



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

**IMPLEMENTACIÓN DE APLICACIÓN MÓVIL PARA REPORTE DE  
CRÍMENES, INTEGRADA CON LA PLATAFORMA FACEBOOK  
UTILIZANDO LA TECNOLOGÍA ANDROID GUATE SEGURA**

**Jhony Anthony Quiem Batz**

**Diego Alberto Salán Calderón**

Asesorados por el Ing. Edgar Estuardo Santos Sutuj

Guatemala, marzo de 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**IMPLEMENTACIÓN DE APLICACIÓN MÓVIL PARA REPORTE DE  
CRÍMENES, INTEGRADA CON LA PLATAFORMA FACEBOOK  
UTILIZANDO LA TECNOLOGÍA ANDROID GUATE SEGURA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**JHONY ANTHONY QUIEM BATZ**

**DIEGO ALBERTO SALÁN CALDERÓN**

ASESORADOS POR EL ING. EDGAR ESTUARDO SANTOS SUTUJ

AL CONFERÍRSELES EL TÍTULO DE

**INGENIEROS EN CIENCIAS Y SISTEMAS**

GUATEMALA, MARZO DE 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
EXAMINADOR	Ing. Marlon Francisco Orellana López
EXAMINADOR	Ing. José Alfredo González Díaz
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

## **HONORABLE TRIBUNAL EXAMINADOR**

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

### **IMPLEMENTACIÓN DE APLICACIÓN MÓVIL PARA REPORTE DE CRÍMENES, INTEGRADA CON LA PLATAFORMA FACEBOOK UTILIZANDO LA TECNOLOGÍA ANDROID GUATE SEGURA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, en septiembre de 2013.

**Diego Alberto Salán Calderón**

**Jhony Anthony Quiem Batz**

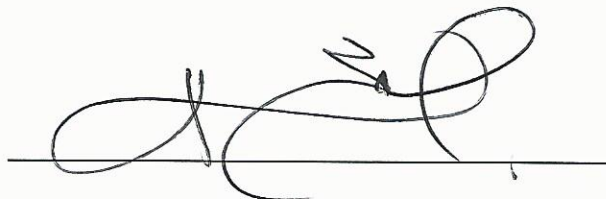
Guatemala, 18 de Noviembre de 2014

**Ingeniero  
Marlon Antonio Pérez Türk  
Director  
Escuela de Ciencias y Sistemas  
Facultad de Ingeniería  
Universidad de San Carlos de Guatemala**

Ingeniero Pérez Türk:

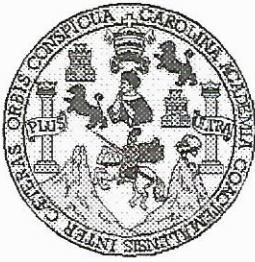
Me complace saludarle, haciendo referencia al trabajo de graduación titulado "IMPLEMENTACIÓN DE APLICACIÓN MÓVIL PARA REPORTE DE CRÍMENES INTEGRADA CON LA PLATAFORMA FACEBOOK UTILIZANDO LA TECNOLOGÍA ANDROID GUATE SEGURA", desarrollado por el estudiante universitario Diego Alberto Salán Calderón con número de carné 200815403 y el estudiante universitario Jhony Anthony Quiem Batz con número de carné 200819401, que como asesor apruebo el contenido del mismo.

Para su conocimiento y efectos, sin otro particular, me suscribo.

A handwritten signature in black ink, consisting of a series of loops and flourishes, positioned above a horizontal line.

Atentamente,  
Ing. Edgar Santos  
No. Colegiado: 5266  
Asesor

*Edgar Santos*  
INGENIERO EN CIENCIAS Y SISTEMAS  
Colegiado 5266



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

Guatemala, 11 de Febrero del 2015

Ingeniero  
**Marlon Antonio Pérez Türk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **DIEGO ALBERTO SALÁN CALDERÓN** con carné **2008-15403**, y **JHONY ANTHONY QUIEM BATZ** con carné **2008-19401**, titulado: **"IMPLEMENTACIÓN DE APLICACIÓN MÓVIL PARA REPORTE DE CRÍMENES INTEGRADA CON LA PLATAFORMA FACEBOOK UTILIZANDO LA TECNOLOGÍA ANDROID –GUATE SEGURA–"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

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

Atentamente,

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



UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
TEL: 24767644

E  
S  
C  
U  
E  
L  
A

D El Director de la Escuela de Ingeniería en Ciencias y  
E Sistemas de la Facultad de Ingeniería de la Universidad de  
C San Carlos de Guatemala, luego de conocer el dictamen del  
I asesor con el visto bueno del revisor y del Licenciado en  
E Letras, del trabajo de graduación **“IMPLEMENTACIÓN  
N DE APLICACIÓN MÓVIL PARA REPORTE DE  
C CRÍMENES, INTEGRADA CON LA PLATAFORMA  
I FACEBOOK UTILIZANDO LA TECNOLOGÍA  
A ANDROID GUATE SEGURA”**, realizado por los  
S estudiantes JHONY ANTHONY QUIEM BATZ y DIEGO  
Y ALBERTO SALÁN CALDERÓN, aprueba el presente trabajo  
y solicita la autorización del mismo.

S  
I  
S  
T  
E  
M  
A  
S

**“ID Y ENSEÑAD A TODOS”**

Ing. Marlon Antonio Pérez Türk

Director, Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 13 de marzo 2015





DTG. 113.2015

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **IMPLEMENTACIÓN DE APLICACIÓN MÓVIL PARA REPORTE DE CRÍMENES, INTEGRADA CON LA PLATAFORMA FACEBOOK UTILIZANDO LA TECNOLOGÍA ANDROID GUATE SEGURA**, presentado por los estudiantes universitarios: **Jhony Anthony Quiem Batz y Diego Alberto Salán Calderón**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Murphy Olympe Paiz Récinos  
Decano

Guatemala, 16 de marzo de 2015

/gdech





## **ACTO QUE DEDICO A:**

<b>Dios</b>	Por brindarme vida, salud, inteligencia y sabiduría. Por guiarme para el logro de esta meta.
<b>Mi madre</b>	Por el apoyo y confianza que me ha dado para alcanzar este gran éxito en mi vida, y por ser una persona digna de admirar en todo sentido.
<b>Mi familia</b>	Hermanos, padres, abuelos, tíos, primos, sobrinos, y demás familia. Por brindarme el apoyo necesario.
<b>Todas las personas</b>	Que de alguna manera han formado parte del cumplimiento de esta meta alcanzada; realizada con mucho esfuerzo, dedicación y perseverancia.

**Jhony Anthony Quiem Batz**

## **AGRADECIMIENTOS A:**

### **Dios**

Por la vida, la salud, la sabiduría e inteligencia necesaria que me ha brindado en cada momento para alcanzar esta meta. Por estar conmigo en todo momento apoyándome y guiándome para poder salir adelante.

### **Mi madre**

Francisca Batz, por todo el apoyo y amor incondicional que me ha brindado siempre. Por enseñarme que la mejor forma de obtener el éxito es con mucho esfuerzo y dedicación, sin rendirse nunca hasta que se alcancen las metas y los sueños propuestos. Por ser un ejemplo digno de superación y perseverancia. Por la confianza y apoyo brindado en todo momento. Gracias por todo madre, la amo.

### **Mis hermanos**

Por todos esos momentos que pasamos juntos. Por esos días de mucho esfuerzo y desvelos para alcanzar nuestros objetivos. Por los éxitos y derrotas que supimos disfrutar y afrontar, apoyándonos e integrándonos como un gran equipo.

**Universidad de  
San Carlos**

Especialmente a la Facultad de Ingeniería, por todo el conocimiento adquirido para culminar esta carrera profesional que tanto amo, y por permitirme cumplir mis sueños.

**Jhony Anthony Quiem Batz**

## **ACTO QUE DEDICO A:**

<b>Dios</b>	Por brindarme vida, salud, inteligencia y sabiduría, además de guiarme para el logro de esta meta propuesta en mi vida.
<b>Mis padres</b>	Por el apoyo y confianza que me brindaron para poder alcanzar este gran éxito en mi vida. Así como el amor incondicional hacia su hijo mayor.
<b>Mi familia</b>	Hermanas, abuelos, tíos, primos, y demás familia. Por brindarme el apoyo necesario.
<b>Todos aquellos</b>	Que me motivaron e incentivaron para el logro de esta meta.

**Diego Alberto Salán Calderón**

## **AGRADECIMIENTOS A:**

### **Dios**

Por la vida, la salud, la sabiduría e inteligencia necesaria que me ha brindado en cada momento para poder alcanzar esta meta. Por estar conmigo en todo momento apoyándome y guiándome para poder salir adelante durante todo el período universitario.

### **Mis padres**

Juan Carlos Salán y Violeta Calderón, por el apoyo y amor incondicional que me han brindado siempre. Por guiarme y aconsejarme en momentos difíciles. Gracias a ustedes logré alcanzar uno de los objetivos propuestos en mi vida.

### **Mis hermanas**

Alejandra y Natalia Salán Calderón, por el apoyo brindado siempre. Por la comprensión, cuidado y compañía en las noches de desvelo.

### **Mis abuelos**

Por los consejos y motivación brindada para seguir adelante. Por abrirme las puertas de su casa cada vez que fue necesario, así como por la confianza depositada en mi persona como ejemplo hacia mis primos.

**Mis amigos**

Brian Estrada, Erick Díaz, Pablo Figueroa, Lyonel Rosales, Darick Tuquer, Fredy Velásquez, Karen González, Giancarlo Guilá, Josué Flores y Diego López, amigos que con el tiempo se convirtieron como hermanos, apoyándonos y ayudándonos unos a otros a salir adelante. Sin ustedes este logro no hubiera sido posible.

**Universidad de  
San Carlos**

Especialmente a la Facultad de Ingeniería, por el conocimiento adquirido a lo largo de mi periodo universitario, con el cual alcancé uno de mis mayores sueños como profesional.

**Diego Alberto Salán Calderón**



## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO .....	VII
RESUMEN.....	XI
OBJETIVOS.....	XIII
INTRODUCCIÓN .....	XV
1. APLICACIONES MÓVILES .....	1
1.1. Sistema operativo Android.....	1
1.2. Arquitectura de Android .....	2
1.3. Dispositivos móviles en Latinoamérica y Guatemala.....	4
1.4. Aplicaciones móviles y seguridad ciudadana .....	7
1.5. Tendencias de las aplicaciones móviles.....	8
2. IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN PLANTEADA .....	11
2.1 Descripción del problema .....	11
2.2 <i>Benchmark</i> de la aplicación.....	12
2.2.1 Espantacacos .....	13
2.2.2 CrimePush.....	14
3. DISEÑO DE LA APLICACIÓN BAJO LA NECESIDAD IDENTIFICADA .....	17
3.1 Diseño de interfaces y usabilidad .....	17

4.	DOCUMENTACIÓN BASE PARA EL DESARROLLO DE LA APLICACIÓN .....	29
4.1	Documentación base para el desarrollo de la aplicación móvil.....	29
4.1.1	Tecnologías utilizadas.....	29
4.1.1.1	Google Maps v2 .....	29
4.1.1.2	Java.....	30
4.1.1.3	Facebook.....	30
4.1.1.4	MySQL .....	31
4.1.1.5	Metodología REST .....	31
4.1.2	Arquitectura de la aplicación web.....	32
4.1.2.1	Modelo.....	32
4.1.2.2	Vista .....	32
4.1.2.3	Controlador.....	33
4.1.3	Requerimientos para el desarrollo.....	33
4.1.3.1	Android SDK.....	33
4.1.3.2	Android Development Tools (ADT).....	34
4.1.4	Instalación y configuración de herramientas .....	34
4.1.4.1	Descarga e instalación de Android SDK.....	34
4.1.4.2	Android SDK Manager .....	34
4.1.4.3	Creación de un proyecto nuevo en Eclipse.....	35
4.1.4.4	Ambiente real de pruebas .....	36
4.1.5	Conceptos generales para desarrollo en Android ...	37
4.1.5.1	Actividad.....	38
4.1.5.2	Servicios.....	38
4.1.5.3	Intención.....	39
4.1.5.4	Android Manifest .....	39

	4.1.5.5	Ciclo de vida de componentes.....	39
4.2		Diseño de base de datos .....	40
	4.2.1	Modelo de base de datos .....	40
	4.2.2	Descripción de tablas .....	41
	4.2.3	Descripción de relaciones.....	45
4.3		Documentación de código fuente .....	46
	4.3.1	Arquitectura de software .....	47
	4.3.2	Patrones de diseño .....	48
	4.3.2.1	Data Transfer Object (DTO).....	49
	4.3.2.2	Data Access Object (DAO) .....	49
	4.3.3	Agrupación de clases .....	49
	4.3.4	Descripción de clases.....	51
	4.3.4.1	Clase de conexión a base de datos.....	52
	4.3.4.2	Clases de transporte de datos .....	53
	4.3.4.3	Clases de acceso a datos.....	56
	4.3.4.4	Implementación de servicios web .....	58
	4.3.5	Aplicación móvil .....	59
	4.3.5.1	Interfaz gráfica .....	59
	4.3.5.2	Google Maps API v2 .....	63
	4.3.5.3	Consumidor de servicios web .....	68
CONCLUSIONES .....			71
RECOMENDACIONES .....			73
BIBLIOGRAFÍA.....			75
APÉNDICES .....			79
ANEXO .....			103



## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Arquitectura Android .....	3
2.	Sistemas operativos en Suramérica.....	8
3.	Sistemas operativos en América Latina .....	9
4.	Crecimiento de telefonía fija y móvil.....	12
5.	Interfaz GuateSegura.....	17
6.	Pantalla inicial .....	18
7.	Listado de reportes .....	19
8.	Reportes sobre mapa.....	19
9.	Ingreso de reporte.....	20
10.	Llenado de formulario de reporte .....	21
11.	Componente de fecha y hora .....	21
12.	Ubicación de reporte .....	22
13.	Navegación de mapa .....	23
14.	Detalle de reporte.....	24
15.	Llamar a las autoridades.....	24
16.	Búsqueda general.....	25
17.	Filtros de búsquedas.....	26
18.	Delimitación de perímetro .....	26
19.	Despliegue de resultados.....	27
20.	Android SDK Manager .....	35
21.	Nueva aplicación Android .....	36
22.	Selección de dispositivo.....	37
23.	Ciclo de vida de una actividad Android .....	40

24.	Modelo relacional de base de datos .....	41
25.	Relación entre medio del delito y denuncia .....	45
26.	Relación entre tipo de denuncia y denuncia .....	45
27.	Relación entre denuncia y ubicación .....	46
28.	Relación entre ubicación y comisaría .....	46
29.	Paquetes definidos en el proyecto del servicio web .....	50
30.	Clases incluidas dentro de los paquetes del proyecto .....	51
31.	Pantalla inicial de aplicación .....	60
32.	Pantalla de ingreso de reportes .....	61
33.	Pantalla para búsquedas de reportes .....	62
34.	Librerías de soporte para Android .....	63
35.	Creación de proyecto en Google APIs.....	64
36.	Activación de servicios Google Maps .....	64
37.	Creación de un nuevo Android Key .....	65
38.	Generación de API Key de GuateSegura .....	66
39.	Credenciales registradas para GuateSegura.....	67
40.	API Key en AndroidManifest.xml de GuateSegura .....	67
41.	Permisos en AndroidManifest.xml de GuateSegura .....	68



## **GLOSARIO**

<b>Base de datos</b>	Colección de información organizada, la cual puede ser manipulada fácilmente.
<b>Campo</b>	Se refiere a cada una de las columnas que conforman una tabla en una base de datos.
<b>Chatear</b>	Acción y efecto de utilizar herramientas tecnológicas en la charla dos o más personas, por medio de dispositivos electrónicos.
<b>Consistencia</b>	Es una propiedad que indica que se debe empezar únicamente aquello que se puede acabar.
<b>Core</b>	Componente principal o esencial de algo, al que se suman o acoplan otros elementos para conformar una totalidad o conjunto.
<b>Debuggear</b>	Proceso de identificar y corregir errores de programación.
<b>Drivers</b>	Programa informático que posibilita la interacción entre el sistema operativo de algún dispositivo electrónico y un periférico o hardware.

<b>Durabilidad</b>	Asegura que una vez realizada una operación, esta persistirá con el tiempo y no existirán pérdidas de datos.
<b>Esquema</b>	Hace referencia a un esquema de base de datos conformado por un conjunto de tablas, con sus respectivos campos o atributos y relaciones.
<b>Framework</b>	Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver problemas de índole similar.
<b>Hardware</b>	Conjunto de componentes que conforman la parte material (física) de una computadora o de algún otro dispositivo electrónico.
<b>Integridad</b>	Es la propiedad que se encarga de asegurar que los datos son exactos y consistentes, es decir siempre intactos, sin que cambien o se deformen.
<b>Kernel</b>	Parte principal del sistema operativo.
<b>Llave foránea</b>	Es una limitación referencial entre dos tablas. Identifica una columna de una tabla, la cual hace referencia a otra columna de otra tabla, generalmente la llave primaria.

<b>Llave primaria</b>	Es una clave diseñada para identificar de manera única a los registros de una tabla a través de toda la estructura de la base de datos.
<b>Market</b>	Herramienta de software en la cual se encuentran almacenadas aplicaciones de diversa índole donde los usuarios pueden efectuar descargas libremente.
<b>Modelo relacional</b>	Es un modelo de datos de nivel lógico, que permite generar un esquema de base de datos a partir de su diseño.
<b>Persistencia</b>	Hace referencia a la propiedad de los datos, de sobrevivir de alguna manera en el tiempo.
<b>Redundancia</b>	Almacenamiento de los mismos datos, varias veces en diferentes lugares. Esto puede provocar incremento de trabajo, desperdicio de espacio e inconsistencia de los datos.
<b>Registro</b>	Se refiere a cada una de las filas en que se divide una tabla, conformadas por un conjunto de campos previamente definidos.
<b>Relación</b>	Hace referencia a la conexión que existe entre dos tablas o entidades en una base de datos.

<b>Requerimiento</b>	Atributo necesario dentro de un sistema, que puede representar una capacidad, característica o factor de calidad del sistema, de manera útil a los usuarios.
<b>Script</b>	Es un conjunto de comandos que se ejecutan para realizar una operación requerida.
<b>Sistema</b>	Conjunto de elementos que se interrelacionan para el logro de un objetivo específico.
<b><i>Smartphone</i></b>	Teléfono móvil con herramientas de software de diferentes tipos, que cuenta con un sistema operativo que hace que el dispositivo sea inteligente.
<b>Software</b>	Programa de computadora que realiza tareas específicas.
<b>Tabla</b>	Se refiere al tipo de modelado de datos en donde se guardan los mismos; su estructura permite organizar y manipular la información.
<b><i>Tablets</i></b>	Dispositivo móvil que permite procesar información con componentes informáticos en su estructura.

## RESUMEN

El proyecto GuateSegura es un software para dispositivos móviles que permite divulgar información sobre robos, crímenes y cualquier acto delictivo en la ciudad capital, para que los usuarios tomen las medidas preventivas necesarias.

La aplicación utilizará diversas tecnologías con las que cuentan ahora los dispositivos móviles como GPS e implementará los diferentes *frameworks* para poder utilizar las APIs de Google y así mostrar diferentes ubicaciones geográficas.

El proyecto es importante ya que se concentra en uno de los problemas que más afectan a nuestra sociedad en estos días, dado a que la delincuencia, puede decirse que tiene atemorizada a la sociedad guatemalteca y si no se toman medidas, el problema puede agravarse con el tiempo.

Uno de los propósitos de la aplicación es formar comunidades más seguras, ya que esto permitirá realizar denuncias de manera más simple, mostrando la comisaría de policía o la manera de establecer comunicación telefónica con las autoridades.

Se espera que el software reporte un beneficio para la sociedad, debido a que se estará proporcionando un instrumento con tecnología moderna para aplicarse, con el afán de minimizar los problemas planteados anteriormente.





## **OBJETIVOS**

### **General**

Desarrollar por medio de la tecnología un software para dispositivos móviles que permita denunciar sobre robos, crímenes y cualquier acto delictivo en la ciudad capital, a la vez informar a los usuarios sobre la problemática de violencia que se vive actualmente en la ciudad.

### **Específicos**

1. Ubicar las zonas más golpeadas por la delincuencia y violencia dentro de la ciudad capital.
2. Establecer cuáles son los principales actos de violencia en el perímetro citadino.
3. Notificar a los usuarios cuál es la aplicación utilizada sobre cualquier reporte realizado dentro de un perímetro específico.



## INTRODUCCIÓN

A medida que el tiempo avanza la tecnología se vuelve cada vez más barata y accesible. Los dispositivos se producen de manera masiva, disminuyendo así su costo y tamaño físico. Esto se ha enfatizado más en los últimos 15 años. Por esta razón ahora es más fácil contar con dispositivos inteligentes tales como *tablets* o *smartphones*.

La Universidad de San Carlos de Guatemala, como única universidad estatal, pretende poner a disposición de todas las personas estas nuevas tecnologías, buscando el progreso a través del fácil acceso a la información. La Escuela de Ciencias y Sistemas ha creado un grupo de nuevas tecnologías, en un intento de promover la innovación para solucionar problemas comunes a un grupo y población en general. Este documento contiene una guía técnica en donde se aplican conceptos técnicos tales Google Maps, bases de datos, servicios web, entre otros conceptos aplicables para dispositivos móviles.

El documento está dividido en cuatro capítulos. En el primero se expone una descripción de los conceptos con los que se están tratando, así como la razón por la que se incluyen en la solución. El segundo capítulo contiene la problemática y la manera en que se resolverá la misma. El tercero contiene una introducción sobre las tecnologías que se utilizaron en el desarrollo del presente trabajo de graduación. Por último se describe la manera y procedimiento que se siguió para crear la solución de software, así como el uso corriente de la aplicación.



# **1. APLICACIONES MÓVILES**

El auge que han tenido los dispositivos móviles como *tablets* o teléfonos inteligentes en los últimos años, ha permitido que una gran cantidad de necesidades que surgen en el día a día del ser humano, se vean solucionadas por medio de aplicaciones instaladas en este tipo de dispositivos, tales como las aplicaciones de agendas electrónicas, noticias, directorios y mapas para ubicar ciertos lugares, etc.

Actualmente, una gran diversidad de usuarios finales optan por utilizar más un dispositivo móvil que una computadora portátil o de escritorio. Esto se ve reflejado en que la tasa de conexiones a internet desde una computadora ha disminuido en los últimos años, razón por la cual existen muchas aplicaciones para dispositivos móviles enfocadas a las diferentes plataformas que utiliza cada uno de ellos.

## **1.1. Sistema operativo Android**

Este es un sistema operativo basado en Linux utilizado por millones de dispositivos móviles principalmente por aquellos que tienen pantalla táctil. Cabe mencionar que junto con el núcleo Linux trabaja Dalvik, que es una máquina virtual de Java que se ejecuta sobre el núcleo del sistema.

Al contrario de otros sistemas operativos, este se desarrolla de forma abierta y muchas veces se puede acceder al código fuente o la lista de incidencias de algunas aplicaciones, donde se pueden ver problemas aún no resueltos o reportar ciertas situaciones específicas.

Actualmente se considera uno de los modelos de negocio de mayor éxito; esto se ve reflejado en las cifras que presentó International Data Corporation (IDC), donde Android está en cada tres de cuatro teléfonos que se vendieron en el tercer cuarto del 2012. Esto es un crecimiento del 91.5 % en un año donde el mercado creció casi al doble al anterior, cuando obtuvo un 46.4 % del mercado durante el 2011.

Es también el sistema operativo móvil que más se utiliza por parte de las compañías de telecomunicaciones y que más aceptación tiene por parte de los usuarios, ya que hay 136 millones de teléfonos Android (a nivel mundial) en el período de julio a septiembre del 2012, lo que equivale al 75 % del total de teléfonos vendidos en ese lapso (181 millones).

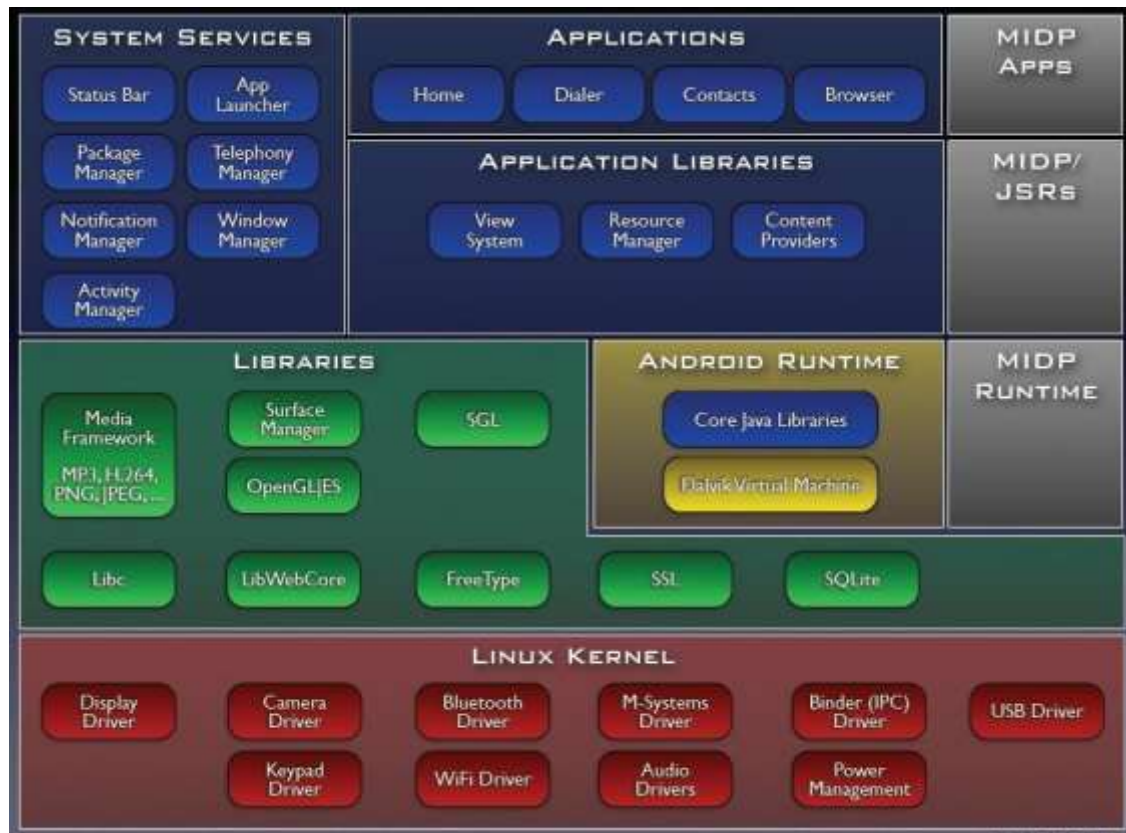
Cuenta también con una plataforma para crear aplicaciones y un sitio para la distribución de estas (*marketplace*).

## **1.2. Arquitectura de Android**

La principal plataforma que utiliza Android es ARM, que es una arquitectura RISC (ordenador con conjunto de instrucciones reducidas) de 32 bits. El núcleo del sistema es Linux en su versión 2.6.

Una de las grandes ventajas que muestra la arquitectura de este sistema operativo, es que esta se encuentra formada por diferentes capas, lo cual es de beneficio para los desarrolladores ya que por medio de librerías no es necesario programar a bajo nivel las funcionalidades de su aplicación.

Figura 1. **Arquitectura Android**



Fuente: <http://www.codeproject.com/Articles/803452/Article-Very-Basic-Introduction-to-Android>. Consulta: 13 de noviembre de 2013.

Como se puede observar en la figura anterior, se tienen diferentes niveles o capas en la arquitectura Android, las cuales son:

- **Aplicaciones:** en esta sección se encuentran todas las aplicaciones que trae por defecto el sistema operativo, así como las que el usuario vaya agregando conforme pase el tiempo. Todas estas corren sobre la máquina virtual Dalvik para garantizar la seguridad del sistema.

- *Framework* de aplicaciones: esta capa consiste básicamente en todas las clases y servicios que utilizan de manera directa las aplicaciones para poder realizar sus funciones. Diseñada principalmente para simplificar la reutilización de componentes, está formada por librerías Java que acceden a los recursos de capas anteriores.
- Librerías: comprendida por librerías en C/C++ usadas en diferentes componentes del sistema, estas están compiladas en código nativo del procesador, las principales se encargan de tareas como el motor gráfico, formatos de audio, imagen y vídeo, cifrado de comunicaciones, fuentes de texto, etc.
- Tiempo de ejecución de Android: se encuentra la máquina virtual Java Dalvik junto con algunas librerías *core* del sistema. No es una capa como tal esta sección de la arquitectura, se implementa así, debido a que las aplicaciones se compilan una única vez y se cuenta con la garantía que podrán ejecutarse en cualquier dispositivo Android que cuente con la versión del sistema operativo mínimo que requiera la aplicación.
- *Kernel* Linux: utiliza el núcleo de Linux 2.6 como una capa de abstracción entre el *hardware* y el resto de capas de la arquitectura. Contiene los *drivers* necesarios para cualquier componente de *hardware* que pueda ser utilizado mediante llamadas del sistema.

### 1.3. Dispositivos móviles en Latinoamérica y Guatemala

De acuerdo con Euromonitor International, Latinoamérica tuvo el mercado con el mayor crecimiento en volumen de ventas al por menor en cuanto a dispositivos móviles en el 2012, con un aumento de 61 % en *tablets* y 414 % en



teléfonos inteligentes respecto del año anterior.<sup>1</sup> Las estadísticas reflejan que Brasil y México son los países con el mercado más grande para la telefonía móvil en relación con unidades vendidas. Algunos otros países o regiones donde se reflejó este fenómeno son:

- Brasil: 227.4 millones de celulares, estadísticamente 11.65 celulares por habitante.
- México: 96.2 millones de celulares, estadísticamente 8.87 celulares por habitante.
- América Central: 52 millones de habitantes de celulares, estadísticamente 12.24 celulares por habitante.
- El Caribe: 22.9 millones de celulares, estadísticamente 5.88 celulares por habitante.

De los celulares vendidos durante este período, el 40 % fueron destinados a personas entre 25 y 35 años de edad, respectivamente.<sup>2</sup> El incremento de usuarios que utilizan este tipo de dispositivos repercute en muchos aspectos. Uno de ellos es el uso que se le da a una conexión a internet desde una computadora, ya que años atrás las personas se conectaban con mayor frecuencia desde una computadora para realizar ciertas actividades como *chatear*, redes sociales, correo electrónico. Actualmente las principales actividades son el correo electrónico, bancas en línea y compras en sitios web.

---

<sup>1</sup> Americaeconomica. <http://www.americaeconomia.com/analisis-opinion/el-boom-de-los-dispositivos-moviles-en-america-latina>.

<sup>2</sup> Elwebmarketer. <http://elwebmarketer.com/estado-de-la-telefonía-movil-en-latinoamerica-2012/2012/05/>.

Chatear y ver redes sociales se ha disminuido considerablemente, ya que es uno de los principales atractivos para un usuario al momento de adquirir un dispositivo móvil, así como lo es también el poder enviar y recibir imágenes o videos.

Según CentralAmericaData, en Guatemala durante el 2012 la telefonía celular alcanzó casi los 21 millones de usuarios, incrementándose así un 6.7 % en comparación con el 2011.<sup>3</sup>

Alexis Batres en el diario “El Periódico” afirma que en el primer trimestre de este año se contabilizaban un poco más de 19.5 millones de líneas activas, lo que equivale a una caída de 4.6 % en comparación con el censo anterior; este fue el primer retroceso en la activación de líneas móviles desde la apertura de comunicaciones hace ya 16 años; esto se debe a que muchas personas a finales del 2011 contaban con 2 o 3 celulares, por lo que el reacomodo en cuanto a cantidad de teléfonos activos es comprensible y previsible.<sup>4</sup>

El país ha tenido un crecimiento significativo en cuanto al uso de tecnología celular. Esto no solo beneficia a los ciudadanos para tener una comunicación sino que las facilidades que proporcionan ciertas aplicaciones, así como el acceso a información, ya que se cuenta con una conexión a internet en todo momento.

---

<sup>3</sup>Centralamericadata. [http://www.centralamericadata.com/es/article/home/Guatemala\\_Telefonia\\_mvil\\_crece\\_67](http://www.centralamericadata.com/es/article/home/Guatemala_Telefonia_mvil_crece_67)

<sup>4</sup> elperiodico <http://www.elperiodico.com.gt/es//economia/221349>

#### **1.4. Aplicaciones móviles y seguridad ciudadana**

Anteriormente se han desarrollado proyectos o investigaciones que ataquen de cierta forma la delincuencia y violencia que afecta a la sociedad, pero ninguna ha tenido el éxito esperado. Por la misma razón, no se cuenta con una herramienta que presente un beneficio significativo a la población.

Cabe mencionar también que Guatemala no ha sido el único país que ha tenido la iniciativa de realizar aplicaciones móviles que ayuden a los ciudadanos a tomar ciertas precauciones o consideraciones sobre la violencia y delincuencia que los rodea diariamente.

Crime Push es una aplicación estadounidense desarrollada para los y Android, la cual fue lanzada este año y ha tenido mucha aceptación en los Estados Unidos de América. Permite realizar diferentes funciones como notificar a los usuarios deseados que se ha arribado a algún destino sin ningún percance, llamar a las autoridades en modo silencioso con solo agitar el dispositivo. Así como reportar crímenes con evidencia como fotos, vídeos o grabaciones de audio.

Actualmente se están desarrollando otros proyectos como Ciudad Segura en Bolivia, donde se pretende registrar actos de violencia y poder realizar una denuncia anónima a la policía. Y como El botón del pánico que es una aplicación que envía a todos los contactos que se encuentran registrados en el teléfono, la ubicación exacta de donde se encuentra, puede ser configurable para que los envíe vía SMS, por redes sociales o correo electrónico.

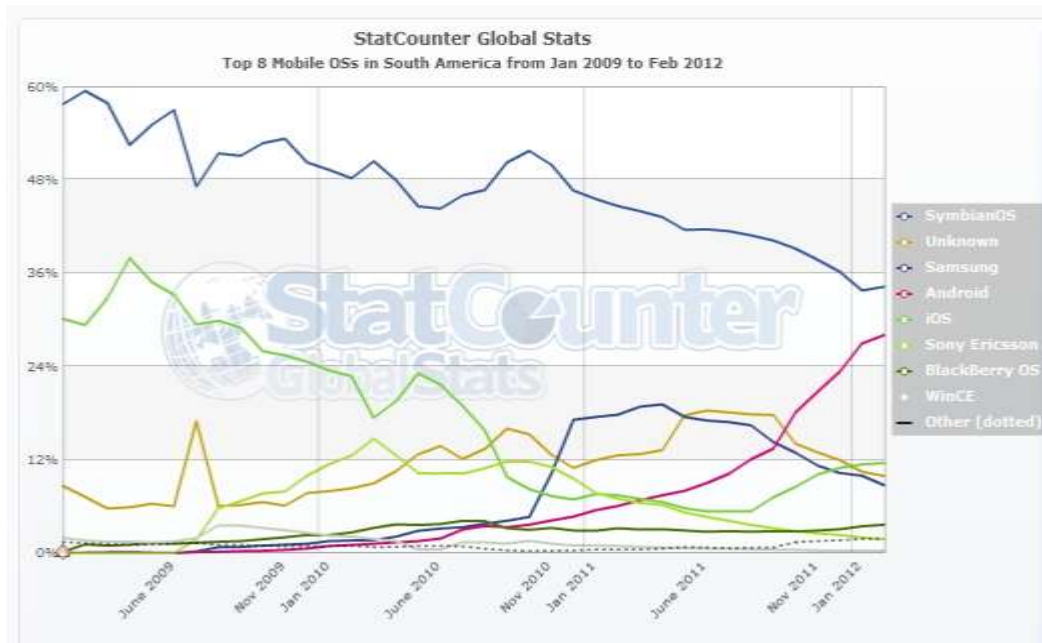
Cabe recalcar que con el uso adecuado de una aplicación de este tipo puede beneficiarse a los ciudadanos de una manera significativa, ya que se

pueden tomar medidas preventivas, dado el alto nivel de violencia y delincuencia por el que se ve afectado nuestra sociedad día a día.

### 1.5. Tendencias de las aplicaciones móviles

En América Latina, inicialmente, con el lanzamiento de los teléfonos inteligentes, Symbian OS era el líder del mercado. Incluso a principios del 2012, se puede ver también el crecimiento bastante rápido de Android y cómo fue descendiendo junto con Symbian OS, lo que puede apreciarse en la siguiente figura.

Figura 2. **Sistemas operativos en Suramérica**



Fuente: <https://www.wayerless.com/2012/02/los-sistemas-operativos-moviles-mas-usados-para-navegar-en-hispanoamerica/>. Consulta: 13 de noviembre de 2013.

Hugo Barra confirmó a inicios del presente año que se tiene un promedio de 1,3 millones de celulares con el sistema operativo Android nuevos por día, convirtiéndose así en el sistema operativo móvil más utilizado a nivel mundial; Es importante mencionar que con el lanzamiento del Iphone 5 se tiene una proyección de ventas para finales de este año de 527 millones de unidades. Es decir alcanzará la marca de Android el próximo año; esto en cuanto a unidades activadas, no así en usuarios que utilicen el sistema operativo.

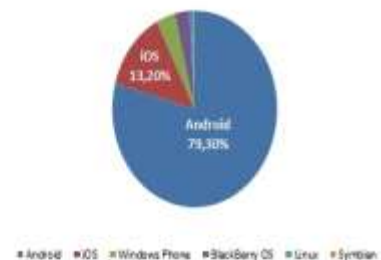
De acuerdo con las publicaciones de IDC del 18 de agosto del 2013, se puede ver claramente que Android es el sistema operativo líder a nivel mundial, seguido por iOS.

Figura 3. **Sistemas operativos en América Latina**

Top Smartphone Operating Systems, Shipments, and Market Share, 2013 Q3 (Units in Millions)

Operating System	2Q13 Unit Shipments	2Q13 Market Share	2Q12 Unit Shipments	2Q12 Market Share	Year-over-Year Change
Android	187.4	79.3%	108	69.1%	73.5%
iOS	31.2	13.2%	26	16.6%	20.0%
Windows Phone	8.7	3.7%	4.9	3.1%	77.6%
BlackBerry OS	6.8	2.9%	7.7	4.9%	-11.7%
Linux	1.8	0.8%	2.8	1.8%	-35.7%
Symbian	0.5	0.2%	6.5	4.2%	-92.3%
Others	N/A	0.0%	0.3	0.2%	-100.0%
<b>Total</b>	<b>236.4</b>	<b>100.0%</b>	<b>156.2</b>	<b>100.0%</b>	<b>51.3%</b>

Q2 2013 Market Share



Fuente: <http://9to5mac.com/2013/08/07/idc-android-reaches-80-of-market-in-q2-due-to-low-cost-offerings-ios-falls-to-13/>. Consulta: 13 de noviembre de 2013.

Según los análisis de esta compañía, Windows Phone que es uno de los sistemas más recientes en el mercado, tendrá un alza en los próximos dos años. Siendo posiblemente, otro fuerte competidor en el mercado en cuanto a dispositivos móviles.

Según la compañía estadounidense Gartner las tiendas de aplicaciones móviles alcanzarán este año un total de 102 millones de descargas ante los 64 millones del año pasado, de las cuales el 91 % de las descargas han sido aplicaciones gratuitas.

Brian Blau, el director de investigaciones de dicha empresa expuso en una conferencia: “las aplicaciones gratuitas representan actualmente alrededor del 60 % de las descargas sobre aplicaciones móviles y el 80 % del total de las aplicaciones disponibles en las tiendas de aplicaciones de Apple y Google play, respectivamente”.<sup>5</sup>

Según estudios se espera que para el 2017, las tiendas de Apple, junto con los diferentes *markets* de tiendas Android, acaparen el 90 % de descargas a nivel mundial en aplicaciones para móviles.

---

<sup>5</sup> Siliconnews. <http://www.siliconnews.es/2013/09/19/en-2013-habra-102-000-millones-de-descargas-de-aplicaciones-moviles/>.

## **2. IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN PLANTEADA**

GuateSegura es una aplicación que permite a los usuarios tomar ciertas medidas preventivas sobre ciertos lugares que transitan o recurren frecuentemente. Ya que por medio de esta será posible visualizar los lugares donde se cometen más crímenes, o los reportes alrededor de un lugar específico.

### **2.1. Descripción del problema**

El proyecto GuateSegura es tan importante, que se concentra en un problema que tiene vigencia actualmente. Debido a que la delincuencia puede decirse que tiene de rodillas a la sociedad guatemalteca y si no se toman las medidas necesarias este problema puede agravarse con el tiempo.

Asimismo es necesario plantear que este problema afecta a toda la sociedad guatemalteca, tomando en cuenta los factores económicos, sociales, psicológicos y de salud de los afectados.

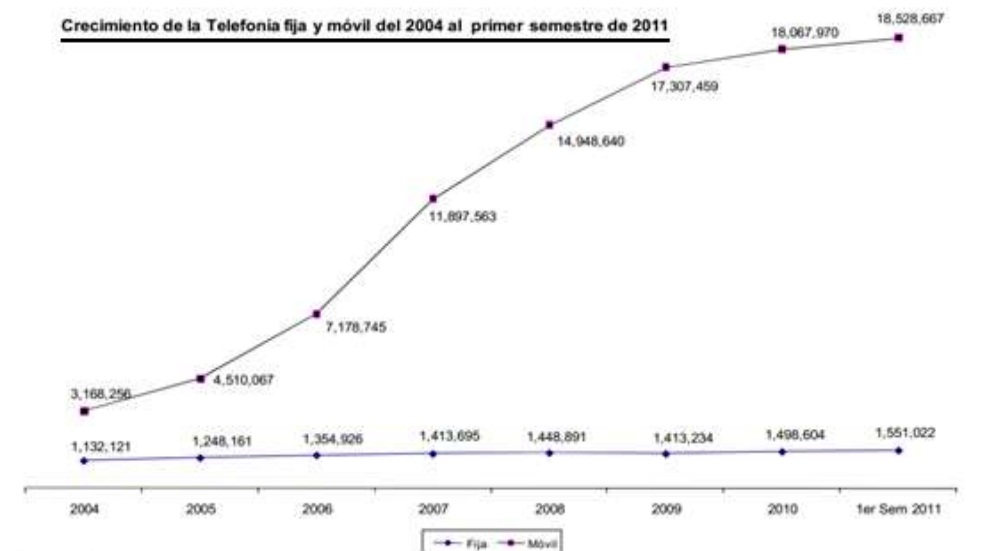
Anteriormente, para solventar la problemática descrita se han realizado investigaciones y proyectos previos como Espantacacos; a través del cual se podía denunciar de manera pública cualquier tipo de acto delictivo; lamentablemente esa aplicación, después de ser lanzada, fue sacada del mercado por motivos desconocidos.

Es importante recalcar que si bien ya existen ciertas aplicaciones que tienen funcionalidades similares a GuateSegura, las mismas están orientadas a un uso específico o particular. Además, no permiten que alguna entidad externa amplíe la funcionalidad con la que ya cuentan. Ese es un agregado que tendrá la aplicación donde se muestra el funcionamiento y la lógica interna para que otras personas puedan beneficiarse de ella.

## 2.2. **Benchmark de la aplicación**

En esta sección se listan aplicaciones móviles que tienen relación o funcionalidades similares en relación con la seguridad ciudadana. Se presentan las aplicaciones más grandes, que son las que mejor aceptación han tenido por parte del mercado. En la siguiente figura se puede ver el crecimiento de la telefonía fija y móvil del 2004 al 2011.

Figura 4. **Crecimiento de telefonía fija y móvil**



Fuente: Superintendencia de Telecomunicaciones de Guatemala, 2011.



### **2.2.1. Espantacacos**

Esta aplicación provee una interfaz gráfica donde se puede denunciar todo acto delictivo. Esta permite ingresar el tipo de crimen, ubicación, hora, adjuntar imágenes que se tomaron durante el crimen, etc.

Una vez ingresado el reporte, esta se almacenaba en una base de datos, donde después podía ser visualizado por otros usuarios. Estos se mostraban de forma gráfica por medio de Google Maps.

Otra de las funcionalidades importantes con las que cuenta la aplicación es el envío de alertas a los usuarios. Esta funcionalidad se refiere a ingresar una ubicación específica y a partir de lo ingresado una cantidad determinada (que ingresa también el usuario) de kilómetros a la redonda, notificará a los usuarios sobre cualquier reporte registrado en la aplicación, vía correo electrónico. Esto también puede filtrarse, dependiendo de la categoría de crimen que desee ingresar el usuario.

Cabe mencionar que una de las deficiencias de la aplicación era el entorno gráfico, ya que no era intuitivo para el usuario final. Esto dificultaba un poco su uso, ya que muchas veces se querían realizar acciones y no era tan fácil encontrar la manera de hacerlo.

Se realizaron campañas publicitarias para promocionar esta herramienta y que tuviera aceptación en el mercado guatemalteco. Esta aplicación fue desarrollada por un grupo llamado Jóvenes contra la violencia; la cual cuenta también con una página web que se alimenta de la misma base de datos que la aplicación móvil. Lamentablemente, por motivos desconocidos la aplicación ya no tuvo soporte y se encuentra prácticamente fuera del mercado.

### **2.2.2. CrimePush**

La misión de CrimePush se centra en empoderar a los ciudadanos con las herramientas necesarias que necesitan para ayudar a construir comunidades más seguras. Esto lo dan a conocer en su página web oficial.

Esta herramienta fue desarrollada por usuarios estadounidenses para ese país es un poco más compleja en cuanto al alcance de la aplicación. Además de estar desarrollada únicamente para dispositivos Apple, también cuenta con una versión para utilizarla con dispositivos que utilicen el sistema operativo Android.

Una de sus principales características es que cuenta con mensajería *distress*, la cual permite a los usuarios enviar varios mensajes de socorro o pidiendo ayuda, llevan una etiqueta GPS para saber la ubicación exacta de la víctima en caso de un percance, esto con tan solo presionar un botón.

Cuenta también con un mecanismo de notificaciones donde las autoridades pueden enviar un mensaje a cierta población en particular, alertando que hay peligro, por ejemplo en la universidad se pueden alertar a los estudiantes que hay un hombre armado en cierto edificio.

Otra funcionalidad que posee la aplicación es que en caso de que un usuario se encuentre en un entorno o situación peligrosa. El agitar el teléfono activa un mecanismo en el cual se realiza una llamada al 911. Esta sección también contiene una caja de seguridad cargada con herramientas como mapas, linterna y números de contacto de emergencias.

Además tiene una sección de entrada, que permite notificar a los amigos, familiares o cualquier otra persona, que ha arribado a su destino sin ninguna novedad, o que están atrasados para evitar preocupaciones. Por último, esta la sección de reportería de crímenes, en la cual se puede informar sobre todo tipo de delitos desde el vandalismo hasta robos discretos y eficientes (crimen organizado). Ofrece a las autoridades una forma más clara y rápida de un crimen específico, así como adjuntar fotos, vídeos o audio que formen parte de la evidencia recaudada sobre un crimen en particular que se desee reportar.

Forbes, revista internacional reconocida como líder innovadora en denuncia de delitos ante las autoridades y de envío de mensajes de auxilio para la familia y amigos en caso de alguna emergencia. Señala que una de las características que más se menciona es la cualidad de la herramienta de asistencia rápida, discreta e inteligente.

El alcance que tiene esta aplicación es bastante considerable, ya que actualmente al instalar la herramienta no puede está vinculada o conectada con la policía local, médicos o autoridades estatales. Tampoco se cuenta con todas las asistencias de cada una de las estaciones de emergencia o crímenes; los informes que se generan al momento de reportar un crimen por parte de los usuarios pueden ser enviados a las autoridades, única y exclusivamente si estas se han vinculado con CrimePush.

En el *market* de Android, se puede encontrar la aplicación buscándola por su nombre. Puede verse que la puntuación que le ha dado la audiencia que ha utilizado la herramienta ha sido bastante buena, ya que tiene un total de 4.4 estrellas. Mostrando de esta manera que es una aplicación bastante buena, la cual ha tenido muy buena aceptación a nivel de mercado, delimitándola en el territorio de los Estados Unidos de América.

Es importante ver el crecimiento que ha tenido con el tiempo esta aplicación, ya que inicialmente fue lanzada hacia un mercado específico: los usuarios particulares. Actualmente contienen una sección para empresas como comisarías de policía, universidades, etc.

### 3. DISEÑO DE LA APLICACIÓN BAJO LA NECESIDAD IDENTIFICADA

#### 3.1. Diseño de interfaces y usabilidad

Al iniciar el programa se despliega una pantalla de inicio, donde se carga todo lo necesario para que la aplicación tenga un desempeño óptimo. Se realizó una pantalla de bienvenida en lo que se iniciaban estos componentes como el *web service* o la base de datos.

Figura 5. Interfaz GuateSegura



Fuente: elaboración propia.

La primera pantalla que despliega la aplicación es sobre su posición inicial, permitiendo visualizar los reportes ingresados alrededor de la posición actual. Además es posible realizar diversas acciones tales como denunciar, hacer búsquedas de reportes, navegar en el mapa para ver reportes desplegados en el mapa, entre otros. Vea figura 6.

Figura 6. **Pantalla inicial**



Fuente: elaboración propia.

Al momento de ver las noticias o últimos reportes ingresados se despliega una lista con todos los reportes que se encuentran en la base de datos, por medio de un tap se puede visualizar el detalle de estos.

Figura 7. **Listado de reportes**



Fuente: elaboración propia.

Al posicionarse sobre la parte del mapa, pueden verse los reportes pero en la ubicación donde ocurrieron. Estos se desplegarán con un marcador y siempre se mostrará la posición de quien reporte.

Figura 8. **Reportes sobre mapa**



Fuente: elaboración propia.

Para ingresar un reporte es necesario que el usuario ingrese cierta información sobre el crimen. Los parámetros son un título, el tipo de delito, medio del delito fecha y hora, así como una descripción y ubicación donde ocurrió el crimen.

Figura 9. **Ingreso de reporte**



The screenshot shows a mobile application interface for reporting a crime. The title bar at the top is black with white text that reads 'Ingresar reporte'. To the right of the title are two icons: a magnifying glass and a checkmark. Below the title bar, there are several input fields and dropdown menus. The first field is labeled 'Titulo' and has a blue underline. Below it is a dropdown menu labeled 'Tipo de delito' with 'Abuso' selected. Below that is another dropdown menu labeled 'Medio de delito' with 'Bus' selected. Below these are two date and time pickers: 'Fecha del delito' showing '17 de nov.' and 'Hora del delito' showing '17:21'. At the bottom is a large text area labeled 'Descripción del delito'.

Fuente: elaboración propia.

Se utilizaron listas de datos donde se muestran las diferentes categorías a las que podría pertenecer el crimen a reportar, se utilizó el mismo tipo de componente para la sección del medio del delito. Esto puede apreciarse en la siguiente figura.



Figura 10. **Llenado de formulario de reporte**



Tipo de delito

- Abuso
- Abuso
- Amenaza
- Asalto
- Asesinato
- Estafa

Fuente: elaboración propia.

Para la fecha se utilizó un componente tipo calendario, donde se selecciona por medio de diferentes listas el día, mes y año cuando ocurrió el crimen a reportar. Para la sección de la hora se utilizó un componente del mismo tipo.

Figura 11. **Componente de fecha y hora**



Fijar fecha

16	oct.	2012
17	nov.	2013
18	dic.	2014

Cancelar Fijar

Fuente: elaboración propia.

Por último se detalla la sección de la ubicación, ya que esta puede ser ingresada vía texto donde solo es necesario pulsar el campo de texto para que se despliegue el teclado virtual y poder ingresarla vía manual.

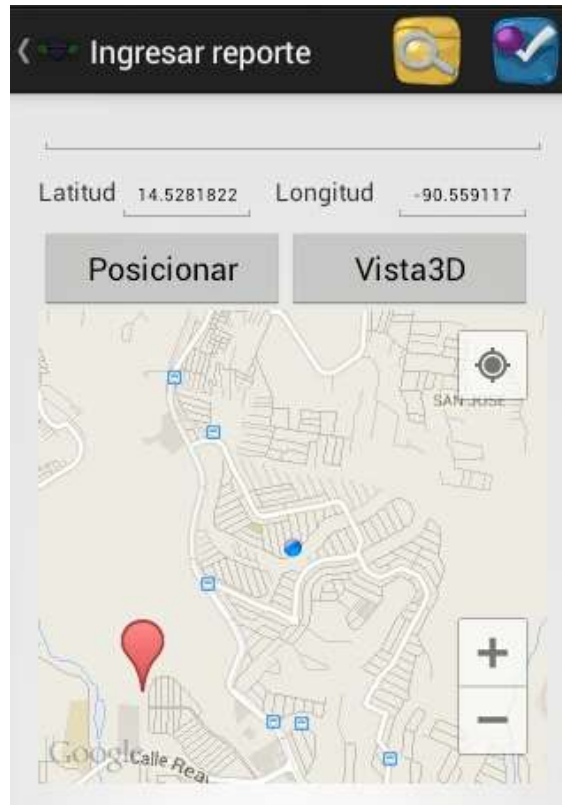
Figura 12. **Ubicación de reporte**



Fuente: elaboración propia.

La aplicación permite que la ubicación, en caso de ser desconocida, se obtenga por medio de coordenadas GPS. También puede ubicarse por medio de un mapa desplegado en la misma pantalla. En el caso de que la ubicación sea distinta a donde se cometió el delito, se puede mover el marcador para posicionarlo en la dirección correcta.

Figura 13. **Navegación de mapa**



Fuente: elaboración propia.

Al finalizar se puede ver el detalle del reporte que se ha ingresado. Es la misma vista que se utiliza al momento que un usuario da tap en alguno de los reportes o denuncias que se despliegan en los resultados de las búsquedas o de noticias.

Cabe mencionar que al finalizar de ingresar el reporte se da la opción de poder llamar a la Policía Nacional Civil o a la campaña Cuéntaselo a Waldemar en caso de querer ser anónimos. Esto puede observarse en la siguiente figura.

Figura 14. **Detalle de reporte**

Detalle de reporte

Título  
Asalto en taxi

Tipo de delito  
Asalto

Medio de delito  
Automovil

Fecha del delito  
2013-10-16

Hora del delito  
16:08:00

Descripción del delito  
Se realizó un asalto con pistola a un taxista en zona 9

Fuente: elaboración propia.

Figura 15. **Llamar a las autoridades**

Su reporte se ha ingresado exitosamente.

Realice su denuncia:  
Llamando a la PNC

110

Cuéntaselo a Waldemar

1561

Fuente: elaboración propia.

Para la realización de búsquedas se despliega una interfaz amigable que permite que la intuición humana entre en acción, donde se pueden utilizar diversos filtros de búsqueda para obtener los resultados más acertados a lo que se desea. Por último se despliegan en pantalla como un listado y pueden ser visualizados en un mapa.

Figura 16. **Búsqueda general**



Fuente: elaboración propia.

Los filtros pueden utilizarse para búsquedas más sofisticadas. Se utilizaron listas de datos para minimizar el error y que tuviera una interfaz más amigable e intuitiva. Se utilizó el mismo componente para los filtros del tipo de delito como el medio. Ver figura 17.

Figura 17. **Filtros de búsquedas**



Fuente: elaboración propia.

Se cuenta también con una lista de datos ya predefinidos, para que el usuario delimite el perímetro de la región donde se buscarán los reportes con los parámetros especificados en los demás filtros.

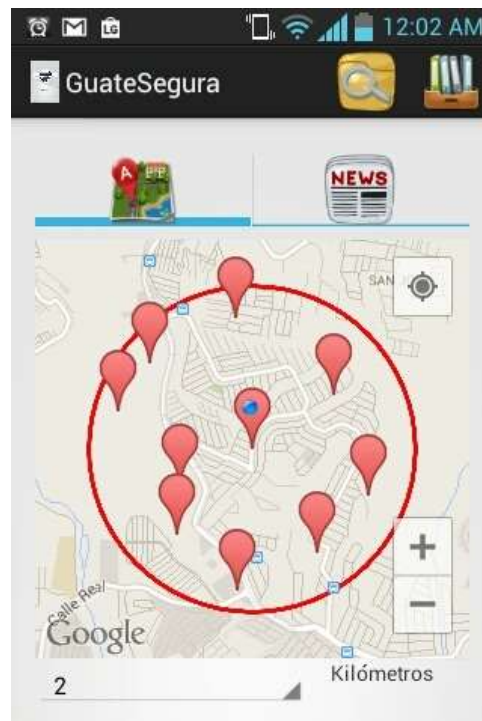
Figura 18. **Delimitación de perímetro**



Fuente: elaboración propia.

Por último se muestran los resultados de las búsquedas, tanto en un listado de datos como en el mapa.

Figura 19. **Despliegue de resultados**



Fuente: elaboración propia.





## **4. DOCUMENTACIÓN BASE PARA EL DESARROLLO DE LA APLICACIÓN**

### **4.1. Documentación base para el desarrollo de la aplicación móvil**

La aplicación GuateSegura se sirve de un *web service* que contiene los servicios varios para almacenar, administrar y proveer información de los puntos y los usuarios. El objetivo de este es concentrar la información para que pueda ser segura y persistente.

Este servicio web puede decirse que es la base fundamental para que la aplicación funcione de manera adecuada, ya que por medio de este es que se comunica la aplicación con la base de datos.

#### **4.1.1. Tecnologías utilizadas**

En esta sección se detallan las tecnologías que se implementaron durante el desarrollo del proyecto GuateSegura.

##### **4.1.1.1. Google Maps v2**

Este es un servicio gratuito de Google, el cual consiste en un servidor de aplicaciones de mapas en la web. Por medio de este servicio pueden desplegarse mapas desplazables en diferentes dispositivos, así como la interacción con estos.

Con esta herramienta se pueden realizar diversas acciones como mostrar ubicaciones, utilizar marcadores para identificar diversos lugares, implementar selección de áreas para poder marcar cierto perímetro o área alrededor de un punto específico, etc.

Tiene la capacidad de hacer acercamientos o alejamientos para mostrar el mapa. Este puede ser controlado mediante los dedos (para dispositivos *touch* como móviles), también se pueden obtener las coordenadas de latitud y longitud sobre un punto especificado. Esto dependerá de la funcionalidad que se le quiera dar al mapa como tal.

#### **4.1.1.2. Java**

Esta es una tecnología que se usa para el desarrollo de aplicaciones orientadas a objetos. Cuenta con diversos *frameworks* o APIs que lo hacen ser una herramienta bastante robusta y poderosa; en este caso se utilizó para desarrollar en Android, utilizando la herramienta Android *adt-bundle*.

#### **4.1.1.3. Facebook**

Este es un sitio web de redes sociales que más éxito ha tenido en los últimos años. En este se permite interactuar con otras personas de amigos, realizar publicaciones sobre pensamientos, eventos, publicar fotografías o vídeos, etc.

Para GuateSegura se utilizará como un medio publicitario donde se pueda dar a conocer la aplicación, así como un punto de viralización donde se puedan publicar en los perfiles de los usuarios algún reporte en específico.

#### **4.1.1.4. MySQL**

Sistema administrador de base de datos relacional de código abierto propiedad de Oracle Corporation. Soporta las transacciones que cumplen con el estándar ACID, haciéndolo una opción confiable para almacenar datos de servicios web.

Al ser una base de datos relacional se pueden archivar datos en tablas separadas en vez de colocarlos todos en un gran archivo, permitiendo así mayor velocidad y flexibilidad. Las tablas están conectadas entre sí por medio de relaciones definidas que hacen posible combinar datos de diferentes tablas.

#### **4.1.1.5. Metodología REST**

Es un estilo de arquitectura de software para sistemas distribuidos como los que se encuentran en la web. Se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Estos se resumen como los principios que han sido definidos y diseccionados.

REST no es un estándar, ya que es solamente un estilo de arquitectura, pero está basado en estándares como HTTP, URL, representación de los recursos tales como XML, HTML, entre otros, así como tipos MIME (text/xml, text/html, etc.). Esta metodología permite una escalabilidad en cuanto a la interacción de componentes; esto puede visualizarse por medio de la variedad de clientes que pueden acceder vía web. De la misma manera proporciona una compatibilidad con componentes intermedios, siendo los más populares los diferentes tipos de proxys, reduciendo así la latencia de interacción y reforzando la seguridad y encapsulación de otros sistemas.

#### **4.1.2. Arquitectura de la aplicación web**

Para la aplicación móvil se utilizó un patrón de diseño llamado modelo vista controlador (MVC), ya que de esta manera es que trabaja el desarrollo en Android.

Utilizando MVC se separan los datos de la parte lógica de negocio y de la interfaz del usuario; estos trabajan de manera independiente donde se puede trabajar de manera más sencilla, ya que facilita el mantenimiento fomentando la reutilización de código.

A continuación se detallan cada una de estos tres principales componentes:

##### **4.1.2.1. Modelo**

Este es el encargado de manejar la lógica de negocio, es totalmente independiente a la manera en que se almacenan y presentan los datos. Controla el flujo de información desde que el usuario ingresa los datos hasta el almacenamiento de la misma. Procesa y retorna a la vista una respuesta para que pueda ser desplegada y leída por el usuario final.

##### **4.1.2.2. Vista**

Se encarga de desplegar y solicitar datos al usuario. Esta sección tiene comunicación con el modelo, ya que los datos que percibe o muestra al usuario son enviados y procesados por este, para que le genere una respuesta acorde a las necesidades y solicitudes del usuario final.

#### **4.1.2.3. Controlador**

Es la sección intermedia entre el modelo y la vista, cada petición de la vista llega al modelo a través del controlador. Su principal tarea es la de traducir la respuesta que envía el modelo para que pueda ser agregada a la parte de la vista.

#### **4.1.3. Requerimientos para el desarrollo**

En esta sección se especifican los requerimientos para el desarrollo de la solución llamada GuateSegura.

##### **4.1.3.1. Android SDK**

El Android SDK (Android software development Kit) provee un API con las herramientas para compilar y probar aplicaciones desarrolladas en Android.

Una de las grandes ventajas que provee esta herramienta a los desarrolladores son los dispositivos virtuales Android AVD, ya que permite realizar cualquier tipo de operación, obteniendo los mismos resultados que se tendrían en un dispositivo real. El Android AVD permite emular un dispositivo real definiendo las especificaciones de hardware y software del dispositivo virtual.

El SDK cuenta con la herramienta Android Debug Bridge (ADB) que permite realizar una depuración de la aplicación en un dispositivo real o virtual; esta herramienta funciona como un cliente-servidor que ejecuta en segundo plano un proceso en el emulador o el dispositivo móvil.

#### **4.1.3.2. Android Development Tools (ADT)**

Es una extensión del IDE Eclipse que provee todas las herramientas necesarias para el desarrollo, depuración y ejecución de aplicaciones para Android, provee editores especializados para XML que permiten al desarrollador una vista previa del archivo XML ,que representa una interfaz de usuario final.

#### **4.1.4. Instalación y configuración de herramientas**

Durante este apartado se da una explicación detallada sobre cómo instalar las herramientas que se utilizaron, así como la configuración que necesita cada una de ellas para un buen desempeño y funcionamiento.

##### **4.1.4.1. Descarga e instalación de Android SDK**

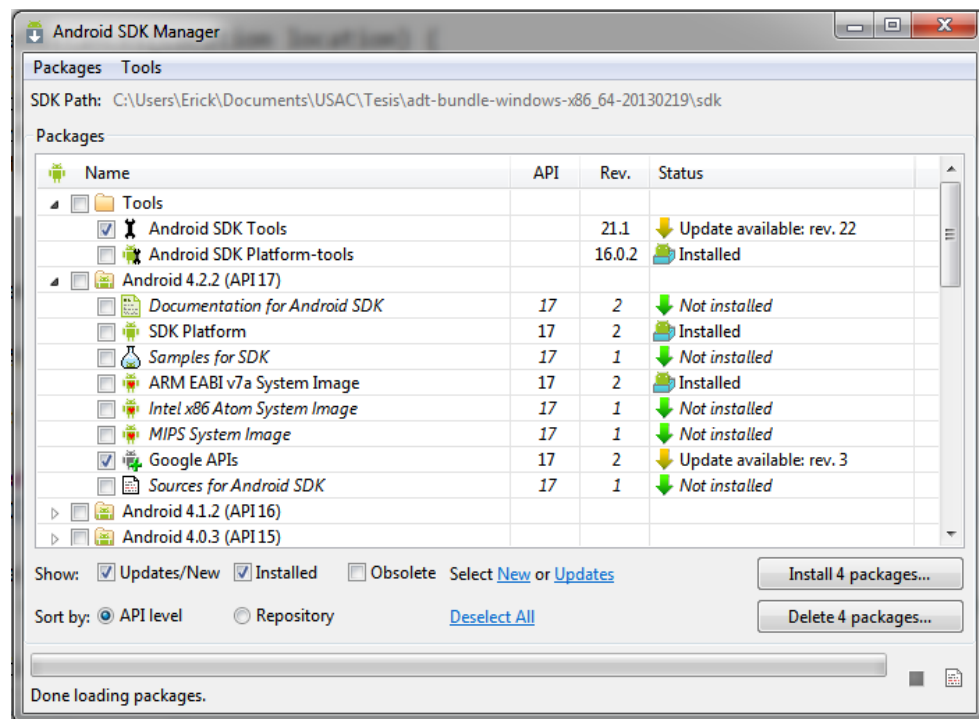
Desde la página oficial de Android developer (<http://developer.android.com/sdk/index.html>) se puede descargar el IDE Eclipse, ADT *plugin* con Android SDK Tools y todas las herramientas necesarias que se utilizan para el desarrollo de aplicaciones Android. Posteriormente, se debe descomprimir en cualquier ubicación del ordenador el archivo descargado. Esta será la ruta donde quede instalado el SDK. Se generarán dos directorios uno que contiene el SDK de Android y el otro Eclipse.

##### **4.1.4.2. Android SDK Manager**

Esta herramienta permite al usuario descargar diferentes paquetes que puede utilizar el desarrollador. Puede iniciarse de dos formas: la primera es inicializando Eclipse y entrar a la sección de menú Windows, posteriormente Android SDK Manager; esto levantará la interfaz gráfica donde se pueden

seleccionar los paquetes deseados para comenzar la descarga. La segunda forma es entrar a la carpeta del SDK que generó la extracción del paquete descargado.

Figura 20. **Android SDK Manager**



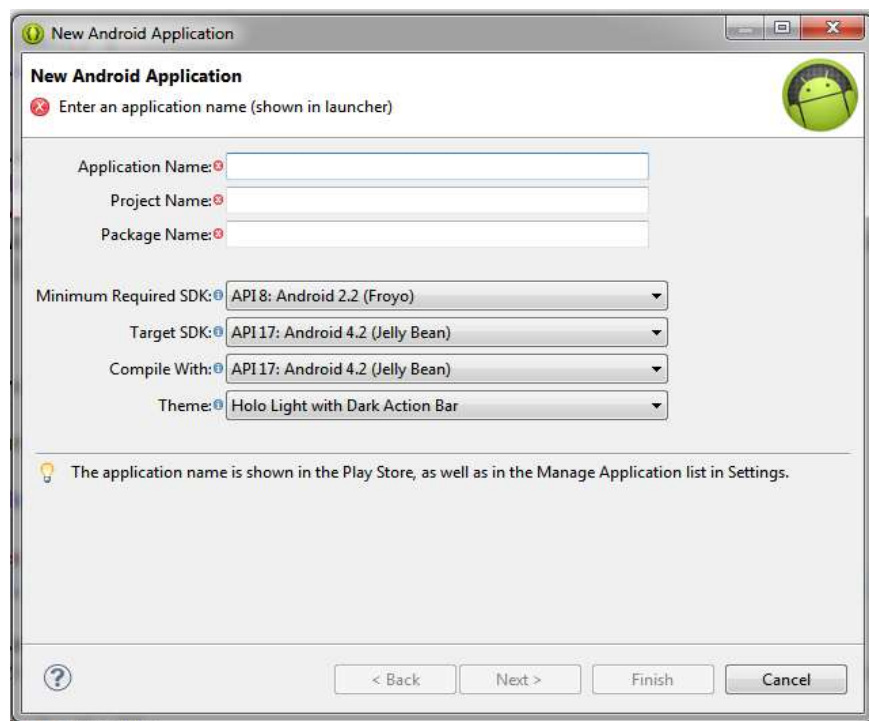
Fuente: elaboración propia.

#### 4.1.4.3. Creación de un proyecto nuevo en Eclipse

Para la creación de un nuevo proyecto en Eclipse se debe acceder al menú File seguido de New y por último en Android Project, se desplegará una ventana donde se solicita el nombre del proyecto. Otra información necesaria es mantener los nombres que autocompletan la herramienta y que el nombre de la aplicación comience con una mayúscula.

En esta misma sección se deben de elegir algunos otros parámetros que debe de llevar el proyecto, tales como el SDK mínimo y máximo requerido, la versión con que se compilará la aplicación, y si esta llevará un tema específico por *default* para el entorno gráfico.

Figura 21. Nueva aplicación Android



Fuente: elaboración propia.

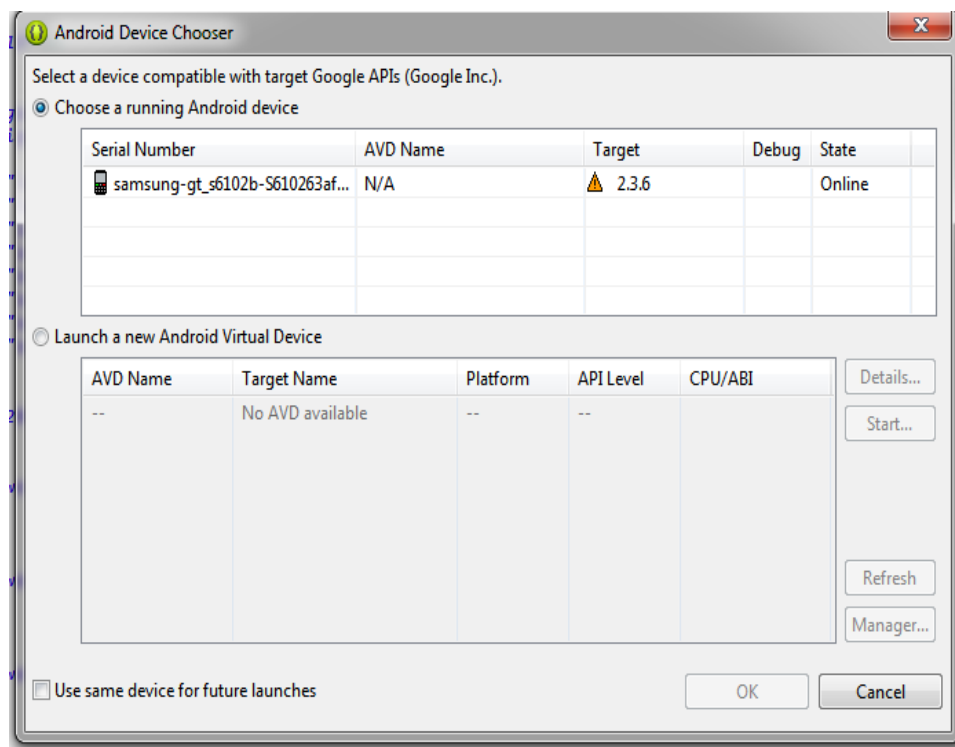
#### 4.1.4.4. Ambiente real de pruebas

Al momento de querer correr la aplicación para la realización de pruebas a la misma, es posible realizarlo en un dispositivo físico. Para esto solo se debe tener conectado el dispositivo vía USB al ordenador y habilitar en él el apartado de permitir aplicaciones que no sean del *market*.



Es importante si se desea *debuggear* el código, ya que para esto se debe entrar al menú de configuración del dispositivo, seguido de las aplicaciones en la sección de desarrollo y habilitar la opción de USB Debugging, una vez realizado esto sí se podrá *debuggear* el código.

Figura 22. **Selección de dispositivo**



Fuente: elaboración propia.

#### 4.1.5. **Conceptos generales para desarrollo en Android**

Para el desarrollo de aplicaciones en el sistema operativo Android es necesario que se conozcan ciertos conceptos que facilitan el uso y comprensión de su funcionamiento.

#### **4.1.5.1. Actividad**

Por medio de estas se puede permitir que el usuario realice diversas acciones; por cada actividad creada, existe una acción que se puede realizar.

Las ventanas que se crean a partir de las actividades pueden colocarse en cualquier parte de la interfaz gráfica por medio del método setContentView. Esto permitirá manejar qué objeto contiene a cuál. Existen dos métodos que implementan todas las actividades que son parte básica para el manejo de estas:

- onCreate(): por medio de este se inicia la actividad
- onPause(): se utiliza para que el usuario pueda salir de la actividad

#### **4.1.5.2. Servicios**

Componente que permite a la aplicación ejecutar algún tipo de acción que se ejecute en segundo plano, es decir que no requiera la interacción directa con el usuario final. Se necesita que este permanezca activo aun cuando el usuario cambia entre actividades.

En Android los servicios tienen una función doble, la primera permite indicar al sistema que el elemento que se ha creado se mantendrá en ejecución en segundo plano. Esto se logra mediante el método startService() donde se indica que se ejecutará de manera indefinida hasta que se le indique lo contrario. La segunda función permite la interacción con otras aplicaciones, para que ciertas funciones puedan ser llamadas desde otras aplicaciones, mediante el método bindService(). Se establece una conexión con el servicio y se invoca alguno de los métodos ofrecidos.

Un servicio se ejecuta en el hilo principal del proceso de la aplicación, por lo cual necesita un uso intensivo de recursos del CPU.

#### **4.1.5.3. Intención**

Sirve principalmente para invocar componentes, es decir actividades o servicios que se están ejecutando en segundo plano. Básicamente permiten llamar aplicaciones externas a la que aquí se propone, para que se lancen eventos a los que otras aplicaciones puedan responder o utilizar.

Por medio de estos también es posible lanzar mensajes de tipo *broadcast* para identificar cuando ciertos eventos han ocurrido.

#### **4.1.5.4. Android Manifest**

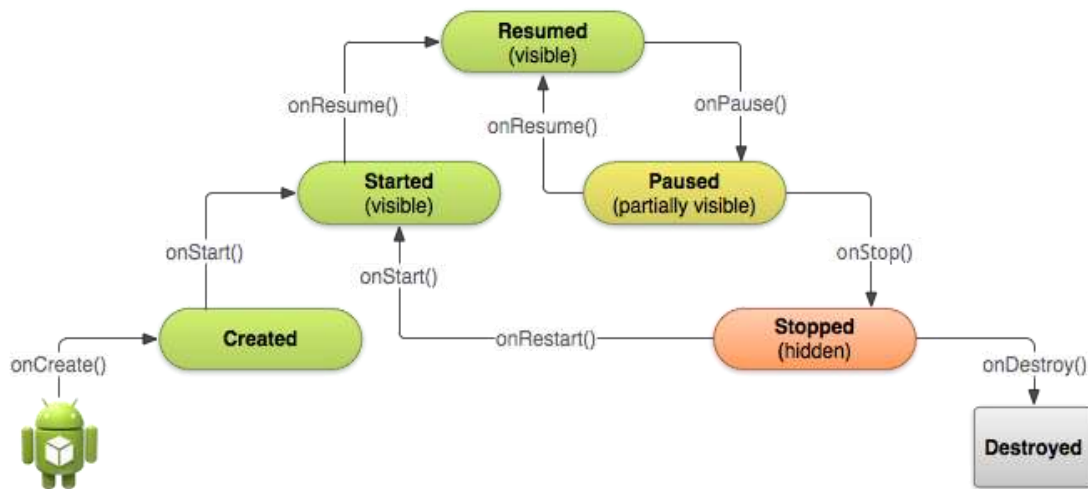
Archivo en formato XML que se encuentra ubicado en el directorio raíz de la aplicación con el nombre `Androidmanifest.xml`. Este es el encargado de informar al sistema los componentes que tiene la aplicación, así como el nivel mínimo del API requerido, y los requerimientos de hardware que necesita la aplicación para su ejecución.

#### **4.1.5.5. Ciclo de vida de componentes**

El ciclo de vida de cada componente hace posible ejecutar muchas aplicaciones de manera simultánea. Para esto Android implementó una serie de llamadas a cada componente para manipular su ciclo de vida.

El ciclo de vida de las actividades comienza al momento de la llamada `onCreate()` y finaliza cuando se ejecuta el método `onDestroy()`.

Figura 23. **Ciclo de vida de una actividad Android**



Fuente: <http://www.informaweb.it/it/en/android/android-activity-lifecycle-qual-e-ciclo-vita-app-android>. Consulta: noviembre de 2013.

## 4.2. **Diseño de base de datos**

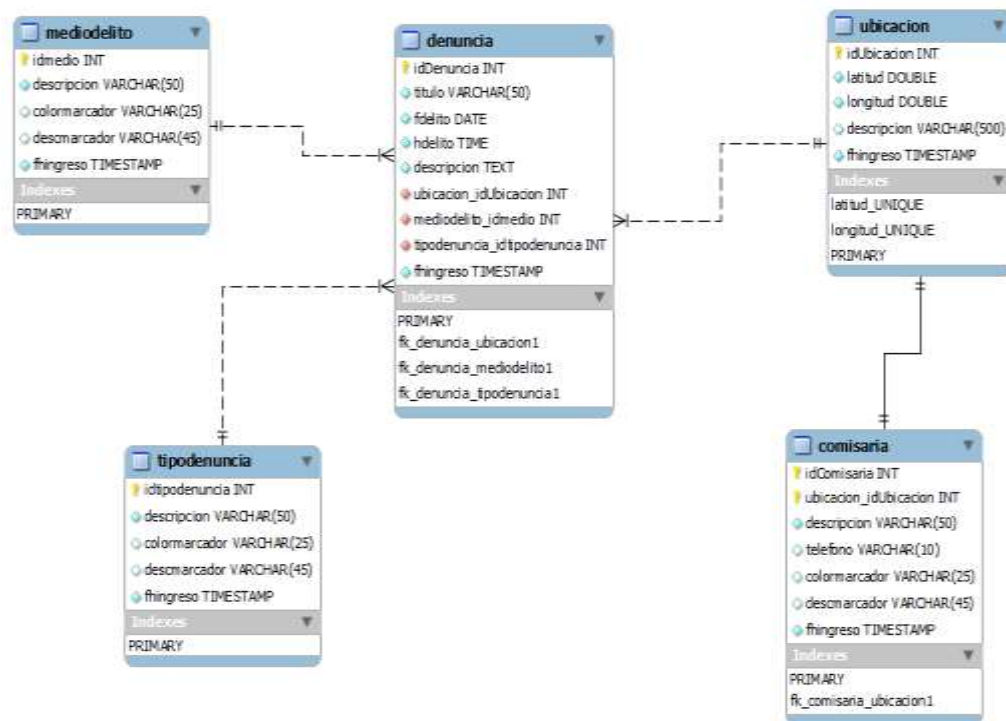
El análisis realizado para el diseño de base de datos relacional permite acceder a la información de manera precisa. El modelo relacional diseñado para este sistema permite adaptarse a las necesidades de la aplicación, de manera que facilite la modificación de la información cuando sea necesario. Dicho modelo no permite la redundancia de los datos, y asegura la durabilidad y consistencia de la información.

### 4.2.1. **Modelo de base de datos**

El modelo relacional de base de datos diseñado para el manejo adecuado de la información del sistema de GuateSegura cuenta con cinco

tablas, que mediante el uso de restricciones permite asegurar la integridad de los datos.

Figura 24. **Modelo relacional de base de datos**



Fuente: elaboración propia.

#### 4.2.2. Descripción de tablas

En esta sección se detallará cada una de las tablas diseñadas para el sistema de GuateSegura, según el modelo relacional (ver figura 24), dando una definición de la misma y brindando una descripción para cada uno de sus campos:

- **MEDIODELITO:** contiene una lista de medios que indican cómo es que ocurrió el crimen o acto delictivo, representando el entorno en el cual se encontraban los criminales. Por ejemplo: en autobús, en la calle, en automóvil, motocicleta o en una pasarela. Esta tabla cuenta con cinco campos que se describen a continuación:
  - **idmedio:** es un identificador único de la tabla.
  - **descripcion:** muestra el nombre del medio del delito.
  - **colormarcador:** indica el nombre del color que se utiliza dentro de la aplicación, para identificar los medios de delito, representados como un marcador en el mapa.
  - **descmarcador:** contiene la descripción que se muestra dentro del mapa al ubicar un marcador específico.
  - **fningreso:** campo de control que contiene la fecha y hora de cuando se ingresó el registro a la tabla.
  
- **TIPODENUNCIA:** contiene una lista de tipos de delitos con el que se identifican las denuncias. Estas pueden ser: abuso, amenaza, asalto, asesinato, extorsión, fraude, homicidio, robo, secuestro u otros. Esta tabla cuenta con cinco campos que se describen a continuación:
  - **idtipodenuncia:** es un identificador único de la tabla.
  - **descripcion:** muestra el nombre del tipo de denuncia.
  - **colormarcador:** indica el nombre del color que se utiliza dentro de la aplicación, para identificar los tipos de delito dentro de un mapa, representados como un marcador.
  - **descmarcador:** contiene la descripción que se muestra dentro del mapa al ubicar un marcador específico.

- fhingreso: campo de control que contiene la fecha y hora de cuando se ingresó el registro a la tabla.
- UBICACIÓN: esta tabla es la encargada de contener entre sus registros las distintas ubicaciones registradas a través de georreferenciación (latitud, longitud). Esta tabla cuenta con cinco campos que se describen a continuación:
  - idUbicacion: es un identificador único de la tabla.
  - latitud: almacena la coordenada angular norte y sur.
  - longitud: almacena la coordenada angular este y oeste.
  - descripcion: almacena la dirección de la ubicación, según su nombre.
  - fhingreso: campo de control que contiene la fecha y hora de cuando se ingresó el registro a la tabla.
- DENUNCIA: esta tabla contiene los datos sobre la denuncia, la cual es ingresada como un reporte por los usuarios. Cuenta con nueve campos que se describen a continuación:
  - idDenuncia: es un identificador único de la tabla
  - titulo: almacena el título con el que se reporta la denuncia
  - fdelito: fecha en que se efectuó el crimen o acto delictivo
  - hdelito: hora en que se efectuó el crimen o acto delictivo
  - descripcion: almacena la descripción reportada por el usuario, de lo ocurrido respecto del delito.
  - ubicacion\_idUbicacion: es el identificador único de un registro contenido en la tabla ubicación, el cual tiene relación con el delito, ya que un reporte deberá contener una ubicación.

- `mediodelito_idmedio`: es el identificador único de la tabla de medios de delito con el cual tiene relación, ya que en un reporte debe indicarse cómo ocurrió el acto.
- `tipodenuncia_idtipodenuncia`: es el identificador único de la tabla tipo de denuncias, con la cual tiene relación, ya que en un reporte debe de indicarse el tipo del delito para identificar el tipo de denuncia.
- `fhingreso`: campo de control que contiene la fecha y hora de cuando se ingresó el registro a la tabla.
- **COMISARIA**: esta tabla contiene la información sobre las distintas comisarías que han sido registradas, para poder ser ubicadas desde la aplicación. Cuenta con siete campos que se describen a continuación:
  - `idComisaria`: identificador único del registro.
  - `ubicacion_idUbicacion`: identificador único de un registro contenido en la tabla ubicación, el cual tiene relación con una comisaría, dado que debe de indicarse la dirección de la comisaría en esta tabla.
  - `descripcion`: contiene el nombre de la comisaría.
  - `telefono`: contiene el número de teléfono de la comisaría.
  - `colormarcador`: Indica el nombre del color que se utiliza dentro de la aplicación para identificar las comisarías.
  - `descmarcador`: contiene la descripción que se muestra dentro del mapa al ubicar un marcador específico.
  - `fhingreso`: campo de control que contiene la fecha y hora de cuando se ingresó el registro a la tabla.

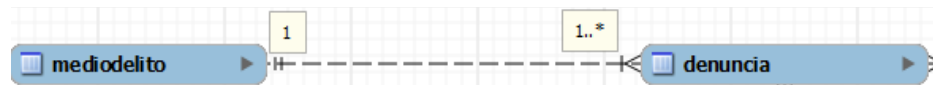


### 4.2.3. Descripción de relaciones

En esta sección se detallará cada una de las relaciones existentes entre las tablas diseñadas para el sistema de GuateSegura, según el modelo relacional.

- Medio delito-denuncia: una denuncia debe contener un solo medio de delito. Mientras que un medio de delito puede estar en muchos registros de denuncias.

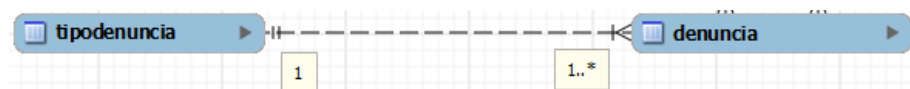
Figura 25. Relación entre medio del delito y denuncia



Fuente: elaboración propia.

- Tipo denuncia-denuncia: una denuncia debe contener un solo tipo de denuncia. Mientras que un tipo de denuncia puede estar en muchos registros.

Figura 26. Relación entre tipo de denuncia y denuncia



Fuente: elaboración propia.

- Ubicación–denuncia: una denuncia debe contener una sola ubicación que identifique en dónde ocurrió el acto delictivo. Mientras que una ubicación puede ser utilizada para varios reportes de denuncias. Ver figura 27.

Figura 27. **Relación entre denuncia y ubicación**



Fuente: elaboración propia.

- Ubicación-comisaría: una comisaría debe identificar su ubicación por medio de una dirección. Luego, una ubicación va a corresponder a una única comisaría.

Figura 28. **Relación entre ubicación y comisaría**



Fuente: elaboración propia.

### 4.3. Documentación de código fuente

Para el desarrollo de la aplicación móvil y el servicio web, la base del código fuente, hecho en un lenguaje de programación Java, se rigió bajo ciertos estándares tales como el uso de una arquitectura de software orientada a servicios, definición de clases utilizando patrones de diseño y conceptos básicos de programación orientada a objetos como la herencia, encapsulación, abstracción y polimorfismo.

Por lo que a continuación se detalla cada uno de los elementos, estándares y conceptos utilizados para el desarrollo del proyecto de GuateSegura.

#### **4.3.1. Arquitectura de software**

La implementación del proyecto de software se basa en una arquitectura orientada a servicios denominada SOA. Las razones son las siguientes:

- Esta arquitectura permite responder rápidamente a las condiciones cambiantes que ocurren en el negocio, permitiendo reutilizar operaciones que son accedidas a través de servicios, y por lo tanto se mejoran los tiempos para realizar cambios en los procesos.
- Permite la interdependencia de las plataformas e infraestructura de tecnologías, dando lugar a desarrollar aplicaciones sin importar lenguajes programación y dejando por un lado la lógica del negocio.
- Facilita la integración de sistemas y aplicaciones diferentes, lo cual mejora la comunicación y la capacidad de respuesta con sistemas externos.

SOA propone una arquitectura de capas para la implementación de aplicaciones, que promueve la separación de la lógica de negocio y la de diseño. Las capas propuestas para la solución en las aplicaciones del sistema de GuateSegura son las siguientes:

- Capa de presentación: se denomina capa de usuario debido a que es la que ve el usuario y con la cual interactúa. Esta se comunica únicamente con la capa de negocio.

- Capa de negocio: es la encargada de proporcionar las funcionalidades que necesita la capa de presentación, y es la responsable de gestionar la lógica del negocio de la aplicación, a la vez representa el contenedor lógico donde se ubican los servicios compartidos.
- Capa de datos: en esta capa residen los datos y se encarga de acceder a los mismos. Se conforma por uno o varios gestores de bases de datos que reciben requerimientos de almacenamiento o recopilación de información de la misma.

#### **4.3.2. Patrones de diseño**

Los patrones de diseño son la base para la búsqueda de soluciones a problemas que son muy comunes en el diseño para un desarrollo de software. El objetivo de los patrones de diseño se describe a continuación:

- Proporcionar catálogos de elementos reutilizables
- Formalizar un vocabulario común entre diseñadores
- Estandarizar el modo en que se realiza el diseño
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente

Los proyectos de software de GuateSegura cuentan con dos patrones de diseño básicos en su implementación, que pretenden cumplir con los objetivos mencionados anteriormente. Además, mejorar el acceso y la protección a los datos.

#### **4.3.2.1. Data Transfer Object (DTO)**

Son considerados objetos de transporte, ya que se encargan de contener los datos que serán transferidos entre los procesos.

Estos objetos son simples ya que no deben contener ninguna lógica del negocio o validaciones. En los proyectos de GuateSegura estos objetos se encuentran definidos dentro de un paquete denominado `com.guatesegura.dto`. Ver figura 29.

#### **4.3.2.2. Data Access Object (DAO)**

Son los objetos que proporcionan una interfaz para poder acceder a algún tipo de base de datos, permitiendo esta acción a través de un conjunto de métodos definidos en los DAO.

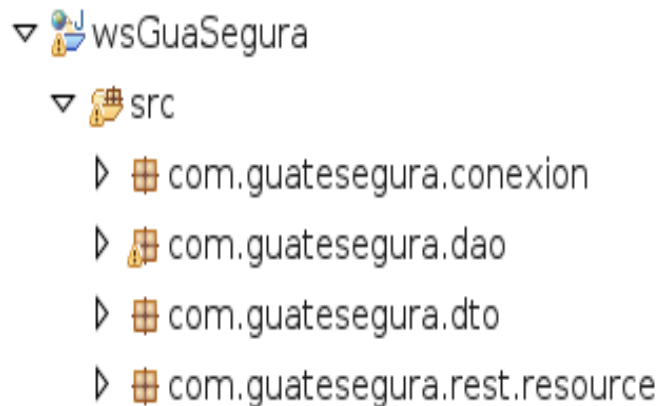
La implementación de este patrón en los proyectos de GuateSegura, se diseñó utilizando un objeto principal que es el controlador de datos, el cual contiene la lógica del negocio y se encarga de acceder a los DAO que contienen las diferentes operaciones que permiten agregar, modificar y eliminar información en la base de datos directamente. De esta manera se logra la separación entre el controlador (capa de negocio) y el modelo (capa de datos).

#### **4.3.3. Agrupación de clases**

En el proyecto desarrollado para implementar los servicios web de GuateSegura, las clases están distribuidas en diferentes paquetes según la lógica descrita en el capítulo cuatro.

La figura 29 muestra una representación de cómo se encuentran estructurados dichos paquetes dentro del proyecto de software.

Figura 29. **Paquetes definidos en el proyecto del servicio web**



Fuente: elaboración propia.

Para el desarrollo del servicio web se ha considerado el uso de cuatro paquetes, los cuales se describen a continuación:

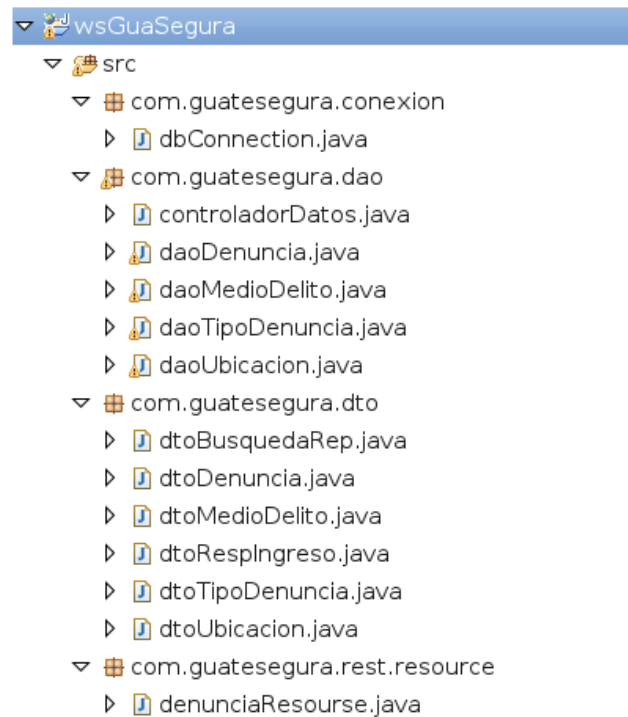
- `com.guatesegura.conexion`: contiene las clases utilizadas para realizar las conexiones a las bases de datos.
- `com.guatesegura.dao`: contiene las clases que brindan el acceso, manejo y control de los datos.
- `com.guatesegura.dto`: contiene todas las clases utilizadas para el transporte de la información entre los procesos.

- `com.guatesegura.rest.resource`: contiene la agrupación de las clases que implementan los servicios web utilizando la arquitectura REST.

#### 4.3.4. Descripción de clases

El servicio web fue desarrollado utilizando trece clases para su implementación, cada una de ellas agrupadas en diferentes paquetes. La siguiente imagen muestra cómo se encuentran agrupadas las clases en cada uno de los paquetes respectivos.

Figura 30. **Clases incluidas dentro de los paquetes del proyecto**



Fuente: elaboración propia.

La clase principal de este proyecto se encuentra dentro del paquete *resource* con el nombre *denunciaResource.java*, el cual contiene los servicios que son consumidos desde la aplicación móvil de GuateSegura.

#### **4.3.4.1. Clase de conexión a base de datos**

Esta clase se encuentra dentro del paquete *com.guatesegura.conexion* con el nombre *dbConnection.java*. Cuenta con un único método denominado *getConnection*, el cual se encarga de crear las conexiones a la base de datos mediante el uso de un JDBC (Java database connectivity), tecnología utilizada por el lenguaje de programación Java que permite crear conexiones a diferentes bases de datos.

El proyecto incluye la librería de conectividad dentro de un JAR (Java archive o archivo de Java), que se utiliza desde la aplicación como un controlador para realizar las conexiones a la base de datos de MySQL, gestor de base de datos utilizado. La versión utilizada puede encontrarse dentro de la carpeta de librerías del proyecto con el nombre: *mysql-connector-java-5.1.18-bin.jar*.

Para utilizar el controlador mencionado anteriormente, deben definirse ciertas variables en el proyecto que representen la siguiente información:

- Dirección IP del servidor de base de datos.
- Puerto que permite el acceso al servidor de base de datos.
- Nombre de la clase utilizada por el controlador de base de datos.
- Usuario de base de datos que tenga definidos los privilegios de consulta, ingreso y modificación de datos.
- Contraseña del usuario mencionado en el punto anterior.



El detalle de la clase que incluye el código fuente puede consultarse en la sección de apéndices.

#### **4.3.4.2. Clases de transporte de datos**

El grupo de clases utilizadas para el transporte de los datos se encuentran dentro del paquete `com.guatesegura.dto.`, las cuales se describen a continuación:

- `dtoBusquedaRep`: esta clase contiene datos que permiten la búsqueda de reportes, los atributos incluidos son:
  - Valores de la latitud y longitud utilizada como ubicación inicial sobre la cual se realizará la búsqueda de reportes.
  - Valor con la cantidad de kilómetros que delimita la búsqueda alrededor de la ubicación indicada.
  - Cantidad del límite de registros a devolver en una búsqueda.
  - Tipo de búsqueda, siendo una búsqueda por medio de filtros o de manera general.
  - Por último se tiene una lista del objeto denuncia en donde se devuelven todos los reportes consultados.

- **dtoDenuncia:** esta clase se encarga de transportar la información de las denuncias ingresadas por los usuarios; los atributos incluidos son:
  - Número único identificador de la denuncia
  - Título referente a la denuncia reportada
  - Descripción de la denuncia, en donde se detalla lo ocurrido
  - Tipo de denuncia según el reporte ingresado
  - Descripción del medio del delito
  - Ubicación de donde ocurrió el crimen o acto delictivo
  
- **dtoMedioDelito:** esta clase se encarga de transportar la información de cómo fue que ocurrió el crimen o delito que se está reportando. Los atributos incluidos son:
  - Número único: identificador del medio de delito.
  - Color del marcador: este se muestra dentro del mapa al realizar búsquedas, indicando el medio de delito.
  - Descripción: se indica el nombre del medio del delito, el cual es utilizado para mostrarse en las aplicaciones.
  - Descmarcador: es un campo que contiene la descripción que se muestra en conjunto con el marcador en el mapa, al momento de realizar búsquedas por medio del delito.

- `dtoTipoDenuncia`: esta clase se encarga de transportar información sobre el tipo de delito. Los atributos incluidos son:
  - Número único identificador del tipo del delito.
  - Color del marcador que se muestra dentro del mapa al realizar búsquedas indicando el tipo de la denuncia.
  - Descripción: en donde se indica el nombre del tipo de delito, y que es utilizado para mostrarse en las aplicaciones.
  - Descmarcador: es un campo que contiene la descripción que se muestra junto con el marcador en el mapa, al momento de realizar búsquedas por tipo de denuncia.
- `dtoUbicacion`: esta clase se encarga de transportar información referente al lugar donde ocurrió el crimen o acto delictivo. Los atributos incluidos son:
  - número único: identificador del registro en la tabla de ubicación.
  - Latitud: este campo contiene el valor de la latitud asignada a la ubicación reportada.
  - Longitud: contiene el valor de la longitud asignada a la ubicación reportada.
  - Descripción: contiene la dirección de la ubicación indicada en el reporte. Por último se tiene un campo con el nombre existe, que indica si la ubicación ya se encuentra registrada dentro de la base de

datos. Con este dato se puede hacer uso del identificador único de la ubicación, evitando el uso de los valores de latitud y longitud.

- **dtoRespIngreso:** esta clase se encarga de transportar información sobre la respuesta del ingreso de denuncias. Los atributos incluidos son:
  - El campo con nombre **blngresado** indica si la solicitud de nuevo reporte fue procesada exitosamente o denegada.
  - El campo **sMsg**, contiene un mensaje donde describe si la solicitud fue ingresada exitosamente o el detalle que indica la razón por la cual fue denegado el ingreso.

El detalle técnico que incluye el código fuente de las clases de transporte de datos, descritas anteriormente, puede encontrarse en la sección de apéndices.

#### **4.3.4.3. Clases de acceso a datos**

Las clases que permiten el acceso a los datos se encuentran dentro del paquete **com.guatesegura.dao**; las cuales se describen a continuación:

- **daoMedioDelito:** esta clase contiene los métodos que permiten acceder y manejar los datos que se encuentran dentro de la tabla medio del delito en la base de datos, realizando operaciones de consulta, ingreso, modificación y eliminación de registros en dicha tabla.
- **daoTipoDelito:** esta clase contiene los métodos que permiten acceder y manejar los datos que se encuentran dentro de la tabla tipo de denuncia

en la base de datos, realizando operaciones de consulta, ingreso, modificación y eliminación de registros en dicha tabla.

- daoUbicacion: esta clase contiene los métodos que permiten acceder y manejar los datos que se encuentran dentro de la tabla ubicación en la base de datos, realizando operaciones de consulta, ingreso, modificación y eliminación de registros en dicha tabla.
- daoDenuncia: esta clase contiene los métodos que permiten acceder y manejar los datos que se encuentran dentro de la tabla denuncia, realizando operaciones de consulta, ingreso, modificación y eliminación de los registros en dicha tabla.
- controladorDatos: es la clase que actúa como interfaz para el acceso a la capa de datos, creando una instancia de conexión a la base de datos, la cual es trasladada a los diferentes manejadores de información de las tablas de denuncias, medio del delito, tipo de denuncias y ubicación, es decir las clases DAO mencionadas anteriormente. Los métodos incluidos en esta clase se describen a continuación:
  - Ingresar denuncias: esta función permite ingresar denuncias a la base de datos, realizando validaciones previas que aseguren la consistencia de los datos y se permita el correcto almacenamiento de la información proporcionada. Este método puede encontrarse dentro de la clase con el nombre de ingresarDenuncia.
  - Búsqueda de denuncias por filtro: este método permite realizar búsquedas de reportes, indicando el tipo de denuncia y medio del delito; además, se tiene la opción de proporcionar una ubicación para

realizar una búsqueda perimetral de una cantidad dada de kilómetros a la redonda de la ubicación indicada.

- Búsqueda de todas las denuncias: esta función permite realizar la búsqueda de todos los reportes ingresados, indicando un límite de registros a devolver dentro de la lista.

El detalle de la clase que incluye el código fuente puede encontrarse en la sección de apéndices.

#### **4.3.4.4. Implementación de servicios web**

La clase que contiene los servicios web implementados se encuentran dentro del paquete `com.guatesegura.rest.resource`, con el nombre `denunciaResource`, la cual contiene una instancia del controlador de datos que permite acceder a la capa de datos y realizar las operaciones establecidas.

A continuación se describen los servicios que pueden ser consumidos desde cualquier aplicación cliente del *web service*:

- El primer servicio permite obtener un listado con todas las denuncias ingresadas; el mismo se ordena de forma descendente con base en la fecha y hora en que se ingresó el reporte. Este puede consumirse con el nombre `listageneral`; debe tomarse en consideración que la cantidad de registros devueltos debe indicarse entre los parámetros que se envían al momento de invocar el servicio.
- El segundo servicio permite obtener un listado de denuncias realizando una búsqueda por filtros. Este servicio puede consumirse con el nombre

busqfiltros indicando entre los parámetros que recibe, los valores de los campos medio del delito, latitud, longitud, número de kilómetros a la redonda y cantidad de reportes a devolver en el listado, los cuales son utilizados para realizar la búsqueda.

- Por último se incluye un servicio que tiene la función de realizar el ingreso de los reportes. Los datos a ingresarse son enviados mediante una petición POST, por lo que se adjuntan al cuerpo del mensaje de la petición.

El detalle de esta clase con el código fuente que implementa los servicios puede encontrarse en la sección de apéndices.

#### **4.3.5. Aplicación móvil**

El código fuente generado para el desarrollo de la aplicación móvil consta de una interfaz gráfica para capturar la información, integración de la aplicación con Google APIs y un consumidor de servicios web utilizado como cliente para enviar u obtener información de la base de datos de GuateSegura.

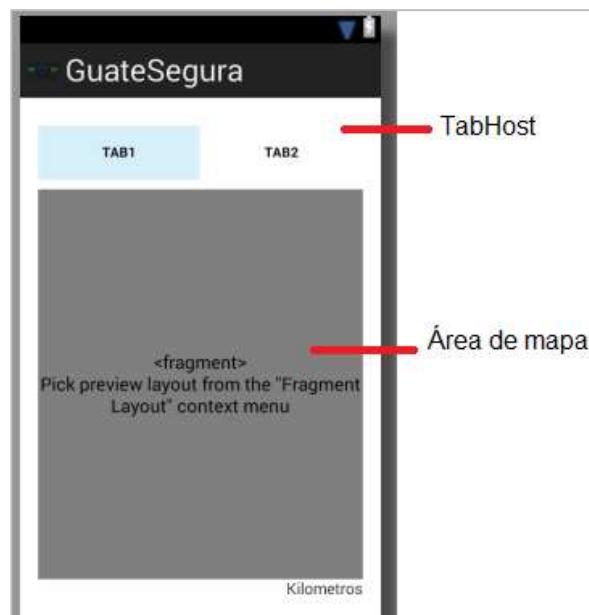
##### **4.3.5.1. Interfaz gráfica**

La interfaz gráfica de la aplicación móvil de GuateSegura fue diseñada para recopilar y mostrar la información necesaria en tres pantallas. La primera muestra una serie de reportes, los cuales son ordenados por fecha de ingreso al sistema de GuateSegura de forma descendente. Estos reportes son buscados en un perímetro de cinco kilómetros a la redonda de la ubicación actual del usuario. Dicha ubicación es obtenida por la aplicación al cargar los valores iniciales.

Los reportes son mostrados en un contenedor para vista de ventanas con pestañas, el cual se denominada TabHost. Este objeto tiene dos hijos donde el primero es un conjunto de etiquetas en las cuales el usuario hace clic para seleccionar una pestaña específica y el segundo objeto es un FrameLayout que muestra el contenido de esta pestaña.

En la primera pestaña se muestran los reportes como marcadores en un mapa y en la segunda pestaña los mismos son mostrados en un listado. La siguiente imagen muestra el diseño de la pantalla principal de la aplicación, que es donde se despliegan los reportes en cada pestaña, tal y como se describió anteriormente.

Figura 31. **Pantalla inicial de aplicación**



Fuente: elaboración propia.



La segunda pantalla fue diseñada para recopilar la información que ingresen los usuarios al momento de reportar algún crimen o acto delictivo.

Esta pantalla contiene los campos necesarios para ingresar un reporte, tales como el título, tipo y medio del delito, fecha y hora en que ocurrió, la descripción del mismo, y por último se muestra una sección con un del mapa en el cual se debe indicar la ubicación donde ocurrió dicho acto.

La siguiente imagen muestra el diseño utilizado para recopilar la información de los reportes al ser ingresado por los usuarios.

Figura 32. **Pantalla de ingreso de reportes**



The screenshot displays the 'GuateSegura' application interface for reporting a crime. The form includes the following fields and options:

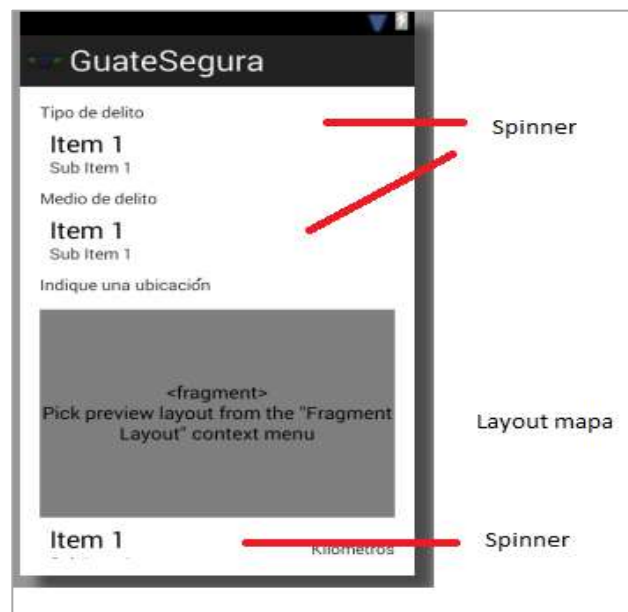
- Título:** A text input field.
- Tipo de delito:** A dropdown menu with 'Item 1' selected, showing a 'Sub Item 1' option below it.
- Medio de delito:** A dropdown menu with 'Item 1' selected, showing a 'Sub Item 1' option below it.
- Fecha del delito:** A date picker set to 'DD-MM-YYYY'.
- Hora del delito:** A time picker set to 'HH:MM:SS'.
- Descripción del delito:** A large text area for the crime description.
- Ubicación donde se cometió el delito:** A field at the bottom for the crime location, which is partially obscured by a map.

Fuente: elaboración propia.

Por último se tiene una tercera pantalla, la cual fue diseñada para generar los reportes por parte del usuario. Esta contiene los campos utilizados como filtros de búsqueda tales como el tipo y medio del delito, la ubicación sobre la cual se realizará la búsqueda y la cantidad de kilómetros a la redonda de la ubicación indicada, que se utilizará para formar el perímetro de búsqueda de reportes.

La siguiente imagen muestra el diseño utilizado para la pantalla de búsqueda de reportes, en donde se utilizan los controles de selección denominados “Spinner”. Estos son equivalentes a las listas desplegables para la selección de algún elemento que se encuentre dentro del listado. También se tiene un Layout que contiene el mapa utilizado para indicar la ubicación donde se desea dar seguimiento de los reportes ingresados.

Figura 33. **Pantalla para búsquedas de reportes**



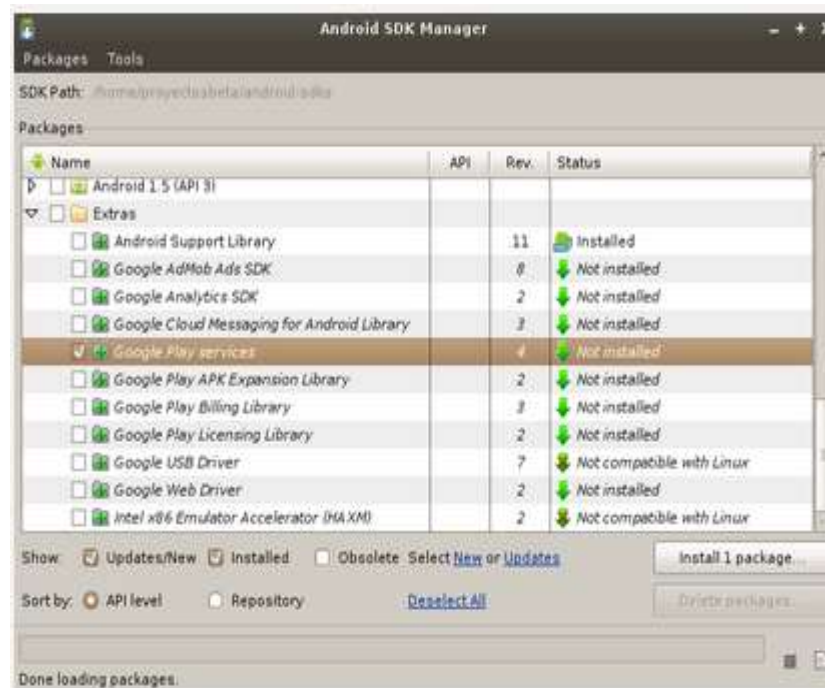
Fuente: elaboración propia.

Para trabajar con estos controles es indispensable proporcionar un adaptador de datos por medio del método `setAdapter()` y proveer un listener para manipular las opciones que el usuario elige a través del método `setOnItemSelectedListener()`.

#### 4.3.5.2. Google Maps API v2

Para llevar a cabo integración de Google Maps API v2 con la aplicación de GuateSegura, primeramente se realizó la instalación de Google Play Services desde el Android SDK Manager, descargando el paquete llamado Google Play Services.

Figura 34. Librerías de soporte para Android



Fuente: <http://proyectosbeta.net/2012/12/configurar-y-usar-google-maps-api-v2-para-android/>.

Consulta: noviembre de 2013.

Posteriormente a la instalación de los paquetes, es necesario obtener un API Key que permita utilizar los servicios de mapas de Google en la aplicación.

Esta clave se obtiene accediendo a la consola de APIs de Google (<https://code.google.com/apis/console/b/0/?pli=1>), creando un nuevo proyecto y activando el servicio Google Maps Android v2 pulsando el botón ON/OFF.

Figura 35. **Creación de proyecto en Google APIs**



Fuente: <http://proyectosbeta.net/2012/12/configurar-y-usar-google-maps-api-v2-para-android/>.

Consulta: marzo de 2014.

Figura 36. **Activación de servicios Google Maps**



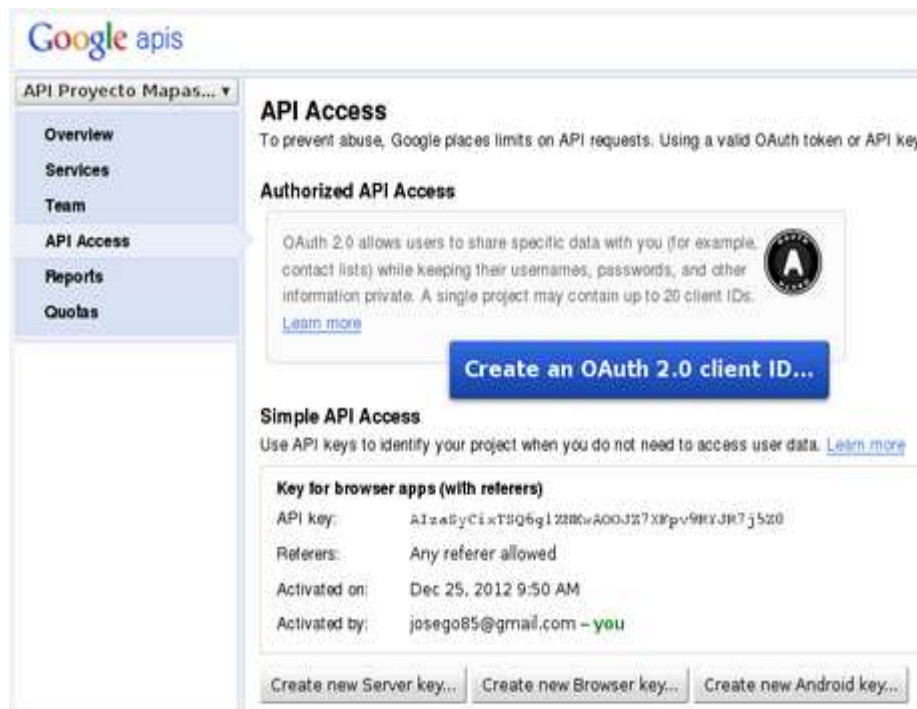
Fuente: <http://proyectosbeta.net/2012/12/configurar-y-usar-google-maps-api-v2-para-android/>.

Consulta: marzo de 2014.

Una vez hecho esto, se selecciona la opción API Access donde se tendrá la posibilidad de obtener la nueva API Key que permite utilizar el servicio de mapas desde la aplicación de GuateSegura. Luego se pulsa el botón Create new Android key..., donde se deberán ingresar dos datos identificativos de la aplicación:

- La huella digital (SHA1) del certificado con el que se firmó la aplicación
- El paquete de java utilizado

Figura 37. Creación de un nuevo Android Key



Fuente: <http://proyectosbeta.net/2012/12/configurar-y-usar-google-maps-api-v2-para-android/>.

Consulta: marzo de 2014.

Toda aplicación Android debe ir firmada para poder ejecutarse en un dispositivo; este proceso de firma es uno de los pasos que se deben ejecutar siempre antes de distribuir públicamente una aplicación. De hecho, durante el desarrollo de las pruebas y la depuración del código, se utiliza un certificado de pruebas para la firma de la aplicación.

Se puede saber la carpeta en la cual se ha almacenado el certificado accediendo desde Eclipse en el menú Eclipse/Preferences/Android/Build. En la consola del sistema se ingresan las siguientes líneas:

```
cd (path del certificado de pruebas)
keytool -list -v -keystore debug.keystore -alias androiddebugkey
```

Esto generará una llave SHA1, la cual se agrega en la sección de obtención de la API Key, separando con punto y coma por el paquete Java que se vaya a utilizar en la aplicación. Ver figura 38.

Figura 38. **Generación de API Key de GuateSegura**



Fuente: elaboración propia.

Una vez creado el API Key, este queda registrado para ser utilizado por la aplicación, las credenciales pueden verse desde la consola de APIs de Google.

Figura 39. **Credenciales registradas para GuateSegura**

Clave para las aplicaciones Android

CLAVE DE LA API	AIzaSyDHQqrFD235IYr6JYK3dKSzI6BiVfLbdX4
APLICACIONES ANDROID	43:34:47:AB:C2:05:5F:8A:63:30:D6:51:71:5E:04:B0:BA:B7:D9:21;com.example.guatesegura
FECHA DE ACTIVACIÓN	20 de oct. de 2013 15:23:00

Fuente: elaboración propia.

Luego de la generación de API Key, se debe proceder a agregarlo al proyecto de GuateSegura. Esto se realiza modificando el archivo AndroidManifest.xml del proyecto de Android; dentro de la etiqueta <application> se debe agregar un nuevo elemento <meta-data>.

Figura 40. **API Key en AndroidManifest.xml de GuateSegura**

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher3"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="AIzaSyDHQqrFD235IYr6JYK3dKSzI6BiVfLbdX4" />
```

Fuente: elaboración propia.

Adicionalmente se deben agregar permisos en el archivo AndroidManifest.xml que permitan acceder a los servicios web de Google a internet, y el almacenamiento externo del dispositivo que permite guardar en cache los mapas.

Figura 41. **Permisos en AndroidManifest.xml de GuateSegura**

```
<permission
    android:name="com.example.guatesegura.permission.MAPS_RECEIVE"
    android:protectionLevel="signature" />

<uses-permission android:name="com.example.guatesegura.permission.MAPS_RECEIVE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.ACCESS_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_GPS" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION" />
```

Fuente: elaboración propia.

#### 4.3.5.3. Consumidor de servicios web

Para la comunicación entre el cliente y el servidor es decir entre la aplicación Android y los servicios web, se ha utilizado la técnica de transferencia de estados representacional conocida como REST por sus siglas en inglés, *Representational state transfer*, utilizando operaciones HTTP en particular GET y POST.

Se utilizaron instancias de las clases definidas en el paquete dto para almacenar la información que se transporta entre el cliente y servidor.



Estos datos almacenados en las instancias se transmiten del servidor en formato JSON (JavaScript Object Notation), que es un formato ligero para el intercambio de datos; utiliza una notación literal de objetos de JavaScript que no requiere del uso de XML.

REST, como se mencionó anteriormente, es una técnica que hace uso de protocolos web estándar para implementar un servicio web. Con ello se pueden realizar servicios en forma de métodos que realicen acciones para presentar, recuperar o modificar los datos del servidor.

Se implementó el siguiente código como parte del cliente, el cual se encarga de enviar los parámetros al servidor, se considera a esta clase como la parte más importante del cliente. Esta se coloca dentro de cada clase que se utilizará como cliente, la cual extiende de la clase `AsyncTask` que permite que un proceso se ejecute en un subproceso independiente. Si la comunicación con el servicio es el hilo principal de la aplicación Android, la interfaz de usuario bloquea el proceso, en lo que espera los resultados del servidor.

Con esta clase se pueden definir los parámetros que se pasan a la instancia de la `AsyncTask`. Esto se hace mediante la función `doInBackground` del `WebServiceTask`. Este código utiliza objetos `HttpClient` de Android, aplicando el método GET que utiliza el `HttpGet` y para el método POST, el `HttpPost`.

La clase `AsyncTask` incluye otros dos métodos que el usuario tiene la opción de sobrescribir. Uno de ellos es `OnPreExecute()`, que se puede utilizar para preparar el proceso de fondo, y el otro es `OnPostExecute()`, que se puede utilizar para hacer cualquier proceso, luego de que el proceso de fondo haya concluido.



## CONCLUSIONES

1. Al implementar aplicaciones que ayuden a que las personas llamen vía telefónica a la Policía Nacional Civil, se ayuda a que la tasa de crímenes negros disminuya.
2. En vista de que el delito más cometido en la sociedad son los asaltos, una aplicación que permita visualizar las áreas más afectadas y los crímenes que se dan a nuestro alrededor. Incita a que la población tome medidas de precaución sobre los lugares o rutas que transitan en su diario vivir.
3. Es importante explotar el campo de la tecnología en todos los ámbitos posibles, en este sentido tanto los usuarios finales como las autoridades se verán beneficiadas con un uso adecuado sobre la aplicación.
4. La principal vulnerabilidad de la aplicación es que no sea aceptada por parte de la sociedad, esto debido a que su fuente de alimentación de datos son los ingresos por parte de los usuarios.
5. Dado a las características del proyecto, siempre se desplegarán los reportes dentro de un perímetro establecido. Este puede acoplarse a cualquier comunidad o región.



## RECOMENDACIONES

1. Brindar apoyo en cuanto a publicidad para la aplicación, dado que se hicieron intentos de contactar a los personeros de Espantacacos, pero lamentablemente no se pudo darle seguimiento a los mismos.
2. Brindar apoyo por parte de los diferentes medios que brindan seguridad nacional como la Policía Nacional Civil (PNC). Se podrían agregar funcionalidades que ayuden a dicha entidad a dar seguimiento a los casos que se reporten.
3. Calificar en cuanto a la criticidad de los delitos. Esta calificación podría darse por parte de los usuarios y que los grupos sociales o PNC puedan darle seguimiento a los reportes que se consideren más críticos en cuanto a la seguridad nacional.
4. Obtener apoyo por parte de otras entidades para uso de la aplicación, ya que por la demanda que tendría en cuanto a la publicación de reportes y búsquedas de los mismos. Se necesitaría un buen servidor que brinde la disponibilidad completa para que los servicios tengan un funcionamiento eficiente.



## BIBLIOGRAFÍA

1. Androideity. *Arquitectura y componentes de Android*. [en línea].  
<<http://androideity.com/2011/07/04/arquitectura-de-android/>>.  
[Consulta: 24 de octubre de 2013].
2. \_\_\_\_\_. *Desarrollar actividades en aplicaciones Android*. [en línea].  
<<http://androideity.com/2011/07/10/actividades-activities-en-android/>>  
[Consulta: 2 de noviembre de 2013].
3. \_\_\_\_\_. *Trabajando con Intents en Android I: Overview*. [en línea].  
<<http://androideity.com/2011/10/17/trabajando-con-intents-en-android-overview/>> [Consulta: 2 de noviembre de 2013].
4. Android Developers. *Página de desarrolladores de Android*. [en línea].  
<<http://developer.android.com/reference/android/app/Activity.html>>  
[Consulta: 2 de noviembre de 2013].
5. Artiteca. *Android el sistema operativo más usado y vulnerable*. [en línea].  
<<http://artiteca.blogspot.com.es/2012/04/android-el-sistema-operativo-mas-usado.html>>. [Consulta: 24 de octubre de 2013].
6. CeroPolitica. *Android el sistema operativo móvil más usado del mundo*. [en línea].  
<<http://ceropolitica.com/android-es-el-sistema-operativo-movil-mas-usado-del-mundo/>>. [Consulta: 28 de octubre de 2013].

7. El Economista. *Las 10 tendencias de la tecnología móvil, según IDC*. [en línea]. <<http://eleconomista.com.mx/tecnociencia/2013/04/17/las-10-tendencias-tecnologia-movil-segun-idc>>. [Consulta: 25 de octubre de 2013].
8. El Financiero. *Tendencias del mundo de sistemas operativos móviles que obligan a desarrollar aplicaciones en diversas plataformas*. [en línea]. <[http://www.elfinancierocr.com/tecnologia/Desarrollo\\_de\\_apps-iOS-Android\\_0\\_259174103.html](http://www.elfinancierocr.com/tecnologia/Desarrollo_de_apps-iOS-Android_0_259174103.html)>. [Consulta: 26 de octubre de 2013].
9. Marketing Directo. *iOS y Android se disputan el podio del sistema operativo de Smartphone más utilizado en el mundo*. [en línea]. <<http://www.marketingdirecto.com/especiales/marketing-movil/ios-y-android-se-disputan-el-podio-del-sistema-operativo-de-smartphone-mas-utilizado/>>. [Consulta: 26 de octubre de 2013].
10. Nacho Carnés. *Ideas de marketing. Mobile marketing: top 5 marcas smartphone y tablet*. [en línea]. <<http://nachocarnes.wordpress.com/tag/cuota-de-mercado-sistemas-operativos-moviles/>>. [Consulta: 26 de octubre de 2013].
11. Tendencias en marketing. *Tendencias de los sistemas operativos móviles*. [en línea]. <<http://tendenciasenmarketing.wordpress.com/2011/11/23/sistemas-operativos-moviles/>>. [Consulta: 26 de octubre de 2013].



12. Universidad Cardenal Herrera. *Ranking de sistemas operativos más usados en 2013*. [en línea]. <<http://blog.uchceu.es/informatica/ranking-de-sistemas-operativos-mas-usados/>>. [Consulta: 24 de octubre de 2013].
13. Universidad Politécnica de Valencia. *Programación de aplicaciones para móviles*. [en línea]. <<http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/99-arquitectura-de-android>>. [Consulta: 26 de octubre de 2013].
14. \_\_\_\_\_. *Introducción a los servicios en Android*. [en línea]. <<http://www.androidcurso.com/index.php/tutoriales-android/38-unidad-8-servicios-notificaciones-y-receptores-de-anuncios/160-los-servicios-en-android>> [Consulta: 2 de noviembre de 2013].
15. Xataka móvil. *Sistemas operativos y software para teléfonos móviles*. [en línea]. <<http://www.xatakamovil.com/categoria/sistemas-operativos>> [Consulta: 26 de octubre de 2013].



## APÉNDICES

### Apéndice 1. Investigación descriptiva

Se ha realizado una investigación estadística para obtener información acerca de la aceptación que tendría una aplicación móvil como GuateSegura. Se aplicó una encuesta con cinco preguntas integradas en un formulario de Google Drive, seleccionando una muestra de treinta y nueve personas, a partir de la cual se realizaron los cálculos estadísticos.

Las preguntas realizadas en la encuesta son las siguientes:

1. ¿Utilizaría una aplicación para celulares donde pueda ver los lugares donde más asaltan en Guatemala?
2. ¿Reportaría algún robo o asalto con su celular por medio de una aplicación?
3. ¿Considera útil que le notifiquen a su celular en caso que hay algún crimen cerca de donde se encuentra?
4. ¿Qué le gustaría poder ver/hacer en una aplicación donde se puedan reportar asaltos o robos?
5. ¿Considera que las autoridades de nuestro país podrían beneficiarse con una aplicación como la descrita, para brindar más seguridad a los ciudadanos?

Esta encuesta fue contestada por diferentes personas a través de internet, a quienes se les proporcionó la página de la misma, como se muestra en esta sección.

Figura A1. **Formulario de encuesta realizada sobre aplicación**

## GuateSegura

¿Utilizaría una aplicación para celulares donde pueda ver los lugares donde más asaltan en Guatemala?

- ☐ Sí
- ☐ No
- ☐ Tal vez

¿Reportaría algún robo o asalto con su celular por medio de una aplicación?

- ☐ Sí
- ☐ No
- ☐ Tal vez

¿Considera útil que le notifiquen a su celular en caso que hay algún crimen cerca de donde se encuentra?

- ☐ Sí
- ☐ No

¿Qué le gustaría poder ver/hacer en una aplicación donde se puedan reportar asaltos o robos?

¿Considera que las autoridades de nuestro país podrían beneficiarse con una aplicación como la descrita para brindar más seguridad a los ciudadanos?

- ☐ Sí
- ☐ No

Never submit passwords through Google Forms.

Teniendo los datos recopilados, se pudo realizar un análisis más amplio sobre las características que debería tener la aplicación; los resultados obtenidos de la encuesta para cada una de las preguntas mencionadas anteriormente se muestran a continuación:

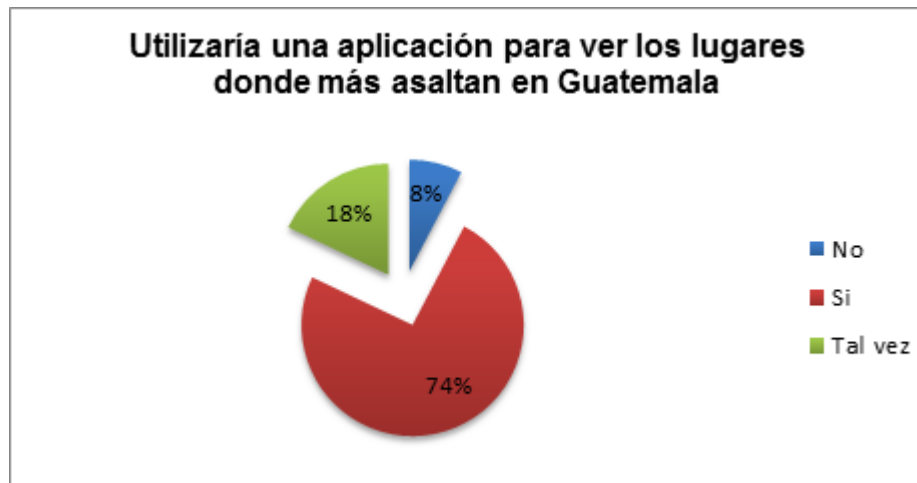
- Pregunta 1: ¿Utilizaría una aplicación para celulares donde pueda ver los lugares donde más asaltan en Guatemala?

Tabla A1. **Resultados obtenidos de la pregunta 1**

Respuesta	Cantidad
No	3
Sí	29
Tal vez	7
<b>TOTAL</b>	<b>39</b>

Fuente: elaboración propia.

Figura A2. **Gráfico porcentual de la pregunta 1**



Fuente: elaboración propia.

Se puede observar en el gráfico anterior que un 74 % de las personas encuestadas estarían dispuestas a utilizar una aplicación para ver los lugares en donde se frecuentan más asaltos en Guatemala. Por lo que se consideraría que GuateSegura tendría una gran aceptación por parte de la población.

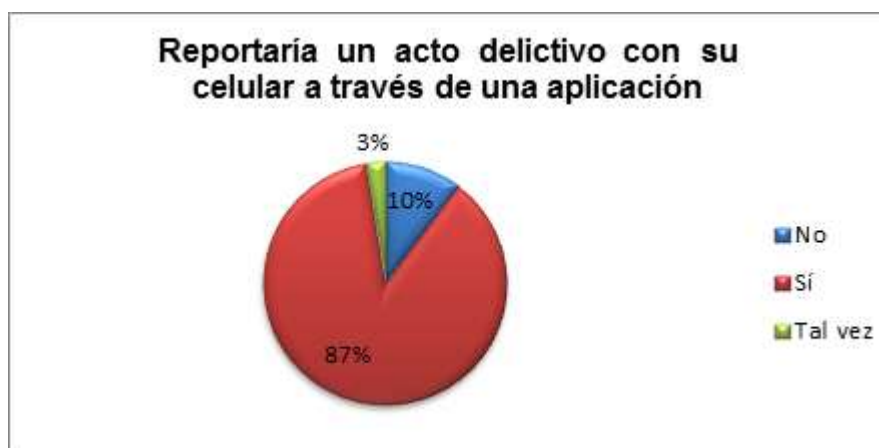
- Pregunta 2: ¿Reportaría algún robo o asalto con su celular por medio de una aplicación?

Tabla A2. **Resultados de la pregunta 2**

Respuesta	Cantidad
No	4
Sí	34
Tal vez	1
<b>TOTAL</b>	<b>39</b>

Fuente: elaboración propia.

Figura A3. **Gráfico porcentual de la pregunta 2**



Fuente: elaboración propia.

Se puede observar que el 87 % de las personas encuestadas estaría dispuesto a utilizar la aplicación para reportar algún tipo de acto delictivo, un 10 % no reportaría ningún tipo de delito y otro 3 % respondió que tal vez podría hacerlo. Por lo que se puede deducir que la aplicación tendría un uso constante por parte de las personas que tengan instalado este aplicativo en sus teléfonos inteligentes.

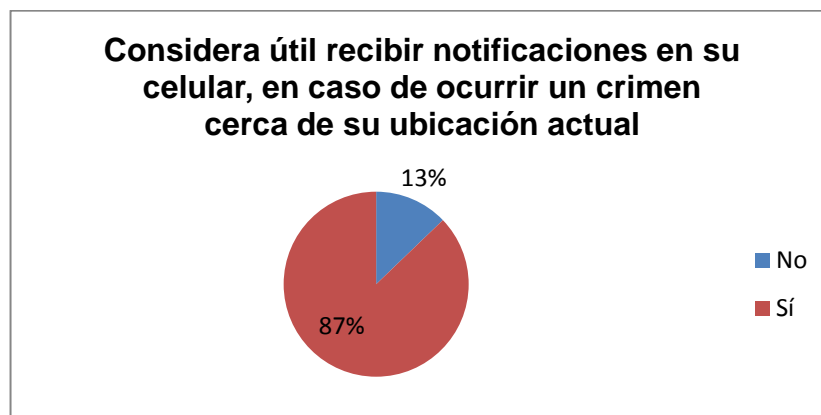
- Pregunta 3: ¿Considera útil que le notifiquen a su celular en caso que hay algún crimen cerca de donde se encuentra?

Tabla A3. **Resultados de la pregunta 3**

Respuesta	Cantidad
No	5
Sí	34
<b>TOTAL</b>	<b>39</b>

Fuente: elaboración propia.

Figura A4. **Gráfico porcentual de la Pregunta No.3**



Fuente: elaboración propia.

13 % de las personas encuestadas no considerarían útil este tipo de notificaciones, mientras que un 87 % respondió que sí considerarían útil el ser notificado a su celular cuando haya ocurrido algún tipo de crimen cerca de su ubicación actual, la razón es que podrían estar con más precaución al estar prevenidos por la aplicación.

- Pregunta 4: ¿Qué le gustaría poder ver/hacer en una aplicación donde se puedan reportar asaltos o robos? A continuación se describen las observaciones obtenidas por las personas encuestadas, acerca de las características que debería incluir la aplicación:
  - Un mapa donde se señale las principales zonas rojas.
  - Notificaciones de otros usuarios en cuanto a los crímenes cometidos recientemente.
  - Avisar directamente a la Policía Nacional Civil (PNC) que se está cometiendo un acto delictivo, y que ellos puedan movilizar sus unidades a ese lugar para tener la posibilidad de capturar a los criminales.
  - Tener una clasificación por tipo de asalto (con arma blanca, arma de fuego), cuántas personas fueron involucradas, placas asociadas por ejemplo, en el caso de asaltantes en motocicleta.
  - Poder visualizar donde ha habido situaciones peligrosas últimamente, no solamente en el instante que suceden.
  - Tener un *ranking* con los lugares más peligrosos de Guatemala



- Poder visualizar en una galería de imágenes todo lo relacionado con los delitos.
- Realizar llamadas directamente a la jefatura más cercana de la PNC, según la ubicación actual al momento de reportar el crimen.
- Mostrar un mapa con información de las jefaturas cercanas al lugar, incluyendo información de dirección y teléfonos. Esto para realizar las llamadas directamente a una comisaria que se encuentre cerca de la ubicación actual al momento de realizar el crimen.
- Generar estadísticas de asaltos por zonas y horas.
- Permitir un rastreo del celular a través de la aplicación.
- Tener un indicador de peligro, que recomiende si es aconsejable ir a determinado lugar por ciertas rutas, o dar rutas alternas según los acontecimientos reportados recientemente, en lugares cercanos al lugar de destino.
- Estar integrado con Facebook para poder notificar en las redes sociales sobre los crímenes ocurridos recientemente.
- Cambiar el estado del teléfono a modo silencioso, para prevenir que al estar en una zona peligrosa pueda sonar o vibrar, al tener una llamada entrante o recibir un mensaje.
- Recibir notificaciones preventivas cuando se está acercando a una zona peligrosa. De esa manera se puede estar con más precaución.

- Pregunta 5: ¿Considera que las autoridades de nuestro país podrían beneficiarse con una aplicación como la descrita para brindar más seguridad a los ciudadanos?

Tabla A4. **Resultados de la pregunta 5**

Respuesta	Cantidad
No	1
Sí	38
<b>TOTAL</b>	<b>39</b>

Fuente: elaboración propia.

Figura A5. **Gráfico porcentual de la pregunta 5**



Fuente: elaboración propia.

Un 97 % de las personas considera que las autoridades sí podrían obtener algún tipo de beneficio a través de la información que se maneje en GuateSegura, permitiendo de esa manera tomar acciones para contrarrestar la delincuencia en los lugares más afectados, según los reportes realizados por la población por medio de la aplicación.

## A2. Scripts de base de datos

- Diseño físico: el esquema físico de base de datos, obtenido a través del modelo relacional que se generó en la fase de diseño lógico de la base de datos de GuateSegura, se describe a continuación:
  - Eliminación y creación del esquema de base de datos: en este *script* se verifica primero la existencia del esquema para ser eliminado y así posteriormente ser creado nuevamente.

Figura A6. **Script de creación de esquema de base de datos**

```
DROP SCHEMA IF EXISTS `guatesegura` ;  
CREATE SCHEMA IF NOT EXISTS `guatesegura` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci ;  
USE `guatesegura` ;
```

Fuente: elaboración propia.

- Creación de la tabla tipo de denuncia y medio del delito: en los scripts de ambas tablas se verifica primero la existencia de ellas para ser eliminadas y posteriormente ser creadas nuevamente. Con los campos “idtipodenuncia” y “idtipomedio” para las tablas tipo denuncia y medio delito, respectivamente, siendo del tipo entero con la característica de ser autoincrementable, el cual funciona como una secuencia, generando un número correlativo distinto cada vez que se ingresa un nuevo registro para cada tabla; estos campos son utilizados como llave primaria de las tablas. Por otra parte el campo fhingreso es del tipo fecha/hora que por defecto coloca como valor la fecha y hora actual del servidor de base de datos, al momento de ingresar el registro.

Figura A7. **Script de creación de tabla tipo denuncia**

```
-- -----  
-- Table `guatesegura`.`tipodenuncia`  
-- -----  
  
DROP TABLE IF EXISTS `guatesegura`.`tipodenuncia` ;  
  
CREATE TABLE IF NOT EXISTS `guatesegura`.`tipodenuncia` (  
  `idtipodenuncia` INT NOT NULL AUTO_INCREMENT ,  
  `descripcion` VARCHAR(50) NOT NULL ,  
  `colormarcador` VARCHAR(25) NULL ,  
  `descmarcador` VARCHAR(45) NULL ,  
  `fhingreso` TIMESTAMP NOT NULL DEFAULT current_timestamp ,  
  PRIMARY KEY (`idtipodenuncia`) )  
ENGINE = InnoDB;
```

Fuente: elaboración propia.

Figura A8. **Script de creación de tabla medio del delito**

```
-- -----  
-- Table `guatesegura`.`mediodelito`  
-- -----  
  
DROP TABLE IF EXISTS `guatesegura`.`mediodelito` ;  
  
CREATE TABLE IF NOT EXISTS `guatesegura`.`mediodelito` (  
  `idmedio` INT NOT NULL AUTO_INCREMENT ,  
  `descripcion` VARCHAR(50) NOT NULL ,  
  `colormarcador` VARCHAR(25) NULL ,  
  `descmarcador` VARCHAR(45) NULL ,  
  `fhingreso` TIMESTAMP NOT NULL DEFAULT current_timestamp ,  
  PRIMARY KEY (`idmedio`) )  
ENGINE = InnoDB;
```

Fuente: elaboración propia.

- Creación de la tabla ubicación: esta tabla tiene un identificador autoincrementable que se utiliza como llave primaria; los campos de latitud y longitud son definidos en conjunto como un índice único dentro de la tabla, para no permitir el ingreso de dos registros con los mismos valores de latitud y longitud, una vez estos ya hayan sido ingresados anteriormente. De esta manera se evita la duplicidad de información, no permitiendo la redundancia de los datos.

Figura A9. **Script de creación de tabla ubicación**

```
-- Table `guatesegura`.`ubicacion`  
-----  
DROP TABLE IF EXISTS `guatesegura`.`ubicacion` ;  
  
CREATE TABLE IF NOT EXISTS `guatesegura`.`ubicacion` (  
  `idUbicacion` INT NOT NULL AUTO_INCREMENT ,  
  `latitud` DOUBLE NOT NULL ,  
  `longitud` DOUBLE NOT NULL ,  
  `descripcion` VARCHAR(500) NULL ,  
  `fhingreso` TIMESTAMP NOT NULL DEFAULT current_timestamp ,  
  UNIQUE INDEX `latitud_UNIQUE` (`latitud` ASC) ,  
  UNIQUE INDEX `longitud_UNIQUE` (`longitud` ASC) ,  
  PRIMARY KEY (`idUbicacion`) )  
ENGINE = InnoDB;
```

Fuente: elaboración propia.

- Creación de la tabla de comisaría: esta tabla tiene un identificador autoincrementable aplicado al campo “idComisaria”, el cual forma parte de la llave primaria. Se tiene una relación con la tabla ubicación representada como llave foránea, como una restricción denominada “fk\_comisaria\_ubicacion1”.

Figura A10. **Script de creación de tabla comisaría**

```
-----  
-- Table `guatesegura`.`comisaria`  
-----  
  
DROP TABLE IF EXISTS `guatesegura`.`comisaria` ;  
  
CREATE TABLE IF NOT EXISTS `guatesegura`.`comisaria` (  
  `idComisaria` INT NOT NULL AUTO_INCREMENT ,  
  `ubicacion_idUbicacion` INT NOT NULL ,  
  `descripcion` VARCHAR(50) NOT NULL ,  
  `telefono` VARCHAR(10) NULL ,  
  `colormarcador` VARCHAR(25) NULL ,  
  `descmarcador` VARCHAR(45) NULL ,  
  `fhingreso` TIMESTAMP NOT NULL DEFAULT current_timestamp ,  
  PRIMARY KEY (`idComisaria`, `ubicacion_idUbicacion`) ,  
  INDEX `fk_comisaria_ubicacion1` (`ubicacion_idUbicacion` ASC) ,  
  CONSTRAINT `fk_comisaria_ubicacion1`  
    FOREIGN KEY (`ubicacion_idUbicacion` )  
    REFERENCES `guatesegura`.`ubicacion` (`idUbicacion` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Fuente: elaboración propia.

- Creación de la tabla denuncia: esta tiene un identificador autoincrementable aplicado al campo “idDenuncia”, el cual se define como llave primaria de la tabla. Las relaciones se representan restricciones aplicando llaves foráneas a cada uno de los campos de las tablas con las cuales se tiene dicha relación.
- El campo “ubicacion\_idUbicacion” tiene una restricción “fk\_denuncia\_ubicacion1”, el cual representa una relación con la tabla ubicación al campo “idUbicacion”.

- El campo “mediodelito\_idmedio” tiene una restricción “fk\_denuncia\_mediodelito1”; representa una relación con la tabla de medios de delito al campo “idmedio”.
- El campo “tipodenuncia\_idtipodenuncia” tiene una restricción “fk\_denuncia\_tipodenuncia1”; representa una relación con la tabla de tipos de denuncia al campo “idtipodenuncia”.

Figura A11. **Script de creación de tabla denuncia**

```

-----
-- Table `guatesegura`.`denuncia`
-----
DROP TABLE IF EXISTS `guatesegura`.`denuncia` ;

CREATE TABLE IF NOT EXISTS `guatesegura`.`denuncia` (
  `idDenuncia` INT NOT NULL AUTO_INCREMENT ,
  `titulo` VARCHAR(50) NOT NULL ,
  `fdelito` DATE NOT NULL ,
  `hdelito` TIME NOT NULL ,
  `descripcion` TEXT NOT NULL ,
  `ubicacion_idUbicacion` INT NOT NULL ,
  `mediodelito_idmedio` INT NOT NULL ,
  `tipodenuncia_idtipodenuncia` INT NOT NULL ,
  `fhingreso` TIMESTAMP NOT NULL DEFAULT current_timestamp ,
  PRIMARY KEY (`idDenuncia`) ,
  INDEX `fk_denuncia_ubicacion1` (`ubicacion_idUbicacion` ASC) ,
  INDEX `fk_denuncia_mediodelito1` (`mediodelito_idmedio` ASC) ,
  INDEX `fk_denuncia_tipodenuncial` (`tipodenuncia_idtipodenuncia` ASC) ,
  CONSTRAINT `fk_denuncia_ubicacion1`
    FOREIGN KEY (`ubicacion_idUbicacion`)
    REFERENCES `guatesegura`.`ubicacion` (`idUbicacion`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_denuncia_mediodelito1`
    FOREIGN KEY (`mediodelito_idmedio`)
    REFERENCES `guatesegura`.`mediodelito` (`idmedio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_denuncia_tipodenuncial`
    FOREIGN KEY (`tipodenuncia_idtipodenuncia`)
    REFERENCES `guatesegura`.`tipodenuncia` (`idtipodenuncia`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Fuente: elaboración propia.

- Ingreso de datos en la tabla de tipo de denuncias y medio del delito: los registros ingresados para las tablas de tipo de denuncias y medios del delito son definidos y configurados previamente para el uso de la aplicación. Estos son ingresados manualmente a la base de datos como se muestra en las siguientes figuras, para las tablas tipo de denuncia y medio del delito, respectivamente.

Figura A12. **Script de ingreso de registros en tipo denuncia**

```
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Abuso','AMARILLO','ABUSO');
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Amenaza','AZUL','AMENAZA');
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Asalto','ROJO','ASALTO');
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Asesinato','ANARANJADO','ASESINATO');
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Estafa','VERDE','ESTAFA');
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Extorsión','VIOLETA','EXTORSIÓN');
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Fraude','ROSA','FRAUDE');
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Homicidio','CIAN','HOMICIDIO');
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Robo','AZURE','ROBO');
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Secuestro','AZURE','SECUESTRO');
INSERT INTO tipodenuncia (descripcion,colormarcador,descmarcador) VALUES ('Otros','','OTROS');
```

Fuente: elaboración propia.

Figura A13. **Script de ingreso de registros en medios del delito**

```
INSERT INTO mediodelito(descripcion,colormarcador,descmarcador) VALUES ('Autobus','AMARILLO','AUTOBUS');
INSERT INTO mediodelito(descripcion,colormarcador,descmarcador) VALUES ('Calle','VIOLETA','CALLE');
INSERT INTO mediodelito(descripcion,colormarcador,descmarcador) VALUES ('Automovil','ROJO','CARRO');
INSERT INTO mediodelito(descripcion,colormarcador,descmarcador) VALUES ('Motocicleta','AZUL','MOTO');
INSERT INTO mediodelito(descripcion,colormarcador,descmarcador) VALUES ('Pasarela','VERDE','PASARELA');
INSERT INTO mediodelito(descripcion,colormarcador,descmarcador) VALUES ('Otros','','OTROS');
```

Fuente: elaboración propia.

Es necesario indicar la descripción del registro, ya que este campo es el que se muestra como opciones en un listado dentro de la aplicación. También se debe definir un color y descripción para el marcador, el cual es reflejado en el mapa al realizar las búsquedas por medio de estos campos.



### A3. Código fuente del desarrollo del software

- Variables constantes: los proyectos de GuateSegura, tanto cliente como servidor utilizan una clase con el nombre constante.java, que se encuentra dentro del paquete com.guatesegura.vars de un proyecto externo denominado gs-lib.

Figura A14. Código fuente de clase constante

```
package com.guatesegura.vars;

import java.util.Properties;

public class constante {

    public static final String constConnectionURL="jdbc:mysql://localhost:3306/guatesegura";
    public static final String constConector = "com.mysql.jdbc.Driver";
    public static final String constUsrDB = "root";
    public static final String constPassDB = "1234";
    public static final String SERVICE_URL = "http://192.168.0.15:8080/wsGuaSegura/rest/denuncia";

    public static final int DATE_DIALOG_ID = 100;
    public static final int TIME_DIALOG_ID = 999;
    public static final int RQS_GooglePlayServices = 1;

    public static final int POST_TASK = 1;
    public static final int GET_TASK = 2;
    public static final String TAG = "WebServiceTask";

    public static final int CONN_TIMEOUT = 3000;
    public static final int SOCKET_TIMEOUT = 5000;

    public final static String DENTITULO = "dentitulo";
    public final static String DENDESC= "dendescripcion";
    public final static String DENFDELITO = "fdelito";
    public final static String DENHDELITO = "hdelito";
    public final static String DENDIRECCION = "dendireccion";
    public final static String DENTIPDENUNCIA = "dentipoden";
    public final static String DENMEDIODELITO = "denmediodel";
    public final static String DENUBILATITUD = "denublatitud";
    public final static String DENUBILONGIT = "denublongitud";

}
```

Fuente: elaboración propia.

- Conexión a base de datos: para la conexión a la base de datos se utiliza una clase denominada dbConnection.java, que contiene un método de nombre “getConnection”, el cual retorna un objeto Connection, utilizado para realizar las operaciones en la base de datos.

Figura A15. **Código fuente de clase de conexión a base de datos**

```
package com.guatesegura.conexion;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import com.gs.vars.constante;

public class dbConnection {
    public Connection getConnection() throws Exception {
        try {
            Connection connection = null;
            Class.forName(constante.constConector).newInstance();
            connection = DriverManager.getConnection(
                constante.constConnectionURL,
                constante.constUsrDB,
                constante.constPassDB);
            System.out.println("Conexion exitosa a la BD");
            return connection;
        } catch (SQLException e) {
            throw e;
        } catch (Exception e) {
            throw e;
        }
    }
}
```

Fuente: elaboración propia.

- Clases de transporte de datos: a continuación se presenta cada una de las clases utilizadas para el desarrollo de la aplicación móvil y servidor web de GuateSegura.

Figura A16. Clase transporte datos de denuncia

```
package com.guatesegura.dto;

public class dtoDenuncia {
    private long id;
    private String titulo;
    private dtoTipoDenuncia tipoDenuncia;
    private dtoMedioDelito medioDelito;
    private String fechaDelito,horaDelito;
    private String descripcion;
    private dtoUbicacion ubicacion;

    public dtoDenuncia(){
        tipoDenuncia = new dtoTipoDenuncia();
        medioDelito = new dtoMedioDelito();
        ubicacion = new dtoUbicacion();
    }

    public long getId() {
    public void setId(long id) {
    public String getHoraDelito() {
    public void setHoraDelito(String horaDelito) {
    public String getTitulo() {
    public void setTitulo(String titulo) {
    public dtoTipoDenuncia getTipoDenuncia() {
    public void setTipoDenuncia(dtoTipoDenuncia tipoDenuncia) {
    public dtoMedioDelito getMedioDelito() {
    public void setMedioDelito(dtoMedioDelito medioDelito) {
    public String getFechaDelito() {
    public void setFechaDelito(String fechaDelito) {
    public String getDescripcion() {
    public void setDescripcion(String descripcion) {
    public dtoUbicacion getUbicacion() {
    public void setUbicacion(dtoUbicacion ubicacion) {
    public void getResumenDenuncia(){
}
```

Fuente: elaboración propia.

Figura A17. **Clase transporte datos tipo denuncia**

```
package com.guatesegura.dto;

public class dtoTipoDenuncia {
    private int idTipoDenuncia;
    private String colormarcador;
    private String descripcion;
    private String descmarcador;

    public int getIdTipoDenuncia() {

    }

    public void setIdTipoDenuncia(int idTipoDenuncia) {

    }

    public String getDescripcion() {

    }

    public void setDescripcion(String descripcion) {

    }

    public String getColormarcador() {

    }

    public void setColormarcador(String colormarcador) {

    }

    public String getDescmarcador() {

    }

    public void setDescmarcador(String descmarcador) {

    }
}
```

Fuente: elaboración propia.

Figura A18. **Clase transporte datos de medio de delito**

```
package com.guatesegura.dto;

public class dtoMedioDelito {

    private int idTipoDelito;
    private String colormarcador;
    private String descripcion;
    private String descmarcador;

    public int getIdTipoDelito() {

    }

    public void setIdTipoDelito(int idTipoDelito) {

    }

    public String getDescripcion() {

    }

    public void setDescripcion(String descripcion) {

    }

    public String getColormarcador() {

    }

    public void setColormarcador(String colormarcador) {

    }

    public String getDescmarcador() {

    }

    public void setDescmarcador(String descmarcador) {

    }
}
```

Fuente: elaboración propia.

Figura A19. **Clase transporte datos ubicación**

```
package com.guatesegura.dto;

public class dtoUbicacion {
    private double latitud;
    private double longitud;
    private String descripcion;
    private int idUbicacion;
    private boolean existe;

    public boolean isExiste() {
    public void setExiste(boolean existe) {
    public int getIdUbicacion() {
    public void setIdUbicacion(int idUbicacion) {
    public double getLatitud() {
    public void setLatitud(double latitud) {
    public double getLongitud() {
    public void setLongitud(double longitud) {
    public String getDescripcion() {
    public void setDescripcion(String descripcion) {
    }
}
```

Fuente: elaboración propia.

Figura A20. **Clase transporte datos de comisaría**

```
package com.guatesegura.dto;

public class dtoComisaria {
    int idComisaria;
    float colomarcador;
    String telefono;
    String descripcion;
    String descmarcador;
    dtoUbicacion ubicacion;

    public int getIdComisaria() {
    public void setIdComisaria(int idComisaria) {
    public float getColomarcador() {
    public void setColomarcador(float colomarcador) {
    public String getTelefono() {
    public void setTelefono(String telefono) {
    public String getDescripcion() {
    public void setDescripcion(String descripcion) {
    public String getDescmarcador() {
    public void setDescmarcador(String descmarcador) {
    public dtoUbicacion getUbicacion() {
    public void setUbicacion(dtoUbicacion ubicacion) {
    }
}
```

Fuente: elaboración propia.

Figura A21. Clase transporte datos de búsqueda reportes

```
package com.guatesegura.dto;

import java.util.ArrayList;
import java.util.List;

public class dtoBusquedaRep {
    String sKmAlrededor;
    String sLatitudActual, sLongitudActual;
    String sLimitReportes;
    int iTipoBusqueda;
    List<dtoDenuncia> listaDenuncias;

    public dtoBusquedaRep(){
        listaDenuncias = new ArrayList<dtoDenuncia>();
    }

    public List<dtoDenuncia> getListaDenuncias() {
    public void setListaDenuncias(List<dtoDenuncia> listaDenuncias) {
    public String getsKmAlrededor() {
    public void setsKmAlrededor(String sKmAlrededor) {
    public String getsLatitudActual() {
    public void setsLatitudActual(String sLatitudActual) {
    public String getsLongitudActual() {
    public void setsLongitudActual(String sLongitudActual) {
    public String getsLimitReportes() {
    public void setsLimitReportes(String sLimitReportes) {
    public int getiTipoBusqueda() {
    public void setiTipoBusqueda(int iTipoBusqueda) {

}
```

Fuente: elaboración propia.

- Métodos de clases de acceso a datos: a continuación se presentan los métodos que realizan los procesos más importantes desde las clases que permiten el acceso a datos.

Figura A22. Ingreso ubicación georreferencial a base de datos

```
public boolean setDenuncia(dtoDenuncia denuncia){
    PreparedStatement preparedStmt;
    try {
        Statement st = connection.createStatement();
        String sSQL="INSERT INTO denuncia "
            +"(titulo,fdelito,hdelito,descripcion,ubicacion_idUbicacion,"
            +"mediodelito_idmedio,tipodenuncia_idtipodenuncia)"
            +"values ("+"denuncia.getTitulo()"
            +", "+"denuncia.getFechaDelito()"
            +", "+"denuncia.getHoraDelito()"
            +", "+"denuncia.getDescripcion()"
            +", "+"denuncia.getUbicacion().getIdUbicacion()"
            +", "+"denuncia.getMedioDelito().getIdTipoDelito()"
            +", "+"denuncia.getTipoDenuncia().getIdTipoDenuncia()+")";
        st.executeUpdate(sSQL);
        System.out.println("Registro de denuncia ingresado exitosamente.");
        return true;
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
}
```

Fuente: elaboración propia.

Figura A23. Ingreso ubicación georreferencial

```
public boolean setUbicacion(dtoUbicacion ubicacion) {
    try {
        Statement st = connection.createStatement();
        String sSQL="INSERT INTO ubicacion (latitud, longitud, descripcion)"
            +"values ("+"ubicacion.getLatitud()"
            +", "+"ubicacion.getLongitud()"
            +", "+"ubicacion.getDescripcion()+")";
        st.executeUpdate(sSQL);
        System.out.println("Registro de ubicacion, ingresado exitosamente.");
        return true;
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
}
```

Fuente: elaboración propia.

Figura A24. **Ingreso ubicación georreferencial a base de datos**

```
public dtoUbicacion getUbicacion(dtoUbicacion ubicacion) {
    try {
        String sSQL = "SELECT u.idUbicacion, u.latitud, u.longitud, u.descripcion "
            + " FROM ubicacion u WHERE u.latitud="
            + ubicacion.getLatitud()
            + " and longitud="
            + ubicacion.getLongitud();
        PreparedStatement ps = connection.prepareStatement(sSQL);
        ResultSet rs = ps.executeQuery();
        boolean isEmpty=false;
        while (rs.next()) {
            ubicacion.setIdUbicacion(rs.getInt("idUbicacion"));
            ubicacion.setDescripcion(rs.getString("descripcion"));
            isEmpty=true;
        }
        ubicacion.setExiste(isEmpty);
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return ubicacion;
}
```

Fuente: elaboración propia.

Figura A25. **Constructor de clase controladora de datos**

```
private Connection connection;
private daoDenuncia handlerDenuncia;
private daoMedioDelito handlerMedioDelito;
private daoTipoDenuncia handlerTipoDenuncia;
private daoUbicacion handlerUbicacion;
private List<dtoDenuncia> alListaDenuncias;

public controladorDatos(){
    try {
        // Se crea una sola conexion a BD de GuateSegura
        dbConnection database = new dbConnection();
        connection = database.getConnection();
        alListaDenuncias = new ArrayList<dtoDenuncia>();
        // Todos los handler apuntan a la conexion creada por el controlador de datos
        handlerDenuncia = new daoDenuncia(connection);
        handlerMedioDelito = new daoMedioDelito(connection);
        handlerTipoDenuncia = new daoTipoDenuncia(connection);
        handlerUbicacion = new daoUbicacion(connection);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Fuente: elaboración propia.



Figura A26. Ingreso de reporte de denuncias

```
public dtoRespIngreso ingresarDenuncia(dtoDenuncia denuncia){
    System.out.println("ControladorDatos: Realizara el ingreso de la denuncia.");
    dtoUbicacion oUbicacion = handlerUbicacion.getUbicacion(denuncia.getUbicacion());
    dtoRespIngreso oRespuesta;
    if (oUbicacion.isExiste()) {
        System.out.println("Ya existe la ubicacion.");
        denuncia.setUbicacion(oUbicacion);
        handlerDenuncia.setDenuncia(denuncia);
        oRespuesta = new dtoRespIngreso();
        oRespuesta.setbIngresado(true);
        oRespuesta.setsMsg("Ingresado exitosamente");
        return oRespuesta;
    }else{
        if (handlerUbicacion.setUbicacion(oUbicacion)) {
            oUbicacion = handlerUbicacion.getUbicacion(denuncia.getUbicacion());
            if (oUbicacion.isExiste()){
                denuncia.setUbicacion(oUbicacion);
                handlerDenuncia.setDenuncia(denuncia);
                oRespuesta = new dtoRespIngreso();
                oRespuesta.setbIngresado(true);
                oRespuesta.setsMsg("Ingresado exitosamente");
                return oRespuesta;
            }else{
                oRespuesta = new dtoRespIngreso();
                oRespuesta.setbIngresado(false);
                oRespuesta.setsMsg("No se pudo ingresar la ubicacion");
                return oRespuesta;
            }
        }else{
            oRespuesta = new dtoRespIngreso();
            oRespuesta.setbIngresado(false);
            oRespuesta.setsMsg("No se pudo ingresar la ubicacion");
            return oRespuesta;
        }
    }
}
```

Fuente: elaboración propia.



## **ANEXO**

### **Talento aplicado para el bien social**

La Guatemala de hoy, inmersa entre grandes retos por resolver, se caracteriza porque el 80 % de su población es menor de 30 años, y entre otras características, cuenta con una gran penetración de la tecnología de comunicaciones —un poco más de 20 millones de líneas telefónicas—.

El último mes revisé una serie de tesis de la carrera de Ingeniería en Ciencias y Sistemas de la Universidad de San Carlos, que proponen lo que, a mi juicio, es el enfoque correcto para valernos de la ciencia y la tecnología a fin de proponer soluciones a nuestros problemas. Comparto algunas de estas propuestas.

Con la gran cantidad de lindos lugares turísticos que tenemos en el país ¿cómo le facilitamos al turista conocer paquetes de viajes? La propuesta Guate2go es una aplicación para celulares móviles que facilita a los extranjeros tener códigos de búsquedas sencillos para conocer paquetes turísticos del país.

En seguridad ciudadana, ¿cómo podemos dar a conocer los lugares con más problemas de seguridad o conocer qué tan seguro es el lugar donde nos encontramos? La propuesta es la aplicación GuateSegura para celulares, que permite registrar un hecho delictivo con la posición georreferenciada del mismo, a la vez también permite buscar los hechos delictivos más cercanos a cierto número de kilómetros a la redonda de una posición.

Otra propuesta interesante es cómo conocer los supermercados donde se encuentran los mejores precios, la aplicación Superbarato nos lo permite.

Hay enfermedades en la sociedad donde la información no es tan accesible o por temor a la acusación no es buscada. Un grupo de alumnos propone una aplicación celular para obtener información sobre VIH, permite suscribirse a recibir información actualizada de la enfermedad y su tratamiento, además de tener opción en cinco idiomas mayas.

La solución Guatextendida es otra aplicación móvil que permite tener en un solo lugar la información más importante de un municipio del país, da a conocer a los turistas los lugares más bellos de un municipio, con su posición geográfica. En fin, quién mejor que nosotros los guatemaltecos que conocemos nuestras necesidades para que también propongamos soluciones a nuestros retos.

Fuente: Ing. Edgar Santos. <http://www.s21.com.gt/edgar-santos/2013/12/17/talento-aplicado-para-bien-social>. Consulta: octubre de 2014.