ❖ (b). En el buscador deberás de ingresar el nombre o responsable del reporte y hacer click en el icono de buscar.
- ❖ (c). En el botón rojo al hacer click podrá eliminar el registro de antiguo responsable del reporte.

Pantalla de inicio de reporte (Adicionar el responsable del reporte)

Registrar Administrador

Nombre(*):

Tipo Documento(*):

Direccion:

Email:

Login(*):

Privilegios:

- Registrar
- Reportes
- Geolocalización
- Listar Hospitales
- Eliminar
- Actualizar

Numero de Documento(*):

Telefono:

Cargo:

Clave(*):

Imagen:

Ningún archivo seleccionado

- ❖ (a). En el campo de texto se registrarán los datos personales del nuevo administrador según como muestra el indicador.
- ❖ (b) En este sector de privilegios se podrá seleccionar los privilegios que tendrá el nuevo administrador.
- ❖ (c). El botón de guardar al hacer click registrara los datos del nuevo administrador.
- ❖ (d). El botón rojo de cancelar enviara a la pantalla de inicio de administración de registro.
- ❖ (e). El botón seleccionar archivo se podrá subir la foto de perfil del nuevo administrador desde un archivo del ordenador.

Pantalla de inicio de registro (Ver privilegios)

Permisos

Copy Excel CSV PDF

Nombre:

Reportes

Registrar

Listar Hospitales

Geolocalización

Eliminar

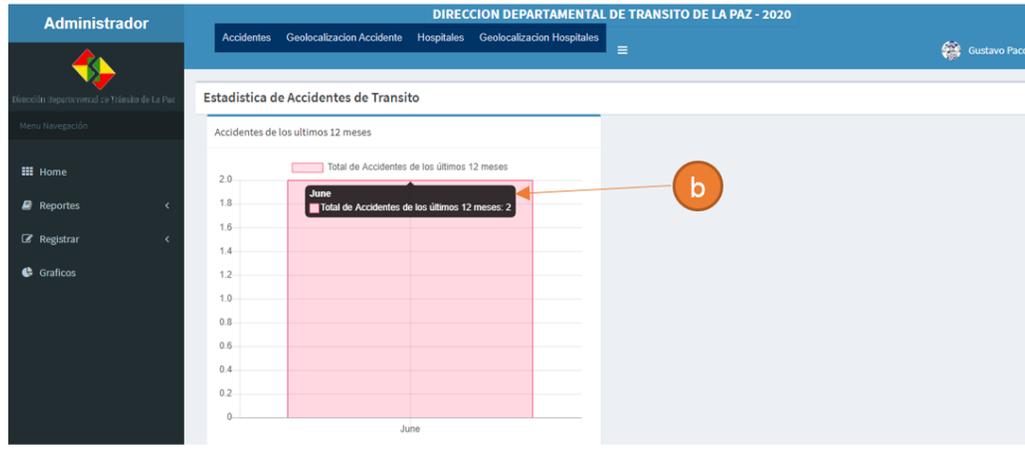
Nombre:

Mostrando 1 a 5 de 7 entradas

Anterior Siguiente

- ❖ (a). El botón PDF se podrá realizar la descarga del archivo
- ❖ (b). En esta sección se podrá ver los privilegios de cada administrador
- ❖ (c). En campo de buscador se realizara la buscado del nombre del administrador.

Pantalla de inicio de Gráficos



- ❖ (a). Aquí nos muestra el grafico de cada mes el registro de los accidentes

Pantalla de Registro de Accidentes

Administrador DIRECCION DEPARTAMENTAL DE TRANSITO DE LA PAZ

Accidentes Geocalcizacion Accidente Hospitales Geocalcizacion Hospitales Gustavo Paco

Panel de Registros de Accidentes de Tránsito

Registro envío de Datos del Lugar del Accidente

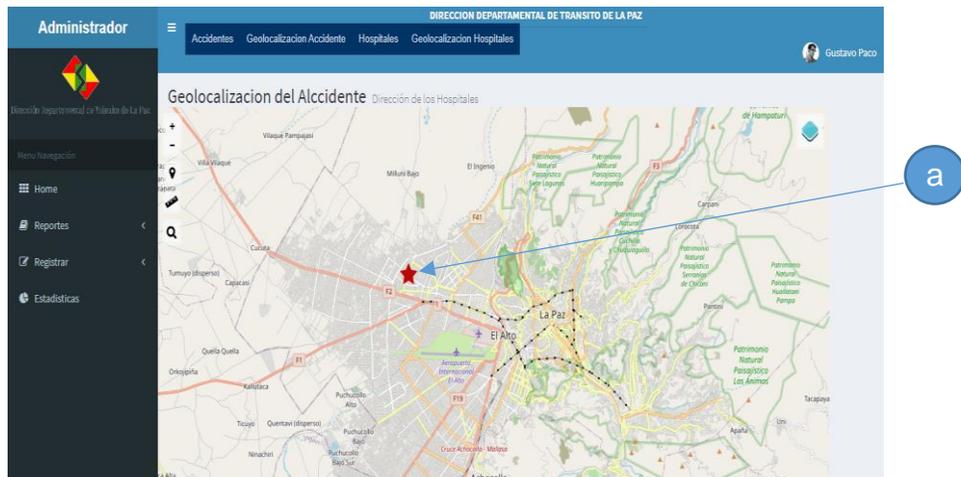
buscar ci o intensidad Buscar Notificar del Accidente

Id	Fecha	Velocidad	Latitud	Longitud	Precision	Ci	Intensidad
2	10/07/2020 15:23/10	0.15	-16.508186	-68.161915	320.2	6780638	64.785644396187
1	10/07/2020 09:45:26	0.18	-16.477258	-68.219685	220.0	6780638	64.008643565376

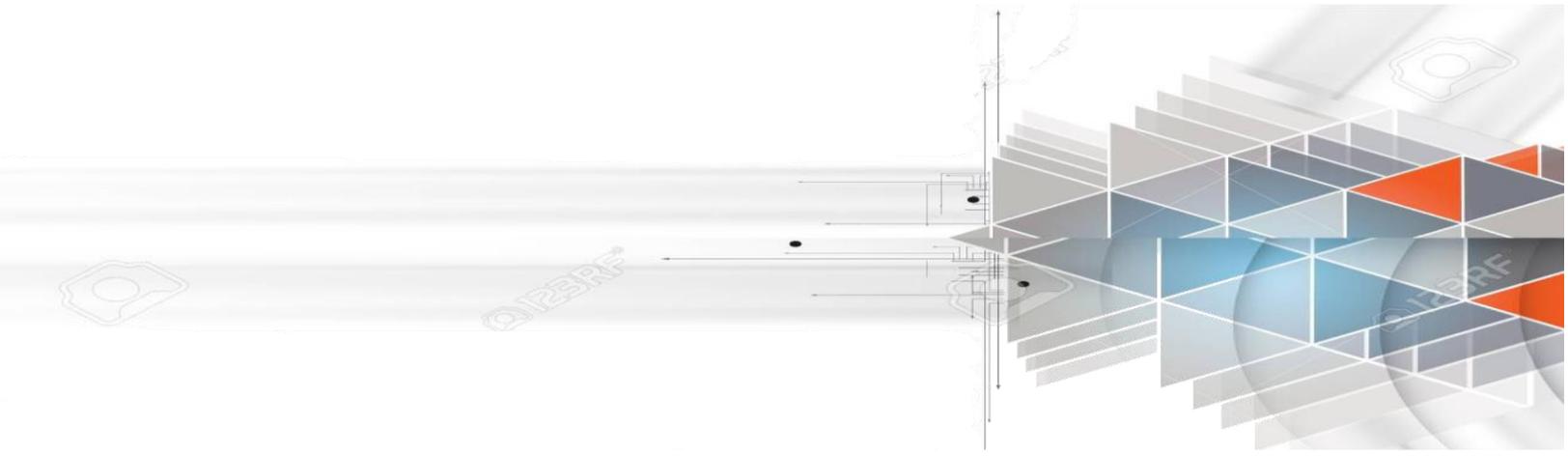
- ❖ Notificar accidente. al hacer click enviara los datos del conductor a través de notificaciones a todos los usuarios que tengan instalada la aplicación.

- ❖ En la pantalla se muestra todos los datos del lugar del hecho, que servirá de análisis del accidente para el tránsito
- ❖ En el campo de texto buscar se podrá ingresar CI. Para una búsqueda rápida de los registros del accidente

Geolocalización del Accidente



- ❖ En la parte del mapa se muestra el lugar del accidente, con los respectivos datos del conductor.



ANEXO E

MANUAL DE USUARIOS DE LA APLICACIÓN ANDROID

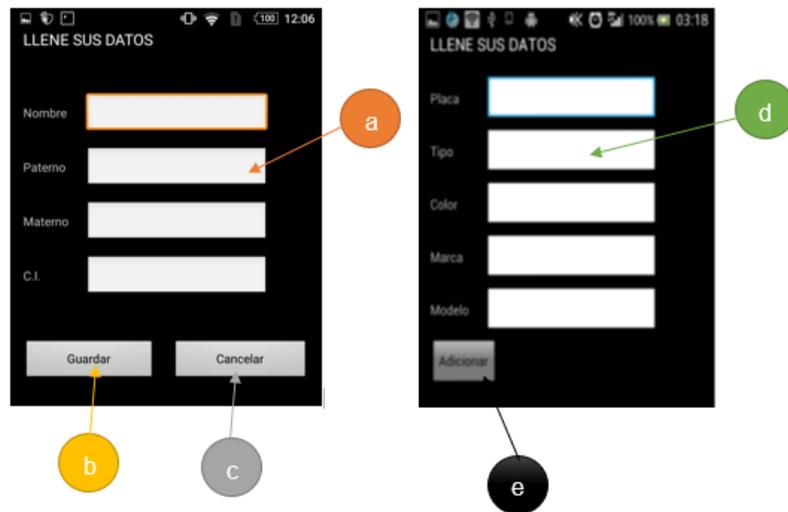


Pantalla de inicio de aplicación móvil



- ❖ (a). El botón azul, es el acceso a la pantalla principal donde el usuario podrá ver el mapa donde esta ubicados todos los hospitales con sus respectivos información y también se podrá ver el velocímetro.
- ❖ (b). El rojo al hacer click cierra sesión de la aplicación
- ❖ (c). Al hacer click en ese botón desplegara otra pantalla para el acceso de logeo.
- ❖ (d). Al hacer click mostrara la información o ayuda de la aplicación.

Pantalla de acceso de



- ❖ (a). El campo de texto se registrará todos los datos personales del usuario tal como indica la información.
- ❖ (b). El botón guardar al hacer click realizara la acción de guardar los datos registrados.
- ❖ (c). El botón cancelar al hacer click, enviara a la pantalla principal.
- ❖ (d). El campo de texto se registrará todos los datos o descripción de la movilidad.
- ❖ (e). El botón Adicionar al hacer click realizara la acción de adicionar los datos del vehículo.

Pantalla de acceso donde se monitoreará al usuario

Este botón permite visualizar una pantalla con dos divisiones: el lado izquierdo tiene un velocímetro que permite mostrar la velocidad con lo que va el vehículo y un botón que permite simular el accidente. En el lado derecho se visualiza la ubicación del vehículo en un mapa. Dentro del mapa existen dos botones: Visualizar u Ocultar Hospitales y una opción para ir al Hospital más cercano desde la ubicación del vehículo.



- ❖ (a). El botón recorrido al hacer click mostrara la información de recorrido
- ❖ (b). Muestra la información de velocidad con la que esta conduciendo en Km/h
- ❖ (c). El botón mostrar/ocultar hospitales visualizara los puntos donde se encuentra los hospitales en el mapa.
- ❖ (d). El botón hospital más cercano al presionar click mostrara el punto del hospital más cercano a la ubicación del usuario.
- ❖ (e). El botón selector de mantener la ubicación, al esta seleccionado en esta opción, mantendrá la ubicación actual del usuario.
- ❖ (f). El botón selector Estoy conduciendo mostrara la información del usuario y de su vehículo.
- ❖ (g). Son las marcaciones de todos los hospitales que visualizara en el mapa.
- ❖ (h). El mapa que se podrá desplegar, acercar y alejar

Pantalla de opciones (llamar hospital/trazar la ruta al hospital)



- ❖ (a). El botón Llamar al hacer click realizara la llamada al punto del hospital que se haya seleccionado.
- ❖ (b). El botón como llegar, al hacer click realizara el trazado de la ruta desde el punto de ubicación del usuario hacia punto marcado del hospital seleccionado.

Pantalla de Trazado de cómo llegar a un hospital determinado (al momento de hacer click al botón (como llegar))



- ❖ (a). El botón (Mostrar/ocultar) al hacer click ocultara la ruta trazado en el mapa.
- ❖ (b). Es el punto de ubicación de usuario
- ❖ (c). La línea roja es el trazado que se ha realizado a un punto del hospital
- ❖ (d). Es el punto de ubicación de hospital seleccionado.

Pantalla de login para acceder a la pantalla de principal



- ❖ (a). El botón Login al hacer click enviará a la pantalla principal del usuario, donde podrá adicionar los datos de la movilidad.
- ❖ (b). Es el campo de texto donde se registrará el CI del cliente registrado.
- ❖ (c). El botón Cancelar, al hacer click enviara a la pantalla principal.

AVAL DE TUTOR REVISOR

La Paz, 10 de julio de 2020

Señor
Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER II
Presente.

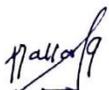
Ref.: AVAL DE CONFORMIDAD

Distinguido Ingeniero,

Mediante la presente tengo a bien comunicarle mi conformidad con el perfil de tesis de grado "MODELO WEB DE LOCALIZACIÓN Y REPORTES DE ACCIDENTES DE TRANSITO BASADO EN ANDROID" CASO: Dirección Departamental de Transito de La Paz que propone el postulante Univ. Gustavo Paco Machaca, con cedula de identidad 6780638 LP. Para su defensa publica, la evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente a la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,


LIC. María Magdalena Aguilar Guanto
TUTOR REVISOR

cc. Arch.

AVAL DE TUTOR ESPECIALISTA

La Paz, 09 de julio de 2020

Señor
Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER II
Presente.

Ref.: AVAL DE CONFORMIDAD

Distinguido Ingeniero,

Mediante la presente tengo a bien comunicarle mi conformidad con el perfil de tesis de grado **“MODELO WEB DE LOCALIZACIÓN Y REPORTE DE ACCIDENTES DE TRANSITO BASADO EN ANDROID” CASO: Dirección Departamental de Transito de La Paz**, que propone el postulante **Univ. Gustavo Paco Machaca**, con cedula de identidad 6780638 LP. Para su defensa publica, la evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente a la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



LIC. Norma Mamani Quispe
TUTOR ESPECIALISTA

cc. Arch.

AVAL DE INSTITUCIÓN



POLICIA BOLIVIANA

DIRECCIÓN DEPARTAMENTAL
TRANSITO Y TRANSPORTE Y SEGURIDAD VIAL DE LA CIUDAD DE LA PAZ



La Paz 9 de julio de 2020

Señor:
Ing. Enrique Flores Baltazar
Universidad Pública de El Alto
Carrera Ingeniería de Sistemas
Tutor Metodológico Taller de Licenciatura II

Presente:

Ref.: AVAL DE CONFORMIDAD

De mi mayor consideración: Tengo a bien dirigirme a su persona, para darle a conocer que luego de efectuar el seguimiento del proyecto titulado: **"Modelo Web de localización y reportes de Accidentes de Tránsito basado en Android"**, elaborado por el Universitario: GUSTAVO PACO MACHACA, con C.I. 8780838, en calidad de Encargado de Soporte Técnico de Área de mantenimiento (Unidad de Infraestructura de Sistemas) expreso mi conformidad con el proyecto en funcionamiento de prueba y la forma de trabajo, dando mi Aval de Conformidad, para que el postulante pueda realizar la defensa de la Tesis de Grado.

Sin otro particular, me suscribo de su persona con las consideraciones más distinguidas.


Eloy Mamani Mamani
Sargento Segundo
Unidad de Infraestructura de Sistemas

