

2016

GEOLOCALIZACIÓN DE EVENTOS DELICTIVOS EN APLICACIÓN MÓVIL SOCIAL Y COMUNITARIA

RAMÍREZ PÉREZ, MANUEL JESÚS

Universidad Técnica Federico Santa María

<http://hdl.handle.net/11673/13613>

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTRÓNICA

VALPARAÍSO – CHILE



**GEOLOCALIZACIÓN DE EVENTOS DELICTIVOS A
TRAVÉS DE UNA APLICACIÓN MÓVIL SOCIAL Y
COMUNITARIA**

MANUEL JESÚS RAMÍREZ PÉREZ

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL TELEMÁTICO

PROFESOR GUÍA:

Marcos Zuñiga Barraza

PROFESORA CORREFERENTE:

Cecilia Reyes Covarrubias

NOVIEMBRE 2016

“Dedicado a Dios, mi familia y futura esposa...”

Agradecimientos

A Dios, por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor. Agradezco a mi familia por acompañarme en todos los momentos importantes de mi vida. A mis padres, Margarita y Manuel, por su cariño y apoyo incondicional, por creer en mi y alentarme cada vez que lo necesité, por enseñarme a enfrentar los obstáculos con optimismo, por mostrarme que la perseverancia y el esfuerzo son el camino para lograr objetivos. A mi novia Bárbara, que siempre ha estado a mi lado, entregando su amor, cariño y apoyo incondicional, siempre alentándome a cumplir mis sueños y metas. A mis hermanas, Margarita, Claudia y Evelyn por su cariño y por confiar siempre en mí. A mis amigos, gracias, por sus demostraciones de afecto y apoyo, en especial a Carlos Zamorano, amigo y socio que constantemente me ha dado consejos, aprendiendo mucho de él. A mi profesor guía, Marcos Zuñiga, por brindarme su apoyo y conocimientos, no tan solo en el transcurso de esta memoria, sino que también en los diferentes ramos que cursé con él, entregando en cada momento un nuevo conocimiento con mucho entusiasmo y alegría. A mis profesores y a todos los que en estos siete años aportaron en mi crecimiento personal y profesional, formando una parte muy importante de esta gran etapa de mi vida. A mi tía Pía Baeza que estuvo hasta el último minuto apoyando en la revisión de esta memoria. A mis amigos de VersusCrime Miguel, Angello y José en conjunto al equipo completo de Memorias Multidisciplinarias que hicieron posible el desarrollo de esta memoria. A todos ellos, Muchas Gracias.

Resumen

El objetivo de este trabajo, es que mediante una aplicación móvil, se registre un mapa de las zonas inseguras de las ciudades, logrando que la ciudadanía pueda hacer uso de esta información y tomar los resguardos necesarios. Así como también, tiene por objetivo interesar a futuros clientes, a los cuales sea de utilidad esta información, como lo son las empresas de seguridad, cadenas comerciales, las constructoras y organismos del estado, que deben velar por la seguridad social. La innovación del proyecto es desarrollar una herramienta de reporte delictivo rápida y eficiente, disponible para los usuarios que posean un smartphone, aprovechando al máximo esta tecnología para potenciar su seguridad. Para el desarrollo de este proyecto se implementó un sistema de arquitectura Cliente-Servidor. Estos dos módulos son: una aplicación cliente en el equipo Android, encargada de capturar y enviar los reportes delictivos, con sus coordenadas geográficas; y un servidor web, encargado de almacenar y procesar la información de posición geográfica de los distintos delitos. La aplicación se conecta a Google Maps para descargar los mapas, y marcar sobre ellos la posición de los últimos delitos reportados por los usuarios del sistema. El desarrollo de este trabajo se llevó a cabo a través del programa de Memorias Multidisciplinarias, de la UTFSM en su primera versión; este programa tiene por objetivo la creación sistemática y sustentable de productos de innovación y emprendimientos en equipos de estudiantes de distintas carreras.

Palabras Claves: Android, Aplicaciones móviles, Seguridad ciudadana, Reportes delictivos.

Abstract

The aim of this work is to register a map of insecure areas of the cities using a smartphone accomplishing that the citizenship make good use of this information been able to take the necessary safeguard actions. Likewise, its objective is to attract future customers for whom this information is useful, such as security agencies, retail chains, construction companies as well as state entities who must ensure the security of the population. The novelty of the project is to develop a fast and efficient tool available for users who own a smartphones, taking full advantage of this technology to enhance their security. For the development of this project was implement a Client-Server architecture system. These two modules are: a client application on the Android device, responsible for capturing and sending criminal reports, with their geographic coordinates; and a web server, responsible for storing and processing information on geographical position of the various crimes. The application connects to Google Maps to download maps, and marks on them the position of the latter offenses reported by users of the system. The development of this work was carried out through the *Memorias Multidisciplinarias* training program for systematic and sustainable development of products linked to technology innovation.

Keywords: Android, Mobile Applications, Public Safety, Criminal Reports.

Índice de Contenidos

Resumen	IV
Abstract	V
Índice de Contenidos	VI
Índice de Tablas	XI
Índice de Figuras	XIV
Glosario	XVI
1. Introducción	1
1.1. Propósito	1
1.2. Alcance	1
1.3. Contexto	2
1.3.1. Definición del Problema	3
Victimización general	4
Principales motivos para no denunciar (Hurto)	5
Exposición al delito 2014	5
1.4. Objetivos generales	6
1.5. Objetivos específicos	6
1.6. Estructura del documento	7
2. Estado del Arte	8
2.1. Introducción a Android	8
2.1.1. Inicios de Android	8
2.1.2. Arquitectura	9
2.2. Situación Actual	12
2.2.1. Situación actual del mercado de aplicaciones móviles	12
Análisis de sistemas operativos	12
2.2.2. Migración móvil en Chile	13
2.2.3. Análisis del tipo de conexión de internet móvil	13

2.2.4.	Descargas de aplicaciones	14
	Por sistema operativo en el mundo	14
	Por sistema operativo en Chile	15
2.2.5.	Ingresos de aplicaciones por sistema operativo	16
2.2.6.	Crecimiento del uso de smartphone por categoría	17
2.2.7.	Panorama actual de los dispositivos móviles en Chile	18
2.2.8.	Demanda potencial	19
2.2.9.	Proyecciones	20
	Usuarios y penetración de internet en Latinoamérica	20
	Usuarios y penetración de smartphones en Latinoamérica	21
	Ingresos de las aplicaciones móviles en el mundo	22
2.3.	Mercado objetivo	23
2.3.1.	Empresas	23
2.3.2.	Personas	24
	Características generales	24
	Tipos de consumidores por adopción de producto	24
2.4.	Análisis del mercado	25
2.4.1.	Competencia	25
3.	Análisis de Requerimientos	26
3.1.	Requisitos del Sistema	26
3.1.1.	Requisitos funcionales	26
3.1.2.	Requisitos de Interfaces	27
3.2.	Requisitos de Ambiente	29
3.2.1.	Descripción del Hardware de desarrollo	29
3.2.2.	Descripción del Software de desarrollo	29
4.	Análisis y Diseño	30
4.1.	Arquitectura del Sistema	30
4.1.1.	Diagrama de Flujos	31
4.2.	Descripción de módulos	32
4.2.1.	Módulo login y registro (C1)	32
4.2.2.	Módulo vista principal (C2)	32
4.2.3.	Módulo formulario de reporte delictivo (C3)	32
4.2.4.	Módulo API REST (C4)	32
4.2.5.	Módulo Servidor (C5)	33
4.3.	Matriz de requisitos funcionales y componentes	33
5.	Gestión de riesgos	34
5.1.	Supuestos	34
5.2.	Dependencias	34
5.3.	Restricciones	34

5.4.	Riesgos	35
5.4.1.	Confiabilidad de los datos	35
5.4.2.	Tiempo de consulta al servidor	35
5.5.	Mitigaciones	35
5.5.1.	Confiabilidad de los datos	35
5.5.2.	Tiempo de consulta al servidor	35
6.	Implementación del Sistema	36
6.1.	Diagrama de Contexto	36
6.1.1.	Existen 3 entes externos con los que el sistema puede interactuar:	37
6.2.	Enumeración de Módulos	37
6.3.	Enumeración de los Requisitos Funcionales	38
6.4.	Matriz de Requisitos Funcionales y Módulos	38
6.5.	Módulo Geolocalización	39
6.5.1.	Definición del módulo	39
6.5.2.	Declaraciones Públicas	39
	Constantes Públicas	39
	Tipos de Datos Públicos	40
	Variables Públicas	40
6.5.3.	Funciones Públicas	40
6.5.4.	Estructuras de Datos	41
6.5.5.	Diseño Detallado de las Funciones	41
	Función getProviders	41
	Función isProviderEnabled	42
	Función getLastKnownLocation	42
6.6.	Módulo APP	43
6.6.1.	Definición del módulo	43
6.6.2.	Declaraciones Públicas	44
	Constantes Públicas	44
	Tipos de Datos Públicos	44
	Variables Públicas	45
6.6.3.	Funciones Públicas	45
6.6.4.	Estructuras de Datos	45
6.6.5.	Funciones Privadas	45
6.6.6.	Diseño Detallado de las Funciones	46
	Función getMap	46
	Función postCrime	46
	Función getCrimes	47
6.7.	Módulo API	48
6.7.1.	Definición del módulo	48
6.7.2.	Declaraciones Públicas	48

Constantes Públicas	48
Tipos de Datos Públicos	48
Variables Públicas	48
6.7.3. Funciones Públicas	48
6.7.4. Estructuras de Datos	49
6.7.5. Funciones Privadas	49
6.7.6. Diseño Detallado de las Funciones	49
Función load_crimes	49
Función new_crime	50
6.8. Módulo Server	51
6.8.1. Definición del módulo	51
6.8.2. Declaraciones Públicas	51
Constantes Públicas	51
Tipos de Datos Públicos	51
Variables Públicas	52
6.8.3. Funciones Públicas	52
6.8.4. Estructuras de Datos	52
6.8.5. Funciones Privadas	52
6.8.6. Diseño Detallado de las Funciones	53
Función getCrimes	53
Función saveCrime	53
6.9. Módulo GCM	54
6.9.1. Definición del módulo	54
6.9.2. Declaraciones Públicas	54
Constantes Públicas	54
Tipos de Datos Públicos	54
Variables Públicas	54
6.9.3. Funciones Públicas	55
6.9.4. Estructuras de Datos	55
6.9.5. Funciones Privadas	55
6.9.6. Diseño Detallado de las Funciones	56
Función sendGoogleCloudMessage	56
6.10. Aplicación Móvil VersusCrime	57
6.11. Panel de Gestión para entidades suscritas al sistema	60
6.11.1. Panel	61
6.11.2. Mis reportes	61
6.11.3. Agregar nuevo reporte	62
7. Verificación y Validación	63
7.1. Análisis y estudio de mercado sobre la intención de uso en aplicaciones móviles de mapas de seguridad	63

7.1.1.	Definición categoría de la aplicación móvil	63
7.1.2.	Análisis de intención de uso	64
7.1.3.	Análisis de mercado	65
	Necesidad de información y objetivo de la investigación	65
	Investigación exploratoria	66
	Atributos sin clasificar:	67
	Atributos clasificados, los que se van a utilizar en la encuesta:	69
7.2.	Resultados Generales	70
8.	Conclusiones	79
8.1.	Conclusiones Generales	79
8.2.	Valores del Sistema	80
8.3.	Trabajo futuro	81
	Bibliografía	83
A.	VersusCrime en los medios de comunicaciones	86
B.	Retrofit	88
B.0.1.	¿Qué es Retrofit?	88
B.0.2.	Primeros Pasos	89
C.	Android Studio	90
C.0.3.	Primeros Pasos: Migrando desde Eclipse a Android Studio	90
D.	Fragment	92
D.0.4.	¿Por qué utilizar Fragment?	92
D.0.5.	Primeros Pasos	93
E.	Material Design	96
E.0.6.	¿Qué es Material Design?	96
E.0.7.	¿En qué se basa Material Design?	96
E.0.8.	Elementos ordenados e imágenes claras	97
E.0.9.	Luz y sombras dan sensación de jerarquía	97
E.0.10.	El movimiento es la mejor forma de guiar al usuario	97
F.	Laravel	98
F.0.11.	¿Qué es Laravel?	98
F.0.12.	Ventajas al emplear Laravel en el desarrollo web	98
F.0.13.	¿Por qué utilizar Laravel?	99
G.	Resultados Modelo TAM	101

Índice de Tablas

2.1. Demanda potencial por segmento. Fuente: Elaboración propia	20
2.2. Contrastando la competencia. Fuente: Elaboración Propia.	25
3.1. Tabla de eventos externos. Fuente: Elaboración Propia	28
3.2. Tabla de respuestas de los eventos. Fuente: Elaboración Propia	28
4.1. Matriz de requisitos funcionales y componentes. Fuente: Elaboración Propia.	33
6.1. Módulos de la Arquitectura del Sistema. Fuente: Elaboración Propia. .	37
6.2. Matriz de requisitos funcionales y módulos. Fuente: Elaboración Propia.	38
6.3. Módulo geolocalización: Matriz de constantes públicas. Fuente: Elabo- ración Propia.	39
6.4. Módulo geolocalización: Matriz de tipos de datos públicos. Fuente: Ela- boración Propia.	40
6.5. Módulo geolocalización: Matriz de variables públicas. Fuente: Elabora- ción Propia.	40
6.6. Módulo geolocalización: Matriz de funciones públicas. Fuente: Elabora- ción Propia.	40
6.7. Matriz de la función getProviders. Fuente: Elaboración Propia.	41
6.8. Matriz de la función isProviderEnabled. Fuente: Elaboración Propia. . .	42
6.9. Matriz de la función getLastKnownLocation. Fuente: Elaboración Propia.	42

6.10. Módulo APP. Matriz de constantes públicas. Fuente: Elaboración Propia.	44
6.11. Módulo APP. Matriz de tipos de datos públicos. Fuente: Elaboración Propia.	44
6.12. Módulo APP. Matriz de variables públicas. Fuente: Elaboración Propia.	45
6.13. Módulo APP. Matriz de funciones públicas. Fuente: Elaboración Propia.	45
6.14. Matriz de la función getMap. Fuente: Elaboración Propia.	46
6.15. Matriz de la función postCrime. Fuente: Elaboración Propia.	46
6.16. Matriz de la función getCrimes. Fuente: Elaboración Propia.	47
6.17. Módulo API. Matriz de funciones públicas. Fuente: Elaboración Propia.	49
6.18. Módulo API. Matriz función load_crimes. Fuente: Elaboración Propia. .	49
6.19. Módulo API. Matriz función new_crime	50
6.20. Módulo Server. Matriz de las funciones públicas. Fuente: Elaboración Propia.	52
6.21. Módulo API. Matriz función getCrimes. Fuente: Elaboración Propia. . .	53
6.22. Módulo API. Matriz función saveCrime. Fuente: Elaboración Propia. .	53
6.23. Módulo GCM. Matriz de constantes públicas. Fuente: Elaboración Propia.	54
6.24. Módulo GCM. Matriz de tipos de datos públicos. Fuente: Elaboración Propia.	55
6.25. Módulo GCM. Matriz de funciones públicas. Fuente: Elaboración Propia.	55
6.26. Módulo GCM. Matriz de la función sendGoogleCloudMessage. Fuente: Elaboración Propia.	56
B.1. Duración de las peticiones REST. Fuente: Elaboración Propia.	88
G.1. Tabla con resultados pregunta número 1 Modelo TAM	102
G.2. Tabla con resultados pregunta número 2 Modelo TAM	103
G.3. Tabla con resultados pregunta número 3 Modelo TAM	103
G.4. Tabla con resultados pregunta número 4 Modelo TAM	104

G.5. Tabla con resultados pregunta número 5 Modelo TAM	105
G.6. Tabla con resultados pregunta número 6 Modelo TAM	105
G.7. Tabla con resultados pregunta número 7 Modelo TAM	106
G.8. Tabla con resultados pregunta número 8 Modelo TAM	106
G.9. Tabla con resultados pregunta número 9 Modelo TAM	107

Índice de Figuras

1.1. Victimization General	4
1.2. Fuente: ENUSC 2014, Subsecretaría de Prevención del Delito	5
1.3. Fuente: ENUSC 2014, Subsecretaría de Prevención del Delito	5
2.1. Arquitectura de Android, compuesta por cuatro capas. Fuente: Androi- deric [13]	12
2.2. Descargas de aplicaciones. Fuente: App Annie	14
2.3. Ingresos de aplicaciones por sistema operativo. Fuente: App Annie . . .	17
2.4. Crecimiento del uso de smartphone por categoría. Fuente: Furry Analytics	18
2.5. Usuarios y penetración de internet en Latinoamérica. Fuente: eMarketer	21
2.6. Usuarios y penetración de smartphones en Latinoamérica. Fuente: eMar- keter	22
2.7. Ingresos de las aplicaciones móviles en el mundo. Fuente: Statista . . .	23
4.1. Infraestructura del sistema. Fuente: Elaboración Propia	30
4.2. Diagrama de flujos de la aplicación. Fuente: Elaboración Propia.	31
6.1. Diagrama de Contexto Sistema VersusCrime. Fuente: Elaboración Propia.	36
6.2. Vista menú lateral con las principales funciones	57
6.3. Vista despliegue de reportes delictivos	57
6.4. Vista formulario de ingreso de un delito	58
6.5. Vista selección opción de reportar un delito	58

6.6. Vista selección de filtros por periodos de fechas	59
6.7. Vista formulario de ingreso de un delito	59
6.8. Sistema VersusCrime	60
6.9. Vista Principal: Panel	61
6.10. Vista Mis Reportes	61
6.11. Vista Agregar Reporte	62
7.1. Atributos de preferencias	70
7.2. Atributos seleccionados por los usuarios	70
7.3. Percepción de exactitud de la ubicación	71
7.4. Percepción de satisfacción	72
7.5. Percepción de calidad del servicio	73
7.6. Percepción de movilidad	74
7.7. Percepción de desempeño esperado	75
7.8. Percepción de actitud	76
7.9. Percepción estado de flujo	77
7.10. Percepción de otros aspectos	78
8.1. Registros de usuarios en los primeros 3 meses	80
8.2. Distribución de los delitos registrados por el sistema	81
A.1. Fuente: Lun.com 13-04-2015.	87
A.2. La RED.	87
D.1. Fragments dentro de un activity.	92
E.1. Vistas Material Design.	97
F.1. Porcentaje de preferencias	100

Glosario

IDC	International Data Corporation
INE	Instituto Nacional de Estadísticas
TAM	Technology Acceptance Model
UTAUT2	Unified Theory of Acceptance and Use of Technology 2
IDE	Integrated Development Environment
GCM	Google Cloud Messaging
GPS	Global Positioning System
SMS.	Short Message Service
iOS	Iphone Operating System
SQL	Structured Query Language
VPS	Virtual Private Server
REST	Representational State Transfer
API	Application Programming Interface
OOP	Object Oriented Programming
POJO	Plain Old Java Object
JSON	JavaScript Object Notation
MVC	Model-View-Controller

Capítulo 1

Introducción

A través del programa *Memorias Multidisciplinarias* [1], se analizó e implementó el desafío propuesto por la empresa SecuritySat [2] *Geolocalización de eventos delictivos a través de una aplicación móvil social y comunitaria*. Para llevar a cabo el desarrollo de este desafío se formó un equipo multidisciplinario, con especialidades de Diseño por José Alvarez, Comercial por Angello Núñez y Técnica por Manuel Ramírez.

1.1. Propósito

El propósito de esta memoria, es desarrollar un mapa delictivo creado a partir de la información geolocalizada que pueden entregar los usuarios que utilizan un smartphone. Toda persona que sea afectada u observe un delito, debe ser capaz de emitir una alerta de emergencia a sus redes de contacto rápidamente. Esta información es capturada y registrada para: informar a los usuarios las zonas más riesgosas y alertar a las policías de los lugares vulnerables a comisión de delitos, de manera de contar con indicadores certeros para este cometido.

1.2. Alcance

El alcance del proyecto es implementar una aplicación móvil, con un alto nivel tecnológico, que permita registrar los actos delictivos reportados por los usuarios. Esta

información permite crear un mapa, disponible en la misma aplicación móvil, identificando las zonas que registran un alto nivel de hechos delictivos, siendo ésta, una información válida para los organismos de seguridad. Además, el proyecto permite la interacción entre los usuarios que han reportado un delito, como también con otros usuarios que no lo han sufrido. También es importante rescatar dentro del alcance, la implementación de un panel de control, disponible para los organismos de seguridad, donde podrán interactuar en tiempo real con los datos registrados.

1.3. Contexto

Al comienzo de la memoria se disponía de una versión inicial básica de VersusCrime. Hoy, la aplicación se encuentra en una versión funcional avanzada en Android. Se han desarrollado módulos independientes, que permiten la escalabilidad en el desarrollo de nuevas funcionalidades. En ellas, ya se encuentran desarrolladas la autenticación y registro. Además los módulos de reportar evento con una posición geográfica y descripción, desplegar la información reportada por medio de un mapa, filtrar por categorías los eventos y la función de contactar al plan cuadrante de carabineros, esto se realiza de forma directa desde la aplicación móvil, que de acuerdo al punto geográfico del usuario permite determinar el plan cuadrante correspondiente a la zona.

Para el desarrollo del proyecto se utiliza un smartphone que actúa de intermediario, entre el servidor, que almacena los datos obtenidos, y el usuario, que facilita dicha información. Además, contempla una plataforma web que permite entregar información clasificada a las distintas entidades que necesitan de su uso para la toma de decisiones.

La aplicación móvil es para dispositivos smartphone, con red de datos y con sistema operativo Android 4.1 o superior. Para la implementación del proyecto, se utiliza el IDE Android Studio [Anexo C]. Además se ocupan las bibliotecas provistas por Google y

Square [3], tales como Maps V2, Cloud Messaging, Retrofit y Picasso.

1.3.1. Definición del Problema

En las empresas de seguridad, existe la necesidad de detectar zonas con mayor riesgos de ocurrencia de delitos, con el fin de destinar recursos de ventas en zonas de potenciales consumidores como también, un despliegue más eficiente de sus recursos de seguridad en sectores donde son más necesarios para ellos. Por parte de organizaciones está la necesidad de valorizar la adquisición de bienes o presencia geográfica.

Las personas en Chile, poseen una alta percepción de inseguridad, existiendo la necesidad de informar e informarse de hechos delictivos que suceden en su entorno.

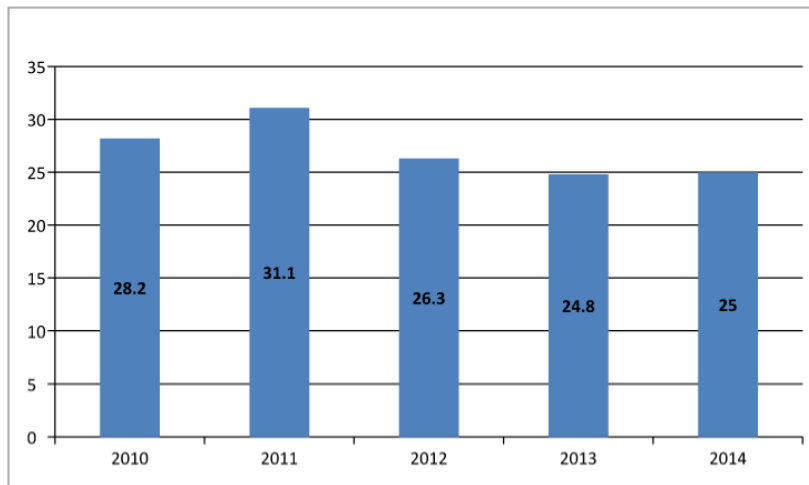
La seguridad es una de las industrias más rentables en Chile que mueve más de 1.000 millones de dólares al año y va en crecimiento con los índices de delitos que no han disminuido [5].

El nivel de temor, producto de la delincuencia en Chile el año pasado (2015) fue el más alto en 14 años, según un estudio de Paz Ciudadana [6]. Esto ha provocado que incluso los vecinos se hayan organizado y a la fecha hayan creado grupos de WhatsApp o Facebook para advertirse de posibles delitos.

La criminología sostiene que, los patrones de delincuencia se reiteran en el mismo lugar y horario a falta de control, por ejemplo por una zona mal iluminada o por falta de carabineros asignados.

Las encuestas en Chile dicen que el 37 % de las veces las personas no denuncian a la policía cuando se sufre un delito, por lo tanto, las estadísticas no reflejan los hechos.

Victimización general



Fuente: ENUSC 2014, Subsecretaría de Prevención del Delito

Figura 1.1 – Victimization General

- Un 25 % de las personas sufrieron algún tipo de delito en el año 2014.
- 3,6 % de los hogares sufrió Robo con fuerza.
- 53,3 % de los delitos de Robo con fuerza fue denunciado.
- 8,7 % de las personas se le ha cometido hurto.
- 21,5 % de los delitos de Hurto fue denunciado.

Principales motivos para no denunciar (Hurto)

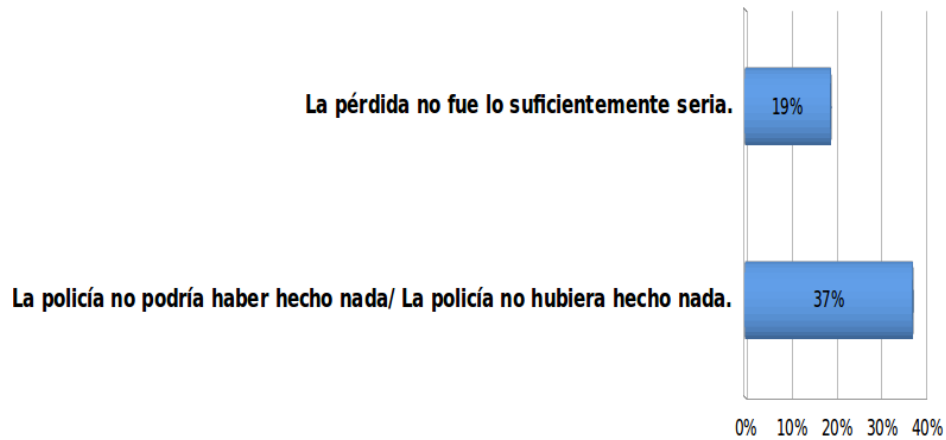


Figura 1.2 – Fuente: ENUSC 2014, Subsecretaría de Prevención del Delito

Exposición al delito 2014

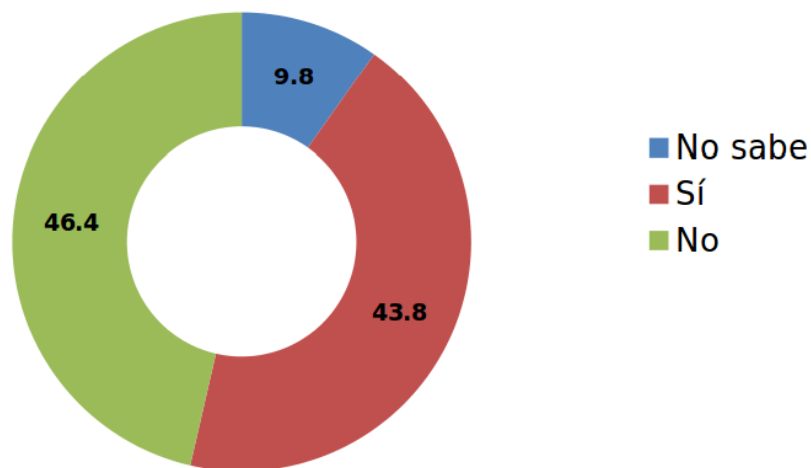


Figura 1.3 – Fuente: ENUSC 2014, Subsecretaría de Prevención del Delito

- 80 % de las personas perciben que la delincuencia en el país aumentó.
- 43,8 % creen que pueden ser víctimas de un delito.

1.4. Objetivos generales

- Desarrollar una aplicación móvil social y comunitaria, para la captura de diferentes delitos.
- Disminuir la percepción de inseguridad en la ciudadanía.
- Mejorar la toma de decisiones, a las empresas u organismos del Estado que deben velar por la seguridad social.

1.5. Objetivos específicos

- Potenciar la intención de uso de la aplicación móvil VersusCrime.
- Establecer mecanismos y métodos eficientes para el desarrollo de la aplicación móvil, con un enfoque hacia la seguridad ciudadana.
- Evaluar y seleccionar las herramientas adecuadas para el desarrollo de la aplicación, con el fin de agilizar los procesos internos.
- Evaluar y seleccionar las bibliotecas adecuadas para otorgar una mejor experiencia usuaria.
- Habilitar la plataforma para que las organizaciones que cuentan con registros propios, pueda alimentar el sistema como también nutrirse de ella.
- Proporcionar una plataforma web con indicadores y mapas de calor, para la toma de decisiones de las empresas de seguridad.

1.6. Estructura del documento

Esta memoria está organizada de la siguiente manera: El capítulo 2 corresponde al Estado del Arte. En él se realizará una introducción a Android, explicando a grandes rasgos sus inicios, arquitectura, tipos de dispositivos, entre otras cosas. Además se estudiarán las aplicaciones similares en el mercado tanto nacional como internacional. En el capítulo 3 se presenta un listado clasificado con los distintos requerimientos funcionales y no funcionales. Además se presentará un listado de los requerimientos técnicos en hardware y software. En el capítulo 4 se analizará la arquitectura del sistema y se describirá los distintos módulos que se componen. En el capítulo 5 se detallará los posibles riesgos que podría afectar el desarrollo e implementación. Además de analizar qué estrategias y mitigación podrían prevenir dichos riesgos. En el capítulo 6 se detallará la implementación del sistema, especificando cada módulo que compone la aplicación móvil. En el capítulo 7 se realizará un estudio por medio de una encuesta para validar el éxito de la aplicación VersusCrime. Finalmente, en el capítulo 8 se presentan las conclusiones obtenidas a partir de los análisis e implementaciones previas.

Capítulo 2

Estado del Arte

En este capítulo se dará a conocer una breve descripción del sistema operativo Android. Se comenzará con una introducción, hablando de sus inicios, su arquitectura y la evolución que ha tenido con el tiempo. Además se hablará sobre las principales competencias de esta aplicación móvil, en el mercado nacional e internacional.

2.1. Introducción a Android

Android es un sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles táctiles, tales como smartphones y tablets.

2.1.1. Inicios de Android

El comienzo del que ahora es el sistema operativo móvil con mayor penetración en el mundo no fue fácil, pero ha conseguido cautivar a millones de usuarios de telefonía móvil.

Android, Inc. fue fundada en Palo Alto, California en Octubre del 2003 por Andy Rubin, Rich Miner, Nick Sears y Chris White [7]. Su objetivo era desarrollar dispositivos móviles más inteligentes, con un mayor foco en la localización del dueño y en su personalización.

Google compró a Android Inc. el 17 de Agosto del 2005 [8]. Poco se sabía sobre

esta compañía para ese entonces, ya que estuvo funcionando de forma secreta, sin dar a conocer detalles sobre lo que desarrollaban. Muchos asumían que Google estaba planeando entrar al mercado de dispositivos móviles. De ahí en adelante los esfuerzos de Google se enfocaron en conversaciones con fabricantes, con la promesa de proveer un sistema flexible y escalable.

Sin embargo, la aparición del iPhone el 9 de Enero del 2007 [10] tuvo un efecto disruptivo en el desarrollo de Android. Hasta el momento se contaba con un prototipo, el cual se acercaba más a lo que podría ser un teléfono BlackBerry, sin pantalla táctil y con un teclado físico. Por lo que se comenzó inmediatamente un trabajo de reingeniería del sistema operativo y del prototipo para que fuese capaz de competir con el iPhone.

El 6 de Noviembre del 2007 [11] fue fundada la Open Handset Alliance, una alianza comercial liderada por Google con compañías tecnológicas como HTC, Sony y Samsung, operadores de carriers como Nextel y T-Mobile y fabricantes de chips, con el objetivo de desarrollar estándares abiertos para dispositivos móviles. El primer smartphone disponible que funcionaba sobre Android fue el HTC Dream, lanzado el 22 de Octubre del 2008.

2.1.2. Arquitectura

La arquitectura del sistema Android [12], también llamado stack, se puede apreciar en la figura 2.1 y está compuesta por cuatro capas:

- **Kernel Linux:** La capa más profunda es su núcleo Linux, un sistema operativo abierto, portable y seguro. Para cada pieza de hardware, como la cámara o el bluetooth, existe un driver dentro del kernel, que permite a la capa superior hacer uso de ella, por lo que funciona como una capa de abstracción. El kernel además se encarga de la gestión de los diversos recursos del dispositivo, como la

energía o la memoria, elementos de comunicación, procesos.

- **Bibliotecas:** La segunda capa en el stack contiene bibliotecas nativas, las cuales están escritas en C o C++, y son compiladas para la arquitectura específica del dispositivo. En la mayoría de los casos el fabricante es quien se encarga de instalarla en su dispositivo. Las bibliotecas incluidas en esta capa son: motor gráfico OpenGL, sistema de gestión de base de datos SQLite, cifrado de comunicaciones SSL, motor de manejo de tipos de letra FreeType, entre otras.
- **Marco o Framework de Aplicaciones:** La tercera capa está compuesta por todas las clases y servicios que se utilizan al momento de programar aplicaciones. Los componentes que posee son:
 - **Administrador de Actividades (Activity Manager):** Gestiona la pila de *Activities* de la aplicación, como también su ciclo de vida. En el desarrollo en Android, se llama *Activity* a cada una de las pantallas con las que el usuario interactúa. Por lo que se puede asumir que una aplicación en la mayoría de los casos va a tener varias *Activities*. Estas son uno de los componentes más importantes dentro de una aplicación.
 - **Administrador de ventanas (Windows Manager):** Organiza lo que se mostrará en pantalla. Crea las superficies en la pantalla, que posteriormente estarán ocupadas por las *Activities*.
 - **Proveedor de contenidos (Content Provider):** Encapsula los datos que pueden ser compartidos por las aplicaciones, facilitando la comunicación entre éstas.
 - **Vistas (Views):** Son los elementos que permiten construir las interfaces de usuario, como listas, botones, textos, hasta otros elementos más avanzados como visores de mapas.

- **Administrador de notificaciones (Notification Manager):** Provee los servicios que notifican al usuario, mostrando alertas en la barra de estado. También permite activar el vibrado, reproducir alertas de sonido y utilizar las luces del dispositivo.
 - **Administrador de paquetes (Package Manager):** Gestiona la instalación de nuevos paquetes y además permite obtener información sobre los que ya están instalados.
 - **Administrador de telefonía (Telephony Manager):** Permite realizar llamadas, como también el envío y recepción de SMS.
 - **Administrador de recursos (Resource Manager):** A través de este administrador se podrá acceder a los elementos que no forman parte del código, como imágenes, sonidos, layouts.
 - **Administrador de ubicaciones (Location Manager):** Permite obtener la posición geográfica actual del dispositivo a través de GPS o redes.
 - **Administrador de sensores (Sensor Manager):** Permite la manipulación de distintos sensores del dispositivo, como el acelerómetro, giroscopio, brújula, sensor de proximidad.
 - **Cámara:** Permite el uso de la cámara del dispositivo para la obtención de fotografías o vídeos.
 - **Multimedia:** Permite la visualización y reproducción de imágenes, vídeos y audio.
- **Aplicaciones:** En esta capa se encuentran todas las aplicaciones del dispositivo, tanto las preinstaladas, como aquellas instaladas por el usuario. También está la aplicación principal del sistema, el Inicio o launcher, desde donde se inician todas las aplicaciones.

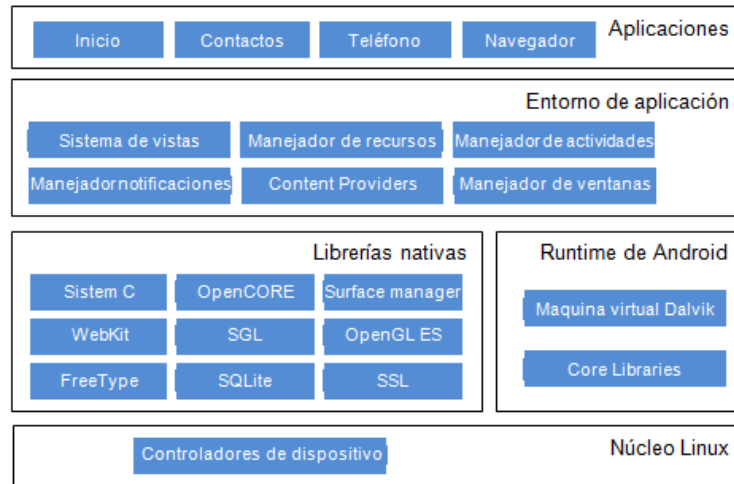


Figura 2.1 – Arquitectura de Android, compuesta por cuatro capas. Fuente: Andri-deric [13]

2.2. Situación Actual

2.2.1. Situación actual del mercado de aplicaciones móviles

Análisis de sistemas operativos

Se ha estudiado la distribución de sistemas operativos en los smartphones con el fin de saber qué es lo que están necesitando los usuarios y en dónde hay que dar énfasis en la aplicación. De acuerdo a la International Data Corporation (IDC) [14], los productores de smartphones vendieron 334,4 millones de unidades en el primer trimestre del 2015 a nivel mundial. Bajo este mismo contexto, el sistema operativo Android sigue manteniendo su férreo liderazgo, con un 78 % de participación de mercado, aventajando por mucho a su principal competidor que es iOS de Apple alcanzando un 18,3 %, una leve mejora respecto al primer trimestre del año anterior, donde tenía un 15,2 %. El resto de la participación de mercado, equivalente al 3,7 % se dividen entre los otros sistemas operativos que son conocidos pero no populares, entre el que destaca Windows, obteniendo un 2,7 %. Esto significó que aumentara en 0,2 % respecto al año anterior.

Chile no es la excepción a estos datos. A pesar que no representa una masa que pueda contribuir en dichos porcentajes, la preferencia por Android e iOS sigue superando por mucho a los demás. Según comScore en su estudio Futuro Digital Chile 2015 [15], la participación de tráfico de smartphones y tablets está distribuida de la siguiente manera: el sistema operativo Android alcanza un 80,5 %, consagrándose nuevamente como el líder. Le sigue iOS con un 13,4 % y Windows tiene una participación mayor que su participación mundial promedio, obteniendo un 4,2 %. Este último tuvo un aumento de 1,7 % respecto al año anterior.

2.2.2. Migración móvil en Chile

La población está cambiando sus hábitos tecnológicos. Al desarrollarse teléfonos móviles que pueden realizar las mismas funciones que un computador, las personas están empezando a utilizar más los smartphones por la comodidad que brindan. El análisis que hace comScore es según las páginas vistas por dispositivo, donde se puede valorar el notorio aumento en la utilización de smartphones para este fin y la disminución del uso de computadores: en mayo de 2015, el 81,2 % de las personas utilizó un PC para este fin, los celulares representaron un 16 %, los Tablet un 2,5 % y otros dispositivos solo tienen un 0,4 %. Las variaciones más notorias las obtuvieron los PC y los celulares, con un -8,9 % y un 7,6 % respectivamente.

2.2.3. Análisis del tipo de conexión de internet móvil

La información obtenida es brindada por las estadísticas de la Subtel hasta el mes de marzo del 2015 [9], respecto al tipo de conexión predominante en los celulares que poseen internet móvil. De acuerdo a los datos, se tiene que la conexión 4G ha tenido un salto explosivo en nuestro país, lo que permite hacer un uso más eficiente del internet en dispositivos móviles. Las conexiones de 3G le siguen, alcanzando 7.533.097 dispositivos y la 2G, que se está quedando atrás con 1.217.618, que representa una

disminución de un 41,2 %. Respecto a la metodología utilizada para obtener estos datos se explica lo siguiente:

1. Las conexiones móviles corresponden al número de usuarios distintos que se han conectado a internet en forma dedicada en las redes móviles celulares dentro del período que se informa. Se contabiliza para cada cliente y/o usuario sólo una conexión, independiente del número de veces que se conecte a Internet dentro del período a informar (mes).
2. Para efectos de clasificación por tecnología, se considera la tecnología más avanzada dentro del mes, por ejemplo, si un usuario se conecta por 2G dentro del mes y además se conecta como 3G, sólo se contabiliza una conexión por 3G.

2.2.4. Descargas de aplicaciones

Por sistema operativo en el mundo

Considerando solo los sistemas operativos que tienen mayor peso, en el 2° trimestre del año 2015, las descargas de Google Play superaron en un 85 % a las de iOS [16]. Esta gran diferencia aparenta ser debido a los nuevos mercados emergentes que están tomando fuerza como Brasil, India y países del sudeste asiático y lo baratos que son los dispositivos de Android capaces de utilizar aplicaciones.

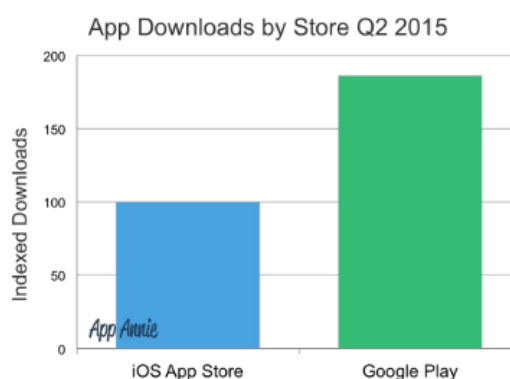


Figura 2.2 – Descargas de aplicaciones. Fuente: App Annie

Ahora, es importante hacer notar que no en todos los países el sistema operativo que predomina es Android. Se puede ver una tendencia que los países desarrollados prefieren iOS y los que están en vías de desarrollo, prefieren Android. Siguiendo esta línea, la empresa de inteligencia de aplicaciones móviles App Annie [16] brinda rankings, como por ejemplo:

- Los países que descargan más aplicaciones en iOS, encontrándose aquí China, EEUU y Japón.
- Los países que descargan más aplicaciones en Android, encontrándose aquí EEUU, Brasil e India.
- Categorías de aplicaciones móviles más descargadas en iOS, teniendo las categorías de Juegos, Fotos y videos, y en 5° lugar estilo de vida.
- Categorías de aplicaciones móviles más descargadas en Android, teniendo Juegos, Herramientas y Comunicación.

Por sistema operativo en Chile

Se analiza cuáles son las aplicaciones más descargadas en el país, en los sistemas operativos más fuertes. La aplicación móvil de VersusCrime en Android está clasificada en la categoría *Social*, por lo que se mostrarán el ranking de esta categoría y, en el caso de iOS, estaría en la categoría de *Estilo de vida*. Existe la posibilidad de utilizar otro filtro: si en la aplicación se paga por adquirir funcionalidades extras o no, ya que VersusCrime no considera el pago dentro de la aplicación. Para Android, las aplicaciones gratis que lideran las descargas son Facebook, Instagram, Facebook Lite, Twitter y Snapchat, casi todas inmersas en el ámbito de redes sociales, donde solo la última aplicación, Snapchat, se distingue del resto al ser de Mensajería. Cabe señalar que predomina en la conducta de los usuarios, la acción de compartir a través

del celular utilizando estas aplicaciones. Las aplicaciones en las que hay que pagar para descargarlas, está Badoo Premium, otra red social para conocer gente; Fenix para Twitter, una versión mejorada de Twitter; y en 3° lugar se encuentra TagsForLikes Pro, una aplicación que complementa a los usuarios de Instagram. Se aprecia que nuevamente se repite el patrón de que predominan las aplicaciones relacionadas con redes sociales.

Analizando las aplicaciones descargadas en iOS, el panorama es completamente distinto. Se nota que el comportamiento del usuario está enfocado hacia otra área, no a redes sociales. En cuanto a las aplicaciones gratuitas, 4 de las 5 aplicaciones que lideran el ranking de las más descargadas son de compras: Ali Express Shopping App, Yapo.cl, Wish Shopping Made Fun y MercadoLibre. Y la aplicación que queda afuera de esta categoría de compras es Tinder, que es utilizada para conocer personas. En las aplicaciones que se pagan, se tiene que la tendencia va hacia aplicaciones que están en la categoría de *Estilo de vida*, ya sean para animar el smartphone o para cuidar a su bebe. Las más descargadas son: Black Elements Use our cool wallpapers, Pimp Your Screen, Urban Farming Assistant, Baby Monitor 3G y Baby Monitor & Alarm.

2.2.5. Ingresos de aplicaciones por sistema operativo

Nuevamente se consideran solos los sistemas operativos que tienen mayor volumen. Los ingresos de iOS aventajan en un 70 % a los de Google Play en el mismo periodo de tiempo. Se cree que la razón de esta ventaja es que en China está predominando los dispositivos con iOS.

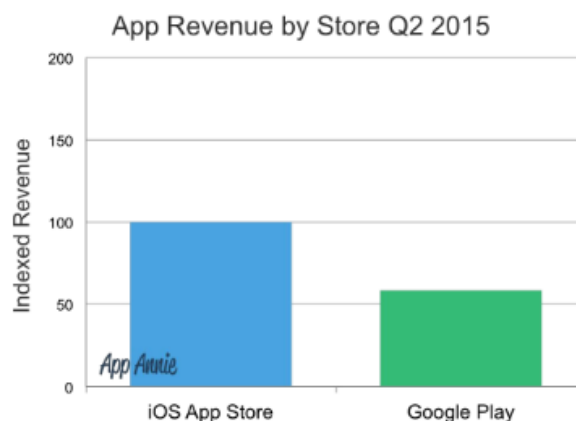


Figura 2.3 – Ingresos de aplicaciones por sistema operativo. Fuente: App Annie

De acuerdo a las estimaciones hechas por App Annie, en iOS los países que gastan más en aplicaciones son EEUU, Japón, China, Reino Unido y Australia, que corresponden a economías consolidadas. Para el caso de Android, donde los ingresos no son tan significativos si se comparan con los de iOS, los principales países son Japón, EEUU, Corea del Sur, Taiwan, y en el último puesto, Alemania.

En lo que respecta a las categorías de las aplicaciones de smartphones, las que predominaron por los altos ingresos obtenidos en el 2° trimestre de 2015 en iOS son Juegos, Redes sociales, Música, Entretenimiento y Educación. En Android la tendencia es parecida y las principales categorías son Juegos, Comunicación, Social, Herramientas y Entretenimiento.

2.2.6. Crecimiento del uso de smartphone por categoría

El año 2014 presenció un cambio en el comportamiento de usuarios de aplicaciones móviles. En este año, hubo un fuerte aumento en las aplicaciones relacionadas con el comercio, siendo que hasta el año pasado había dominado lo que era vinculado al entretenimiento. Esto no quiere decir que lo último haya dejado de crecer, sino que su crecimiento no fue tan radical como el de comercio. De acuerdo a Flurry

Analytics [17], las 3 categorías que obtuvieron el mayor crecimiento fueron: estilo de vida y compras, que alcanzó un 174 % a nivel global y un 220 % solamente en android; utilidad y productividad, con un 121 %; y como última categoría, mensajería, cerrando las categorías que tienen un crecimiento de 3 dígitos, con un 103 %. Cabe mencionar que las otras clases también alcanzaron una gran subida, las que están entre un 30 y un 90 % y, junto con esto, que el crecimiento promedio fue de 76 %.

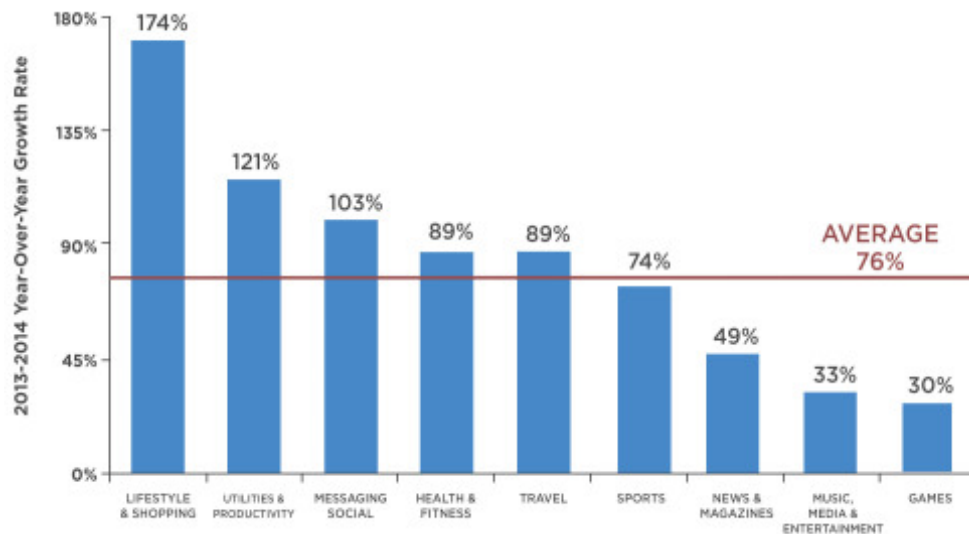


Figura 2.4 – Crecimiento del uso de smartphone por categoría. Fuente: Furry Analytics

2.2.7. Panorama actual de los dispositivos móviles en Chile

Se tiene información dada por el estudio Quinta encuesta sobre acceso, usos, usuarios y disposición de pago por internet en zonas urbanas y rurales de Chile, realizado por el centro de análisis Intelis de la Universidad de Chile para la Subtel en el año 2015 [19]. Los datos relevantes del estudio son los siguientes:

- El 67 % de la población es usuaria de internet.
- El 62 % de los hogares poseen internet.

- 28,3 % de los hogares que declaró poseer internet en su hogar; solo posee internet móvil, donde:
 - 13,8 % corresponde a smartphones.
 - 11,2 % corresponde a banda ancha móvil.
 - 3,3 % posee ambas opciones mencionadas anteriormente.
- 20,4 % de los hogares que declaró poseer internet en su hogar posee tanto internet móvil como internet fijo.

Al cruzar estos datos, se puede concluir que el 32,63 % de las personas en Chile dispone de internet móvil.

2.2.8. Demanda potencial

Se debe tener en cuenta el número de clientes que podrían utilizar la aplicación, lo que se le llama demanda potencial. Para calcularla, se debe tener como primer dato, la población chilena, del Instituto Nacional de Estadística [18] según los segmentos objetivos definidos: para el segmento de 13 a 35 años la población chilena es 6.388.811; para el segmento de 20 a 26 años, 2.071.229 personas; para el segmento de 30 a 40 años, se tienen 2.851.616 personas; y para el último segmento, corresponden a 4.706.061. Luego, del apartado anterior, Panorama actual de los dispositivos móviles en Chile, se obtuvo que el 32,63 % de las personas tiene internet móvil. Por lo que cruzando este número junto al número de personas por segmento objetivo se tiene que la demanda potencial es, por segmento objetivo: 2.084.669 personas para el primer segmento, 675.842 personas para el segundo, 930.482 para el tercero y 1.535.588 para el cuarto segmento.

Demanda potencial segmentos objetivos de edad	Habitantes 2016
13-35	2.084.669
20-26	675.842
30-40	930.482
41-60	1.535.588

Tabla 2.1 – Demanda potencial por segmento. Fuente: Elaboración propia

2.2.9. Proyecciones

A continuación se presentan proyecciones realizadas por eMarketer [20], una empresa consultora de marketing que provee de explicaciones y tendencias relacionadas con marketing digital, medios masivos y comercio y, Statista [21], un portal de estadísticas a nivel mundial.

Usuarios y penetración de internet en Latinoamérica

La consultora calculó que en el final del año 2015, Latinoamérica tenía 309 millones de usuarios de internet y para finales del siguiente año tendría casi 332 millones, lo que significa un crecimiento de un 7,4 %. Debido que Chile no presenta una gran cantidad de población a nivel latinoamericano, no está entre los primeros puestos de la cantidad de usuarios de internet, pero se observa lo que respecta a un número más representativo, es la penetración, ya que es la cantidad de usuarios de internet sobre el total de población, lo que demuestra finalmente que Chile es el país más conectado de la región.

Internet Users and Penetration in Latin America, by Country, 2013-2018						
	2013	2014	2015	2016	2017	2018
Internet users (millions)						
Brazil	99.2	107.7	113.7	119.8	123.3	125.9
Mexico	53.1	59.4	65.1	70.7	75.7	80.4
Argentina	25.0	27.1	29.0	29.8	30.5	31.1
Colombia	24.2	26.5	28.6	29.4	30.5	31.3
Peru	13.0	14.5	16.0	17.6	19.0	20.1
Chile	10.7	11.6	12.3	12.7	12.9	13.2
Other	56.7	62.7	67.0	70.7	73.7	76.2
Latin America	282.0	309.5	331.7	350.6	365.5	378.3
Internet user penetration (% of population in each group)						
Chile	62.4%	66.7%	70.5%	71.7%	72.7%	73.5%
Argentina	58.7%	63.0%	66.8%	68.0%	69.0%	69.8%
Colombia	53.0%	57.3%	61.1%	62.3%	64.0%	65.0%
Brazil	49.3%	53.1%	55.7%	58.2%	59.5%	60.3%
Mexico	44.9%	49.7%	53.8%	57.8%	61.2%	64.5%
Peru	43.5%	48.1%	52.6%	57.1%	61.1%	64.3%
Other	38.9%	42.4%	44.8%	46.7%	48.1%	49.2%
Latin America	46.9%	51.0%	54.1%	56.7%	58.5%	60.0%
<i>Note: individuals of any age who use the internet from any location via any device at least once per month</i>						
<i>Source: eMarketer, Nov 2014</i>						
181961 www.eMarketer.com						

Figura 2.5 – Usuarios y penetración de internet en Latinoamérica. Fuente: eMarketer

Usuarios y penetración de smartphones en Latinoamérica

En lo que respecta a smartphones, eMarketer estimó que 194 millones de latinoamericanos acceden a internet a través de dispositivos móviles y de estos, 126 millones lo hacen por medio de smartphones, a finales del año 2014. Se repite aquí el hecho que Chile es el líder en penetración de smartphones proyectándole un 55,5 % al término del 2015 y un 66,3 % para inicios del año 2019.

Smartphone Users and Penetration in Latin America, by Country, 2013-2018						
	2013	2014	2015	2016	2017	2018
Smartphone users (millions)						
Brazil	27.1	38.8	48.6	58.5	66.6	71.9
Mexico	22.9	28.7	34.2	39.4	44.7	49.9
Colombia	11.7	14.4	16.3	18.2	19.7	20.9
Argentina	8.8	10.8	12.6	14.1	15.6	17.0
Chile	5.2	6.3	7.1	7.9	8.4	8.9
Peru	4.4	5.8	7.1	8.3	9.4	10.5
Other	16.6	21.3	26.7	31.7	36.4	40.8
Latin America	96.6	126.1	152.6	178.0	200.9	219.9
Smartphone user penetration (% of mobile phone users)						
Chile	41.9%	49.7%	55.5%	60.3%	63.7%	66.3%
Colombia	37.9%	45.3%	50.4%	55.1%	58.4%	60.9%
Mexico	34.3%	41.2%	47.2%	52.1%	57.5%	62.6%
Argentina	30.0%	36.0%	41.0%	45.0%	49.0%	52.6%
Peru	25.9%	31.8%	37.1%	42.3%	46.8%	51.0%
Brazil	20.8%	28.3%	34.4%	40.4%	45.0%	47.6%
Other	19.5%	24.3%	29.9%	34.7%	38.9%	42.8%
Latin America	25.5%	31.9%	37.5%	42.7%	47.1%	50.6%
<i>Note: individuals of any age who own at least one smartphone and use the smartphone(s) at least once per month; numbers may not add up to total due to rounding</i>						
<i>Source: eMarketer, Dec 2014</i>						
182940 www.eMarketer.com						

Figura 2.6 – Usuarios y penetración de smartphones en Latinoamérica. Fuente: eMarketer

Ingresos de las aplicaciones móviles en el mundo

Los ingresos de aplicaciones móviles se han estimado en US 76,5 billones para el año 2017. Por lo que debido a las cifras que se están alcanzando en ingresos, indica que se debe mover de la mobile web hacia las aplicaciones móviles. Esto se explica, según comScore, ya que el 32% de los clientes objetivos acceden vía mobile web y el 68% acceden por medio de las aplicaciones.

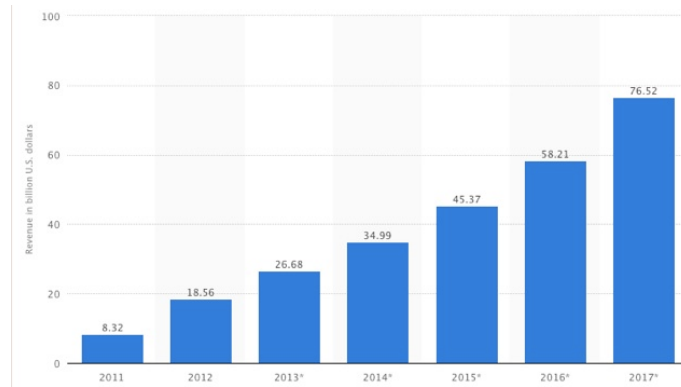


Figura 2.7 – Ingresos de las aplicaciones móviles en el mundo. Fuente: Statista

2.3. Mercado objetivo

Es necesario definir cuáles serán los clientes a los que está enfocada la aplicación. Por esta razón, es que se realiza una revisión detallada de:

2.3.1. Empresas

- Empresas que su rubro sea la seguridad de hogares, negocios y empresas.
- Empresas que necesitan saber hechos delictivos habitacionales.
- Empresas que deban tomar decisiones en las que el acontecer delictivo sea un elemento determinante.
- Instituciones del Estado
- Carabineros y PDI
- Seguridad Nacional
- Municipios
- Seguridad Ciudadana
- MINVU

2.3.2. Personas

Características generales

- Disponen de internet móvil.
- Necesidad de protección de personas y bienes.

Tipos de consumidores por adopción de producto

- **Innovadores:** jóvenes preocupados de su entorno. Personas que tienen entre 20 a 26 años. Poseen un nivel de uso medio-avanzado de smartphones. Interesados por las tecnologías vanguardistas.

Canal de comunicación: redes sociales, páginas web vanguardistas en tecnología.

- **Adaptadores tempranos:** adultos independientes. Personas que tienen entre 13 a 35 años. Están preocupados de su propia seguridad. Dominio medio de smartphones.

Canal de comunicación: redes sociales.

- **Mayoría temprana:** gente joven que tiene familia. Están en un rango de edad entre 30 a 40 años. Se les ve más preocupados de la seguridad de su familia. Su smartphone está enfocado para su trabajo. Nivel medio de uso de smartphone.

Canal de comunicación: redes sociales, televisión, radio.

- **Mayoría tardía:** gente adulta que el principal uso de su smartphone es para llamar. Su edad varía entre 40 a 60 años. Poseen un nivel de uso básico de smartphone. Se les debe instruir en cómo utilizar la aplicación.

Canal de comunicación: redes sociales, televisión, radio, presencial.

2.4. Análisis del mercado

En la siguiente sección, se analiza la competencia del producto en base a las características que posee la aplicación VersusCrime, con el fin de demostrar el valor agregado que entrega ésta. Las características que se analizan son a través de una tabla comparativa:

2.4.1. Competencia

Características	CityCop	LookApp	Muevete Seguro	CommiSur	VersusCrime
Número de descargas	10.000 - 50.000	1.000 - 5.000	10.000 - 50.000	500 - 1.000	5.000 - 10.000
Disponibilidad Android	X	X	X	X	X
Disponibilidad iOS	X		X	X	
Número de plan cuadrante			X		X
Reporte clasificado con descripción de evento	X	X		X	X
Mapa delictivo	X	X			X
Grupos de usuarios o Comunidades				X	
Notificaciones configuradas por ubicación	X				

Tabla 2.2 – Contrastando la competencia. Fuente: Elaboración Propia.

Capítulo 3

Análisis de Requerimientos

En este capítulo se detallarán los requerimientos funcionales y no funcionales del sistema. Además los requerimientos de ambiente que se necesitará tanto Hardware como Software para llevar a cabo el desarrollo de la aplicación móvil VersusCrime.

3.1. Requisitos del Sistema

A continuación se describen requisitos funcionales del sistema, interfaces externas, condiciones de excepción y las clases de pruebas que se harán para que los requisitos sean válidos.

3.1.1. Requisitos funcionales

- RF1 La aplicación debe tener un sistema de registro.
- RF2 La aplicación debe tener un sistema de login.
- RF3 La aplicación debe ser capaz de recibir reportes delictivos del servidor, que se encuentren dentro de un determinado rango.
- RF4 La aplicación debe ser capaz de enviar reportes delictivos al servidor.
- RF5 La aplicación debe ser capaz de entregar información telefónica del Plan Cuadrante más cercano a su geolocalización.

3.1.2. Requisitos de Interfaces

- RF1 La aplicación debe ser capaz de obtener la geolocalización del usuario, a través del GPS, la red móvil o WIFI del dispositivo.
- RF2 La aplicación debe ser capaz de enviar la ubicación al servidor.
- RF3 El servidor debe ser capaz de procesar dicha información y consumir desde la base de datos los reportes que se encuentren dentro de un determinado rango.
- RF4 La aplicación debe ser capaz de consumir la data entregada por el servidor. Posteriormente desplegar los reportes en un mapa.
- RF5 La aplicación debe ser capaz de notificar al usuario, con los reportes delictivos de las zonas previamente establecidas.

El cuadro 3.1 muestra la lista de eventos externos a los que el sistema responde. La primera columna es el Nombre del evento; la segunda es la Descripción del mismo. La tercera columna es el Iniciador, componente externo al sistema que inicia el evento. Cuarta columna, son los Parámetros asociados al evento. Finalmente la quinta columna es la Respuesta, cuya descripción está en el cuadro 3.2.

Evento	Descripción	Iniciador	Parámetros	Respuesta
Evento Externo 1	La aplicación detecta la ubicación del usuario	El instante en que el usuario abre la aplicación	Al abrir la aplicación	La aplicación almacena la localización en una variable local
Evento Externo 2	La aplicación envía la localización al servidor	El instante en que es capturada y almacenada la ubicación	Latitud y Longitud	El servidor almacena dicha información
Evento Externo 3	El servidor procesa la información y consulta a la base de datos	Al recibir la data a través del protocolo HTTP	Latitud longitud	Obtiene una lista con todos los reportes correspondientes.
Evento Externo 4	Despliegue de la data en la aplicación	El instante en que los datos son recibidos por la aplicación	Lista de reportes deictivos	El sistema despliega dicha información en un mapa
Evento Externo 5	Notificación por un evento en algún sector previamente definido	El instante en que es almacenado un reporte que corresponda al rango de dicha ubicación establecida	Aviso del reporte	La aplicación genera una alerta

Tabla 3.1 – Tabla de eventos externos. Fuente: Elaboración Propia

El cuadro 3.2 muestra las respuestas del sistema frente a eventos externos.

Respuesta	Descripción	Parámetros
Respuesta al evento 1	Comienza el envío de la ubicación al servidor	Localización obtenida a través del dispositivo móvil
Respuesta al evento 2	Comienza el procesamiento de la data	Latitud y Longitud
Respuesta al evento 3	Consume la data correspondiente desde la base de datos	Latitud y Longitud
Respuesta al evento 4	Despliegue de la información consumida por el servidor	Lista de reportes
Respuesta al evento 5	Se genera una alerta	Notificación enviada por GCM desde el servidor

Tabla 3.2 – Tabla de respuestas de los eventos. Fuente: Elaboración Propia

3.2. Requisitos de Ambiente

3.2.1. Descripción del Hardware de desarrollo

Para el desarrollo de la aplicación, se requiere un computador que soporte Android Studio. Idealmente un procesador Intel Core i5, con una RAM mínima de 8GB.

Para la ejecución de la aplicación, se requiere de un smartphone con características mínimas de 512MB de RAM y 1GB de almacenamiento.

3.2.2. Descripción del Software de desarrollo

El entorno de desarrollo integrado (IDE) utilizado es Android Studio. Se utiliza el lenguaje de programación JAVA para el desarrollo de las funcionalidades. Para el panel de indicadores se utilizó el FrameWork Laravel en PHP 5.6. El sistema operativo en que se trabaja es Ubuntu 14.04.

Capítulo 4

Análisis y Diseño

4.1. Arquitectura del Sistema

En la figura 4.1 muestra el principal flujo de información a través de todo el sistema.

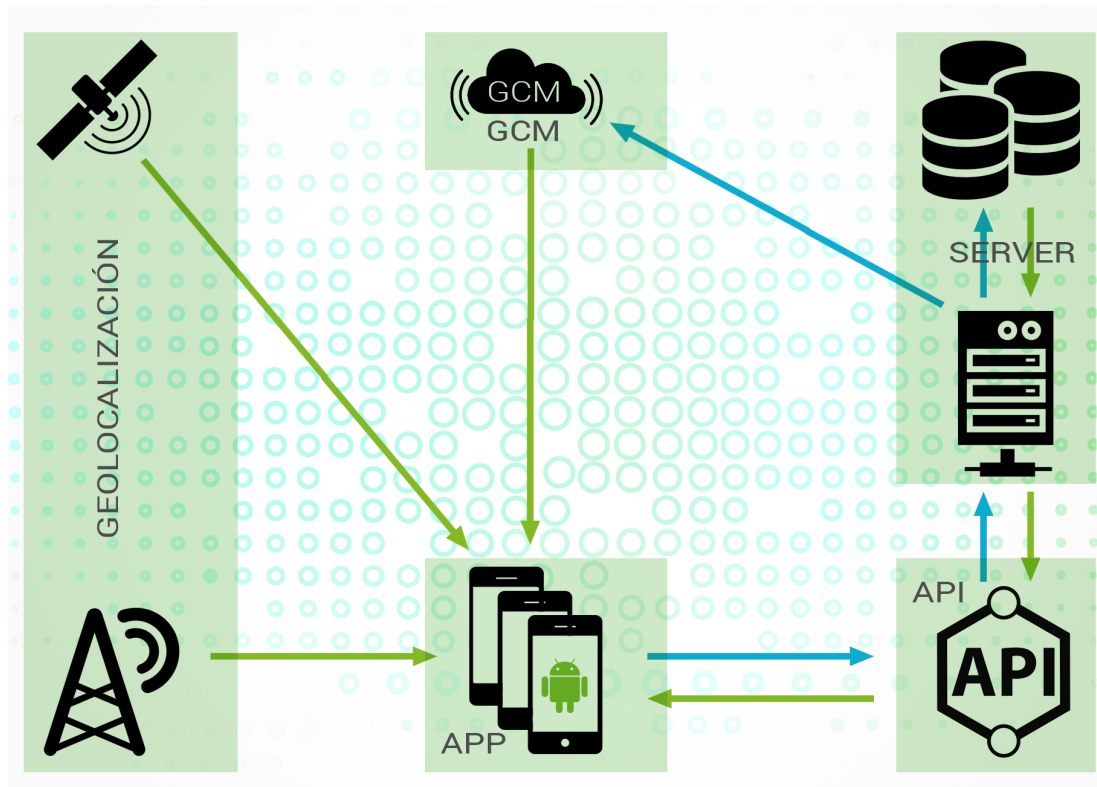


Figura 4.1 – Infraestructura del sistema. Fuente: Elaboración Propia

4.1.1. Diagrama de Flujos

En la figura 4.2 muestra el principal flujo de información al reportar y recibir un reporte delictivo.

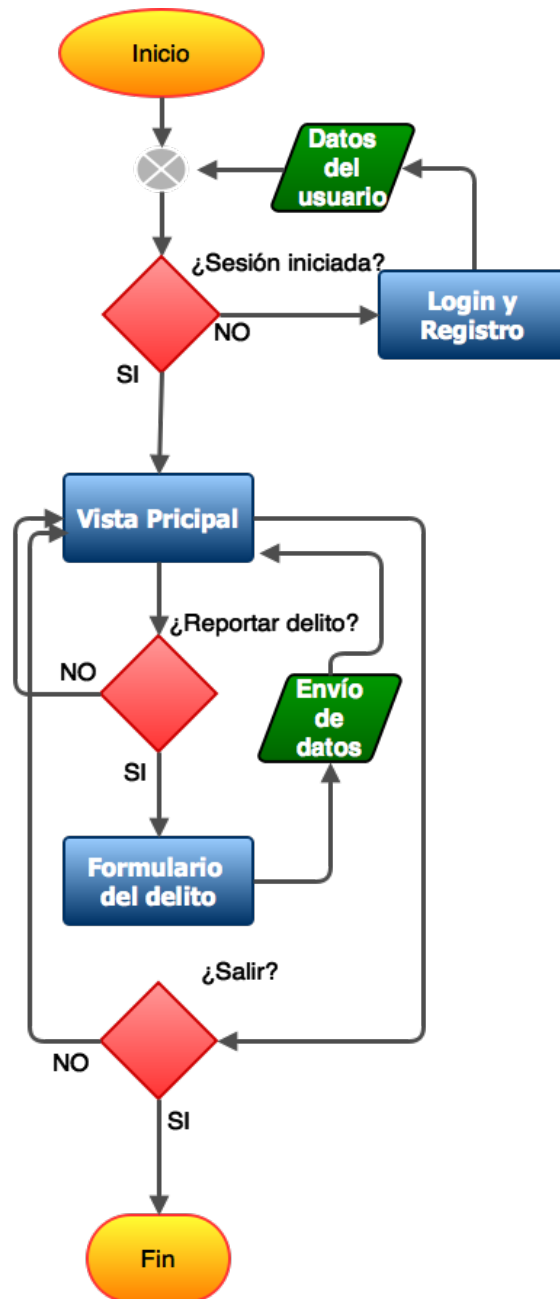


Figura 4.2 – Diagrama de flujos de la aplicación. Fuente: Elaboración Propia.

4.2. Descripción de módulos

Los siguientes componentes son descritos a partir de la representación de la Figura 4.1. Por cada componente se entrega un breve párrafo descriptivo de su función y sus interacciones con otros.

4.2.1. Módulo login y registro (C1)

Al iniciar la aplicación, ésta comprueba si existe una sesión abierta. Si la hay, se procede a actualizar y consumir los datos desde el servidor. En caso contrario, se despliega la vista de login con la opción de registrar una nueva cuenta de usuario.

4.2.2. Módulo vista principal (C2)

En este módulo se despliega en un mapa toda la información consumida a través del servidor por medio de marcadores. Ésta es obtenida por la ubicación geográfica del dispositivo móvil. Además se tiene la opción de realizar filtros, para distintos tipos de reportes. Existe la posibilidad de configurar zonas de interés, para ser notificado en el momento que se registre un nuevo delito en algún sector definido previamente.

4.2.3. Módulo formulario de reporte delictivo (C3)

Permite describir un delito asignándole su respectiva categoría. Además de poder subir alguna fotografía representativa al hecho delictivo. Esta información es almacenada en el servidor y procesada. Para una mayor confiabilidad de la información provista por los usuarios, se realiza un *Match* entre las distintas fuentes.

4.2.4. Módulo API REST (C4)

Interfaz intermediaria, controla la comunicación de la aplicación y la base de datos. Permite procesar dicha data y obtener información clasificada.

4.2.5. Módulo Servidor (C5)

Almacena toda la información delictiva de forma estructurada y relacionada, para su posterior consumo a través de la API REST [Anexo B].

4.3. Matriz de requisitos funcionales y componentes

El cuadro 4.1 muestra qué módulos o componentes, implementan qué requisitos funcionales.

	C1	C2	C3	C4	C5
RF1	X			X	X
RF2	X			X	X
RF3		X		x	X
RF4			X	X	X
RF5		X		X	X

Tabla 4.1 – Matriz de requisitos funcionales y componentes. Fuente: Elaboración Propia.

Capítulo 5

Gestión de riesgos

En este capítulo se estudiará los riesgos asociados al desarrollo de la aplicación móvil VersusCrime. Además de las estrategias y mitigaciones para prevenir dichos riesgos.

5.1. Supuestos

1. Usuario dispone de un smartphone, con los requerimientos mínimos para la ejecución de la aplicación.
2. Usuario dispone de una red de datos, para el consumo y envío de data al servidor.
3. El servidor tiene los requisitos mínimo para soportar la alta demanda de tráfico.

5.2. Dependencias

1. Confiabilidad de la información provista por los usuarios.

5.3. Restricciones

1. Limitaciones del uso del sistema operativo móvil.
2. Limitaciones en el conocimiento de iOS.
3. Limitaciones en la utilización de un Servidor Dedicado.

5.4. Riesgos

5.4.1. Confiabilidad de los datos

Existe la probabilidad de que por razones de incertidumbre, no se confíe en la información delictiva entregada por fuentes desconocidas.

5.4.2. Tiempo de consulta al servidor

Dada la enorme cantidad de información que se necesita consumir y enviar, existe la probabilidad de que se realicen en un tiempo mayor al esperado, o en el peor de los escenarios, ésta se cierre por Time Out.

5.5. Mitigaciones

5.5.1. Confiabilidad de los datos

Para entregar una mayor confiabilidad de los datos recibidos por la comunidad, se propone establecer convenios con empresas que les interese esta información, con fines de servicios u organizaciones que cuentan con registros propios, como lo son las grandes cadenas noticieras del país. Además se tienen las empresas de seguridad, que poseen su propia base de datos de hechos delictivos. Con ellas se podría realizar un *Match* entre la información obtenida desde la aplicación móvil y las organizaciones que se encuentren suscrita, y así puedan alimentar el sistema, como también nutrirse de ella.

5.5.2. Tiempo de consulta al servidor

Para descongestionar y evitar el colapso del servidor, se propone migrar todo el sistema desde el servidor actual (VPS) a un Servidor Dedicado con una mayor cantidad de recursos, que permite soportar una alta demanda de peticiones simultáneas.

Capítulo 6

Implementación del Sistema

En el capítulo 4 de Análisis y Diseño de VersusCrime se especificó su arquitectura inicial. A la luz de un trabajo de diseño más detallado, en este capítulo se presenta la arquitectura definitiva incluyendo mayor profundidad en su descripción, además de la implementación de los diferentes módulos que se compone el sistema.

6.1. Diagrama de Contexto

En el diagrama de contexto 6.1, el sistema está graficado como una caja negra, junto a los principales entes externos que interactúan con él. Las flechas que unen los entes externos con el sistema, indican flujos de datos desde o hacia el sistema.

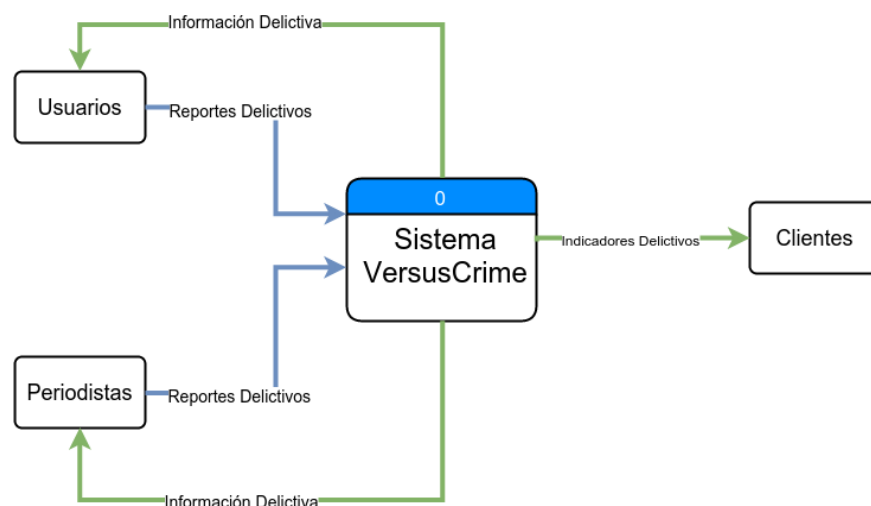


Figura 6.1 – Diagrama de Contexto Sistema VersusCrime. Fuente: Elaboración Propia.

6.1.1. Existen 3 entes externos con los que el sistema puede interactuar:

- **Usuarios:** El usuario es capaz de enviar ciertos parámetros de procesamiento al sistema como lo es su propia geolocalización de forma automática. El sistema procesa dicha información adquirida, permitiendo así que el usuario en tiempo real pueda ver el resultado de salida con toda la data necesaria.
- **Periodistas:** Al igual que el ente Usuario, es capaz de enviar ciertos parámetros al sistema de forma manual. El sistema procesa dichos datos y le entrega al periodista información de utilidad para sus reportes noticiarios.
- **Clientes:** Este ente son todo tipo de empresas que necesitan tomar decisiones valiosas a partir de indicadores delictivos. El sistema le envía de forma manual o periódica dichos indicadores a todas las empresas que se encuentren suscrita a él.

6.2. Enumeración de Módulos

El cuadro 6.1 muestra los módulos de la Figura 4.1. Por cada módulo se entrega una breve descripción de su propósito, además de la sección en donde se especifica el módulo en detalle.

Módulo	Propósito
GEOLOCALIZACIÓN	Obtención de la geolocalización a partir de las redes móviles o GPS.
APP	Despliegue de la información asociada a lo solicitado por el usuario.
API	Procesamiento de la data entrante y saliente.
SERVER	Obtención e inserción de toda la data.
GCM	Sistema de notificación de Google.

Tabla 6.1 – Módulos de la Arquitectura del Sistema. Fuente: Elaboración Propia.

6.3. Enumeración de los Requisitos Funcionales

RF1	La aplicación debe ser capaz de obtener la geolocalización del usuario, a través del GPS o la red móvil del dispositivo.
RF2	La aplicación debe ser capaz de enviar la ubicación al servidor.
RF3	El servidor debe ser capaz de procesar dicha información y consumir desde la base de datos los reportes que se encuentren dentro de un determinado rango.
RF4	La aplicación debe ser capaz de consumir la data entregada por el servidor. Posteriormente desplegar los reportes en un mapa.
RF5	La aplicación debe ser capaz de notificar al usuario, con los reportes delictivos de las zonas previamente establecidas.

6.4. Matriz de Requisitos Funcionales y Módulos

La siguiente matriz muestra los diferentes módulos del sistema y qué requisitos funcionales implementan.

	RF1	RF2	RF3	RF4	RF5
GEOLOCALIZACIÓN	X				
APP		X		X	X
API			X	X	
SERVER			X	X	
GCM					X

Tabla 6.2 – Matriz de requisitos funcionales y módulos. Fuente: Elaboración Propia.

6.5. Módulo Geolocalización

6.5.1. Definición del módulo

Propósito	Este módulo se encarga de obtener la geolocalización del usuario, a través de las redes móviles o GPS.
Alcances	Desde el punto de vista de la arquitectura este módulo funciona como punto de entrada de los datos al sistema, actuando como habilitador para que los demás módulos puedan llevar a cabo su función.
Dependencias	Dado que es el primer módulo del sistema no se necesita de la inicialización de otros módulos para que este funcione.
Supuestos	Para el funcionamiento de este módulo se requiere que esté activado la opción <i>Ubicación</i> del dispositivo móvil. Además de tener algún tipo de conexión a internet.
Restricciones	Este módulo no tiene ninguna restricción.

6.5.2. Declaraciones Públicas

Constantes Públicas

Nombre de la constante	Descripción
ACCURACY_FINE	Geolocalización con precisión alta. Es más lenta, pero entrega con exactitud la posición deseada.
ACCURACY_LOW	Geolocalización con precisión baja. Ésta es más rápida, pero entrega la posición con un margen de error más grande.
NETWORK_PROVIDER	Localización por la red telefónica
GPS_PROVIDER	Localización por GPS

Tabla 6.3 – Módulo geolocalización: Matriz de constantes públicas. Fuente: Elaboración Propia.

Tipos de Datos Públicos

Nombre del tipo	Descripción
LocationManager()	Permite obtener una lista completa de proveedores disponibles. Además se puede acceder a las distintas propiedades tales como precisión, costo, consumo de recursos y velocidad.
Criteria()	Si se desea obtener los proveedores que cumplen unos determinados requisitos entre todos los disponibles. Para ello es necesario definir un criterio de búsqueda, mediante un objeto de tipo Criteria, en el que se puede indicar las características mínimas del proveedor que se necesita utilizar.

Tabla 6.4 – Módulo geolocalización: Matriz de tipos de datos públicos. Fuente: Elaboración Propia.

Variables Públicas

Nombre de la variable	Descripción
LatLng position	Almacena de forma temporal la última posición conocida en LatLng(latitude,longitude)

Tabla 6.5 – Módulo geolocalización: Matriz de variables públicas. Fuente: Elaboración Propia.

6.5.3. Funciones Públicas

Las siguientes funciones son accesibles desde otros módulos. Otros módulos tienen acceso a la funcionalidad de este módulo mediante estas funciones.

Nombre de la función	Descripción
getProviders	Obtiene una lista de todos los proveedores disponible que satisfacen los criterios dados.
isProviderEnabled	Retorna el estado de disponibilidad del proveedor a consultar.
getLastKnownLocation	Obtiene la última localización conocida.

Tabla 6.6 – Módulo geolocalización: Matriz de funciones públicas. Fuente: Elaboración Propia.

6.5.4. Estructuras de Datos

Este módulo ha sido desarrollado en Java y su estructura de datos está basada en Programación Orientada a Objetos(POO). La POO toma las mejores ideas incorporadas en la programación estructurada y las combina con nuevos y potentes conceptos que permiten organizar los programas de forma más efectiva. La POO permite descomponer un problema en subgrupos relacionados. Cada subgrupo pasa a ser un objeto autocontenido que contiene sus propias instrucciones y datos que le relacionan con ese objeto. De esta manera, la complejidad se reduce.

6.5.5. Diseño Detallado de las Funciones

Función getProviders

En el cuadro 6.7 se puede observar los detalles de esta función.

Propósito	Devuelve una lista de los nombres de todos los proveedores que satisfacen los criterios dados.
Dependencias	No depende de otras funciones.
Prototipo	<pre>public List<String> getProviders(Criteria criteria,boolean enabledOnly)</pre>
Parámetro	Explicación
criteria	Los criterios definidos previamente para que los proveedores devueltos deben coincidir.
enabledOnly	Si es TRUE entonces se devuelven sólo los proveedores que están habilitados actualmente.
Retorno	Lista de cadenas que contienen los nombres del proveedor.
Proceso	Se instancia la clase LocationManager, luego se llama el método getProviders(criteria,true). Finalmente retorna la lista con los proveedores.

Tabla 6.7 – Matriz de la función getProviders. Fuente: Elaboración Propia.

Función isProviderEnabled

En el cuadro 6.8 se puede observar los detalles de esta función.

Propósito	Devuelve el estado actual del proveedor determinado.
Dependencias	No depende de otras funciones.
Prototipo	<pre>public boolean isProviderEnabled (String provider)</pre>
Parámetro	Explicación
provider	El nombre del proveedor.
Retorno	True si existe el proveedor y está habilitado. False caso contrario.
Proceso	Se instancia la clase LocationManager. Luego se llama el método isProviderEnabled para algún proveedor seleccionado. Finalmente retorna su estado respectivo.

Tabla 6.8 – Matriz de la función isProviderEnabled. Fuente: Elaboración Propia.

Función getLastKnownLocation

Propósito	Devuelve una ubicación a partir de los datos de la última localización conocida por el proveedor dado.
Dependencias	No depende de otras funciones.
Prototipo	<pre>public Location getLastKnownLocation (String provider)</pre>
Parámetro	Explicación
provider	El nombre del proveedor.
Retorno	La última ubicación conocida por el proveedor, o null si no la hay.
Proceso	Se instancia la clase LocationManager. Luego se llama el método getLastKnownLocation para algún proveedor seleccionado. Finalmente retorna su última ubicación.

Tabla 6.9 – Matriz de la función getLastKnownLocation. Fuente: Elaboración Propia.

6.6. Módulo APP

6.6.1. Definición del módulo

Propósito	Despliega toda la información recibida a partir de ciertos parámetros enviados por el usuario al webservice(API). Además procesa cierta data que será enviada al servidor, como lo son los reportes delictivos.
Alcances	Este módulo corresponde en cierta forma a la finalización del proceso de los datos, dado que el mayor flujo de información solicitada por el usuario se despliegan aquí.
Dependencias	Depende del correcto funcionamiento del módulo GEOLOCALIZACIÓN, debido a que el despliegue de información se acompaña de la ubicación geográfica del usuario.
Supuestos	Correcto funcionamiento del módulo anterior.
Restricciones	No existe restricción para este módulo.

6.6.2. Declaraciones Públicas

Esta sección enumera constantes, tipos y variables del módulo, visibles para otros módulos.

Constantes Públicas

Nombre de la constante	Descripción
MAP	Instancia del objeto GoogleMap.
USER	Instancia de los datos provisto por el usuario al iniciar sesión en la aplicación.
LOAD_DATA	Instancia del objeto LoadData().
REST_CLIENT	Instancia de la biblioteca Retrofit, permite la interacción con la API.
URL_API	URL para acceder a los métodos dentro de la API.

Tabla 6.10 – Módulo APP. Matriz de constantes públicas. Fuente: Elaboración Propia.

Tipos de Datos Públicos

Nombre del tipo	Descripción
GoogleMap()	Esta es la clase principal de la API de Google Maps para Android y es el punto de entrada para todos los métodos relacionados con el mapa.
User()	Objeto de la clase User con los datos necesarios para el procesamiento de los métodos de la aplicación.
LoadData()	Clase que permite la interacción con la API.
RestClient()	Clase de servicios web que se puede utilizar para que sea fácil y segura de consumir los distintos tipos de servicios (POST, GET, DELETE y PUT). Es compatible tanto con la recuperación sincrónica y asincrónica.

Tabla 6.11 – Módulo APP. Matriz de tipos de datos públicos. Fuente: Elaboración Propia.

Nombre de la variable	Descripción
POSITION	Almacena la última posición conocida para el despligue de la data sobre el mapa.

Tabla 6.12 – Módulo APP. Matriz de variables públicas. Fuente: Elaboración Propia.

Variables Públicas

6.6.3. Funciones Públicas

Las siguientes funciones son accesibles desde otros módulos. Otros módulos tienen acceso a la funcionalidad de este módulo mediante estas funciones.

Nombre de la función	Descripción
getMap()	Obtiene el GoogleMap subyacente que está ligado a la vista envuelta por el fragmento.
getCrimes()	Obtiene los delitos geolocalizados a partir de una localización geográfica dada.
postCrime()	Envía un delito geolocalizado hacia el servidor con los datos correspondientes al él.

Tabla 6.13 – Módulo APP. Matriz de funciones públicas. Fuente: Elaboración Propia.

6.6.4. Estructuras de Datos

Al igual que el módulo anterior, éste ha sido desarrollado en Java y su estructura de datos esta basada en Programación Orientada a Objetos.

6.6.5. Funciones Privadas

El módulo no posee funciones privadas.

6.6.6. Diseño Detallado de las Funciones

Función getMap

En el cuadro 6.14 se observa los detalles de esta función.

Propósito	Obtiene el GoogleMap subyacente que está ligado a la vista envuelta por el fragmento.
Dependencias	MapView: Clase pública MapView que extiende de FrameLayout, facilita la instancia del mapa a través del fragmento dentro de la vista.
Prototipo	<pre>public final GoogleMap getMap ()</pre>
Retorno	Devuelve el GoogleMap. Null si la vista del fragmento aún no está lista.
Proceso	Se instancia el MapView, luego la clase GoogleMap. Finalmente se llama al método getMap() retornando el mapa a utilizar.

Tabla 6.14 – Matriz de la función getMap. Fuente: Elaboración Propia.

Función postCrime

En el cuadro 6.15 se observa los detalles de esta función.

Propósito	Enviar un reporte delictivo que el usuario haya acontecido.
Dependencias	LoadData: Clase que se debe instanciar previamente para poder acceder a este método.
Prototipo	<pre>public void postCrime(Context context, Crime crime)</pre>
Parámetro	Explicación
crime	Objeto Crime previamente creado a través de un formulario.
Retorno	No devuelve nada.
Proceso	Se obtiene el Crime ingresado por el usuario. Luego se instancia la clase LoadData. Finalmente se accede al método postCrime() para ser el reporte respectivo.

Tabla 6.15 – Matriz de la función postCrime. Fuente: Elaboración Propia.

Función getCrimes

En el cuadro 6.16 se observa los detalles de esta función.

Propósito	Obtiene una lista de delitos a partir de una ubicación geográfica y con un rango determinado.
Dependencias	LoadData: Clase que se debe instanciar previamente para poder acceder a este método.
Prototipo	<pre>public void getCrimes(Context context, GoogleMap map, String lat, String lon)</pre>
Parámetro	Explicación
context	Permite acceder a los recursos específicos de la aplicación y a sus clases, así como llamar al padre para realizar operaciones a nivel de la aplicación.
map	Es una instancia del fragmento mapa. Permite desplegar la data recibida a través de marcadores geolocalizados con la información de los delitos respectivos.
lat	Latitud: Es uno de los parámetros de la ubicación del usuario.
lon	Longitud: Es uno de los parámetros de la ubicación del usuario.
Retorno	No devuelve nada, dado que se pasa la instancia del mapa y realiza el despliegue de los delitos dentro del método.
Proceso	Se instancia la clase LoadData, luego se accede al método getCrimes(). Finalmente se visualiza los marcadores con los delitos respectivos.

Tabla 6.16 – Matriz de la función getCrimes. Fuente: Elaboración Propia.

6.7. Módulo API

6.7.1. Definición del módulo

Propósito	Es un conjunto de protocolos y estándares que sirven para intercambiar datos entre la aplicación móvil y el servidor.
Alcances	Se definen métodos de comunicación para el flujo de la información a través de un JSON que es un formato para el intercambios de datos.
Dependencias	Depende del correcto funcionamiento del módulo APP, debido a que es éste el que le proporciona los parámetros necesarios para la obtención de la data requerida.
Supuestos	Correcto funcionamiento del módulo anterior
Restricciones	No existe restricción para este módulo.

6.7.2. Declaraciones Públicas

Esta sección enumera constantes, tipos y variables del módulo, visibles para otros módulos.

Constantes Públicas

Este módulo no posee constantes públicas.

Tipos de Datos Públicos

Este módulo no posee tipos de datos públicos.

Variables Públicas

Este módulo no posee variables públicas.

6.7.3. Funciones Públicas

Las siguientes funciones son accesibles desde otros módulos. Otros módulos tienen acceso a la funcionalidad de este módulo mediante estas funciones.

Nombre de la función	Descripción
load_crimes()	Obtiene los delitos geolocalizados a partir de una localización geográfica recibida por POST.
new_crime()	Almacena un delito geolocalizado en la base de datos con los datos correspondiente del delito recibido por POST

Tabla 6.17 – Módulo API. Matriz de funciones públicas. Fuente: Elaboración Propia.

6.7.4. Estructuras de Datos

Este módulo ha sido desarrollado en PHP con el FrameWork Laravel. Éste provee un conjunto de estructuras de datos estándar.

6.7.5. Funciones Privadas

Este módulo no posee funciones privadas.

6.7.6. Diseño Detallado de las Funciones

Función load_crimes

Propósito	Se obtiene los delitos geolocalizados a partir de la información recibida previamente por POST.
Dependencias	Api_Controller: Clase que le entrega una capa de seguridad a las peticiones request.
Prototipo	<hr/> <pre>function load_crimes()</pre> <hr/>
Retorno	Devuelve un JSON con toda la data necesaria. JSON es un acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos.
Proceso	Se recibe los parámetros por POST, luego se procede a obtener los datos necesarios. Finalmente imprime el resultado en formato JSON.

Tabla 6.18 – Módulo API. Matriz función load_crimes. Fuente: Elaboración Propia.

Función new_crime

Propósito	Se almacena el delito geolocalizado a partir de la información recibida previamente por POST.
Dependencias	Api_Controller: Clase que le entrega una capa de seguridad a las peticiones request.
Prototipo	<pre>function new_crime()</pre>
Retorno	Devuelve un JSON con estado TRUE si el resultado fue exitoso. FALSE caso contrario. JSON es un acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos.
Proceso	Se recibe los parámetros por POST, luego se procede a almacenar dicha información en la base de datos. Finalmente imprime el resultado exitoso en formato JSON.

Tabla 6.19 – Módulo API. Matriz función new_crime

6.8. Módulo Server

6.8.1. Definición del módulo

Propósito	Consulta a la Base de Datos para obtener información de lo solicitado por el usuario.
Alcances	La principal función de este módulo es consultar al servidor por la data requerida por el usuario. Además de almacenar la información provista por él mismo.
Dependencias	Dependerá del modulo de API, ya que éste le dará la respectiva información a consultar.
Supuestos	Correcto funcionamiento del módulo anterior.
Restricciones	Posee limitaciones de consultas simultáneas a la base de datos.

6.8.2. Declaraciones Públicas

Esta sección enumera constantes, tipos y variables del módulo, visibles para otros módulos.

Constantes Públicas

Este módulo no posee constantes públicas.

Tipos de Datos Públicos

Este módulo no posee tipos de datos públicos.

Variables Públicas

Este módulo no posee variables públicas.

6.8.3. Funciones Públicas

Las siguientes funciones son accesibles desde otros módulos. Otros módulos tienen acceso a la funcionalidad de este módulo mediante estas funciones.

Nombre de la función	Descripción
getCrimes()	Realiza las consultas SQL a partir de los parámetros provisto por el módulo API, para obtener los datos de delitos solicitado.
saveCrime()	Realiza la inserción en SQL del delito nuevo a almacenar.

Tabla 6.20 – Módulo Server. Matriz de las funciones públicas. Fuente: Elaboración Propia.

6.8.4. Estructuras de Datos

Este módulo ha sido desarrollado en PHP con el FrameWork Laravel. Éste provee un conjunto de estructuras de datos estándar.

6.8.5. Funciones Privadas

Este módulo no posee funciones privadas.

6.8.6. Diseño Detallado de las Funciones

Función getCrimes

Propósito	Se obtienen los datos necesarios a través de las consultas SQL con los parámetros provisto por el módulo API.
Dependencias	CI.Model: Clase que entrega un conjunto de métodos, facilitando las consultas a la base de datos.
Prototipo	<pre>function getCrimes()</pre>
Retorno	Devuelve un ARRAY con toda la data necesaria.
Proceso	Se recibe los parámetros por referencia, luego se procede a obtener los datos necesarios a través de SQL. Finalmente retorna el ARRAY con el resultado arrojado.

Tabla 6.21 – Módulo API. Matriz función getCrimes. Fuente: Elaboración Propia.

Función saveCrime

En el cuadro 6.22 se observa los detalles de esta función.

Propósito	Se almacena el delito provisto por el módulo API a través de una inserción SQL.
Dependencias	CI.Model: Clase que entrega un conjunto de métodos, facilitando las consultas a la base de datos.
Prototipo	<pre>function saveCrime()</pre>
Retorno	Devuelve TRUE si fue exitosa la inserción. FALSE en caso contrario.
Proceso	Se recibe los parámetros por referencia, luego se procede a insertar los datos provisto por la API a través de una sintaxis SQL. Finalmente retorna un Boolean con el estado exitoso o fallido.

Tabla 6.22 – Módulo API. Matriz función saveCrime. Fuente: Elaboración Propia.

6.9. Módulo GCM

6.9.1. Definición del módulo

Propósito	El servicio Google Cloud Messaging permite recibir notificaciones push en los distintos dispositivos móviles Android.
Alcances	Se registra el dispositivo móvil en el módulo GCM, permitiendo el envío de notificación push a través del servidor con el id previamente almacenado.
Dependencias	Dependerá de los módulos APP para el registro del dispositivo y API para el envío de la notificación push a aquel móvil.
Supuestos	Correcto funcionamiento de los módulos anteriores.
Restricciones	No existe restricciones para este módulo.

6.9.2. Declaraciones Públicas

Esta sección enumera constantes, tipos y variables del módulo, visibles para otros módulos.

Constantes Públicas

Nombre de la constante	Descripción
API.KEY	Es un identificador que permite el acceso al envío de notificaciones push al las aplicaciones que posean este mismo ApiKey.
URL	Url de la API de Google https://android.googleapis.com/gcm/send

Tabla 6.23 – Módulo GCM. Matriz de constantes públicas. Fuente: Elaboración Propia.

Tipos de Datos Públicos

En este módulo no existe tipos de datos públicos.

Variables Públicas

Nombre de la variable	Descripción
mensajes	Variable que se setea desde el módulo server para el envío respectivo de la notificación push a un dispositivo determinado.

Tabla 6.24 – Módulo GCM. Matriz de tipos de datos públicos. Fuente: Elaboración Propia.

6.9.3. Funciones Públicas

Las siguientes funciones son accesibles desde otros módulos. Otros módulos tienen acceso a la funcionalidad de este módulo mediante estas funciones.

Nombre de la función	Descripción
sendGoogleCloudMessage	Envío de notificaciones push.

Tabla 6.25 – Módulo GCM. Matriz de funciones públicas. Fuente: Elaboración Propia.

6.9.4. Estructuras de Datos

Este módulo ha sido desarrollado en PHP con el framework Codeigniter. Éste provee un conjunto de estructuras de datos estándar.

6.9.5. Funciones Privadas

Este módulo no posee funciones privadas.

6.9.6. Diseño Detallado de las Funciones

Función sendGoogleCloudMessage

En el cuadro 6.26 se observa los detalles de esta función.

Propósito	Permite enviar notificaciones push en los distintos dispositivos móviles Android.
Dependencias	No posee dependencias.
Prototipo	<pre>function sendGoogleCloudMessage(\$message, \$id)</pre>
Parámetro	Explicación
message	ARRAY con el mensaje a enviar. Ejemplo <code>array("title" => "MiTitulo", "body" => "MiMensaje")</code>
id	Identificador del dispositivo móvil a enviar la notificación push.
Retorno	Devuelve TRUE si fue exitosa la inserción. FALSE en caso contrario.
Proceso	Se setea el Array con el mensaje a enviar. Además se agrega el Id del dispositivo móvil destino. Finalmente es enviada la notificación push.

Tabla 6.26 – Módulo GCM. Matriz de la función sendGoogleCloudMessage. Fuente: Elaboración Propia.

6.10. Aplicación Móvil VersusCrime

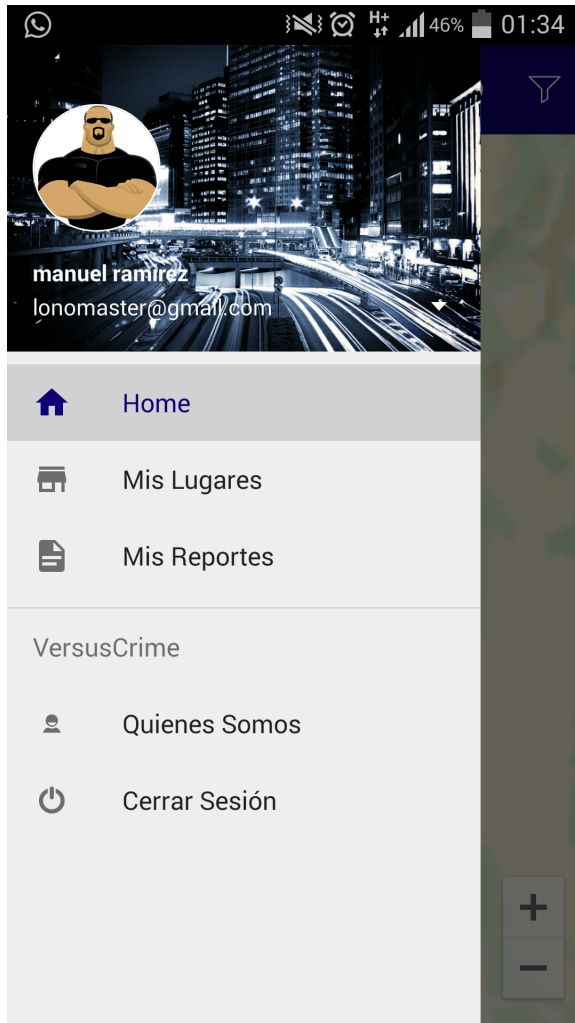


Figura 6.2 – Vista menú lateral con las principales funciones



Figura 6.3 – Vista despliegue de reportes delictivos

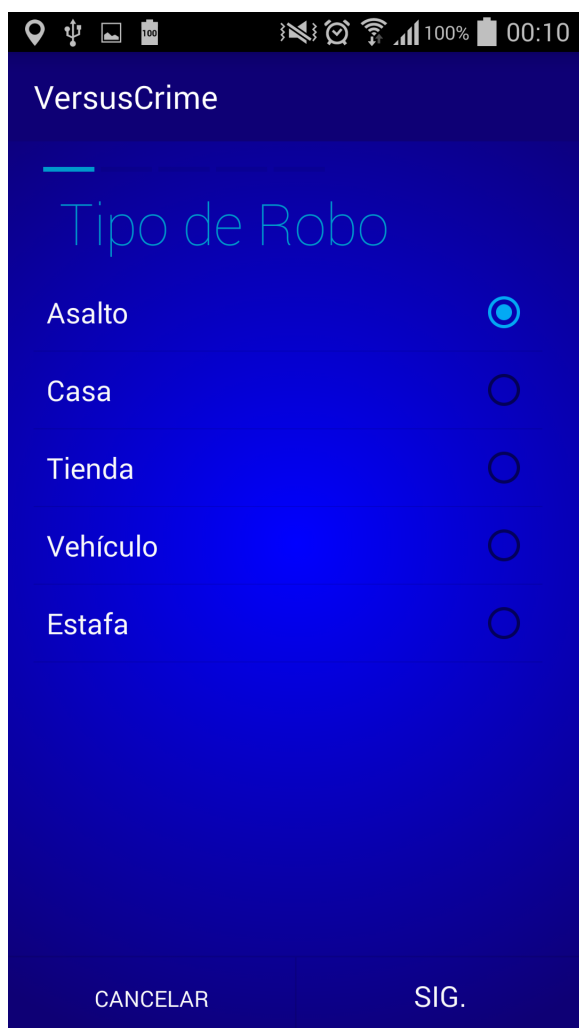


Figura 6.4 – Vista formulario de ingreso de un delito

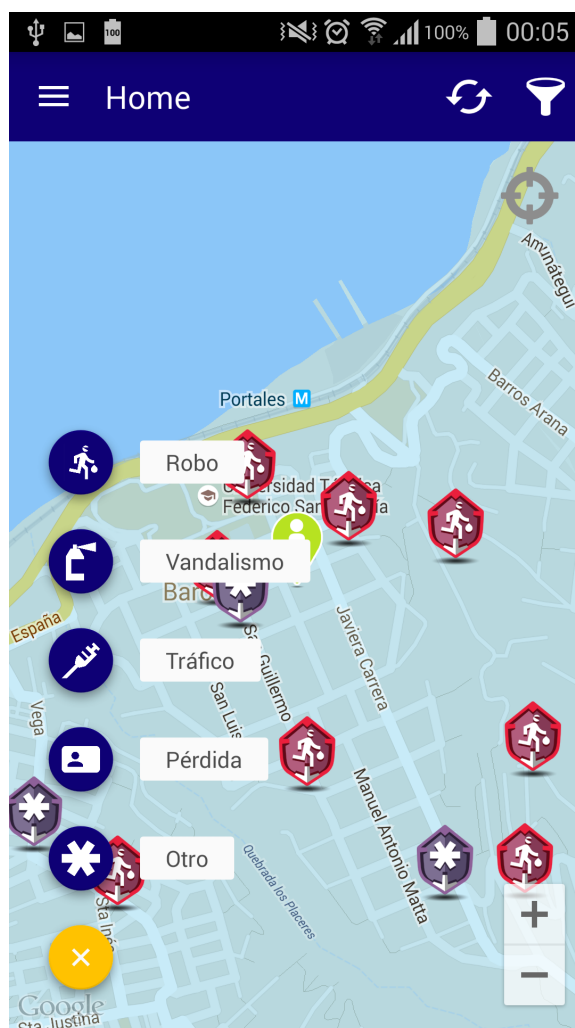


Figura 6.5 – Vista selección opción de reportar un delito

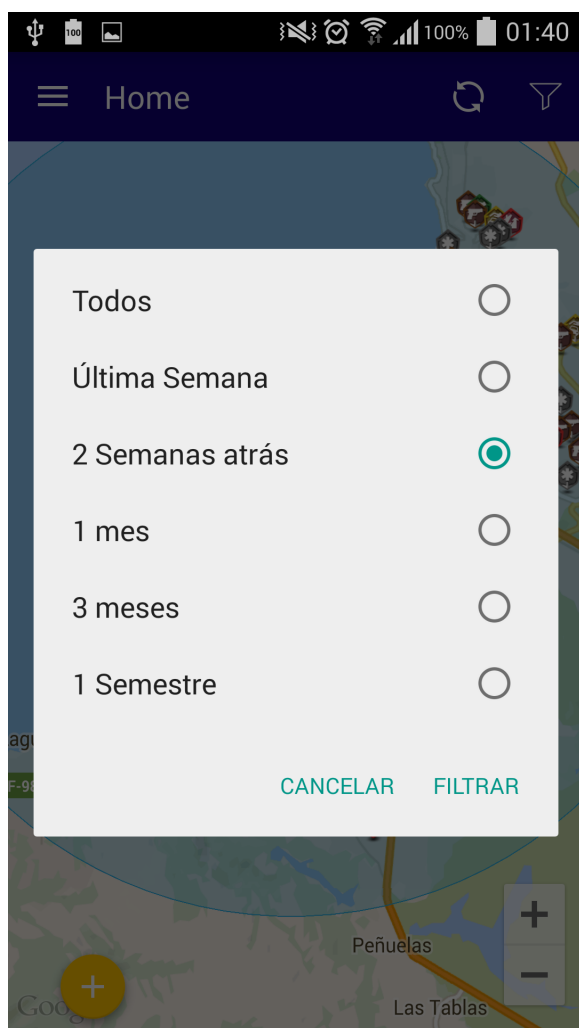


Figura 6.6 – Vista selección de filtros por periodos de fechas

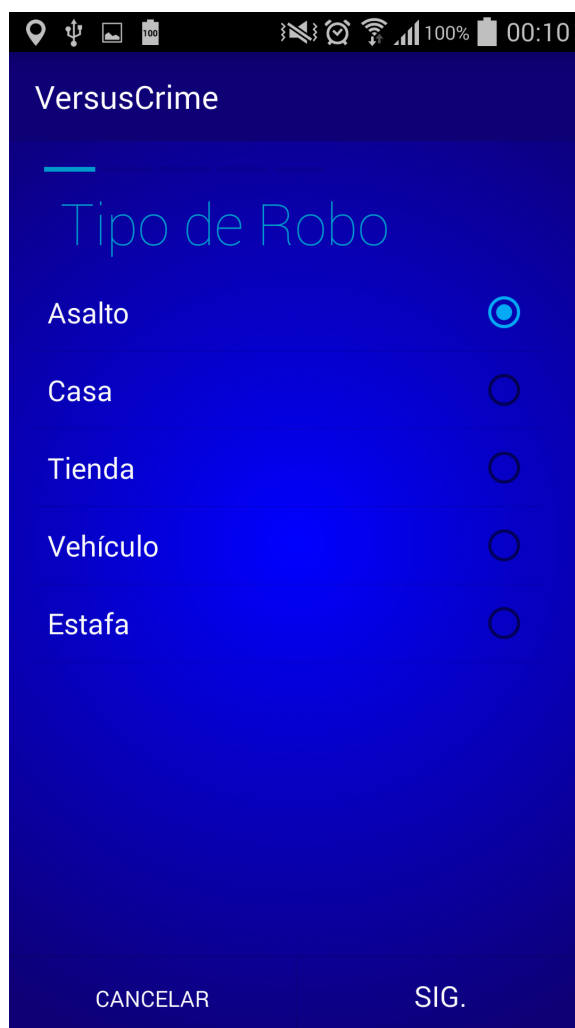


Figura 6.7 – Vista formulario de ingreso de un delito

6.11. Panel de Gestión para entidades suscritas al sistema

Durante este periodo, se comienzan a diseñar las vistas respectiva de la plataforma web, con sus respectivos mantenedores. A continuación se detallan 3 vistas principales: Panel, Mis Reportes y Agregar un Nuevo Reporte.

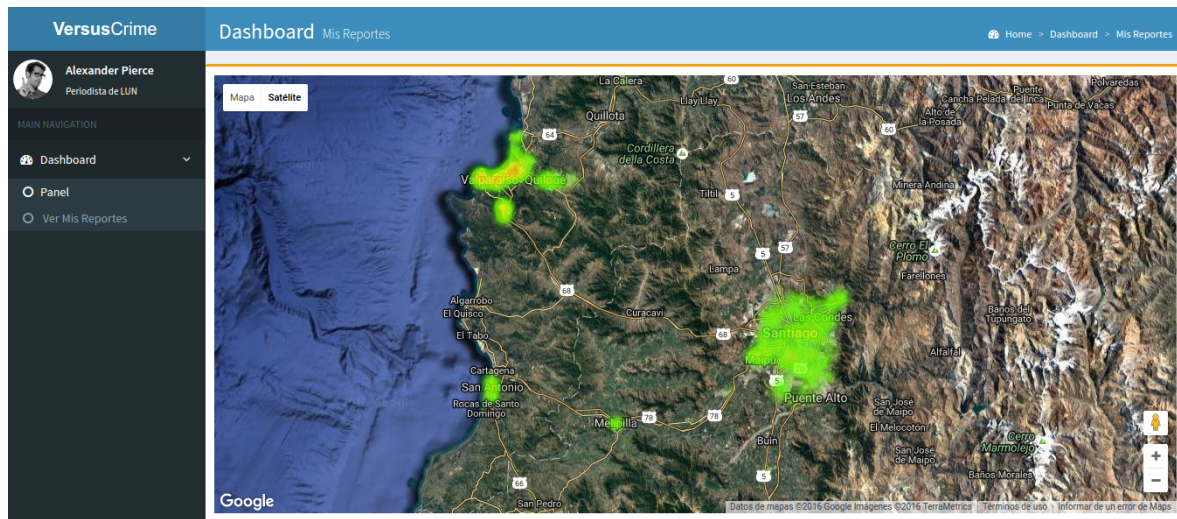


Figura 6.8 – Sistema VersusCrime

6.11.1. Panel

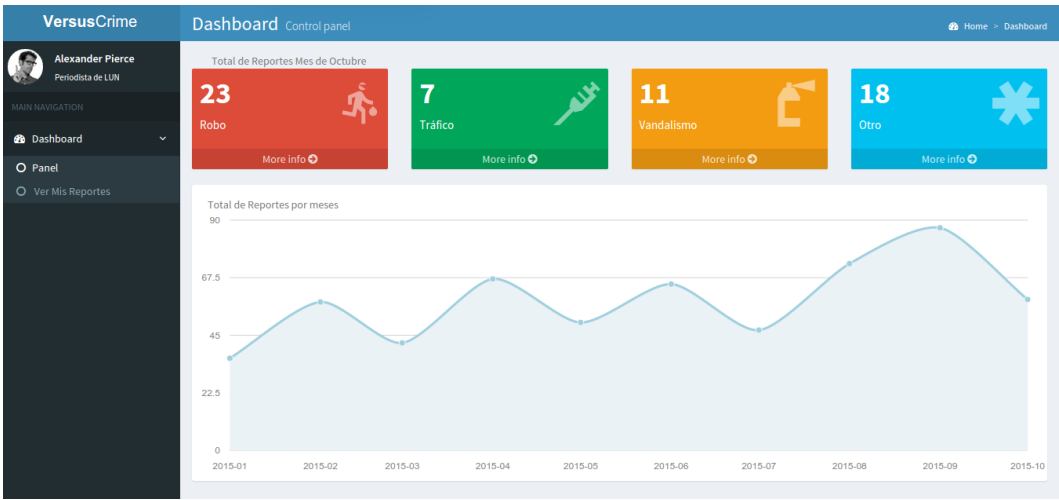


Figura 6.9 – Vista Principal: Panel

En esta vista se puede observar en los cuadrados ubicados en la parte superior, la cantidad de reportes realizado por el usuario durante el mes actual. Además en la siguiente sección se aprecia la cantidades de reportes durante los meses anteriores.

6.11.2. Mis reportes

The screenshot shows the 'VersusCrime' dashboard for user 'Alexander Pierce' in the 'Mis Reportes' view. The top navigation bar includes 'Home', 'Dashboard', and 'Mis Reportes'. The main content area is titled 'Lista de mis reportes' and shows a list of reports. The table has columns for Tipo, Coordenadas, Descripción, Fecha, and Opciones. The 'Opciones' column contains 'Edit' and 'Delete' buttons for each report.

Tipo	Coordenadas	Descripción	Fecha	Opciones
Test	(-33.00,-71.00)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	29-10-2015 22:05:04	Edit Delete
Test	(-33.00,-71.00)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	29-10-2015 22:05:04	Edit Delete
Test	(-33.00,-71.00)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	29-10-2015 22:05:04	Edit Delete
Test	(-33.00,-71.00)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	29-10-2015 22:05:04	Edit Delete
Test	(-33.00,-71.00)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	29-10-2015 22:05:04	Edit Delete
Test	(-33.00,-71.00)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	29-10-2015 22:05:04	Edit Delete
Test	(-33.00,-71.00)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	29-10-2015 22:05:04	Edit Delete
Test	(-33.00,-71.00)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	29-10-2015 22:05:04	Edit Delete
Test	(-33.00,-71.00)	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	29-10-2015 22:05:04	Edit Delete

Figura 6.10 – Vista Mis Reportes

En esta vista se puede observar una tabla con toda la data de los reportes, con sus respectivos opciones como editar y eliminar reporte.

6.11.3. Agregar nuevo reporte

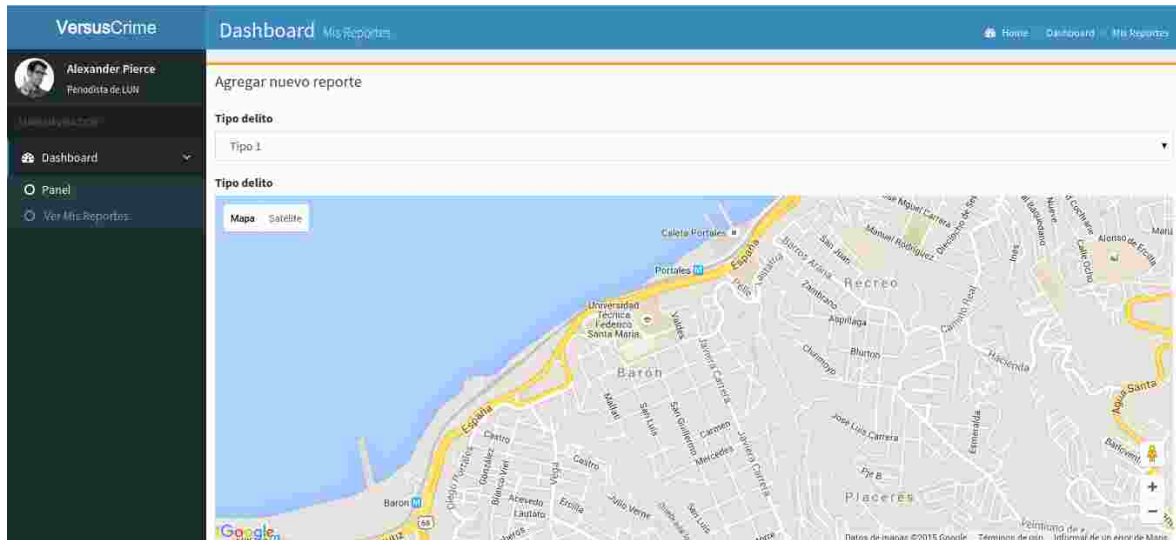


Figura 6.11 – Vista Agregar Reporte

En esta vista se puede apreciar un formulario con todos los campos necesarios para agregar un nuevo reporte a la base de datos.

Capítulo 7

Verificación y Validación

7.1. Análisis y estudio de mercado sobre la intención de uso en aplicaciones móviles de mapas de seguridad

7.1.1. Definición categoría de la aplicación móvil

Debido a la gran magnitud de aplicaciones que existe hoy en día, es que es imprescindible clasificar VersusCrime dentro de una categoría, lo que facilitaría el posicionamiento en la mente del consumidor, a pesar de que sea un tipo de categoría emergente en nuestro país. Siguiendo esta misma línea, es otra ventaja que se obtendrá y que ayudará en el estudio ya que contextualizarán de manera más sencilla y tendrán una idea más clara de qué situación pensar para poder responder las preguntas.

El equipo de VersusCrime determinó que la categoría más representativa para la aplicación es Seguridad Ciudadana. La razón de esto radica en que lo que busca VersusCrime es exactamente eso: una manera en que todos los ciudadanos puedan apoyarse y que favorezca a todos, notificando y alarmando sobre hechos delictivos lo que generará un mayor grado de seguridad en el quehacer diario de la ciudadanía.

7.1.2. Análisis de intención de uso

Para saber si la aplicación VersusCrime está causando un impacto positivo en las personas respecto a si estarían dispuestas a usarla, para ello se realiza un análisis de intención de uso. Se hizo necesario hacer una revisión bibliográfica buscando si es que anteriormente se ha aplicado un modelo de intención de uso a aplicaciones móviles de mapas de seguridad. En esta rama sobre modelos de intención de uso, existe el modelo Technology Acceptance Model (TAM), que mide distintas tecnologías y sobre el cual se han hecho muchas derivaciones. Es aquí donde se encontraron dos modelos para utilizar en esta investigación:

- *Unified theory of acceptance and use of technology 2* (UTAUT2), un modelo que fue desarrollado por un grupo de investigadores, entre ellos Venkatesh, que es reconocido en esta área sobre investigación del uso de tecnologías.
- Un modelo derivado del TAM, del paper llamado *Factors influencing users employment of mobile map services*, que tiene como objetivo investigar las percepciones de los usuarios de aplicaciones de mapas móviles explorando características externas de los servicios.

Para medir la intención de uso de VersusCrime se realiza un cuestionario entre estos dos modelos, ya que se concluyó que se adaptaba mejor a la realidad chilena, ya que por un lado, UTAUT2, presenta constructos muy generales y que no tienen que ver con el objetivo de la encuesta, como por ejemplo:

- Expectativas de esfuerzo, que tiene preguntas como: aprender cómo se utiliza el internet móvil es fácil para mí y, mi interacción con él, es clara y entendible.
- Condiciones facilitadoras, que tiene preguntas como: tengo los recursos necesarios para usar internet móvil y, tengo el conocimiento necesario para usar internet móvil.

Del otro modelo los constructos que generan problemas, ya que son muy específicos e implican un uso y conocimiento claro de la aplicación son:

- Intención de uso, que tiene preguntas como: preferiría usar aplicaciones de mapas móviles que otro tipo de aplicaciones de servicios de información geográficos.
- Utilidad percibida, que tiene preguntas como: creo que los servicios de mapas móviles son útiles en mi trabajo.

Ver [Anexo G] para revisar los resultados obtenidos en detalle.

7.1.3. Análisis de mercado

Necesidad de información y objetivo de la investigación

Considerando el hecho que VersusCrime es una aplicación que ya se encuentra en el mercado, es decir, está disponible para ser descargada en Google Play, se hace indispensable un mayor acercamiento con la opinión de los usuarios y potenciales usuarios para poder fortalecer esta aplicación y con esto, demostrar que se está escuchando a las personas, tomando en cuenta su opinión y que la finalidad de esta aplicación sea ayudar a la población.

De esta forma, se complementa la encuesta anterior con un estudio de preferencias y el objetivo principal es determinar cuáles son los atributos que debería tener esta aplicación según los usuarios y potenciales usuarios, demostrando al equipo de Versus-Crime si las acciones puestas en marcha son evaluadas positivamente. Para esto se va a recabar información sobre cuáles son los potenciales atributos que la gente considera que debe tener esta aplicación móvil.

Investigación exploratoria

El objetivo de esta investigación es conseguir de manera preliminar los atributos más relevantes para después poder hacer una encuesta con estos y sacar conclusiones. Para comenzar el estudio de preferencias, se investigó haciendo un fase exploratoria en la que se consideró:

- Investigar en páginas web, sobre el problema de la delincuencia que acontece en Chile.
- Investigar en Google play y App Store: aquí lo que se hizo fue revisar la competencia de VersusCrime (CityCop, LookApp, Oincs y Muevete Seguro), tanto las funcionalidades que tienen como la opinión que tienen los usuarios, considerando tanto los elementos buenos y malos que tienen estas, además de revisar la opinión de los usuarios de VersusCrime en Google Play.
- Preguntar directamente a potenciales usuarios (personas entre 18 a 30 años), explicándoles que existe una aplicación móvil que combate la delincuencia, la cual posee un mapa y en éste se registran hechos delictivos. Con esta información se les preguntó cuáles son los atributos que consideraban que debería tener esta aplicación móvil.

Posteriormente, se hizo un análisis respecto a los atributos que se mencionaban, debido a que éstos eran muchos pero si solo los nombraba una persona, no tenían el peso suficiente y fue considerado como un atributo subjetivo para una persona en particular. Entonces, el total de atributos recolectados y los que fueron clasificados se mencionan a continuación:

Atributos sin clasificar:

1. Iconos representativos para los delitos: que se entienda claramente el tipo de delito que ocurrió. Por ejemplo, si hubo un asalto a mano armada, que el icono sea una pistola.
2. Geolocalización: la aplicación permita identificar la posición exacta del usuario para que éste no pierda tiempo en buscar en el mapa.
3. Plan Cuadrante: tener la posibilidad de llamar al plan cuadrante más cercano en caso de que suceda un hecho delictivo.
4. Mapas de calor: permite tener una visualización clara de las zonas potencialmente inseguras.
5. Grupos de usuarios: facilitar una plataforma para mantener comunicados a las personas de ciertos sectores y además, lo puedan hacer por medio de redes sociales. Permitir invitar a personas que se integren en estas redes.
6. Notificaciones: determinar zonas en las cuales al usuario le llegue un aviso si ocurre un tipo de delito o el tipo de delito que le interese (filtro). Disponer de una opción, si el usuario desea recibir notificaciones en tiempo real del lugar por donde transita.
7. Filtros: permitir en el mapa mostrar los delitos de acuerdo a los periodos de tiempo que el usuario estime relevante, ya sea hace 1 semana atrás y/o de noche.
8. Rutas seguras: muestra información sobre qué caminos puede tomar el usuario evitando lugares potencialmente peligrosos.
9. Botón de pánico: permite tener contacto con carabineros de manera expedita en caso de que uno presencie un delito. Además, que registre que ocurrió un hecho delictivo y pueda hacer un seguimiento del celular por medio de GPS.

10. Límite de denuncias: disponer de un número de registro de delitos para que los usuarios no abusen del sistema saturándolo.
11. Anónimo por defecto: registrar delitos como anónimo para proteger la seguridad de la persona.
12. Carabineros: Los funcionarios proveen de información a la plataforma. Que sea utilizada por el personal de Carabineros y se les pueda contactar, asimismo se tenga la ubicación de las comisarías y de las radios patrullas.
13. Comentarios y valoración: los usuarios puedan comentar y valorar los registros de otros usuarios. De esta manera, se podrán tener usuarios destacados y también bloqueados, estos últimos en caso de que emitan registros falsos.
14. Actualización en tiempo real de los hechos que se registran.
15. Poder utilizarse sin GPS: ubicar una ciudad por medio de un buscador.
16. Información detallada de los eventos: que la información que se llene dependiendo del tipo de delito sea distinta. Por ejemplo, si se roba un auto, que se entregue información de éste.
17. Web: se tenga una página web con el mapa delictual.
18. Menor consumo de datos de internet y mayor velocidad en la transferencia de información.

Atributos clasificados, los que se van a utilizar en la encuesta:

- (A) Identificar mi ubicación automáticamente en el mapa (geolocalización).
- (B) Comunicarme por medio de grupos sobre hechos delictivos.
- (C) Recibir notificaciones de delitos en lugares que transito y/o en lugares que deseo.
- (D) Usar filtros en el mapa como el tipo de delito que deseo ver, antigüedad u horario (mañana, tarde o noche).
- (E) Visualizar rutas seguras por donde pueda transitar.
- (F) Usar un botón de pánico para marcar que ha ocurrido un delito donde me encuentro y además llamar al plan cuadrante/carabineros.
- (G) Publicar los delitos e información en redes sociales.
- (H) Tener la opción preestablecida de anónimo para registrar delitos.
- (I) Mostrar la ubicación de carabineros en el mapa: comisarías, retenes móviles, etc.
- (J) Eliminar publicaciones falsas o desactualizadas.
- (K) Limitar la cantidad de denuncias diarias que uno puede realizar.
- (L) Actualizar los registros delictivos en tiempo real.
- (M) Generar descripción de los delincuentes.

7.2. Resultados Generales

El universo de estudio considerado, fueron aproximadamente 200 personas, entre estudiantes universitarios en general y personas entre 18 a 30 años de edad.



Figura 7.1 – Atributos de preferencias

A partir de los atributos previos, los usuarios prefieren las siguientes características:

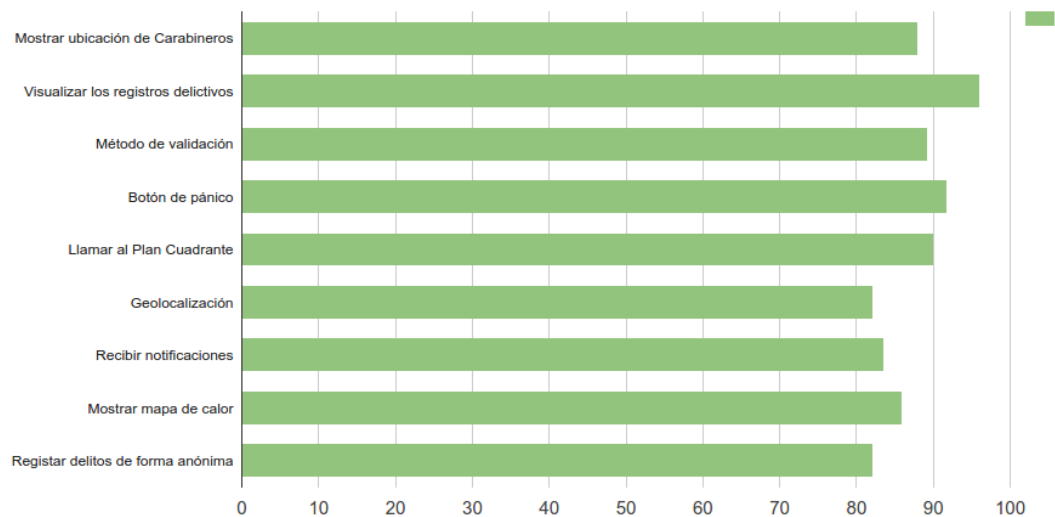


Figura 7.2 – Atributos seleccionados por los usuarios

A continuación se muestra los resultados de percepción en la aplicación, donde 1: Totalmente en desacuerdo, y 5: Totalmente en de acuerdo.

■ Aspectos de exactitud de la ubicación:

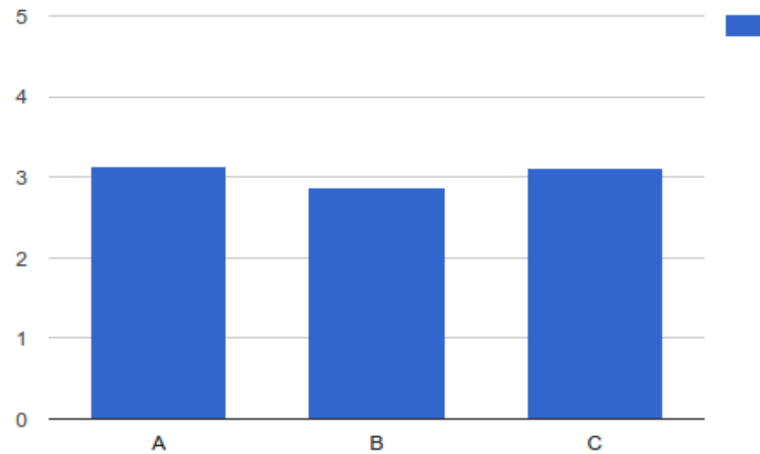


Figura 7.3 – Percepción de exactitud de la ubicación

- (A) Las aplicaciones móviles de mapas de seguridad siempre indican con exactitud la ubicación de eventos delictivos y regiones de interés.
- (B) Las aplicaciones móviles de mapas de seguridad siempre indican con exactitud la ubicación en tiempo real de eventos delictivos y regiones de interés frente a situaciones de movimiento de la zona visualizada.
- (C) Las aplicaciones móviles de mapas de seguridad entregan rutas eficientes y exactitud en los destinos a los que deseo ir, evitando rutas potencialmente peligrosas.

■ Aspectos de satisfacción en la aplicación:

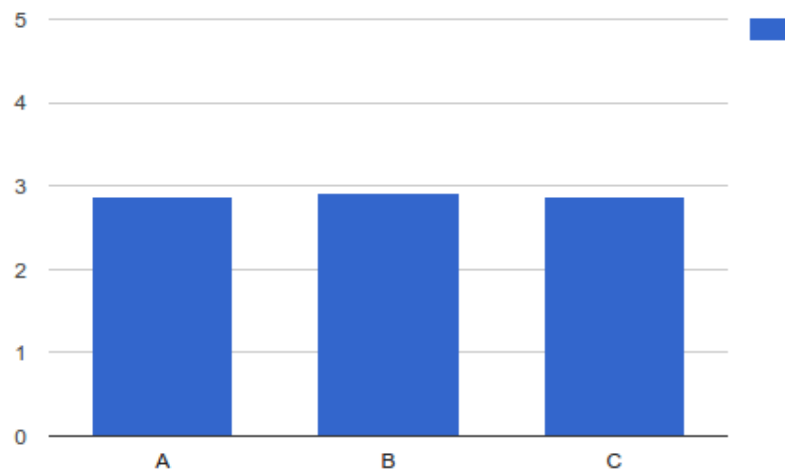


Figura 7.4 – Percepción de satisfacción

- (A) En general, me encuentro satisfecho con las aplicaciones móviles de mapas de seguridad.
- (B) Las aplicaciones móviles de mapas de seguridad que actualmente utilizo cumplen con mis expectativas.
- (C) Recomendando las aplicaciones móviles de mapas de seguridad a otras personas que intentan usar o comprar aplicaciones móviles.

■ Aspectos de calidad del servicio:

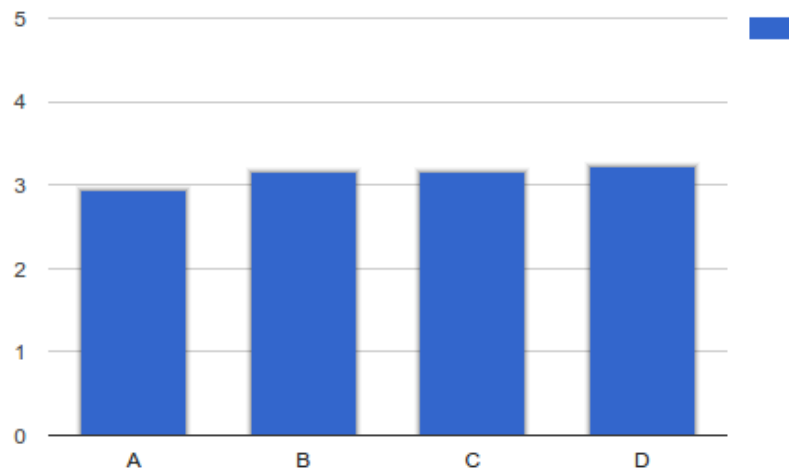


Figura 7.5 – Percepción de calidad del servicio

- (A) Las aplicaciones móviles de mapas de seguridad y la información en pantalla satisfacen completamente mis necesidades.
- (B) No tengo problemas o restricciones en el uso de aplicaciones móviles de mapas de seguridad.
- (C) Las aplicaciones móviles de mapas de seguridad entregan servicios precisos y exactos de acuerdo a su propósito.
- (D) No tengo problemas o restricciones viendo la información en pantalla que exhiben las aplicaciones móviles de mapas de seguridad.

■ Aspectos de movilidad:

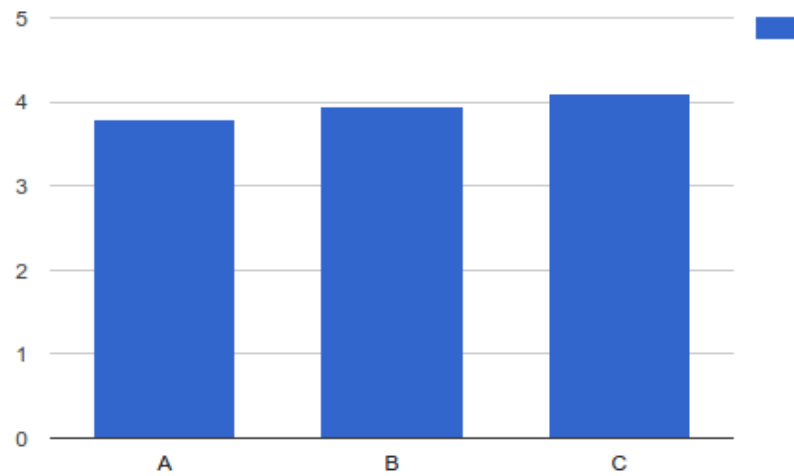


Figura 7.6 – Percepción de movilidad

- (A) La movilidad es una ventaja dentro de las aplicaciones móviles de mapas de seguridad.
- (B) La movilidad hace posible tener en tiempo real información de una ubicación.
- (C) Es importante acceder a las aplicaciones móviles de mapas de seguridad a cualquier hora y lugar.

■ Aspectos de desempeño esperado:

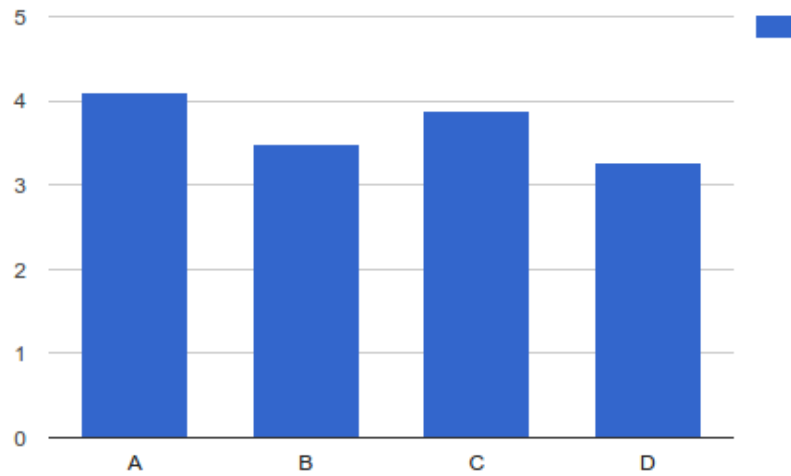


Figura 7.7 – Percepción de desempeño esperado

- (A) Encuentro que el internet móvil es útil en mi vida diaria.
- (B) Usar internet móvil incrementa mis oportunidades de lograr cosas que son importantes para mí.
- (C) Usar internet móvil me ayuda a completar cosas más rápido.
- (D) Usar internet móvil incrementa mi productividad.

■ Aspectos de actitud:

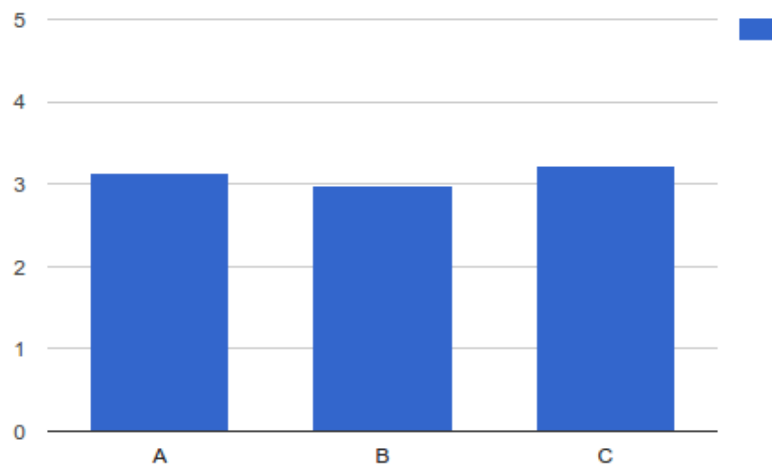


Figura 7.8 – Percepción de actitud

- (A) Pienso que las aplicaciones móviles de mapas de seguridad son más importantes que otras aplicaciones.
- (B) Resulta más fácil o mejor para mí usar las aplicaciones móviles de mapas de seguridad frente a otras aplicaciones móviles o servicios de información geográfica.
- (C) Desde mi punto de vista, es muy conveniente usar aplicaciones móviles de mapas de seguridad, en lugar de otras aplicaciones móviles o servicios de información geográfica.

■ Aspectos estado de flujo:

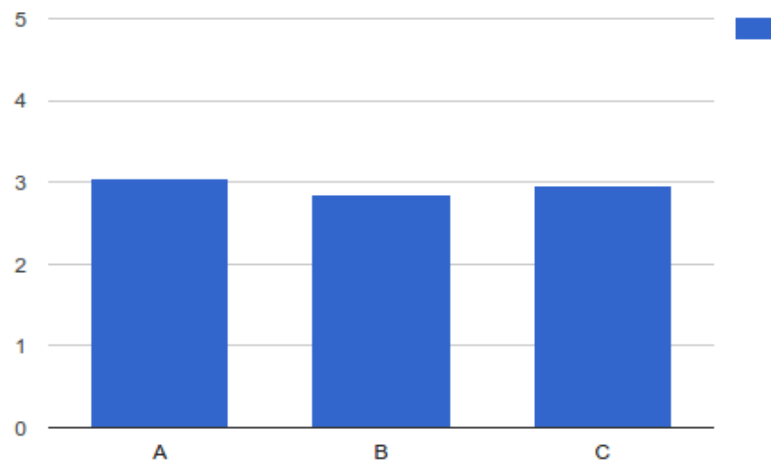


Figura 7.9 – Percepción estado de flujo

- (A) Cuando usé aplicaciones móviles de mapas de seguridad, estuve muy concentrado.
- (B) Estoy convencido que estoy dentro de un mundo diferente cuando uso aplicaciones móviles de mapas de seguridad.
- (C) Cuando uso aplicaciones móviles de mapas de seguridad, tengo pleno control sobre estas.

■ Otros aspectos:

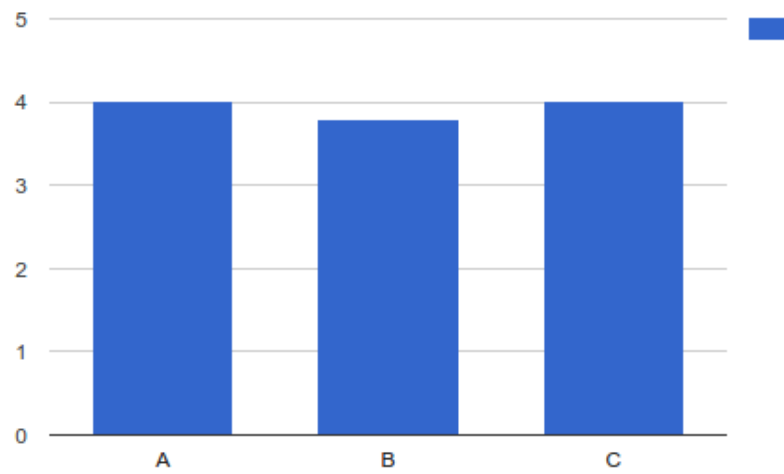


Figura 7.10 – Percepción de otros aspectos

- (A) Intentaré seguir usando internet móvil en el futuro.
- (B) Siempre trataré de usar internet móvil en mi vida diaria.
- (C) Planeo seguir usando internet móvil de manera frecuente.

Capítulo 8

Conclusiones

8.1. Conclusiones Generales

Al finalizar el desarrollo de este proyecto, y revisar los objetivos propuestos inicialmente, es posible darse cuenta que se cumplieron cada uno de ellos, de lo que se puede concluir:

- Se logró identificar la intención de uso de la aplicación móvil VersusCrime. Por medio del modelo TAM (Technology Acceptance Model) se realizó una encuesta de preferencia y de percepción. En el Anexo G se puede visualizar los resultados con mayor detalle.
- Se establecieron mecanismos y métodos eficientes para el desarrollo de la aplicación móvil, con un enfoque hacia la seguridad ciudadana, uno de ellos es la obtención del Plan Cuadrante de acuerdo a la geolocalización del usuario.
- Se evaluaron distintas herramientas, y se seleccionaron las más adecuadas para el desarrollo de la aplicación, con el fin de agilizar los procesos internos. La principal herramienta es Retrofit [Anexo B], que permite realizar peticiones REST hacia el servidor en un menor tiempo. Otra herramienta es Laravel [Anexo F], framework que permite desarrollar una plataforma web con un elevado estándar de seguridad, además facilita la estructuración de la plataforma con una arquitectura

MVC(Modelo-Vista-Controlador).

- Se seleccionó la herramienta provista por Google Material Design, que es la más adecuada para otorgar una mejor experiencia usuaria.
- Se habilitó una plataforma básica, para que las organizaciones que cuentan con registros propios, puedan alimentar el sistema como también, nutrirse de ella. Además proporcionar indicadores y mapas de calor para la toma de decisiones en las empresas u organizaciones suscrita al sistema.

8.2. Valores del Sistema

- Registros de usuarios en los primeros 3 meses (Marzo - Abril 2015):



Figura 8.1 – Registros de usuarios en los primeros 3 meses

Luego de ser lanzada la aplicación móvil al PlayStore, se observa que el día 4 del tercer mes, aumentó considerablemente las descargas como también, el registro de usuarios. Esto se debió principalmente a la publicidad en los medios, ver Anexo A. Actualmente la aplicación ha sido descargada por **+7500** personas, y de ellas un total de **2161** se han registro al sistema. De las cuales **1523** son hombres y **638** mujeres.

- Distribución de los delitos registrados por el sistema a la fecha:

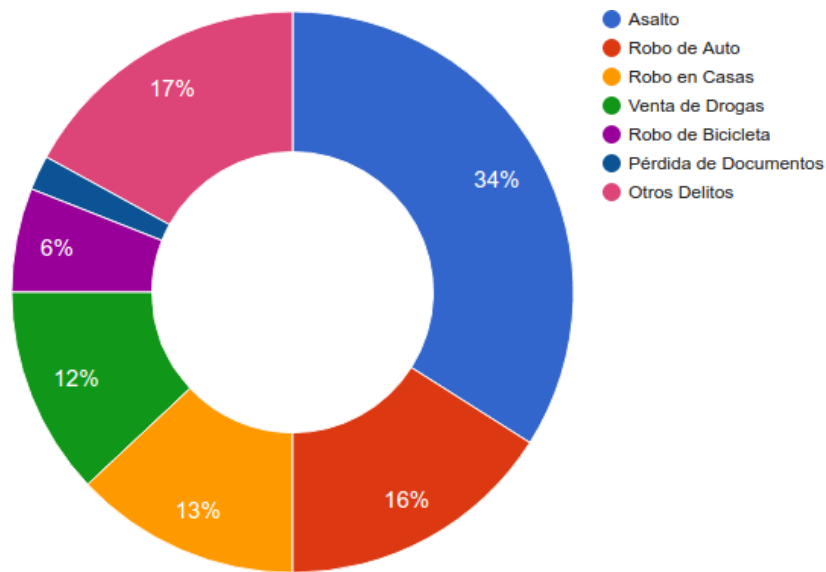


Figura 8.2 – Distribución de los delitos registrados por el sistema

Se observa que el principal delito reportado por los usuarios de VersusCrime es el **Asalto**, luego el **Robo de Auto** y **Robo en Casa**.

8.3. Trabajo futuro

La aplicación VersusCrime está desarrollada en una plataforma sujeta a constantes cambios y actualizaciones, uno de los trabajos futuros radica en la investigación sobre nuevas bibliotecas, con el fin de mejorar constantemente el funcionamiento de la aplicación.

Además se puede realizar los siguientes trabajo futuro con la finalidad de mejorar la aplicación:

- **Aplicación web:**

Brindar la posibilidad de una interfaz web con más funcionalidades para las empresas u organizaciones que deben velar por la seguridad social.

- **Desarrollar la versión iOS de VersusCrime:**

Para abarcar una mayor cantidad del mercado objetivo, es necesario desarrollar la segunda plataforma más utilizada en el mundo iOS.

- **Consumo de batería:**

Es un problema inherente el uso de geolocalización con GPS y es necesario hacer pruebas e investigar nuevas herramientas, que optimicen el consumo de energía en el proceso de obtención de la ubicación, así poder disminuir considerablemente el consumo en el dispositivo móvil.

- **Notificaciones push:**

Una de las posibles funcionalidades arrojada por el modelo TAM, es poder establecer algún grado de comunicación con el resto de los usuarios del sistema, esto se resolvería mediante la incorporación de la opción de llamada telefónica. Una mejor alternativa son las notificaciones push, por medio de grupos de chat creados desde la aplicación, puedan enviar un mensaje su estado o comunicarse con los demás usuarios.

Bibliografía

- [1] Memorias Multidisciplinarias. Fortalecimiento del Desarrollo de Competencias Transversales en la Formación del Profesional USM - Foco Q1 Q2 [en línea]. <<http://competenciastransversales.usm.cl/>>. [Consulta: 16 de Octubre del 2016].
- [2] SecuritySat. Seguridad para hogares y empresas [en línea]. <<http://www2.securitysat.cl/>>. [Consulta: 14 de Marzo del 2016].
- [3] Square. Empresa que desarrolla bibliotecas open source para aplicaciones móviles [en línea]. <<http://square.github.io/>>. [Consulta: 14 de Marzo del 2016].
- [4] Google Inc. Material design [en línea]. <<http://www.google.com/design/spec/material-design/introduction.html>>. [Consulta: 14 de Marzo del 2016].
- [5] SigWeb El Portal de los Expertos en Prevención de Riesgos de Chile [en línea]. <<http://www.sigweb.cl/biblioteca/MercadoSP.pdf>>. [Consulta: 15 de Marzo del 2016].
- [6] Paz Ciudadana. Balance delincuencia en chile [en línea]. <<http://www.pazciudadana.cl/category/publicaciones/balance-delincuencia-en-chile/>>. [Consulta: 17 de Marzo del 2016].
- [7] El Android Libre Los Padres de Android bloomberg[en línea]. <<http://www.elandroidelibre.com/2015/03/los-padres-de-android.html>>. [Consulta: 2 de Abril del 2016].
- [8] Ben Elgin. Google buys android for its mobile arsenal. bloomberg businessweek [en línea]. <<http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>>. [Consulta: 3 de Abril del 2016].
- [9] SubTEL Presentacion Final Sexta Encuesta 2015. [en línea]. <http://www.subtel.gob.cl/wp-content/uploads/2015/04/Presentacion_Final_Sexta_Encuesta_vers_16102015.pdf>. [Consulta: 20 de Abril del 2016].
- [10] The Atlantic. The day google had to start over on android [en línea]. <<http://www.theatlantic.com/technology/archive/2013/12/>>

- the-day-google-had-to-start-over-on-android/282479/>. [Consulta: 22 de Abril del 2016].
- [11] Open Handset Alliance. Industry leaders announce open platform for mobile devices [en línea]. <http://www.openhandsetalliance.com/press_110507.html>. [Consulta: 22 de Abril del 2016].
- [12] Androideity. Arquitectura de android [en línea]. <<http://androideity.com/2011/07/04/arquitectura-de-android/>>. [Consulta: 25 de Abril del 2016].
- [13] Androideric. Arquitectura de android [en línea]. <<http://androideric.blogspot.cl/2013/02/13-arquitectura-de-android.html>>. [Consulta: 5 de Mayo del 2016].
- [14] IDC International Data Corporation [en línea]. <<https://www.idc.com/>>. [Consulta: 7 de Mayo del 2016].
- [15] comScore Compañía de Investigación de Marketing en Internet [en línea]. <<http://www.comscore.com/es1/>>. [Consulta: 10 de Mayo del 2016].
- [16] AppAnnie Compañía de Inteligencia de Aplicaciones Móviles [en línea]. <<https://www.appannie.com/>>. [Consulta: 15 de Mayo del 2016].
- [17] Flurry Compañía de Análisis en Aplicaciones Móviles [en línea]. <<http://flurrymobile.tumblr.com>>. [Consulta: 4 de Junio del 2016].
- [18] INE Instituto Nacional de Estadísticas [en línea]. <<http://www.ine.cl>>. [Consulta: 8 de Junio del 2016].
- [19] SubTEL Sector Telecomunicaciones Cierre 2015 [en línea]. <http://www.subtel.gob.cl/wp-content/uploads/2015/04/PPT_Series_DICIEMBRE_2015_V5.pdf>. [Consulta: 16 de Junio del 2016].
- [20] eMarketer Investigación de Mercado [en línea]. <<http://www.emarketer.com>>. [Consulta: 25 de Junio del 2016].
- [21] Statista Portal de Estadísticas para Datos de Mercado [en línea]. <<http://www.statista.com>>. [Consulta: 3 de Julio del 2016].
- [22] Resultados Modelo TAM Technology Acceptance Model [en línea]. <<https://es.surveymonkey.net/results/SM-2T8F87NJ/>>. [Consulta: 12 de Agosto del 2016].
- [23] POJO Acrónimo de Plain Old Java Object [en línea]. <http://es.wikipedia.org/wiki/Plain_Old_Java_Object>. [Consulta: 4 de Febrero del 2016].
- [24] GRADLE Gradle [en línea]. <<https://gradle.org/>>. [Consulta: 4 de Febrero del 2016].

- [25] ECLIPSE Eclipse [en línea]. <<https://eclipse.org/>>. [Consulta: 4 de Febrero del 2016].
- [26] ANDROID STUDIO Android Studio [en línea]. <<https://developer.android.com/sdk/index.html>>. [Consulta: 4 de Febrero del 2016].
- [27] MAVEN Maven [en línea]. <<https://maven.apache.org/>>. [Consulta: 4 de Febrero del 2016].
- [28] MATÍAS DUARTE Matías Duarte [en línea]. <https://es.wikipedia.org/wiki/Mat%C3%ADas_Duarte>. [Consulta: 5 de Marzo del 2016].
- [29] LARAVEL Laravel [en línea]. <<http://laravel.com/>>. [Consulta: 10 de Marzo del 2016].

Anexo A

VersusCrime en los medios de comunicaciones

Las Últimas Noticias / Lunes 13 de abril de 2015

EMPLEO Y EDUCACIÓN 43

Mapa delictual

Dos inquietudes le gatillaron una idea a Miguel Ángel Alvar, ingeniero comercial de la USM: 1) Los constantes asaltos que sufría un amigo que circulaba a diario en bicicleta en Valparaíso; 2) la inquietud de no saber sobre qué lugares eran más peligrosos, para evitar que le robaran su propia bici. Entonces se le prendió la ampollita: presentó su "mapa virtual del delito" en un concurso de emprendimiento de la universidad y recibió \$400.000 para desarrollarlo. Sumó a Manuel Ramírez (estudiante de Ingeniería Civil Telemática de la USM) y José Ignacio Álvarez (Diseño Gráfico, Duoc UC), quienes le dieron forma a VersusCrime.



Creada por universitarios, despliega en el celular un completo mapa del delito

App chilena muestra dónde es más probable que te asalten

VersusCrime cubre un radio de 10 km para conocer de antemano los lugares donde los propios usuarios han reportado robos.

JAVIER DE LA RIVERA

Enfoque local

Los creadores de VersusCrime tienen claro que su plan de negocios debe vincularse a organismos ciudadanos y de seguridad. La empresa SecuritySat ya mostró interés en la iniciativa. "Nos están asesorando sobre cómo se pueden conciliar sus necesidades con las funciones de esta aplicación. La información delictual recaudada por los usuarios les informará de los lugares con más delitos, para que puedan destinar eficientemente sus recursos de patrullaje", explica Miguel Ángel. Además, están en conversaciones con municipalidades, agencias de turismo y empresas de seguridad de automóviles. Ahora buscan apoyo financiero. "Trabajaremos duro para aportar a la tranquilidad de las familias fortaleciendo esta tecnología", cierra Alvar.



¿Cómo funciona?

VersusCrime está disponible gratis en Android (próximamente en iOS y Windows Phone) y sólo requiere el GPS del smartphone para funcionar. Al abrir la aplicación se despliega un mapa con todos los delitos informados en un radio de 10 km a la redonda; al registrarse se pueden reportar delitos. "Tenemos disponibles asaltos, robo de casas, bicicletas y automóviles, objetos perdidos, venta de drogas u otros, que se deben especificar en el mapa", detalla Miguel Ángel. Luego hay que redactar una breve descripción y asignar la hora del suceso. Así todos los usuarios -inscritos o no- visualizarán el reporte. "La idea es estar conscientes de los focos de delincuencia en cada barrio, para prevenir y actuar a través de la asociación entre amigos, colegas o juntas de vecinos", explica.

Nuevas funcionalidades

En menos de 5 meses, esta plataforma antidelitos ha alcanzado un gran desarrollo. El equipo creativo proyecta a mitad de año tener disponibles nuevas funciones:

- Base de contactos:** todos los números de seguridad del sector donde se active la app (Plan Cuadrante y seguridad ciudadana municipal), para tener pronta reacción en caso de emergencias.
- Personalización:** notificaciones por lugar de interés, clasificaciones de usuarios según su aporte a la comunidad y facilidad para organizar grupos comunitarios de seguridad.
- Búsqueda de objetos perdidos:** posibilidad de ofrecer recompensas, indicando el último lugar donde se vio una billetera o documento valioso.

Figura A.1 – Fuente: Lun.com 13-04-2015.



Figura A.2 – La RED.

Anexo B

Retrofit

B.0.1. ¿Qué es Retrofit?

Retrofit es un cliente REST para Android y Java, desarrollada por Square, muy simple y fácil de aprender.

Esta biblioteca nos permite hacer peticiones REST de una forma sencilla, rápida y elegante, definiendo una simple interfaz con anotaciones. Además por defecto soporta entregar la respuesta de forma síncrona, asíncrona con un callback, y asíncrona mediante reactive, por lo que es bastante versátil.

Permite hacer peticiones GET, POST, PUT, PATCH, DELETE y HEAD , gestionar diferentes tipos de parámetros y parsear automáticamente la respuesta a un POJO [23].

La gran mayoría de las aplicaciones actuales necesitan interactuar con alguna API REST, y para eso hay diversas soluciones.

- Http Request + AsyncTask (Forma nativa).
- Volley por Google.
- Retrofit por Square.

	1 Petición	7 Peticiones	25 Peticiones
AsyncTask	941 ms	4.539 ms	13.957 ms
Volley	560 ms	2.202 ms	4.275 ms
Retrofit	312 ms	889 ms	1.059 ms

Tabla B.1 – Duración de las peticiones REST. Fuente: Elaboración Propia.

Como se puede observar, Retrofit es la mejor alternativa en cuanto rendimiento, ya que destaca por su gran velocidad.

B.0.2. Primeros Pasos

El primer paso es añadir la dependencia al proyecto via GRADLE [24]:

```
compile 'com.squareup.retrofit:retrofit:1.9.0'
```

A continuación se necesitan 3 clases para poder usar Retrofit:

- POJO: Representación del Json proveniente de la API REST.
- Interface (Service): Definición de las diferentes llamadas a la API REST.
- RestAdapter: Cliente que se usa para realizar y configurar las peticiones. Retrofit permite hacer las peticiones de manera sincrónica o asincrónica.

Anexo C

Android Studio

Se comienza cambiando el entorno de desarrollo de Eclipse [25] hacia Android Studio [26]. Eclipse es una plataforma de desarrollo, diseñada para ser extendida a través de plugins. Al no ser un IDE nativo para Android, esto conlleva a provocar errores de compilación durante el desarrollo.

Android Studio fue presentado como la alternativa oficial a Eclipse en Mayo de 2013. A pesar de que ya lleva unos años, fue durante este año que se lanzó la versión estable. Incorpora multitud de mejoras y ayudas en relación a Eclipse.

Una de las ventajas de Android Studio sobre Eclipse, es que utiliza Gradle [24], cuyas características son:

- Facilita reusar código y recursos.
- Facilita configurar, extender y personalizar el proceso.
- Facilita la distribución del código y por tanto trabajar en equipos.
- Gestiona las dependencias de una forma cómoda y potente (esta basado en Maven[27]).
- Permite compilar desde línea de comandos.
- Permite crear distintas versiones de la aplicación, por ejemplo para hacer una versión gratis y otra de pago.

C.0.3. Primeros Pasos: Migrando desde Eclipse a Android Studio

Exportación desde Eclipse.

1. Actualizar el plugin ADT(Android Development Tools) a la ultima versión.

2. En Eclipse, seleccionar **File > Export**.
3. Abrir el apartado Android, y seleccionar la opción, **Generate Gradle build files**.
4. Elegir el proyecto y pulsar **Finish**.

Importación a Android Studio.

1. Cerrar todos los proyectos.
2. Hacer click en **Import Project**.
3. Buscar el proyecto en la ruta del Workspace de Eclipse.
4. Seleccionar el archivo build.gradle y pulsar **OK**.
5. Seleccionar **Use gradle wrapper** y pulsar **OK**.

Con estos 2 procedimientos se tiene el proyecto migrado, desde el entorno de desarrollo Eclipse, a Android Studio.

Anexo D

Fragment

D.0.4. ¿Por qué utilizar Fragment?

La necesidad de usar fragmentos nace con la versión 3.0 (API 11) de Android, debido a los múltiples tamaños de pantalla que aparecieron en el mercado y a la capacidad de orientación de la interfaz (Landscape y Portrait). Estas características necesitaban dotar a las aplicaciones Android de la capacidad para adaptarse y responder a la interfaz de usuario sin importar el dispositivo.

Un fragmento es una sección modular de interfaz de usuario embebida dentro de una actividad anfitriona. Ésta permite versatilidad y optimización de diseño. Se trata de miniactividades contenidas dentro de una actividad principal, manejando su propio diseño y ciclo de vida.

La principal ventaja que se tiene con respecto al modelo Vista-Activity es que dentro de un solo Activity se pueden lanzar varios Fragments que interactúan entre ellos a través de su Activity, como se visualiza en la Figura D.1. De este modo se ahorra de estar cambiando entre distintas actividades, con el consecuente gasto de memoria.

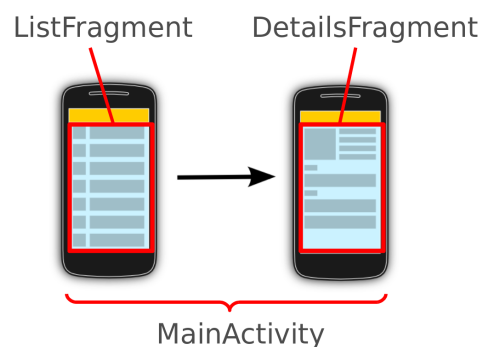


Figura D.1 – Fragments dentro de un activity.

D.0.5. Primeros Pasos

Crear una clase que extienda de Fragment:

```
//MyFragment.java
public class MyFragment extends Fragment{

    public static final String TAG = "MyFragment";
    private FragmentIterationListener mCallback = null;
    public interface FragmentIterationListener{
        public void onFragmentIteration(Bundle parameters);
    }

    public static MyFragment newInstance(Bundle arguments){
        MyFragment f = new MyFragment();
        if(arguments != null){
            f.setArguments(arguments);
        }
        return f;
    }

    //Constructor
    public MyFragment(){

    }

    /**** Aqui van los metodos propios de MyFragment ****/

    //El fragment se ha adjuntado al Activity
    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
    }

    //El Fragment ha sido creado
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    //El Fragment va a cargar su layout, el cual debemos especificar
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
```

```

        Bundle savedInstanceState) {
            super.onCreateView(inflater, container, savedInstanceState);
            View v = inflater.inflate(R.layout.my_fragment_view, container,
                false);
            return v;
        }

        // La vista de layout ha sido creada y ya esta disponible
        @Override
        public void onViewCreated(View view, Bundle savedInstanceState) {
            super.onViewCreated(view, savedInstanceState);
        }

        //La vista ha sido creada y cualquier configuracion guardada esta cargada
        @Override
        public void onViewStateRestored(Bundle savedInstanceState) {
            super.onViewStateRestored(savedInstanceState);
        }

        //El Activity que contiene el Fragment ha terminado su creacion
        @Override
        public void onActivityCreated(Bundle savedInstanceState) {
            super.onActivityCreated(savedInstanceState);
        }

        //El Fragment ha sido quitado de su Activity y ya no esta disponible
        @Override
        public void onDetach() {
            super.onDetach();
        }
    }
}

```

Usando esta estructura de instanciación, cuando se quiera añadir el Fragment a un Activity, solo se debe hacer lo siguiente:

```
//MyActivity.java
public class MyActivity extends Activity{

    @Override
    protected void onCreate(Bundle onSaveInstanceState){
        super.onCreate(onSaveInstanceState);
        String id = "IdQueNecesitaMyFragment";
        Bundle arguments = new Bundle();
        arguments.putString("id", id);
        MyFragment fragment = MyFragment.newInstance(arguments);
        FragmentTransaction ft = getFragmentManager().beginTransaction();
        ft.replace(android.R.id.content, fragment, MyFragment.TAG);
        ft.commit();
    }
}
```

Anexo E

Material Design

E.0.6. ¿Qué es Material Design?

Durante la conferencia que da Google cada año, en Google I/O 2014, se presentaron muchas novedades siendo una de ellas este nuevo diseño. Material Design es un concepto, una filosofía, unas pautas enfocadas al diseño utilizado en Android, pero también en la web y en cualquier plataforma. El encargado de crear Material Design y máximo responsable de diseño en Google es el chileno Matías Duarte [28].

Históricamente Android ha tenido diseño Holo (Normativa de diseño previa a Material Design), a partir de ahora el nuevo estilo y la tendencia en Android es Material Design.

E.0.7. ¿En qué se basa Material Design?

Material Design recibe su nombre por estar basado en objetos materiales. Piezas colocadas en un espacio (lugar) y con un tiempo (movimiento) determinado. Es un diseño donde la profundidad, las superficies, los bordes, las sombras y los colores juegan un papel principal.

Precisamente este diseño basado en objetos es una manera de intentar aproximarse a la realidad, algo que en un mundo donde todo es táctil y virtual es difícil. Material Design quiere guiarse por las leyes de la física, donde las animaciones sean lógicas, los objetos se superpongan pero no puedan atravesarse el uno al otro.

¿Cómo se traslada esto a Android?. Básicamente delimitando el tipo de menús, los botones y los tipos de imágenes a elegir, como se visualiza en la imagen a continuación.

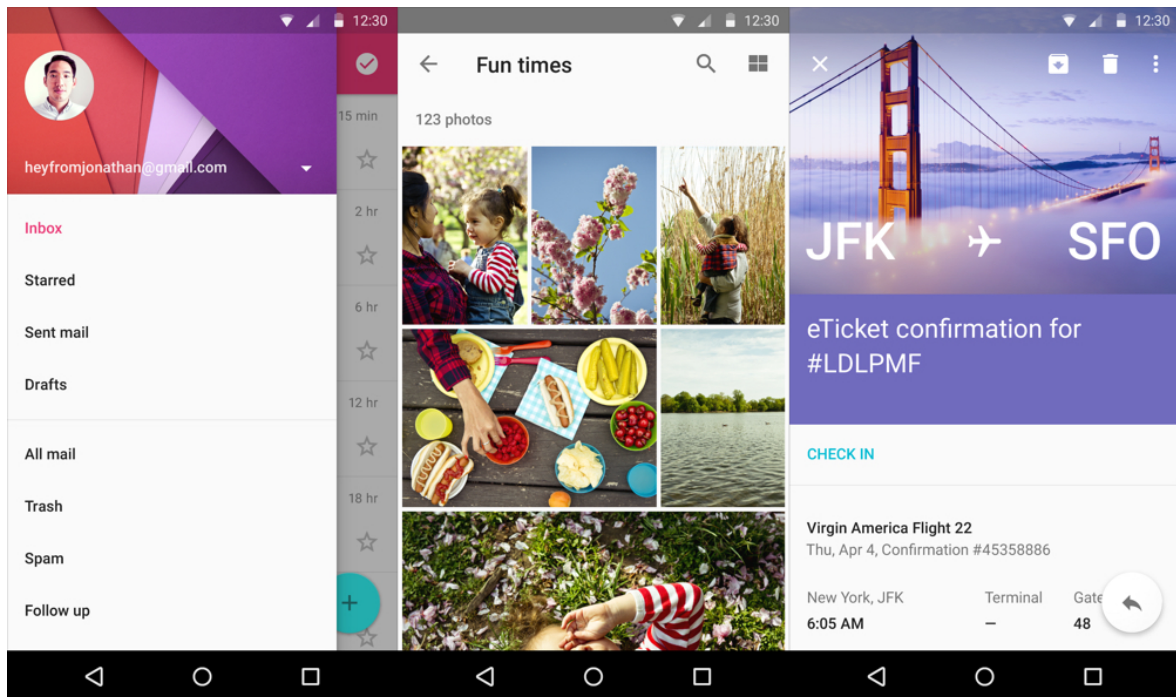


Figura E.1 – Vistas Material Design.

E.0.8. Elementos ordenados e imágenes claras

Material Design es un diseño con una tipografía clara, casillas bien ordenadas, colores e imágenes llamativos. Estas ideas ya se aplican en muchos diseños, pero en Material Design Google ha creado unas normas muy claras de como llevarlo a la práctica.

E.0.9. Luz y sombras dan sensación de jerarquía

Uno de los elementos clave es la luz y las sombras. Una iluminación realista proporciona indicios de cómo se comportará un elemento y en qué nivel se encuentra. Imagínense un cuadrado que tiene su sombra sobre otro cuadrado, el cerebro humano entiende que el que recibe la sombra es el que está debajo. En la Fig. 1 se observa en la primera vista donde el menú de la izquierda además de tener un fondo más claro proyecta su sombra sobre el panel principal, dando la sensación de estar delante.

E.0.10. El movimiento es la mejor forma de guiar al usuario

El movimiento es otro elemento clave, por ejemplo un objeto que parpadea significa que está llamando tu atención, un elemento que se expande es que se acaba de abrir. Todos estos movimientos se crean en una dirección determinada, nuevamente explicada en la guía de Material Design(<https://developer.android.com/design/index.html>).

Anexo F

Laravel

F.0.11. ¿Qué es Laravel?

Laravel es un Framework nuevo programado en PHP, utilizando la arquitectura MVC(Modelo Vista Controlador). Laravel maneja una sintaxis expresiva y elegante, facilitando las tareas comunes, como la autenticación, enrutamiento, sesiones y caché.

Proporciona, potentes herramientas accesibles necesarias para construir grandes aplicaciones robustas. Laravel se puede utilizar para aplicaciones de nivel empresarial enormes ó simples, lo que significa que es perfectamente adecuado para todos los tipos y tamaños de proyectos.

F.0.12. Ventajas al emplear Laravel en el desarrollo web

Trabajando con Laravel como framework base en el desarrollo de los proyectos web se obtiene:

- Reducción de costos y tiempos en el desarrollo y mantenimiento.
- Curva de aprendizaje baja en comparación con otros framework PHP.
- Flexible y adaptable no solo al MVC Tradicional sino que para reducir código propone usar Routes with.
- Abundante documentación en el sitio oficial (<http://laravel.com/docs>).
- Posee una amplia comunidad y foros.
- Es modular y con una amplia gama de sistemas de paquetes y drivers con el que se puede extender la funcionalidad de forma fácil, robusta y segura.
- Hace que el manejo de la data no sea complejo. Mediante Eloquent, que es un ORM basado en el patrón active record, la interacción con las bases de datos es totalmente orientada a objetos, facilitando la migración de nuestros datos de

una forma fácil y segura. Además permite la creación de consultas robustas y complejas.

- Facilita el manejo de ruteo de nuestra aplicación como así también, la generación de url amigables.
- El sistema de plantillas Blade de Laravel, trae consigo la generación de mejoras en la parte de presentación de la aplicación, como la generación de plantillas más simples y limpias en código. Además incluye un sistema de caché que las hace más rápidas, lo que mejora el rendimiento de la aplicación.
- Cuenta con una herramienta de interfaces de líneas de comando llamada Artisan. Provee comandos útiles para el desarrollo de la aplicación. Está impulsada por el poderoso componente Console de Symfony.

F.0.13. ¿Por qué utilizar Laravel?

Es un framework diseñado para trabajar con simplicidad, flexibilidad y robustez. Lo que hizo que rápidamente se pusiera de moda en la comunidad de desarrolladores web, siendo muy demandado. Además tiene una gran comunidad y su curva de aprendizaje es muy corta.

En varias encuestas realizadas a final del año 2014 en portales especializados como codegeekz.com, catswhocode.com y sitepoint.com. Se confirma que Laravel es el Framework PHP preferido por los desarrolladores. Laravel es potente, elegante y fácil de aprender.

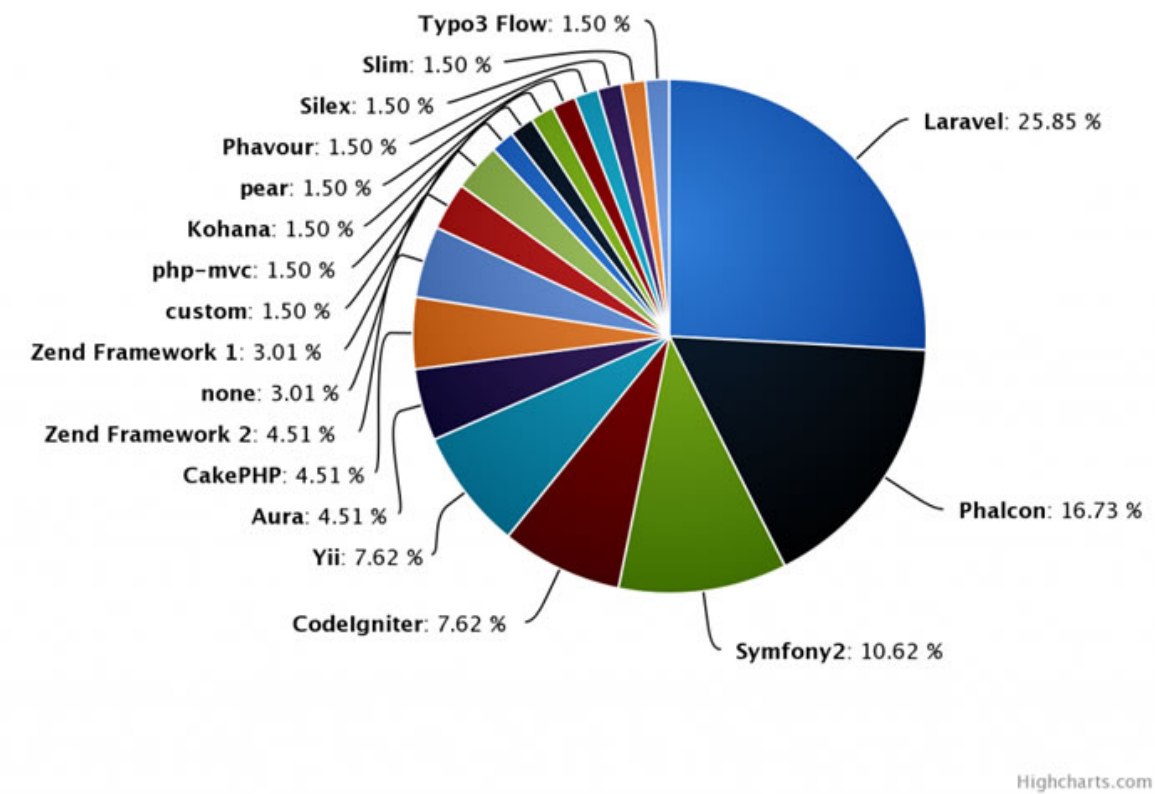


Figura F.1 – Porcentaje de preferencias

Anexo G

Resultados Modelo TAM

1. Evalúe los atributos que se presentan de acuerdo a su percepción, donde 1: No importante y 5: Muy importante.

	1	2	3	4	5
Identificar mi ubicación automáticamente en el mapa (geolocalización)	6,12 %	3,06 %	19,39 %	16,33 %	54,10 %
Disponer y poder llamar los números de plan cuadrante	2,04 %	0,00 %	10,20 %	21,43 %	66,33 %
Comunicarme por medio de grupos sobre hechos delictivos	7,22 %	8,25 %	30,93 %	30,93 %	22,68 %
Recibir notificaciones de delitos en lugares que transito y/o en lugares que deseo	1,03 %	7,22 %	11,34 %	34,02 %	46,39 %
Usar filtros en el mapa como el tipo de delito que deseo ver, antigüedad u horario (mañana, tarde o noche)	3,09 %	6,19 %	24,74 %	26,80 %	39,18 %
Visualizar rutas seguras por donde pueda transitar	2,06 %	3,09 %	10,31 %	31,96 %	52,58 %
Usar un botón de pánico para marcar que ha ocurrido un delito donde me encuentro y además llamar al plan cuadrante/carabineros	1,03 %	2,06 %	5,15 %	20,62 %	71,13 %
Publicar los delitos e información en redes sociales	11,34 %	17,53 %	26,80 %	24,74 %	19,59

	1	2	3	4	5
Tener la opción preestablecida de anónimo para registrar delitos	4,12 %	3,09 %	20,62 %	21,65 %	50,52 %
Mostrar la ubicación de carabineros en el mapa: comisarías, redes móviles, etc.	1,02 %	2,04 %	12,24 %	25,51 %	59,18 %
Eliminar publicaciones falsas o desactualizadas	1,03 %	5,15 %	9,28 %	15,46 %	69,07 %
Limitar la cantidad de denuncias diarias que uno puede realizar	16,49 %	12,37 %	22,68 %	23,71 %	24,74 %
Actualizar los registros delictivos en tiempo real	1,03 %	5,15 %	13,40 %	21,65 %	58,76 %
Generar descripción de los delin-cuentes	2,06 %	8,25 %	8,25 %	34,02 %	47,42 %

Tabla G.1 – Tabla con resultados pregunta número 1 Modelo TAM

2. A continuación, evalúe los siguientes aspectos de exactitud de la ubicación percibida que se presentan, de acuerdo a su percepción, donde 1: Totalmente en desacuerdo, y 5: Totalmente de acuerdo.

	1	2	3	4	5
Las aplicaciones móviles de mapas de seguridad siempre indican con exactitud la ubicación de eventos delictivos y regiones de interés.	4,08 %	16,33 %	48,98 %	22,45 %	8,26 %
Las aplicaciones móviles de mapas de seguridad siempre indican con exactitud la ubicación en tiempo real de eventos delictivos y regiones de interés frente a situaciones de movimiento de la zona visualizada.	4,08 %	24,49 %	59,18 %	4,08 %	8,16 %
Las aplicaciones móviles de mapas de seguridad entregan rutas eficientes y exactitud en los destinos a los que deseo ir, evitando rutas potencialmente peligrosas.	6,12 %	14,29 %	53,06 %	14,29 %	12,24 %

Tabla G.2 – Tabla con resultados pregunta número 2 Modelo TAM

3. A continuación, evalúe los siguientes aspectos de satisfacción que se presentan, de acuerdo a su percepción, donde 1: Totalmente en desacuerdo, y 5: Totalmente de acuerdo.

	1	2	3	4	5
En general, me encuentro satisfecho con las aplicaciones móviles de mapas de seguridad.	8,16 %	16,33 %	61,22 %	8,16 %	6,12 %
Las aplicaciones móviles de mapas de seguridad que actualmente utilizo cumplen con mis expectativas.	10,20 %	14,29 %	57,14 %	10,20 %	8,16 %
Recomiendo las aplicaciones móviles de mapas de seguridad a otras personas que intentan usar o comprar aplicaciones móviles.	18,37 %	6,12 %	53,06 %	14,29 %	8,16 %

Tabla G.3 – Tabla con resultados pregunta número 3 Modelo TAM

4. A continuación, evalúe los siguientes aspectos de calidad de servicio y visualización que se presentan, de acuerdo a su percepción, donde 1: Totalmente en

desacuerdo, y 5: Totalmente de acuerdo.

	1	2	3	4	5
Las aplicaciones móviles de mapas de seguridad y la información en pantalla satisfacen completamente mis necesidades.	10,20 %	16,33 %	55,10 %	6,12 %	12,24 %
No tengo problemas o restricciones en el uso de aplicaciones móviles de mapas de seguridad.	8,16 %	8,16 %	55,10 %	16,33 %	12,24 %
Las aplicaciones móviles de mapas de seguridad entregan servicios precisos y exactos de acuerdo a su propósito.	6,12 %	18,37 %	42,86 %	18,37 %	14,29 %
No tengo problemas o restricciones viendo la información en pantalla que exhiben las aplicaciones móviles de mapas de seguridad.	10,20 %	6,12 %	51,02 %	16,33 %	16,33 %

Tabla G.4 – Tabla con resultados pregunta número 4 Modelo TAM

5. A continuación, evalúe los siguientes aspectos de movilidad percibida que se presentan, de acuerdo a su percepción, donde 1: Totalmente en desacuerdo, y 5: Totalmente de acuerdo.

	1	2	3	4	5
La movilidad es una ventaja dentro de las aplicaciones móviles de mapas de seguridad.	4,08 %	6,12 %	28,57 %	28,57 %	32,65 %
La movilidad hace posible tener en tiempo real información de una ubicación.	2,04 %	6,12 %	24,49 %	30,61 %	36,73 %
Es importante acceder a las aplicaciones móviles de mapas de seguridad a cualquier hora y lugar.	2,04 %	8,16 %	18,37 %	20,41 %	51,02 %

Tabla G.5 – Tabla con resultados pregunta número 5 Modelo TAM

6. A continuación, evalúe los siguientes aspectos de desempeño esperado que se presentan, de acuerdo a su percepción, donde 1: Totalmente en desacuerdo, y 5: Totalmente de acuerdo.

	1	2	3	4	5
Encuentro que el internet móvil es útil en mi vida diaria.	8,16 %	2,04 %	18,37 %	14,29 %	57,14 %
Usar internet móvil incrementa mis oportunidades de lograr cosas que son importantes para mí.	10,20 %	14,29 %	24,49 %	18,37 %	32,65 %
Usar internet móvil me ayuda a completar cosas más rápido.	6,12 %	10,20 %	18,37 %	20,41 %	44,90 %
Usar internet móvil incrementa mi productividad.	16,33 %	12,24 %	26,53 %	18,37 %	26,53 %

Tabla G.6 – Tabla con resultados pregunta número 6 Modelo TAM

7. A continuación, evalúe los siguientes aspectos de actitud que se presentan, de acuerdo a su percepción, donde 1: Totalmente en desacuerdo, y 5: Totalmente de acuerdo.

	1	2	3	4	5
Pienso que las aplicaciones móviles de mapas de seguridad son más importantes que otras aplicaciones.	8,16 %	18,37 %	42,86 %	12,24 %	18,37
Resulta más fácil o mejor para mí usar las aplicaciones móviles de mapas de seguridad frente a otras aplicaciones móviles o servicios de información geográfica.	8,16 %	26,53 %	38,78 %	12,24 %	14,29 %
Desde mi punto de vista, es muy conveniente usar aplicaciones móviles de mapas de seguridad, en lugar de otras aplicaciones móviles o servicios de información geográfica.	6,12 %	20,41 %	36,73 %	18,37 %	18,37 %

Tabla G.7 – Tabla con resultados pregunta número 7 Modelo TAM

8. A continuación, evalúe los siguientes aspectos de estado de flujo que se presentan, de acuerdo a su percepción, donde 1: Totalmente en desacuerdo, y 5: Totalmente de acuerdo.

	1	2	3	4	5
Cuando usé aplicaciones móviles de mapas de seguridad, estuve muy concentrado.	10,20 %	12,24 %	53,06 %	12,24 %	12,24 %
Estoy convencido que estoy dentro de un mundo diferente cuando uso aplicaciones móviles de mapas de seguridad.	18,37 %	16,33 %	42,86 %	6,12 %	16,33 %
Cuando uso aplicaciones móviles de mapas de seguridad, tengo pleno control sobre estas.	10,20 %	16,33 %	51,02 %	12,24 %	10,20 %

Tabla G.8 – Tabla con resultados pregunta número 8 Modelo TAM

9. A continuación, evalúe los siguientes aspectos que se presentan, de acuerdo a su percepción, donde 1: Totalmente en desacuerdo, y 5: Totalmente de acuerdo.

	1	2	3	4	5
Intentaré seguir usando internet móvil en el futuro.	6,12 %	8,16 %	16,33 %	18,37 %	51,02 %
Siempre trataré de usar internet móvil en mi vida diaria.	8,16 %	10,20 %	20,41 %	16,33 %	44,90 %
Planeo seguir usando internet móvil de manera frecuente.	6,12 %	8,16 %	14,29 %	20,41 %	51,02 %

Tabla G.9 – Tabla con resultados pregunta número 9 Modelo TAM