



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

**IMPLEMENTACIÓN DE APLICACIÓN MÓVIL DE HECHOS
DELICTIVOS GUATESEGURA**

Obed Alejandro Mazariegos Castillo

Keny Estuardo Pérez Lemus

Asesorado por el Ing. Edgar Estuardo Santos Sutuj

Guatemala, febrero de 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**IMPLEMENTACIÓN DE APLICACIÓN MÓVIL DE HECHOS
DELICTIVOS GUATESEGURA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

OBED ALEJANDRO MAZARIEGOS CASTILLO

KENY ESTUARDO PÉREZ LEMUS

ASESORADO POR EL ING. EDGAR ESTUARDO SANTOS SUTUJ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, FEBRERO 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	Inga. 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. Luis Fernando Quiñónez López
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. Roberto Estuardo Ruiz Cruz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

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

IMPLEMENTACIÓN DE APLICACIÓN MÓVIL DE HECHOS DELICTIVOS GUATESEGURA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha julio de 2014.



Keny Estuardo Pérez Lemus



Obed Alejandro Mazariegos Castillo

Guatemala, 30 de julio de 2014


Ingeniero
Carlos Azurdia
Revisor de Trabajo de Graduación
Escuela de Ciencias y Sistemas
Facultad de Ingeniería

Respetable Ing. Azurdia

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **OBED ALEJANDRO MAZARIEGOS CASTILLO** y **KENY ESTUARDO PEREZ LEMUS**, titulado: **"IMPLEMENTACIÓN DE APLICACIÓN MÓVIL DE HECHOS DELICTIVOS GUATESEGURA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Sin otro particular, me suscribo de usted.

Atentamente,



Edgar Santos
Ingeniero en Ciencias y Sistemas
Colegiado No. 5266
Asesor de Trabajo de Graduación

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, 3 de Septiembre de 2014

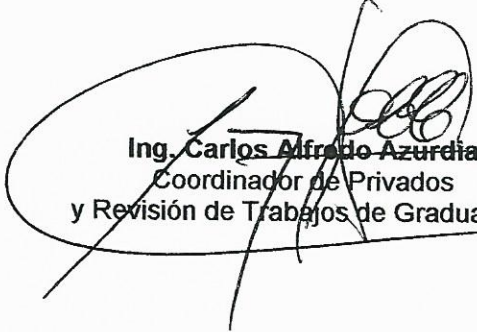
Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **OBED ALEJANDRO MAZARIEGOS CASTILLO** con carné **199910635**, y **KENY ESTUARDO PÉREZ LEMUS** con carné **200112674**, titulado: **"IMPLEMENTACIÓN DE APLICACIÓN MÓVIL DE HECHOS DELICTIVOS GUATESEGURA"**, 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 DE HECHOS DELICTIVOS
C GUATESEGURA”**, realizado por los estudiantes OBED
I ALEJANDRO MAZARIEGOS CASTILLO y KENY
A ESTUARDO PÉREZ LEMUS, aprueba el presente trabajo y
S solicita la autorización del mismo.

Y

“ID Y ENSEÑAD A TODOS”

S
I
S
T
E
M
A
S

Ing. Marlon Antonio Pérez Türk
Director, Escuela de Ingeniería en Ciencias y Sistemas

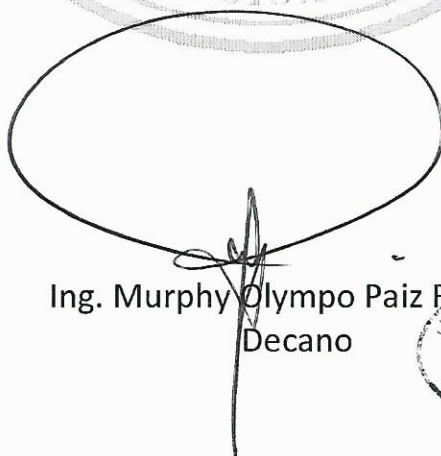


Guatemala, 19 de febrero 2015



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 DE HECHOS DELICTIVOS GUATESEGURA**, presentado por los estudiantes universitarios: **Obed Alejandro Mazariegos Castillo y Keny Estuardo Pérez Lemus**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Ing. Murphy Olympto Paiz Recinos
Decano



Guatemala, 26 de febrero de 2015

/gdech

AGRADECIMIENTOS A:

- Dios** Por darme la oportunidad de la vida, y la capacidad para llegar hasta acá y culminar una etapa más, tanto en mi carrera profesional como en mi desarrollo personal, porque sin Él nada puedo y nada soy.
- Mi madre** Quien en vida me apoyó incondicionalmente, me alentó a seguir adelante y luchar por mis sueños. Gracias mamá, sé que desde la compañía de Dios me has estado apoyando siempre.
- Mi padre** Quien me enseñó con su ejemplo a ser una persona responsable y trabajadora.
- Mi esposa e hija** Quienes me llenan de amor cada día y le dieron un nuevo sentido a mi vida.
- Sergio Castillo** Quien siempre me ha brindado su apoyo y me ha animado a seguir adelante.
- Mis amigos y compañeros** A quienes agradezco su amistad sincera.

Obed Alejandro Mazariegos Castillo

AGRADECIMIENTOS A:

Dios

Por la vida y sabiduría que me dio a lo largo de mi carrera universitaria.

Mi familia

Por el apoyo y amor incondicional para convertirme en un profesional, porque sin su ayuda no sería la persona que actualmente soy. No existirá forma de agradecer un largo camino de sacrificios, esfuerzo y amor que me brindaron para cumplir mi objetivo.

Mis amigos y compañeros

Porque de cada uno de ellos aprendí diferentes conocimientos que me ayudaron a convertirme en un profesional.

Keny Estuardo Pérez Lemus

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	III
GLOSARIO	V
RESUMEN.....	IX
OBJETIVOS.....	XI
INTRODUCCIÓN	XIII
1. MARCO TEÓRICO.....	1
1.1. Calidad de software	1
1.1.1. La gestión de la calidad.....	2
1.1.2. El control de la calidad	2
1.1.3. El aseguramiento de la calidad	3
1.1.4. Métricas de calidad	4
1.1.5. Modelos conocidos	5
1.1.5.1. Modelo de MCCALL.....	6
1.1.5.2. Modelo FURPS	7
1.1.5.3. Modelo Droomey.....	7
1.1.5.4. Normas ISO	8
1.1.6. Aplicaciones móviles.....	9
1.1.7. Sistema operativo Android	10
2. ANÁLISIS DE LA APLICACIÓN	15
2.1. Análisis FURPS a las aplicaciones	18
2.1.1. Características de funcionalidad	18
2.1.2. Características de usabilidad	20
2.1.3. Características de fiabilidad	21

2.1.4.	Características de rendimiento	22
2.1.5.	Características de soporte.....	22
2.2.	Análisis de mejoras.....	23
2.3.	Diseño de mejoras	25
2.3.1.	Diagramas	25
3.	IMPLEMENTACIÓN DE MEJORAS.....	35
3.1.	Entorno gráfico.....	35
3.2.	Pantallas.....	37
3.3.	Tipificación de delitos	41
3.4.	Publicación de aplicación móvil en Google Play	42
	CONCLUSIONES.....	47
	RECOMENDACIONES	49
	BIBLIOGRAFÍA.....	51

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Normas ISO.....	9
2.	Arquitectura de sistema operativo Android.....	11
3.	Aplicación GuateSegura.....	16
4.	Aplicación espantacacos.....	17
5.	Aplicación seguridad en línea.....	18
6.	Diagrama de clases de nueva versión de GuateSegura	26
7.	Diagrama de componentes nueva versión de GuateSegura	27
8.	Diagrama entidad relación nueva versión GuateSegura	27
9.	Diagrama casos de uso nueva versión de GuateSegura	28
10.	Nueva versión de logo GuateSegura	35
11.	Paleta de colores.	35
12.	Opción denunciar/reportar de la nueva versión de GuateSegura	37
13.	Captura de pantalla opción reportes de la nueva versión de GuateSegura.....	38
14.	Captura de pantalla opción mapa de la nueva versión de GuateSegura.....	39
15.	Captura de pantalla de opción de teléfonos de emergencia de la nueva versión de GuateSegura	40
16.	Consola de Google Play para desarrolladores.....	44
17.	Exporta aplicación no firmada.....	44

TABLAS

I.	Clasificación de las métricas de software	5
II.	Modelo de MCCALL	6
III.	Modelo FURPS.....	7
IV.	Modelo Droomey	8
V.	Tabla comparativa de funcionalidades de aplicaciones móviles	24
VI.	Componentes personalizados.....	36
VII.	Tipificación de delitos	41

GLOSARIO

<i>Android</i>	Sistema operativo basado en el <i>kernel</i> de Linux diseñado para dispositivos móviles.
<i>API</i>	Interfaz de programación de aplicaciones (IPA) o API (del inglés <i>Application Programming Interface</i>).
<i>BlackBerry OS</i>	Es un sistema operativo móvil desarrollado para los dispositivos BlackBerry.
C	Es un lenguaje de programación.
C++	Lenguaje de programación extendido del lenguaje de programación C.
Caso de uso	Secuencia de interacciones que se desarrollarán entre un sistema y sus actores.
Código abierto	Es la expresión con la que se conoce al software distribuido y desarrollado libremente.
Diagrama de clases	Tipo de diagrama estático que describe la estructura de un sistema orientado a objetos.
Diagrama de componentes	Representa cómo un sistema de software es dividido en componentes.

Diagrama entidad relación	Representa la estructura y relaciones en una base de datos.
<i>Framework</i>	Conjunto de librerías y funciones que ayudan a desarrollar software.
<i>Georreferencia</i>	Es un neologismo que refiere al posicionamiento con el que se define la localización de un objeto.
<i>Google Maps</i>	Servidor de aplicaciones de mapas en la web que pertenece a Google.
<i>Google Play</i>	Plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android.
<i>Hardware</i>	El término hardware se refiere a todas las partes tangibles de un sistema informático.
Ícono	Gráfico en pantalla que identifica y representa a algún objeto.
Interfaz	Es conexión física y funcional entre dos sistemas o dispositivos de cualquier tipo, dando una comunicación entre distintos niveles. Su plural es interfaces.
<i>IOS</i>	Es un sistema operativo móvil de la empresa Apple Inc.

ISO	Organización Internacional de Normalización.
Java	Es un lenguaje de programación.
Kernel	Es un software que constituye una parte fundamental del sistema operativo.
Linux	Sistema operativo basado en Unix.
Red local	Es una topología de red.
Servidor	Se refiere a las computadoras que van a compartir sus recursos hardware y software con los demás equipos de la red.
SO	Siglas para describir sistema operativo (SO o frecuentemente OS, del inglés <i>Operating System</i>).
Software	Programa informático.
VM	Siglas para describir máquina virtual, siglas en inglés (<i>virtual machine</i>).

RESUMEN

En Guatemala existe mucha información proporcionada por medio de dispositivos móviles; esto debido a la disponibilidad que dan las compañías de telecomunicaciones en el país. Hoy en día se puede visualizar en prensa y televisión que una persona captura el video o imagen de delincuentes. Cada vez existen más personas que utilizan internet y los dispositivos móviles para hacer público los abusos de delincuencia a los que son víctimas o ven en el día.

La Universidad de San Carlos de Guatemala, dentro de sus trabajos de graduación realizó el análisis y prototipo del proyecto GuateSegura, el cual consiste en la prevención ciudadana por medio de los dispositivos móviles Android.

Esta aplicación fue puesta a disposición de la ciudadanía el 24 junio del 2014; a través de ella se pueden realizar denuncias georreferenciales y ser visualizadas en internet; en la misma aplicación, la información se centraliza en un servidor y la ciudadanía puede tomar las prevenciones debidas, según los reportes.

La aplicación brinda un indicador de las zonas peligrosas del país, así como también los hechos delictivos que tienen mayor incidencia. En países como Colombia y México, la tecnología móvil facilita a las autoridades identificar las regiones con más índices de hechos delictivos y los ciudadanos puedan generar una denuncia formal por medio de estas tecnologías.

GuateSegura, que es una aplicación móvil de la Universidad de San Carlos de Guatemala, tiene como objetivo en un mediano plazo convertirse o proporcionar la tecnología y ejemplo de cómo la tecnología podría facilitar la transmisión de denuncias y se pueda actuar de una manera más rápida de lo que se hace hoy en día.

OBJETIVOS

General

Implementar mejoras a la aplicación móvil GuateSegura y permita centralizar la información de hechos delictivos para que sea de beneficio a los ciudadanos y entidades públicas.

Específicos

1. Implementar mejoras de funcionalidad, usabilidad, fiabilidad, rendimiento y soporte a la aplicación en la tecnología móvil Android.
2. Poner al alcance de los ciudadanos de Guatemala la aplicación por medio de Google Play.
3. Hacer un análisis comparativo de aplicaciones similares determinando ventajas y desventajas, describiendo las desventajas de la aplicación GuateSegura

INTRODUCCIÓN

En Guatemala ha surgido mucha información que es proporcionada por medio de un dispositivo móvil. Hoy en día se puede visualizar en prensa, televisión y redes sociales, que una persona captura vídeo o imagen de un hecho delictivo; dicha información es utilizada para alertar a las demás personas y utilizada por las autoridades para la captura de estos delincuentes.

En países como Colombia y México, la tecnología móvil facilita a las autoridades identificar las regiones con más índices de hechos delictivos y los ciudadanos puedan generar una denuncia formal por medio de estas tecnologías.

Dado que en Guatemala no existe una aplicación disponible de este tipo, se implementó el proyecto GuateSegura, que es una aplicación móvil de la Universidad de San Carlos de Guatemala realizada por los estudiantes de Ingeniería. Esta aplicación no llegó a estar disponible a los ciudadanos; por tal motivo surge el presente trabajo de investigación e implementación de mejoras a la aplicación GuateSegura.

En este trabajo de graduación se presenta un análisis de ventajas y desventajas con aplicaciones similares, implementación de mejoras y proceso de publicación.

1. MARCO TEÓRICO

1.1. Calidad de software

La calidad de software se refiere a una serie de características propias del software aquellas que requieren controlar y asegurar, el software es un producto inmaterial que no se fabrica, tampoco se degrada físicamente, sino que se desarrolla. El software puede tener errores, incidencias pero no son similares a lo que cualquier equipo de carácter físico.

La calidad del software se encuentra casi a la par de la calidad tradicional, ligeramente detrás debido a que la calidad tradicional tiene varias décadas de historia, mientras que la calidad de software tiene entre 50 y 30 años de haber surgido.

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, con el fin de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto de la labor de desarrollo, como del control de la calidad del software.

Es la eficiencia y producción de su rendimiento y funcionamiento del equipo “La calidad del software es una preocupación a la que se dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla y si puede, supere las expectativas de los usuarios.”

En la calidad de software existen 3 partes que tienen relación entre ellas mismas;

- Gestión de la calidad
- Control de la calidad
- Aseguramiento de la calidad

1.1.1. La gestión de la calidad

Son aspectos de la función de gestión que determinan y aplican la política de la calidad, los objetivos, responsabilidades y que lo realiza con medios tales como la planificación, el control, la garantía y la mejora de la calidad. Dentro de la gestión de la calidad se observa:

- Gestión de la calidad de software (ISO 9000): conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y responsabilidades, se implanta por medios tales como la planificación, control, aseguramiento (garantía) y la mejora, en el marco del sistema de calidad
- Política de calidad (ISO 9000): directrices y objetivos generales de una organización, relativos a la calidad, tal como se expresan formalmente por la alta dirección.

1.1.2. El control de la calidad

Técnicas y actividades de carácter operativo, utilizadas para verificar los requisitos relativos a la calidad, centrados en mantener bajo control el proceso

de desarrollo y eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

El control de la calidad del software está centrado en dos objetivos fundamentales:

- Mantener bajo control un proceso
- Eliminar las causas de los defectos en las diferentes fases del ciclo de vida

1.1.3. El aseguramiento de la calidad

Es un conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto (software) satisfaga los requerimientos dados sobre calidad.

El aseguramiento de calidad del software se diseña para que cada aplicación antes de comenzar a desarrollarla, cumpla con los requerimientos mínimos que se esperan de la aplicación y que esta brinde confianza a los usuarios.

Las actividades para el aseguramiento de calidad del software se detallan en:

- Métricas de software
- Verificación y calidad del software a lo largo del ciclo de vida (incluye las pruebas y procesos de revisión e inspección)
- La gestión de la configuración del software

1.1.4. Métricas de calidad

La métrica es una medida del grado en que un sistema, componente o proceso posee un atributo dado

Las métricas son un buen medio para entender, monitorear, controlar, predecir y probar el desarrollo de software y los proyectos de mantenimiento.

En general, la medición persigue 3 objetivos: ayudar a entender qué ocurre durante el desarrollo, controlar lo que ocurre en los proyectos y mejorar procesos y productos.

Las métricas del software comprenden un amplio rango de actividades:

- Aseguramiento y control de calidad
- Modelos de fiabilidad
- Modelos y evaluación de ejecución
- Modelos y medidas de productividad

Existen diferentes clasificaciones de las métricas de software, las cuales se presentan en la siguiente tabla.

Tabla I. **Clasificación de las métricas de software**

De complejidad	Métricas que definen la medición de la complejidad: volumen, tamaño, anidaciones, y configuración
De calidad	Métricas que definen la calidad del software: exactitud, estructuración o modularidad, pruebas de mantenimiento.
De competencia	Métricas que intentan valorar o medir las actividades de productividad de los programadores respecto de su certeza, rapidez, eficiencia y competencia.
De desempeño	Métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del SO o hardware.
Estilizadas	Métricas de experimentación y de referencia: estilo de código, convenciones, limitaciones, etc.

Fuente: <http://ingosoftmet.blogspot.com>. Consulta: marzo de 2014.

Según el contexto, recopilan de todos los proyectos, y durante un largo periodo de tiempo. Son caracterizadas por:

- Control y ejecución del proyecto
- Medición de tiempos de las fases
- Permiten evaluar el estado del proyecto

1.1.5. Modelos conocidos

Dentro de los modelos conocidos para las métricas de calidad de software existen los siguientes.

- Modelo de MCCALL (1977)
- Modelo de FURPS (1987)

- Modelo de DROMEY (1996)
- Normas ISO 9000 e ISO/IEC 9126

1.1.5.1. Modelo de MCCALL

Este modelo describe la calidad como un concepto elaborado mediante relaciones jerárquicas entre factores de calidad, con base en criterios definidos. Los factores de calidad se concentran en tres aspectos importantes de un producto de software: características operativas, capacidad de cambios y adaptabilidad a nuevos entornos. Las métricas desarrolladas se relacionan con los factores de calidad y la relación que se establece se mide en función del grado de cumplimiento de los criterios. Identifica una serie de criterios, tales como rastreabilidad, simplicidad, capacidad de expansión, etc.

Tabla II. Modelo de MCCALL

Factor	Criterio	Factor	Criterio
Correctitud	Rastreabilidad Completitud y consistencia	Mantenibilidad	Simplicidad concreción
Confiabilidad	Consistencia Exactitud Tolerancia a fallas	Capacidad de prueba	Simplicidad Instrumentación Autodescriptividad Modularidad
Eficiencia	Eficiencia de ejecución Eficiencia de almacenamiento	Flexibilidad	Autodescriptividad Capacidad de expresión Generalidad Modularidad
Integridad	Control de acceso Auditoría de acceso	Portabilidad	Autodescriptividad Independencia del sistema Independencia de máquina
Usabilidad	Operabilidad Entrenamiento Comunicación	Reusabilidad	Autodescriptividad Generalidad y modularidad Independencia del sistema Independencia de máquina
Interoperabilidad	Modularidad Similitud de comunicación Similitud de datos		

Fuente: <http://ingsoftmet.blogspot.com/>. Consulta: marzo de 2014.

1.1.5.2. Modelo FURPS

Modelo desarrollado por Hewlett Packard (HP) en 1987, empleando un conjunto de factores de calidad de software y sus respectivos atributos, con base en el modelo MCCALL. Se utilizan para establecer métricas de la calidad para todas las actividades del proceso de desarrollo de un software.

Tabla III. Modelo FURPS

Factor	Criterio	Factor	Criterio
Funcionabilidad	Características y capacidad del programa Generalidad de las funciones Seguridad del sistema	Rendimiento	Velocidad del procedimiento Tiempo de respuesta Consumo de recursos Rendimiento efectivo total Eficiencia
Factibilidad de uso	Factores humanos Factores estéticos Consistencia de la interfaz Documentación	Capacidad de soporte	Extensibilidad Adaptabilidad Capacidad de pruebas Capacidad de configuración Compatibilidad Requisitos de instalación
Confiabilidad	Frecuencia y severidad de las fallas Exactitud de las salidas Tiempo medio de fallos Capacidad de recuperación ante fallas Capacidad de predicción		

Fuente: <http://ingosoftmet.blogspot.com>. Consulta: marzo de 2014.

1.1.5.3. Modelo Droomey

Resalta el hecho de que la calidad del producto se determina con los componentes del mismo (incluyendo documentos de requisitos, guías de usuarios, diseños y código).

Sugiere el uso de cuatro categorías que implican propiedades de calidad, que son: correctitud (correcciones), internas, contextuales y descriptivas.

Tabla IV. **Modelo Droomey**

Factor	Criterio
Correctitud	Funcionalidad Confiabilidad
Internas	Mantenibilidad Eficiencia Confiabilidad
Contextuales	Mantenibilidad Reusabilidad Portabilidad Confiabilidad
Descriptivas	Mantenibilidad Reusabilidad Portabilidad Usabilidad

Fuente: <http://ingosoftmet.blogspot.com>. Consulta: marzo de 2014.

1.1.5.4. Normas ISO

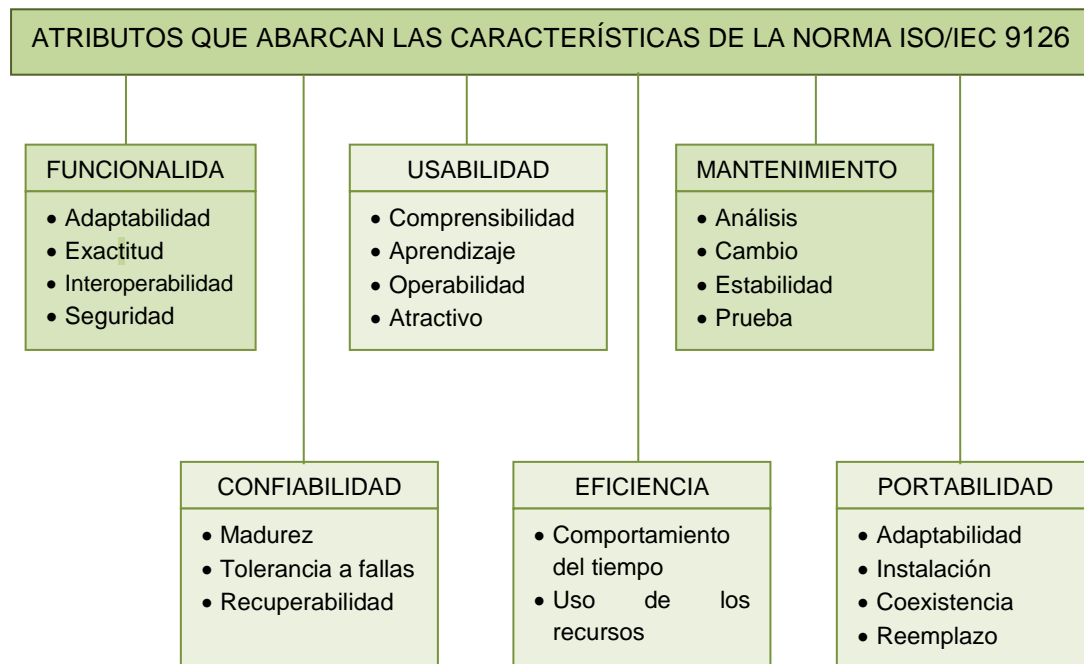
Las normas ISO-9000 son un estándar de calidad para todo tipo de industrias. Contienen una normativa específica para el desarrollo de software, la ISO-9003. Consisten en una serie de cláusulas que deben aplicarse en el marco de trabajo, en el ciclo de vida del proyecto y en las actividades de apoyo al mismo.

La aplicación de las normas ISO en un proyecto ha de demostrarse mediante una evaluación formal por un organismo competente.

Las normas ISO tienen sin embargo ciertos inconvenientes:

- Son estáticas, de escaso valor, lentas y muy caras
- No están plenamente orientadas al software
- En muchos casos se han adoptado por obligación y para “cubrir el expediente”, lo que no mejora demasiado la calidad

Figura 1. **Normas ISO**



Fuente: <http://ingosoftmet.blogspot.com>. Consulta: marzo de 2014.

1.1.6. **Aplicaciones móviles**

Es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles.

Por lo general se encuentran disponibles a través de plataformas de distribución operadas por las compañías propietarias de los sistemas operativos móviles como Android, IOS, BlackBerry OS, Windows Phone, entre otros. Existen aplicaciones móviles gratuitas y otras de pago, donde en promedio del 20 a 30 % del costo de la aplicación se destina al distribuidor, y el resto es para el desarrollador.

El desarrollo de aplicaciones para dispositivos móviles requiere tener en cuenta las limitaciones de estos dispositivos. Los dispositivos móviles funcionan con batería y tienen procesadores menos poderosos que los ordenadores personales. Los desarrollos de estas aplicaciones también tienen que considerar una gran variedad de tamaños de pantalla, datos específicos de software y configuraciones.

El desarrollo de aplicaciones móviles requiere el uso de entorno de desarrollos integrados. Las aplicaciones móviles suelen ser probadas, primero usando emuladores y más tarde se ponen en el mercado en periodo de prueba. Actualmente un gran número de empresas se dedica a la creación profesional de aplicaciones

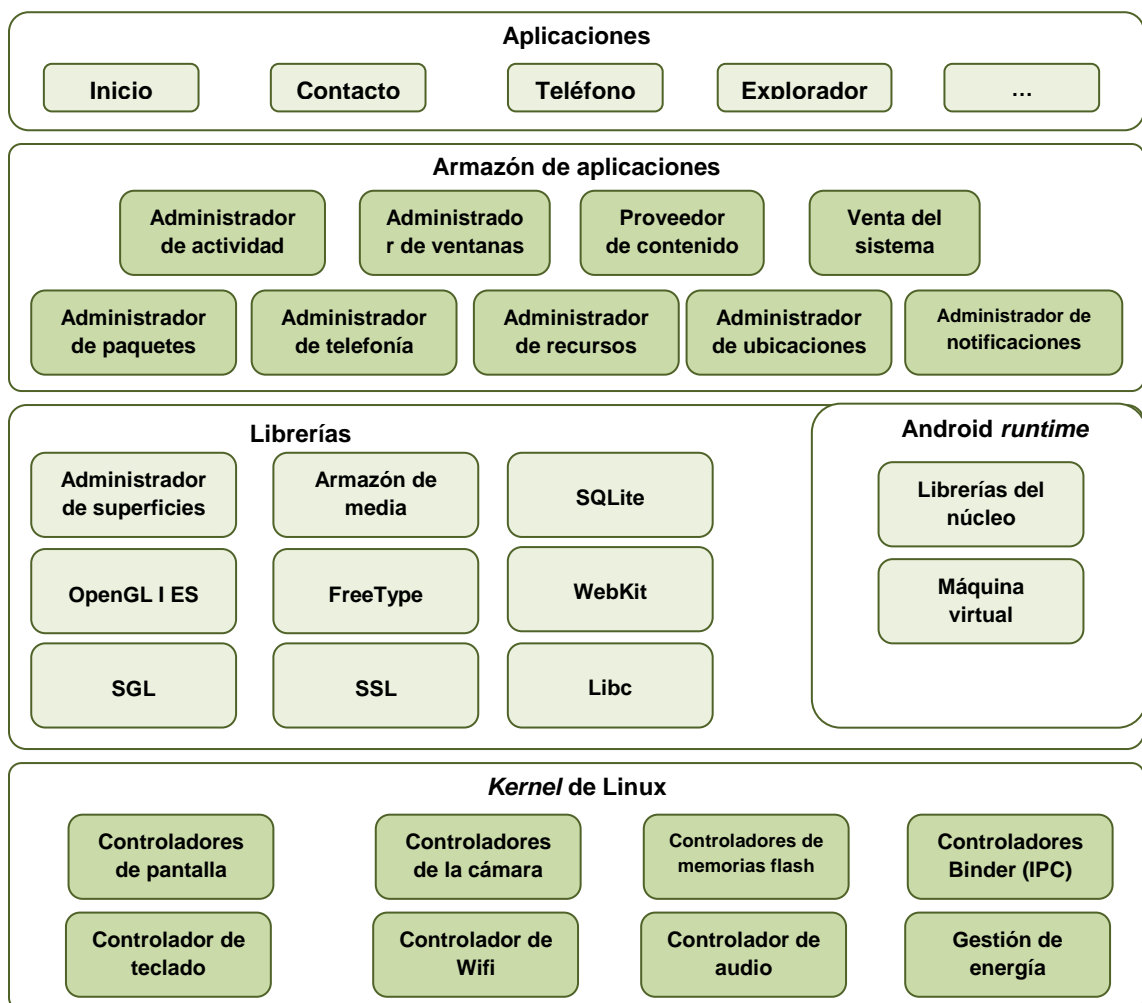
1.1.7. Sistema operativo Android

Es un sistema operativo basado en el *kernel* de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, inicialmente desarrollado por Android; Inc. Android está basado en Linux, un sistema operativo libre, gratuito y multiplataforma.

El lenguaje de programación que se utiliza para desarrollar aplicaciones para este sistema es una variante del lenguaje Java llamada Dalvik.

La existencia de múltiples herramientas gratuitas para empezar a programar aplicaciones para Android ha hecho posible el crecimiento de la comunidad de programadores que han decidido desarrollar para esta plataforma, ofreciendo tan extensa y variada lista de aplicaciones. En el gráfico mostrado a continuación se puede observar la arquitectura de este sistema operativo.

Figura 2. **Arquitectura de sistema operativo Android**



Fuente: <http://ingosoftmet.blogspot.com>. Consulta: marzo de 2014.

- Aplicación: el *framework* de aplicaciones proporciona muchas interfaces de programación. Llamando a las funciones de estas interfaces, las aplicaciones se pueden diseñar fácilmente, simplificando y reutilizando recursos. Además, a través de las funciones proporcionadas por el *framework*, se puede reescribir el *kernel* original de aplicaciones de Android. El *framework* de aplicaciones proporciona las siguientes interfaces funcionales:
 - Proveedor de contenido: para permitir a las aplicaciones acceder a datos y a otros programas de aplicación.
 - Gestor de recursos: para gestionar las partes que no son programa, tales como la codificación multilenguaje, fotos y efectos de sonido.
 - Gestor de notificaciones: para permitir a las aplicaciones notificar a la interfaz de mensajes de usuarios.
- Librerías: las librerías de Android incluyen librerías enlazadas de C y C++, conectadas hacia arriba con el *framework* de aplicaciones, y hacia abajo con el *kernel* del SO. La librería principal incluye una de sistema C, de multimedia, de red, de base de datos, de 3D y de fuentes. La mayoría de librerías mejoran los programas de sistemas embebidos, dándoles alta eficiencia de ejecución, códigos de programas más pequeños, y funciones reducidas.
- *Runtime* de Android: esta capa incluye la librería del *kernel* y la máquina virtual Dalvik, la cual se usa para ejecutar los programas de Java; la VM Dalvik está especialmente mejorada para sistemas embebidos, y sus

programas ocupan menor espacio de memoria; al mismo tiempo que es capaz de ejecutar varios programas simultáneamente.

- *Kernel* Linux: se basa en el *kernel* de Linux, versión 2.6, para los servicios básicos del sistema tales como gestión de memoria y de procesos, pilas de red, seguridad, y modelo de *drivers*. El *kernel* también actúa como una capa de abstracción hardware entre las aplicaciones, y todo el hardware.



2. ANÁLISIS DE LA APLICACIÓN

Utilizando el modelo FURPS se describen atributos de calidad que debe tener un software; a continuación se realiza un análisis a diferentes aplicaciones móviles, utilizando los atributos descritos por el modelo FURPS.

“GuateSegura” es una aplicación desarrollada por estudiantes de la Escuela de Ciencias y Sistemas, Facultad de Ingeniería, de la Universidad de San Carlos de Guatemala (ver tabla I). Desarrollada para denunciar actos delincuenciales con posicionamiento georreferencial entre sus características principales.

Con características funcionales que brindan información al usuario sobre actos delincuenciales cercanos y visualización de los reportes más recientes en un listado o mapa; la aplicación brinda una buena opción para la prevención y seguridad ciudadana.¹

¹ <http://ingsoftmet.blogspot.com/>. Consulta: marzo 2014

Figura 3. Aplicación GuateSegura

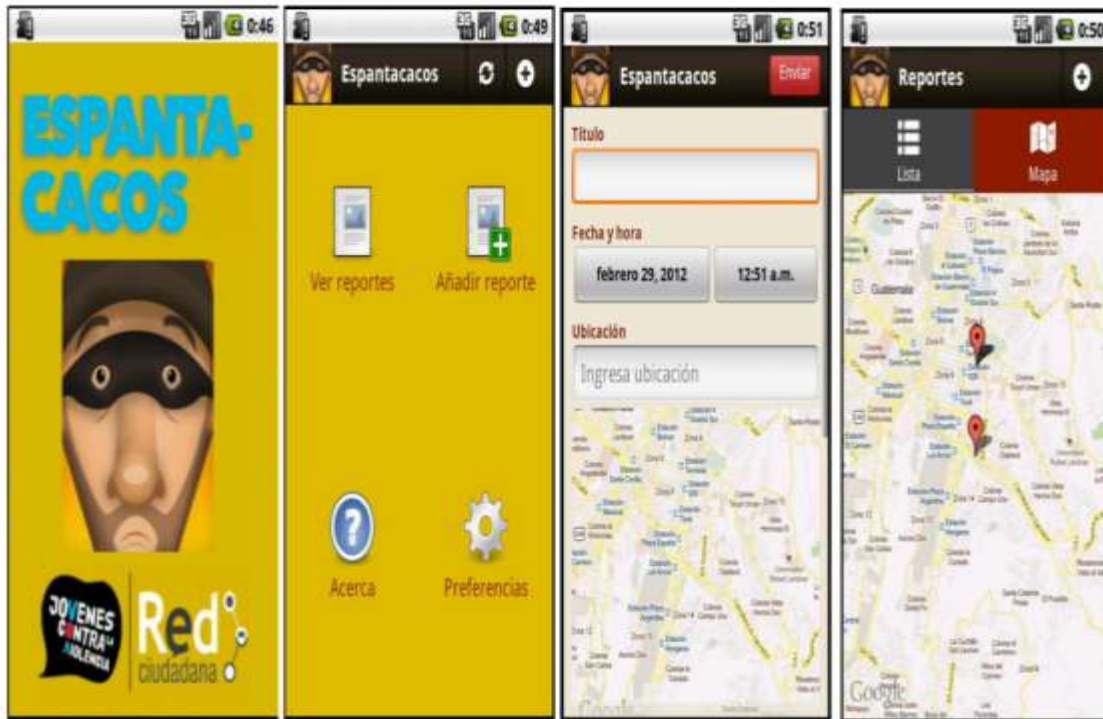


Fuente: imagen proporcionada por SALÁN CALDERÓN, Diego Alberto; QUIEM BATZ, Jhony Anthony.

“Espantacacos es una aplicación desarrollada por la organización jóvenes contra la violencia (ver tabla II), en un esfuerzo de brindar a la población guatemalteca una herramienta para denunciar los actos delincuenciales que viven a diario. Tratando de hacer ver a las autoridades las zonas de mayor riesgo y dónde opera la delincuencia, con la finalidad que se tomen acciones contra la delincuencia”.²

² <http://ingsoftmet.blogspot.com/>. Consulta: marzo 2014

Figura 4. Aplicación espantacacos

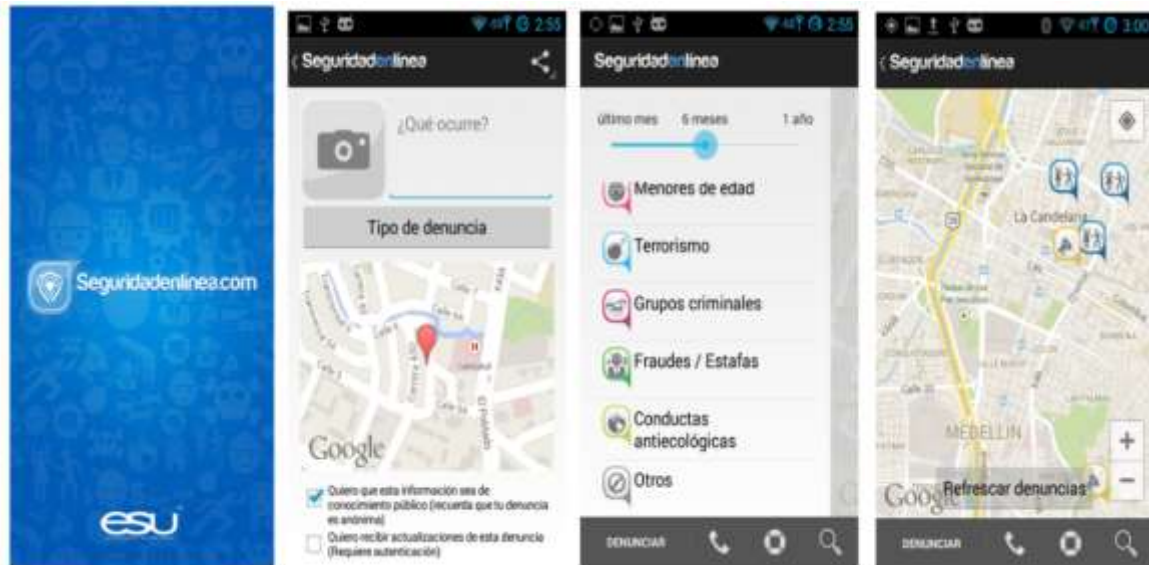


Fuente: <http://www.redciudadana.org/Proyectos/espantacacos/>. Consulta: mayo de 2014.

“Seguridad en línea” es una aplicación desarrollada por las autoridades de la ciudad de Bogotá y Medellín, Colombia (ver tabla III), con el fin de brindarle a los habitantes de dichas ciudades un medio para denunciar y reportar a las autoridades actos delincuenciales.

Una herramienta muy completa que funciona en conjunto con el sitio web en donde cada denuncia es tomada en cuenta por las autoridades de la ciudad, dándole un giro al proceso tradicional de denuncias telefónicas y presenciales; ahora los ciudadanos de Bogotá y Medellín tienen una herramienta tecnológica para realizar las denuncias.

Figura 5. **Aplicación seguridad en línea**



Fuente: <https://seguridadenlinea.com>. Consulta: marzo de 2014.

2.1. **Análisis FURPS a las aplicaciones**

Para la evaluación de las aplicaciones GuateSegura, espantacacos y seguridad en línea, se presenta a continuación un análisis en donde se utiliza el modelo FURPS que detalla las características principales de cada aplicación.

2.1.1. **Características de funcionalidad**

- GuateSegura:
 - Formulario para ingreso de denuncia georeferencial
 - Formulario para visualización de reportes
 - Reportes de denuncias mapeados a un mapa
 - Funcionamiento únicamente dentro de una red local

- No tiene soporte en línea
- Ingreso de denuncia no tiene la opción para adjuntar imagen
- Disponible únicamente para sistemas Android

- Espantacacos:
 - Formulario para ingreso de denuncia georreferencial
 - Formulario para visualización de reportes
 - Es posible adjuntar una imagen desde la cámara o galería a una denuncia.
 - Reporte de denuncias mapeados a un mapa
 - Centralización de reportes en internet
 - Disponible únicamente para sistemas Android

- Seguridad en línea:
 - Formulario para ingreso de denuncia georreferencial
 - Formulario para visualización de reportes
 - Es posible adjuntar una imagen desde la cámara o galería a una denuncia.
 - Reporte de denuncias mapeados a un mapa
 - Centralización de reportes en internet
 - Disponible para SO Android, BlackBerry e IOS
 - Filtros para visualizar los reportes en mapa y listado
 - Directorio de teléfonos de emergencia

2.1.2. Características de usabilidad

- GuateSegura:
 - Utilización de controles estándares y pocos íconos
 - Poca consistencia gráfica
 - El ingreso de denuncia solicita demasiados datos
 - Tipos de delitos únicamente son descritos por texto
 - No existe diferenciación al mapear denuncias a un mapa
 - Colores apagados y pocos amigables
 - Poco intuitivo
 - Contiene filtros configurables por el usuario
 - No existe documentación
 - No utilizable con redes sociales

- Espantacacos
 - Utilización de íconos personalizados
 - Consistencia gráfica en todos los formularios
 - Intuitivo y fácil de usar
 - El ingreso de denuncia solicita demasiados datos
 - Ubicación de la denuncia no es posible cambiarla
 - Visualización de denuncias disponibles en sitio web
 - No utilizable con redes sociales

- Seguridad en línea:
 - Utilización de íconos personalizados
 - Intuitivo y fácil de usar

- Consistencia gráfica en todos los formularios
- Utilización de íconos para cada tipo de delito
- Únicamente disponible para Bogotá y Medellín
- Utilización de redes sociales
- Brinda un mapa de calor en su sitio web, visualizando las áreas de mayores denuncias.

2.1.3. Características de fiabilidad

- GuateSegura:
 - No controla estado de conexión a su servidor local
 - Poca validación a datos de ingreso
 - Disponible y dependiente del servidor local

- Espantacacos
 - Control de excepciones para el estado de conexión a internet
 - Validación a datos de ingreso
 - Utilización de servicios en internet

- Seguridad en línea:
 - Control de excepciones para el estado de conexión a internet
 - Validación a datos de ingreso
 - Utilización de servicios en internet

2.1.4. Características de rendimiento

- GuateSegura:
 - Inicio de la aplicación en tiempo aceptable
 - Envío de denuncia a servicios locales en tiempo aceptable
 - Visualización de reportes en listado y en mapa en tiempo aceptable

- Espantacacos
 - Inicio de la aplicación en tiempo aceptable
 - Envío de denuncia a servicios en internet, en tiempo aceptable
 - Visualización de reportes en listado y en mapa en tiempo aceptable

- Seguridad en línea:
 - Inicio de la aplicación en tiempo aceptable
 - Envío de denuncia a servicios en internet, en tiempo aceptable
 - Visualización de reportes en listado y en mapa en tiempo aceptable
 - Compartir reportes en redes sociales en tiempo aceptable

2.1.5. Características de soporte

- GuateSegura:
 - Fácil de instalar
 - No disponible en Google Play
 - Pocos requisitos para su instalación
 - No existe medio para brindar soporte
 - Poco extensible, clases altamente acopladas

- Espantacacos:
 - Fácil de instalar
 - Disponible en Google Play
 - Pocos requisitos para su instalación
 - Tiene soporte vía correo electrónico

- Seguridad en línea:
 - Fácil de instalar
 - Disponible en Google Play
 - Pocos requisitos para su instalación
 - Tiene soporte vía correo electrónico
 - Funcionalidad extensible al sitio web

2.2. Análisis de mejoras

Para determinar los requerimientos funcionales y no funcionales de la aplicación móvil GuateSegura se realiza una tabla comparativa (ver tabla V) con las características que actualmente tienen las aplicaciones en el mercado.

Tabla V. **Tabla comparativa de funcionalidades de aplicaciones móviles**

Características principales	GuateSegura	Espantacacos	Seguridad en línea
Consistencia gráfica	NO	SÍ	SÍ
Íconos personalizados	NO	SÍ	SÍ
Usabilidad	POCA	REGULAR	BUENA
Redes sociales	NO	NO	SÍ
Georeferencial	SÍ	SÍ	SÍ
Visualización de la información en sitio web	NO	SÍ	SÍ
Soporte	NO	SÍ	SÍ
Adjuntar imagen desde la cámara	NO	SÍ	SÍ
Adjuntar imagen desde la galería	NO	SÍ	SÍ
Pública en Google Play	NO	SÍ	SÍ

Fuente: elaboración propia.

A continuación se presenta el listado de los requerimientos funcionales y no funcionales a implementar en la aplicación GuateSegura.

- Mejorar la consistencia gráfica de la aplicación
- Mejoras en logo y colores base
- Utilizar íconos personalizados
- Personalizar controles (botones, listas, marcadores de mapas, etc.)

- Mejorar la usabilidad de la aplicación, haciendo que el ingreso de la denuncia sea lo más rápido y sencillo para el usuario
- Mejorar la visualización de la información tanto en mapa como en el listado
- Utilizar medios digitales para brindar mayor detalle a las denuncias; utilizando la cámara o la galería de imágenes
- Hacer visible la información en un sitio web
- Utilizar las redes sociales para compartir la información
- Hacer pública la aplicación en Google Play
- Brindar un sitio web para publicaciones de soporte

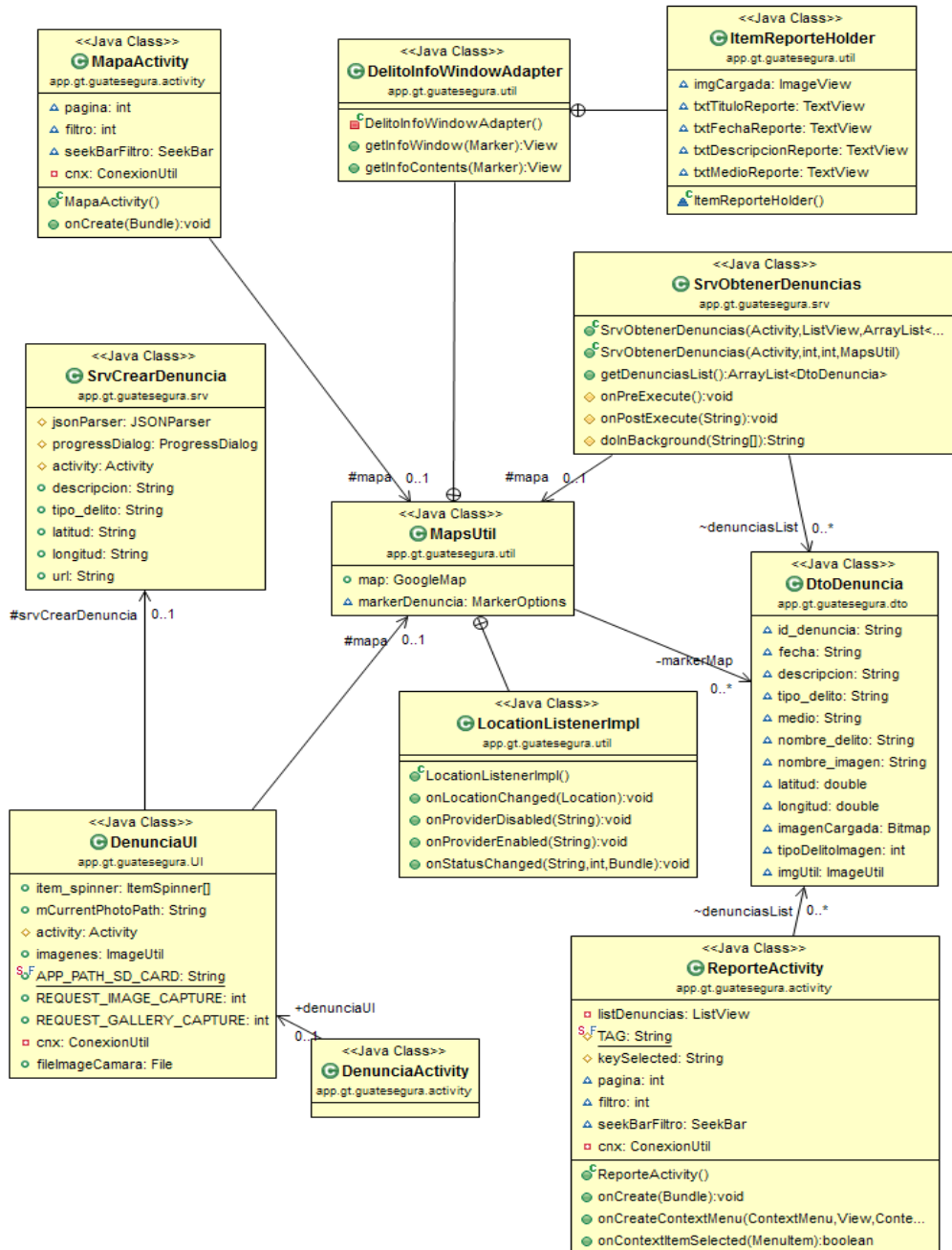
2.3. Diseño de mejoras

Para la implementación de mejoras a continuación se describe en forma gráfica, diferentes modelos conceptuales que soportan las nuevas características funcionales necesarias para brindar una aplicación competitiva en el mercado de aplicaciones móviles.

2.3.1. Diagramas

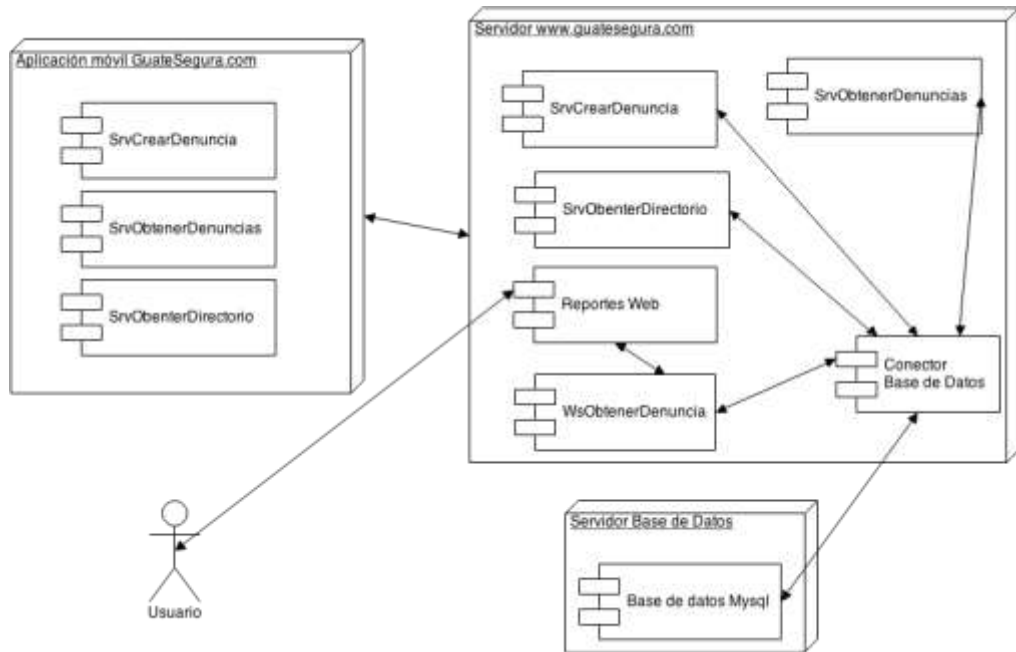
Los diferentes diagramas descritos en esta sección, comprenden modelos y lineamientos de programación, estructura de la información e interacción de los componentes necesarios a implementar. Dichos modelos se apegan a los estándares de programación y brindan un panorama general del funcionamiento de las mejoras en la aplicación.

Figura 6. Diagrama de clases de nueva versión de GuateSegura



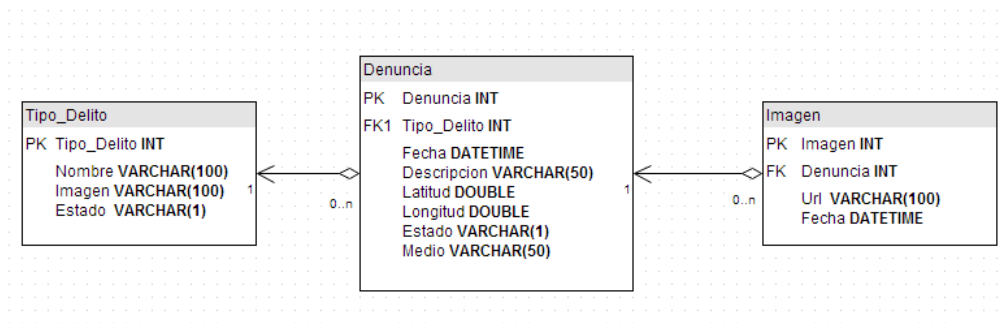
Fuente: elaboración propia.

Figura 7. Diagrama de componentes nueva versión de GuateSegura



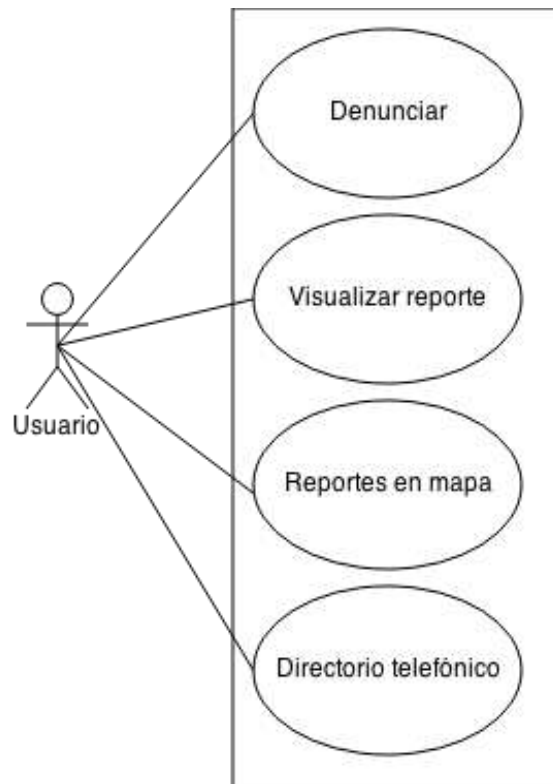
Fuente: elaboración propia.

Figura 8. Diagrama entidad relación nueva versión GuateSegura



Fuente: elaboración propia.

Figura 9. **Diagrama casos de uso nueva versión de GuateSegura**



Fuente: elaboración propia.

- Caso de uso
 - Caso de uso: denunciar
 - Objetivo: reportar actos delincuenciales
 - Actores: aplicación GuateSegura
 - Precondiciones: servicio de internet activo
 - Flujo básico
 - ✓ El usuario ingresa a la aplicación
 - ✓ El usuario selecciona la opción denunciar/reportar

- ✓ El sistema muestra el formulario para reportar Denuncias.
 - ✓ El usuario ingresa descripción
 - ✓ El usuario selecciona el tipo de denuncia
 - ✓ El usuario ubica en el mapa el lugar donde ocurrió la denuncia.(FA1) (FA2).
 - ✓ El usuario presiona el botón reportar
 - ✓ El sistema realiza las validaciones (VA1), (VA2)
 - ✓ El sistema envía el reporte
 - ✓ Fin
- FA1: Adjuntar imagen desde la galería
 - ✓ El usuario presiona el botón adjuntar imagen desde galería.
 - ✓ El sistema muestra la galería de imágenes
 - ✓ El usuario selecciona la imagen
 - ✓ El usuario confirma la selección de imagen
 - ✓ El sistema obtiene la imagen y la adjunta al reporte
 - ✓ El sistema regresa al paso 7 del flujo principal
 - FA2: Adjuntar imagen desde la cámara del dispositivo móvil
 - ✓ El usuario presiona el botón adjuntar imagen desde la cámara.
 - ✓ El sistema abre la interfaz de la cámara
 - ✓ El usuario toma una fotografía
 - ✓ El usuario confirma la fotografía tomada
 - ✓ El sistema obtiene la imagen y la adjunta al reporte

- ✓ El sistema regresa al paso 7 del flujo principal
- VA1: El sistema verifica estado de la conexión a internet
 - ✓ El sistema verifica que el estado de conexión a internet este activo.
 - ✓ Si no existe conexión, el sistema muestra mensaje de error: “No existe conexión a internet”, el sistema retorna al paso 10 del flujo básico.
 - ✓ Sistema retorna al paso 9
- VA2: El sistema valida que exista descripción
 - ✓ El sistema verifica que descripción no este vacía
 - ✓ Si no se ingresó descripción, el sistema muestra mensaje de error: “Debes ingresar una descripción”; el sistema retorna al paso 10 del flujo básico.
 - ✓ Sistema retorna al paso 9
- Poscondiciones: el sistema regresa al menú principal
- Caso de uso: visualizar reporte
 - Objetivo: visualizar los reportes más recientes en un listado
 - Actores: aplicación GuateSegura
 - Precondiciones: servicio de internet activo
 - Flujo básico
 - ✓ El usuario ingresa a la aplicación
 - ✓ El usuario selecciona la opción reportes (VA1)
 - ✓ El sistema obtiene los 5 reportes más recientes

- ✓ El sistema muestra los reportes en un listado (FA1), (FA2)
- ✓ Fin

- FA1: El usuario presiona el botón ver más.
 - ✓ El sistema obtiene los siguientes 5 reportes más recientes
 - ✓ El sistema agrega los reportes al listado
 - ✓ Regresa al paso 5 del flujo principal

- FA2: El usuario filtra la información
 - ✓ El usuario selecciona un filtro (5 km o 10 km)
 - ✓ El sistema obtiene la ubicación actual
 - ✓ El sistema obtiene los reportes más recientes a la distancia de 5 km o 10 km de la posición actual
 - ✓ Se retorna al paso 4 del flujo principal.

- VA1: No existe conexión a internet
 - ✓ El sistema verifica que el estado de conexión a internet este activo
 - ✓ Si no existe conexión, el sistema muestra mensaje de error: "No existe conexión a internet", el sistema retorna al paso 5 del flujo básico
 - ✓ Sistema retorna al paso 3

- Poscondiciones: ninguna

- Caso de uso: reportes en mapa
 - Objetivo: visualizar los reportes más recientes en un mapa
 - Actores: aplicación GuateSegura
 - Precondiciones: servicio de internet activo
 - Flujo básico
 - ✓ El usuario ingresa a la aplicación
 - ✓ El usuario selecciona la opción mapa.(VA1)
 - ✓ El sistema obtiene los 5 reportes más recientes
 - ✓ El sistema muestra los reportes en un mapa (FA1), (FA2)
 - ✓ Fin
 - FA1: El usuario presiona el botón ver más
 - ✓ El sistema obtiene los siguientes 5 reportes más recientes
 - ✓ El sistema agrega los reportes al mapa
 - ✓ Regresa al paso 5 del flujo principal
 - FA2: El usuario filtra la información
 - ✓ El usuario selecciona un filtro (5 km o 10 km)
 - ✓ El sistema obtiene la ubicación actual
 - ✓ El sistema obtiene los reportes más recientes a la distancia de 5 km o 10 km de la posición actual
 - ✓ Se retorna al paso 4 del flujo principal
 - VA1: No existe conexión a internet
 - ✓ El sistema verifica que el estado de conexión a internet este activo

- ✓ Si no existe conexión, el sistema muestra mensaje de error: “No existe conexión a internet”, el sistema retorna al paso 5 del flujo básico
 - ✓ Sistema retorna al paso 3
 - Poscondiciones: ninguna
- Caso de uso: directorio telefónico
 - Objetivo: visualizar números de emergencia en un listado
 - Actores: aplicación GuateSegura
 - Precondiciones: servicio de internet activo
 - Flujo básico
 - ✓ El usuario ingresa a la aplicación
 - ✓ El usuario selecciona la opción “Teléfonos de Emergencia” (VA1)
 - ✓ El sistema obtiene los teléfonos
 - ✓ El sistema muestra la información en un listado (FA1)
 - ✓ Fin
 - FA1: El usuario filtra la información
 - ✓ El sistema obtiene la entidad seleccionada
 - ✓ El sistema obtiene los teléfonos (VA1)
 - ✓ Regresa al paso 4 del flujo principal
 - VA1: No existe conexión a internet
 - ✓ El sistema verifica que el estado de conexión a internet este activo

- ✓ Si no existe conexión, el sistema muestra mensaje de error: "No existe conexión a internet"; el sistema retorna al paso 5 del flujo básico
 - ✓ Sistema retorna al paso 3
-
- Poscondiciones: ninguna

3. IMPLEMENTACIÓN DE MEJORAS

3.1. Entorno gráfico

Se realizó el rediseño de la aplicación GuateSegura para lograr una consistencia gráfica uniforme, a continuación se listan las mejoras realizadas.

- Mejoras en el diseño del logo:

Figura 10. Nueva versión de logo GuateSegura



Fuente: elaboración propia.

- Creación de una paleta de colores a utilizar:

Figura 11. Paleta de colores



Fuente: elaboración propia.

- Personalización de componentes:

Tabla VI. Componentes personalizados

Cuadros de texto	
Lista de selección	
Botones	
Marcadores en mapa	
Listas	
Íconos personalizados	

Fuente: elaboración propia.

3.2. Pantallas

Implementación de mejoras en las pantallas de ingreso, según caso de usos definidos en la sección 2.3.2.

- Pantalla: denunciar

Figura 12. **Opción denunciar/reportar de la nueva versión de GuateSegura**



Fuente: elaboración propia.

- Pantalla: visualizar reporte

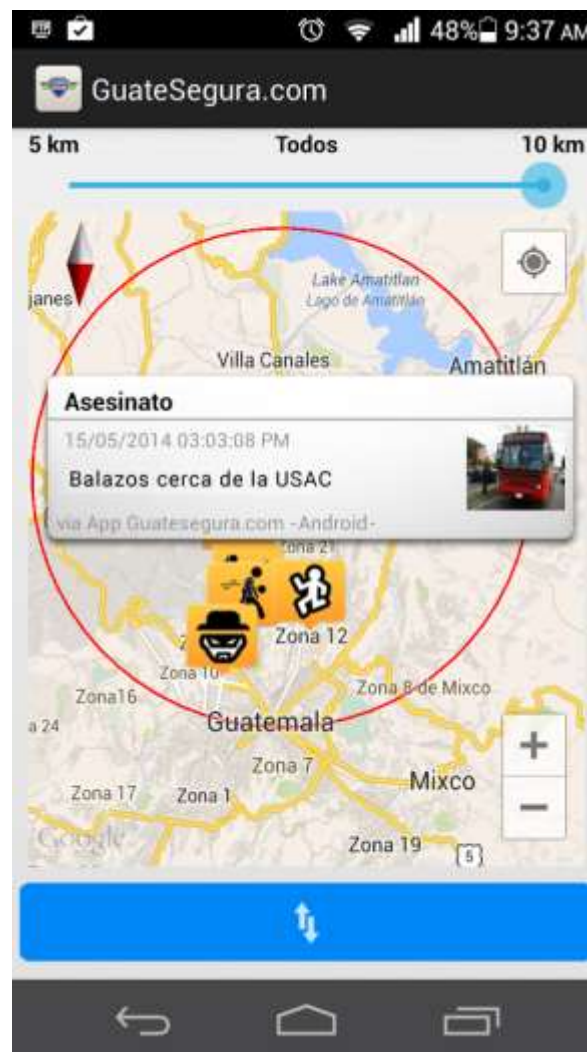
Figura 13. **Captura de pantalla opción reportes de la nueva versión de GuateSegura**



Fuente: elaboración propia.

- Caso de uso: reportes en mapa

Figura 14. **Captura de pantalla opción mapa de la nueva versión de GuateSegura**



Fuente: elaboración propia.

- Caso de uso: directorio telefónico

Figura 15. **Captura de pantalla de opción de teléfonos de emergencia de la nueva versión de GuateSegura**



Fuente: elaboración propia.

3.3. Tipificación de delitos

Se realizó la tipificación de los delitos más comunes reportados en redes sociales.

Tabla VII. Tipificación de delitos

Ícono	Descripción
	Robo
	Asesinato
	Ataque armado
	Accidente vehicular
	Vandalismo
	Secuestro
	Extorsión
	Violación
	Ventas ilegales
	Estado de ebriedad
	Soborno
	Incendio
	Fraude

Fuente: elaboración propia.

3.4. Publicación de aplicación móvil en Google Play

En la siguiente sección se describe el proceso realizado para publicar la aplicación móvil en Google Play y pueda ser pública para la población.³

- Registro como desarrollador
 - Como primer punto se debe contar con una cuenta de correo en Gmail, la cual actuará como titular del registro. El costo del registro tiene un valor de 25 dólares, pago único.
 - Se debe ingresar al siguiente link: <https://play.google.com/apps/publish/> y completar los datos que Google solicitará en un tiempo máximo de 48 horas, después de realizar el pago, el registro en la consola de Google Play para desarrolladores(ver figura 16) estará activa.
 - Una vez está activa la cuenta, se puede publicar la aplicación móvil llenando los datos requeridos para su publicación; entre ellos están: título de la aplicación, descripción, texto promocional, capturas de pantalla, ícono en alta resolución, categorización de la aplicación, información de contacto y políticas de privacidad.
 - Para publicar la aplicación móvil se debe firmar y alinear.

³ <https://support.google.com/googleplay/android-developer/answer/113468?hl=es>. Consulta: febrero de 2014.

Figura 16. Consola de Google Play para desarrolladores

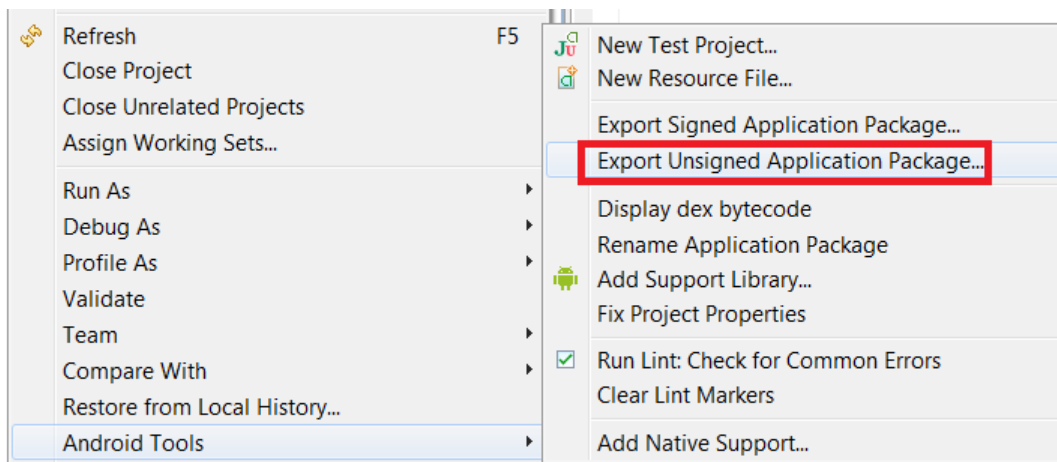


Fuente: elaboración propia

- Firmar y alinear la aplicación móvil
 - Para firmar una aplicación móvil debes generar un firma, que se realiza de la siguiente forma.
 - Abrir la consola de Windows y ejecutar el siguiente comando.`keytool -genkey -v -keystore guatesegura.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 10000`. Donde: guatesegura.keystore: es el nombre de la firma; este nombre puede ser cambiado a alias_name: alias que le asignará a la firma.

- Al aplicar el comando pedirá una contraseña. Se escribe la contraseña sin olvidarla, porque se usará más adelante. Este paso generará un archivo de extensión .keystore
- Exportar la aplicación móvil no firmado
 - Clic derecho sobre el proyecto. Seleccionar Android Tools
 - Seleccionar Export Unsigned Application Package

Figura 17. **Exporta aplicación no firmada**



Fuente: elaboración propia, utilizando herramienta Android.

- Firmar la aplicación: se utiliza el archivo .keystore generado anteriormente, para firmar la aplicación se ejecuta el siguiente comando:
`jarsigner -verbose -sigalg MD5withRSA -digestalg SHA1 -keystoreguatesegura.keystoreguatesegura.apkalias_name.` Entonces:
 - guatesegura.keystore: es el archivo generado anteriormente

- guatesegura.apk: es el nombre de la aplicación no firmada
- alias_name: es el nombre del alias con el que se generó la firma
- Alinear la aplicación móvil: se utiliza la herramienta Zipalign incluida en el SDK tools, ya firmada la aplicación, lo siguiente es alinearla. Para alinearla se debe ejecutar el siguiente comando: `zipalign -v 4 guatesegura.apkguateseguraPro.apk` por tanto:
 - guatesegura.apk: es la aplicación firmada no alineada.
 - guateseguraPro.apk: es la aplicación firmada alineada y lista para ser publicada.

CONCLUSIONES

1. El modelo de calidad FURPS utilizado incluye los requisitos necesarios que deben evaluarse en el desarrollo de aplicaciones móviles. Los diferentes requisitos que propone el modelo de calidad FURPS se adaptan de la mejor manera a las capacidades y condiciones que una aplicación móvil debe cumplir.
2. El *benchmarking* constituye una herramienta que ayuda a que la aplicación móvil a implementar sea competitiva en el mercado al cual se lanzará, determinando a través de esta herramienta los requerimientos funcionales mínimos que se deben tomar en cuenta al momento de desarrollar e implementar una aplicación móvil.
3. El diseño impacta en gran manera en la usabilidad y visualización que el usuario experimenta; el fácil uso y una interfaz gráfica amigable, es de gran relevancia en la implementación y aceptación en el mercado al cual se lanzará la aplicación móvil.

RECOMENDACIONES

1. Utilizar un modelo de calidad de software para determinar requisitos y métricas que se deben evaluar al desarrollar e implementar una aplicación móvil. El uso de un modelo de calidad de software ayuda en gran manera a garantizar el éxito de una aplicación móvil.
2. Desarrollar aplicaciones Android tiene menor costo de implementación y publicación comparado con los sistemas IOS y Windows Phone. La plataforma Android brinda herramientas de código abierto, las cuales hacen que el desarrollo de proyectos en dispositivos móviles tenga un menor costo.
3. Realizar estrategias de promoción para ayudar a hacer llegar la aplicación móvil a los usuarios generadores de información. Una aplicación móvil pierde relevancia si no tiene usuarios y si no genera información; la estrategia de promoción ayuda a llegar a los usuarios potenciales que darán relevancia a la aplicación móvil.
4. La diversidad de dispositivos móviles que existen actualmente es sumamente grande, un factor importante en el éxito de una aplicación móvil lo determina el número de fallos que tenga en los diferentes dispositivos. Es de suma importancia crear diferentes ambientes de pruebas cuando se desarrolla para dispositivos móviles; esto ayudará en gran manera a mitigar errores en la diversidad de los mismos.

BIBLIOGRAFÍA

1. AJIN, Doris. *Guatesegura, una app guatemalteca diseñada para ayudar*. [en línea]. <<http://www.soy502.com/articulo/guatesegura-una-app-guatemalteca-disenada-para-ayudar-0>>. [Consulta: 12 de junio de 2014].
2. *Bitsando*. [en línea]. <<http://www.bitsando.com/2013/01/firmar-aplicaciones-para-publicarlas-en.html>>. [Consulta: 01 de mayo de 2014].
3. GONZÁLEZ DORIA, Heidi. *Métricas en el desarrollo del software*. [en línea]. <[http://catarina.udlap.mx / u_dl_a / tales / documentos /lis/gonzalez _d_h/capitulo4.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo4.pdf)>. [Consulta: 15 de febrero de 2014].
4. GOOGLE. *Registro de desarrolladores*. [en línea]. <<https://support.google.com/googleplay/androiddeveloper/answer/13468?hl=es>>. [Consulta: 15 de febrero de 2014].
5. LÍNEA. *Seguridad en línea*. [en línea]. <<https://seguridadenlinea.com/>>. [Consulta: 03 de marzo de 2014].
6. OZAETA, Paula. *App GuateSegura* [en línea]. <<https://www.youtube.com/watch?v=XDnTVVHsFzc>>. [Consulta: 19 de junio de 2014].

7. Red Ciudadana. *Espantacacos*. [en línea]. <[http:// www.redciudadana.org / Proyectos / espantacacos />. \[Consulta: 10 de febrero de 2014\].](http://www.redciudadana.org/Proyectos/espantacacos/)
8. RIVAS, Jaime Roberto. *La arquitectura Android*. [en línea]. <[http://catedratelefonica.ulpgc.es / blog / arucas / 2012/10 / 18 / la-arquitectura-de-android/>. \[Consulta: 12 de marzo de 2014\].](http://catedratelefonica.ulpgc.es/blog/arucas/2012/10/18/la-arquitectura-de-android/)
9. SALÁN CALDERÓN, Diego Alberto; QUIEM BATZ, Johny Anthony. Entrevista personal. [Consulta: 15 de marzo 2014]. Guatemala. Comunicación personal.
10. SAN GABINO, Niurka; HERNÁNDEZ, Camilo; NORIEGA, Darcy. *Calidad del software*. [en línea]. <[http://www.monografias.com/ trabajos59 /calidad-Software/calidadSoftware.shtml](http://www.monografias.com/trabajos59/calidad-Software/calidadSoftware.shtml)>. [Consulta: 3 de marzo de 2014].
11. SANTOS, Edgar. *GuateSegura, un software ejemplar*. [en línea]. <[http://www.s21.com.gt / agexport / 2014 / 08 / 12 / guatesegura-Software-ejemplar](http://www.s21.com.gt/agexport/2014/08/12/guatesegura-Software-ejemplar)>. [Consulta: 12 de agosto de 2014].
12. SILVA, Guadalupe. *Métricas de calidad de software*. [en línea]. <[http://ingsoftmet.blogspot.com/>. \[Consulta: 13 de marzo de 2014\].](http://ingsoftmet.blogspot.com/)