



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

CTRL – TAXI, NUEVAS TECNOLOGÍAS, GESTIÓN DE TAXIS ANDROID

Leonel Enrique Rojas Chinchilla

María Soledad Lucero Castillo

Asesorado por el Ing. Ricardo Israel Mazariegos Castillo

Guatemala, abril de 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

CTRL – TAXI, NUEVAS TECNOLOGÍAS, GESTIÓN DE TAXIS ANDROID

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

LEONEL ENRIQUE ROJAS CHINCHILLA

MARÍA SOLEDAD LUCERO CASTILLO

ASESORADO POR EL ING. RICARDO ISRAEL MAZARIEGOS CASTILLO

AL CONFERÍRSELES EL TÍTULO DE

INGENIEROS EN CIENCIAS Y SISTEMAS

GUATEMALA, ABRIL DE 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympto Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Walter Rafael Véliz Muñoz
VOCAL V	Br. Sergio Alejandro Donis Soto
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADORA	Inga. Floriza Ávila Pesquera
EXAMINADORA	Inga. Sonia Yolanda Castañeda Ramírez
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

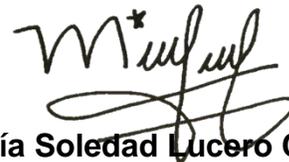
En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presentamos a su consideración nuestro trabajo de graduación titulado:

CTRL – TAXI, NUEVAS TECNOLOGÍAS, GESTIÓN DE TAXIS ANDROID

Tema que nos fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha enero de 2013.



Leonel Enrique Rojas Chinchilla



María Soledad Lucero Castillo

Guatemala 10 de octubre de 2013

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

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que he revisado el Trabajo de Graduación de los estudiantes **Leonel Enrique Rojas Chinchilla** identificado con número de carnet **200819162**, y **María Soledad Lucero Castillo** identificada con número de carnet **200818880**, titulado: "**CTRL – Taxi, Nuevas Tecnologías, Gestión de Taxis Android**", y a mi criterio el mismo cumple con los objetivos propuestos, según el protocolo.

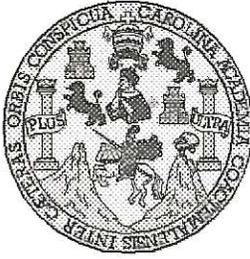
Sin otro particular, me suscribo de usted.

Atentamente,



Ricardo Mazariegos
Ingeniero en Ciencias y Sistemas
Colegiado No.
Asesor de Trabajo de Graduación

Ricardo Israel Mazariegos Castillo
Ingeniero en Ciencias y Sistemas
Colegiado No.: 12451



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

Guatemala, 29 de Enero de 2014

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

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **LEONEL ENRIQUE ROJAS CHINCHILLA**, con carné **2008-19162**, y **MARÍA SOLEDAD LUCERO CASTILLO**, con carné **2008-18880**, titulado: **"CTRL – TAXI, NUEVAS TECNOLOGIAS, GESTION DE TAXIS ANDROID"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

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

Atentamente,


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



E
S
C
U
E
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación CTRL-TAXI, NUEVAS TECNOLOGÍAS, GESTIÓN DE TAXIS ANDROID”, realizado por los estudiantes LEONEL ENRIQUE ROJAS CHINCHILLA Y MARÍA SOLEDAD LUCERO CASTILLO, aprueba el presente trabajo y solicita la autorización del mismo.

“ID Y ENSEÑAD A TODOS”

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



Guatemala, 31 de marzo 2014



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **CTRL – TAXI, NUEVAS TECNOLOGÍAS, GESTIÓN DE TAXIS ANDROID**, presentado por los estudiantes universitarios: **Leonel Enrique Rojas Chinchilla** y **María Soledad Lucero Castillo**, procede a la autorización para la impresión del mismo.

IMPRÍMASE.

Ing.  Olympos Paz Recinos
Decano



Guatemala, abril de 2014

/cc

ACTO QUE DEDICO A:

Dios

Por poner en mi vida tantas oportunidades y la guía para saberlas aprovechar, por la vida y por las fuerzas para lograr mis metas.

Mis padres

Luis Rojas y Marisol Chinchilla, por servirme de soporte durante mi vida, por luchar junto a mí en todo momento, por hacerme sentir querido y apoyado sin importar las circunstancias.

Mis hermanos

Natalia y Juan Pablo Rojas Chinchilla, por su amistad y soporte en momentos difíciles de mi vida.

Mi novia

María Soledad Lucero Castillo, por ser mi compañera de estudios, mi soporte, mi amiga, la persona que me levanta en los momentos de flaqueza.

Leonel Enrique Rojas Chinchilla

ACTO QUE DEDICO A:

Dios	Por la vida maravillosa con la que me ha privilegiado y ser la fuente de todo lo bueno en ella.
Mi madre	Ninet Castillo, por demostrarme siempre su amor incondicional y hacerme la persona que soy.
Mi novio	Leonel Rojas, por proporcionarle a mi vida luz, amor y alegría.
Mi hermana	Cindi Lucero, por ser mi amiga y demostrarme su apoyo y solidaridad en cada fase de mi vida.

María Soledad Lucero Castillo

AGRADECIMIENTOS A:

Dios	Por bendecirme con tanta abundancia en esta vida, dándome una familia ejemplar.
Mis padres	Por darme una formación sin igual, tanto en lo académico como en lo moral, por su amor y apoyo incondicional durante mi vida.
Mi hermano	Por darme esa valiosa compañía en toda la carrera y comprender mi ocupación.
Mi novia	Por su amor, apoyo, comprensión y soporte continuo durante toda la carrera, sin su apoyo hubiera sido imposible este logro.
Mis compañeros	Por siempre estar listos para levantarme en tantos momentos difíciles, por luchar junto a mí y tener siempre esta meta presente, por apoyarme y acompañarme a lo largo de la carrera.
Universidad San Carlos de Guatemala	Por formar profesionales íntegros y capaces, por darme la oportunidad de formarme y crecer en la vida.

Leonel Enrique Rojas Chinchilla

AGRADECIMIENTOS A:

Dios	Por brindarme fortaleza, perseverancia y sabiduría en el trayecto de mi carrera y de toda mi vida.
Mi madre	Ninet Castillo, por su apoyo incondicional, sin el cual no hubiera podido lograr esta meta.
Mi novio	Leonel Rojas, por lograr esta meta a mi lado y permanecer siempre en todos los momentos felices y tristes a lo largo de mi carrera.
Mi hermana	Cindi Lucero, por su ayuda y comprensión en los momentos que lo necesitaba.
Mis amigos	Por haber compartido éxitos, fracasos, alegrías, desvelos, enojos y risas juntos.
Facultad de Ingeniería	Por proporcionarme conocimientos y experiencias para la formación de mi vida profesional.

María Soledad Lucero Castillo

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO	IX
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. HISTORIA DE LOS DISPOSITIVOS MÓVILES	1
1.1. Concepto de dispositivo móvil	1
1.2. Categorías de los dispositivos móviles	2
1.3. Teléfonos inteligentes.....	3
1.4. <i>Tablets</i>	4
2. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO EN GUATEMALA	9
2.1. Modelo de Aceptación Tecnológica (TAM)	10
2.1.1. Concepto	13
2.1.2. Definición de variables de TAM	14
2.1.3. Limitaciones de TAM	15
2.2. Teoría y la relación con la tecnología escogida.....	15
2.2.1. Percepción de fácil uso.....	15
2.2.2. Percepción de utilidad	16
2.2.3. Variables externas	16
2.2.3.1. El internet.....	16
2.2.4. Intención de uso (aceptación).....	17
2.2.5. Uso actual.....	17

3.	IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN QUE LA APLICACIÓN REALIZARÁ	19
3.1.	Antecedentes	19
3.2.	Mercado objetivo (usuarios de la aplicación)	20
3.3.	<i>Benchmarking</i> de la aplicación.....	21
3.4.	<i>Benchmarking</i> aplicado a Ctrl - Taxi.....	22
3.4.1.	Organizaciones líderes.....	22
3.4.2.	Problemas de las organizaciones líderes	24
3.4.3.	Mejores prácticas de las organizaciones líderes	24
3.5.	Comparación entre aplicaciones	25
3.5.1.	Escala de calificación	26
4.	DISEÑO DE LA APLICACIÓN BAJO LA NECESIDAD IDENTIFICADA	27
4.1.	Prototipo.....	27
4.1.1.	Aplicación cliente.....	27
4.1.2.	Aplicación piloto de taxi	32
4.2.	Validaciones y formularios	37
4.3.	Diseño intuitivo y usabilidad	38
4.3.1.	Herramientas utilizadas en la evaluación de usabilidad	39
4.3.1.1.	Evaluaciones con expertos.....	39
4.3.1.2.	Evaluaciones heurísticas.....	40
4.3.1.3.	Recorridos cognitivos	40
4.3.1.4.	Test de usabilidad	41
4.3.1.5.	Equipos montados en la cabeza	41

5.	DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN.....	43
5.1.	Software	43
5.2.	Instalación	45
5.3.	Desarrollo	47
5.4.	Google Maps	53
5.5.	Google <i>Cloud Messaging</i>	65
5.5.1.	Flujo de transmisión.....	65
5.5.2.	Aplicación Android lado del cliente	69
5.5.3.	Aplicación del lado del servidor	77
	CONCLUSIONES	79
	RECOMENDACIONES.....	81
	BIBLIOGRAFÍA.....	83
	APÉNDICE.....	85

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Dispositivos móviles	4
2.	<i>Tablet Sony Xperia Z</i>	5
3.	Modelo original TAM propuesto por Fred Davis, 1985.....	11
4.	Formulación de nueva relación en TAM (Davis, 1993)	12
5.	Versión final de TAM (Venkatesh & Davis, 1996)	13
6.	Menú principal de la aplicación cliente	28
7.	Formulario de registro para el cliente	29
8.	Mapa que muestra la ubicación para la solicitud	30
9.	Confirmación de petición y agregar ubicación a favoritos.....	31
10.	Lista de favoritos.....	31
11.	Creadores de la aplicación	32
12.	Menú principal del piloto de taxi.....	33
13.	Marcar inicio y fin de una ruta.....	34
14.	Costo total de un recorrido	35
15.	Aceptar o rechazar la petición de un cliente	36
16.	Gafas comerciales para test de dispositivos móviles.....	42
17.	Captura de información mediante gafas	42
18.	IDE Eclipse	43
19.	Emulador SDK de Android.....	44
20.	Manejador de SDK	45
21.	Descarga el SDK Android.....	46
22.	ADT-Bundle.zip.....	46
23.	Herramientas desarrollo Android	47

24.	IDE Eclipse para desarrollo Android.....	48
25.	Nuevo proyecto Android	48
26.	Configuración del Proyecto Android	49
27.	Imagen identificativa para la aplicación Android.....	50
28.	Creación de actividad principal.....	51
29.	Nombre actividad principal	52
30.	Descarga de servicios de Google <i>Play</i>	53
31.	Creación de nuevo proyecto en la consola de APIs de Google.....	54
32.	Activación de API de Google maps	55
33.	Obtención de API Key	55
34.	Generación de huella digital de ambiente de desarrollo.....	57
35.	Configuración de nueva API Key.....	58
36.	<i>Android manifest</i> API Key.....	59
37.	Permisos para acceso Google Maps.....	59
38.	Enlace de proyecto externo Google <i>Play Services</i>	60
39.	Creación de vista de mapa.....	61
40.	Actividad controladora del mapa	61
41.	Evento pulsación larga	62
42.	Incorporación de marcador al mapa.....	63
43.	Ventana informativa de marcadores en el mapa	63
44.	Centrar cámara en posición actual.....	64
45.	Tipos de mapa disponibles.....	64
46.	Flujo de transmisión GCM	66
47.	Activación servicio GCM.....	67
48.	Creación de llave de servidor	67
49.	Configuración de llave de servidor para el proyecto.....	68
50.	Llave de servidor creada	69
51.	Descarga de servicios de GCM para la aplicación Android.....	70
52.	Versión mínima Android SDK.....	71

53.	Permisos recepción de mensajes GCM.....	71
54.	Permisos <i>Android-Manifest</i> GCM	72
55.	Configuración <i>Android-Manifest Broadcast Receiver</i>	72
56.	Configuración servicio <i>GCMIntentService</i>	73
57.	Clase <i>GCMIntentService</i> , sobrescribiendo métodos	74
58.	Actividad principal chequeo registro GCM.....	75
59.	Envío de petición al servidor.....	76
60.	Envío de mensaje a dispositivo Android desde servidor.....	77
61.	Recepción de mensajes en aplicación Android	78

TABLAS

I.	Definición de variables de TAM	14
II.	Mercado objetivo	20
III.	Problemas de las organizaciones líderes	24
IV.	Mejores prácticas de organizaciones líderes.....	25
V.	Comparación entre aplicaciones.....	26
VI.	Comparativa entre aplicaciones del mercado.....	26
VII.	Evaluaciones heurísticas	40

GLOSARIO

Android	Sistema operativo para dispositivos móviles, basado en GNU Linux desarrollado por Google.
API	Interfaz de programación de aplicaciones.
Bluetooth	Sistema de interconexión inalámbrica para dispositivos como computadoras, teléfonos móviles, auriculares y otros. Permite la transmisión de datos.
CMD	Consola de comandos.
Dispositivo	Mecanismo o artificio dispuesto para producir una acción prevista.
Dispositivo de entrada	Objeto destinado a proveer de información a un ordenador para que este realice operaciones con la información.
Dispositivo de salida	Objeto destinado a proveer de información a un ordenador para que este realice operaciones con la información.
Emulador	Programa que se encarga de hacer compatibles distintos tipos de plataformas mediante aplicaciones,

un emulador intenta respetar el funcionamiento y visualización del original pero no siempre se logra.

GCM	Mensaje de Google por la nube.
Gesto	Interacción con un dispositivo haciendo algún movimiento específico con algún objeto de entrada como un <i>stylus</i> o los dedos.
Google maps	Plataforma que provee servicios de ubicación desde un dispositivo con sensores de localización como un GPS.
Google play	Aplicación disponible en el sistema operativo Android donde se pueden descargar e instalar aplicaciones compatibles con la versión de Android instalada en el dispositivo móvil.
GPS	Sistema de posicionamiento global.
Heurística	Criterio para encontrar una solución para un problema.
Ícono	Signo que mantiene una relación de semejanza con el objeto representado.
IDE	Entorno de desarrollo integrado.

Interfaz	También conocida como GUI, parte de un sistema informático que sirve para interactuar con el usuario y que este pueda darle instrucciones al sistema informático.
IT	Tecnologías de la información.
Keystore	Es una clave o identificación guardada en un archivo que usa el dispositivo y la aplicación para poder hacer uso del servidor que provee de los datos a la aplicación.
Luz infrarroja	Radiación del espectro luminoso, que tiene mayor longitud de onda y se encuentra más allá del rojo visible; usada en dispositivos móviles para transmisión de datos pero con un alcance muy limitado.
Mensaje <i>Push</i>	Tipo de mensaje en la que el servidor envía una petición al cliente cuando tiene información o notificación nueva, permite un ahorro de recursos y tiempo respecto a la tecnología anterior <i>pull</i> .
Microsoft Office <i>Mobile</i>	Paquete de oficina compuesto por procesador de texto, hojas de cálculo, presentaciones, cliente de correo; con el cual se busca brindar al usuario un paquete de oficina completo en un dispositivo móvil.

Mini-Tackball	Dispositivo de entrada que permite controlar el cursor, con este se pueden realizar acciones como seleccionar o apuntar.
MIT	Instituto tecnológico de Massachusetts, institución de educación superior situada en Cambridge, Massachusetts, Estados Unidos.
Navegador	Es una aplicación que opera a través de Internet, interpretando la información de archivos y sitios web para que éstos puedan ser leídos.
Ordenador	Dispositivo electrónico capaz de procesar información recibido, a través de dispositivos de entrada y genera resultados que serán mostrados o retornados a través de dispositivos de salida.
Paginador	Dispositivo de comunicación muy simple usado para recibir mensajes de texto de tamaño limitado, conocidos también como Beeper.
PDA	Asistente personal digital.
Periférico	Aparatos o dispositivos auxiliares e independientes conectados a la unidad central de procesamiento de una computadora.

Prototipo	Es un producto preliminar que puede ser desechable, una muestra simple de un sistema que no forzosamente tiene que ser funcional.
Red	Es un conjunto de dispositivos interconectados entre sí a través de un medio, que intercambian información y comparten recursos.
SDK	Paquete de software para desarrollo.
Servidor	Ordenador remoto o local que provee de servicios o datos a otros programas presentes en computadoras cliente.
Servlet	Objetos encargados de manejar y procesar la lógica del negocio, solo dando a la interfaz gráfica los datos necesarios para mostrarlos al cliente.
Sistema Operativo	Es el software básico de una computadora que provee una interfaz entre el resto de programas del ordenador, los dispositivos hardware y el usuario.
SMS	Servicio de mensajes cortos.
Stylus	Es un mecanismo de ingresar comandos a una computadora en forma de lápiz, la punta de este objeto es la que entra en contacto con el dispositivo haciendo posible la interacción con el mismo.

<i>Tablet</i>	Dispositivo que tiene características similares a las de un ordenador o computadora pero que se presenta en una sola pieza, contiene todos los componentes esenciales para su funcionamiento de forma autónoma, todo ello comprimido en una sola pieza.
TAM	Modelo de aceptación tecnológica.
<i>Touchpad</i>	Superficie generalmente cuadrada añadida al teclado que es sensible al tacto y cumple las funciones de un mouse.
USB	Conductor universal en serie.
Usuario	Persona que hace uso de un sistema o aplicación informática, cuenta con ciertos roles y permisos para interactuar con el sistema o aplicación.
<i>Wifi</i>	Tecnología de transmisión de datos inalámbrica, comúnmente usada para la conexión a internet por medio de un <i>router</i> o punto de acceso, con alcance diverso que van desde 1 metro hasta los 30 o 40 metros.

RESUMEN

Los dispositivos móviles, son una herramienta con mucho potencial y versatilidad en la realización de tareas que antes se efectuaban con otros medios como el papel. Con la sustitución de estas tareas por un medio virtual se busca ayudar al medio ambiente y automatizar los procesos, dando una eficiencia más alta y mayor control.

La geolocalización es una capacidad de los dispositivos móviles que provee una gran gama de usos, como es el caso de peticiones de taxis. Estas solicitudes pueden realizarse desde un dispositivo con sistema operativo Android, sin hacer llamadas, sin operadores intermedios, solo con un dispositivo móvil y la conexión a internet.

La aprobación de este servicio es analizada con el modelo de aceptación tecnológica, TAM, con este modelo se valida la percepción de fácil uso, percepción de utilidad, variables externas, intención de uso y el uso actual, con lo que se puede tomar una decisión según la evidencia puesta en escena. La conclusión de este estudio es que tendrá una aceptación muy amplia ya que se cubren aspectos faltantes de otras alternativas existentes en el mundo, con lo que se presenta una solución más completa en el ámbito guatemalteco.

Se presenta un prototipo que cubre las necesidades identificadas en el mercado objetivo, este prototipo cuenta con cualidades de utilidad, fácil uso, flexibilidad. Cubriendo el flujo completo de la aplicación y su uso real cuando la plataforma ya está en producción.

Se detalla la implementación de la solución, describiendo paso a paso lo que es requerido para el correcto funcionamiento de la plataforma. Se desarrolla la plataforma del servidor y las aplicaciones del cliente y del piloto, cada quien con roles definidos y acciones correspondientes, con lo que se cuenta con un sistema completo y funcional para la gestión de taxis.

OBJETIVOS

General

Proporcionar una guía detallada y un software funcional que promueva la implementación de una plataforma para la gestión de taxis en la vida cotidiana de los guatemaltecos.

Específicos

1. Conocer la evolución, capacidades, métodos de interacción y ventajas de los dispositivos móviles.
2. Analizar las diversas variables del Modelo de Aceptación Tecnológica, para determinar el grado de aceptación que obtendrá la aplicación en el mercado objetivo.
3. Realizar un análisis comparativo, tomando en cuenta fortalezas y debilidades de las aplicaciones líderes en el mercado, y de esta forma fortalecer la nueva aplicación.
4. Presentar un posible producto o bosquejo que cubra requerimientos específicos, los cuales reflejan las necesidades guatemaltecas en cuanto al servicio de transporte público individual (taxi).

5. Guiar a un emprendedor para el desarrollo de una plataforma para gestión de taxis, aprovechando la movilidad y potencial que tienen los dispositivos móviles con sistema operativo Android.

INTRODUCCIÓN

Guatemala actualmente cuenta con un gran número de usuarios con dispositivos móviles, la mayoría de los cuales usan sus dispositivos inteligentes para tareas muy básicas como llamar, mandar mensajes o revisar el correo electrónico; por lo que se desaprovecha el potencial que tienen los dichos dispositivos para hacer una gran cantidad de tareas en la vida cotidiana de los usuarios. Tareas como el manejo de agenda, compras en línea, reservas de servicios, sincronización de archivos, guía de tránsito con mapas, incluso la gestión para un servicio público como el taxi.

Un claro ejemplo es la implementación de una aplicación móvil en Lima, Perú para la gestión del servicio de taxi, con la cual se puede solicitar un taxi que llegará a la localidad indicada por medio de la aplicación, además el usuario conoce de antemano la información del chofer y vehículo que le darán el servicio. Esta es una forma más segura, más práctica y más cómoda de gestionar un servicio de taxi.

Con base en el ejemplo anterior y analizando el desempeño de proyectos similares en otras partes del mundo, se pretende que mediante este trabajo de investigación y desarrollo se tenga una guía detallada y un software funcional que promueva la implementación de este servicio en la vida cotidiana de los guatemaltecos.

1. HISTORIA DE LOS DISPOSITIVOS MÓVILES

Los dispositivos móviles se han convertido en parte de la vida cotidiana, son utilizados desde niños hasta personas adultas para diversos fines, entre los que se pueden mencionar conexión a internet, chat, redes sociales, intercambio de información entre otros. Este capítulo pretende introducir conceptos importantes de dispositivos móviles, cómo estos se han desarrollado a lo largo de la historia, los tipos de dispositivos móviles que existen y sistemas operativos diseñados para ellos.

1.1. Concepto de dispositivo móvil

Un dispositivo móvil es un aparato de reducido tamaño, con capacidades para realizar procesos, conexión permanente o intermitente a una red, y memoria limitada. Originalmente fueron diseñados para una función específica, pero con el paso del tiempo se les han ido incorporando otras funcionalidades.

Actualmente existe gran variedad de dispositivos móviles, desde reproductores de audio portátiles hasta navegador GPS, pasando por los teléfonos móviles, los PDAs y las *tablets*.

Las principales características que debe tener un dispositivo móvil son:

- Memoria de almacenamiento limitada
- Usualmente de tamaño pequeño
- Diseñados para una función específica, aunque algunas veces pueden realizar otras funciones

- Uso personal
- Fácil transporte
- Funcionamiento autónomo a través de una batería que le proporciona la energía que necesita

Uno de los aspectos más importantes de los dispositivos móviles es el concepto de movilidad, por ello deben ser pequeños y portables. Así la persona podrá trasladarlos de un lugar a otro sin ninguna complicación.

Dentro de los dispositivos móviles no se encuentran únicamente los celulares, como la mayoría de personas piensan, sino que existen una gran variedad de aparatos que entran en esta categoría, pudiéndose mencionar:

- Paginadores
- Comunicadores de bolsillo
- Teléfonos con pantalla para internet (*Internet Screen Phones*)
- Sistemas de navegación de automóviles
- Sistemas de entretenimiento
- Sistemas de televisión e internet (*WebTV*)

1.2. Categorías de los dispositivos móviles

Dado el variado número de niveles de funcionalidad asociado con dispositivos móviles, en el 2005, T38 y *DuPont Global Mobility Innovation Team* propusieron los siguientes estándares para la definición de dispositivos móviles (1):

- Dispositivo móvil de datos limitados (*limited data mobile device*)

- Teléfonos móviles clásicos
- Poseen una pantalla pequeña tipo texto
- Los servicios que proporcionan son limitados a SMS y acceso WAP

- Dispositivo móvil de datos básicos (*basic data mobile device*)
 - Poseen una pantalla mediana
 - Poseen un menú o una navegación basada en íconos
 - Acceso a correo electrónico
 - Navegador web
 - Lista de direcciones
 - SMS

- Dispositivo móvil de datos mejorado (*enhanced data mobile device*)
 - Posee una pantalla de mediana a grande (mayor de 240 x 120 píxeles)
 - Navegación tipo *stylus*
 - Posee las mismas características que el dispositivo de datos básicos, adicional aplicaciones nativas como *Microsoft Office Mobile* y aplicaciones corporativas

1.3. Teléfonos inteligentes

En la actualidad los *smartphones* abarcan casi en su totalidad el mercado de dispositivos móviles. Los *smartphones*, también llamados dispositivos móviles de última generación, son aparatos pequeños con capacidades grandes de procesamiento, conexión de red casi permanente, memoria ampliable y multipropósito.

Los teléfonos son llamados inteligentes por las prestaciones que los hacen diferentes de un teléfono móvil tradicional, haciendo referencia a la interfaz, una pantalla táctil o el sistema operativo móvil que poseen, su utilización se diferencia mediante una exclusiva disposición de menús, teclas o botones.

En los teléfonos inteligentes se pueden instalar aplicaciones con diversos propósitos que abarcan desde funciones diseñadas para el ocio, hasta mejorar el rendimiento en el trabajo.

Figura 1. **Dispositivos móviles**



Fuente: <http://156.35.151.9/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf>. Consulta: 2 de abril de 2013.

1.4. **Tablets**

Una *tablet* es un tipo de computadora portátil, es más grande que un teléfono inteligente y posee una pantalla táctil, no necesita teclado físico o ratón.

Las *tablets* utilizan en su mayoría los mismos sistemas operativos que utilizan los *smartphones*, por consiguiente también permiten la descarga de aplicaciones directamente desde la tienda *online* del sistema.

Las *tablets* no poseen teclado integrado, pero puede conectarse uno mediante *bluetooth* o mediante un cable USB. También existe la *tablet* convertible que dispone de un teclado físico que gira sobre una bisagra o se desliza debajo de la pantalla. Asimismo, existe un formato denominado híbrido, que posee un teclado físico pero puede separarse de él y comportarse como una *tablet* convencional.

Figura 2. **Tablet Sony Xperia Z**



Fuente: <http://techbuzzes.com/2013/03/sony-xperia-tablet-z/> Consulta: 2 de abril de 2013

La interacción es el intercambio de información entre el dispositivo móvil y la persona que lo maneja.

Existe una gran variedad de periféricos para la interacción, los más comunes son:

- Teclado y pantalla: algunos dispositivos móviles traen integrado teclados para facilitar la escritura a los usuarios. La mayoría de este tipo son teléfonos clásicos, como Blackberry.
- Apuntadores: los dispositivos móviles en algunos casos poseen un *touchpad* o *mini-trackball* integrado para poder manipular objetos en la pantalla.
- Voz y sonido: mediante la voz y la lectura de pantalla, la persona no tiene necesidad de ver la pantalla del dispositivo, y con esto se facilita la interacción.
- Rastreo ocular: permite que el usuario interactúe con el dispositivo móvil (movimiento de los ojos) a través del rastreo ocular. El requisito es que el dispositivo contenga una cámara frontal para que el usuario pueda ver la pantalla.
- Pantalla táctil: permiten la interacción con el dispositivo mediante el toque con los dedos sobre su superficie. Existen dos tipos de pantallas, según la tecnología.
 - Resistivas: son económicas y no les afecta el polvo ni el agua, pueden ser utilizadas con un puntero con el dedo, son más precisas.
 - Capacitivas: poseen una buena calidad de imagen, permiten la utilización de varios dedos a la vez, dan la oportunidad de interpretar gestos.

- Lápiz o *stylus*: la forma de interactuar con algunos dispositivos es por medio de un lápiz, puntero o *stylus*, mediante el cual se seleccionan las opciones de la pantalla.
- Otros: el tacto es otra posibilidad para interactuar con el dispositivo. Un ejemplo es una funda con serigrafía que permite el manejo de las funciones básicas mediante el tacto. Esta funda fue diseñada para personas ciegas o con visión reducida.

2. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO EN GUATEMALA

Actualmente la tecnología es parte del sistema de vida de todas las sociedades. A lo largo de los años el destino de la humanidad ha quedado en manos de la tecnología. Es inevitable el avance tecnológico, al igual que es inevitable el avance social y económico de un país.

Los beneficios que trae la tecnología a en Guatemala son muy numerosos, a partir de ella existe una mayor productividad, las empresas guatemaltecas cada vez más desarrollan, actualizan y automatizan sus procesos, lo cual proporciona a los empleados facilidades en su trabajo y mayor tiempo libre.

Por otro lado, existe la posibilidad que personas no acepten ese cambio tecnológico ya que el trabajo podría convertirse en monótono y aburrido, el ser humano necesita realizar algo que estimule su cerebro, su capacidad manual y también necesita variedad.

Es necesario poner en una balanza los beneficios y problemas que trae consigo la tecnología, y de esta forma reducir gradualmente los problemas, para lograr que las personas acepten las nuevas herramientas que harán que Guatemala sea más productiva y tenga un mejor desarrollo económico.

Para lograr la aceptación de nuevas tecnología se han desarrollado modelos para entender por qué los usuarios aceptan o usan las tecnologías. Sin embargo a pesar de lo fértil que ha sido la investigación sobre la adopción tecnológica, el Modelo de Aceptación Tecnológica (TAM) ha destacado entre

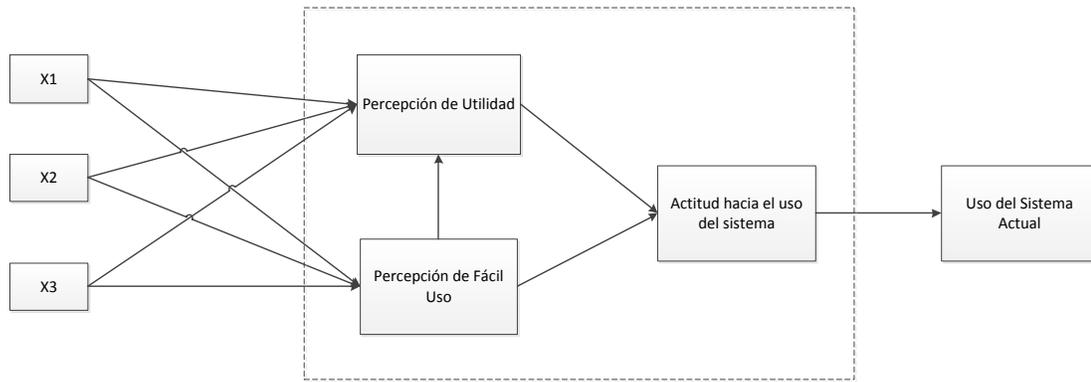
todos los existentes. En la mayoría de situaciones TAM ha confirmado su solidez y capacidad para predecir la adopción tecnológica en usuarios de una organización.

2.1. Modelo de Aceptación Tecnológica (TAM)

Los orígenes de TAM se remontan a la décadas de 1980, por la preocupación que trabajadores no utilizaban productos IT desarrollados para ellos. Los autores de IT comprobaron que la clave para que utilizaran los productos era promover primero su aceptación. El proceso de aceptación se llevaba a cabo preguntando individualmente las intenciones futuras que el usuario tenía para el producto desarrollado. Las organizaciones empleaban esta información para promover la aceptación y por lo tanto aumentar el uso de IT.

En 1985, Fred Davis propuso el Modelo de Aceptación Tecnológica en su tesis de doctorado en MIT *Sloan School of Management*, el modelo se puede observar en la figura 3. Davis propuso que el uso del sistema es una respuesta que puede ser explicada o predicha por la motivación del usuario, que, a su vez, está directamente influenciada por un estímulo externo que consiste en las características y capacidades del sistema real.

Figura 3. **Modelo original TAM propuesto por Fred Davis, 1985**



Fuente: elaboración propia.

Davis (1985) sugirió que la motivación de los usuarios puede ser explicada por tres factores:

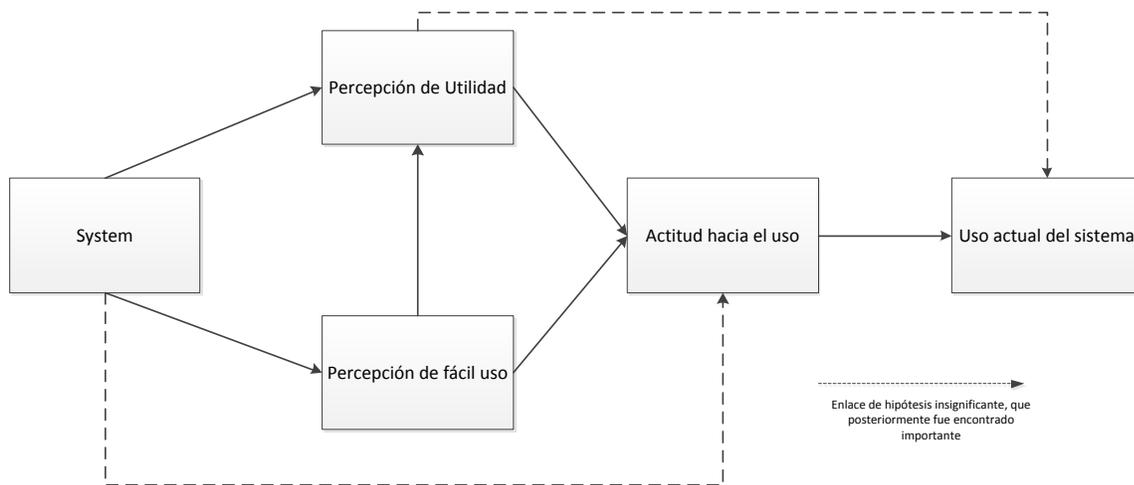
- Percepción de fácil uso
- Percepción de utilidad
- Actitud hacia el uso del sistema

Davis formuló la hipótesis que la actitud de un usuario hacia un sistema era un factor importante y determinante para que utilizara o rechazara el sistema. La actitud del usuario, a su vez, se consideró a ser influenciada por dos principales creencias: percepción de utilidad y percepción de fácil uso. Finalmente ambas creencias son la hipótesis de ser directamente influenciados por las características de diseño del sistema, representado por X1 X2 y X3 en la figura 1.

Davis (1993) sugiere que en contraste con lo que se predijo inicialmente, la utilidad percibida también podría tener una influencia directa sobre el uso real

del sistema. Al mismo tiempo, se encontró que las características del sistema pueden influir directamente en la actitud de una persona para usarlo, sin la necesidad de formar en una persona el pensamiento del sistema como se muestra en la figura 2.

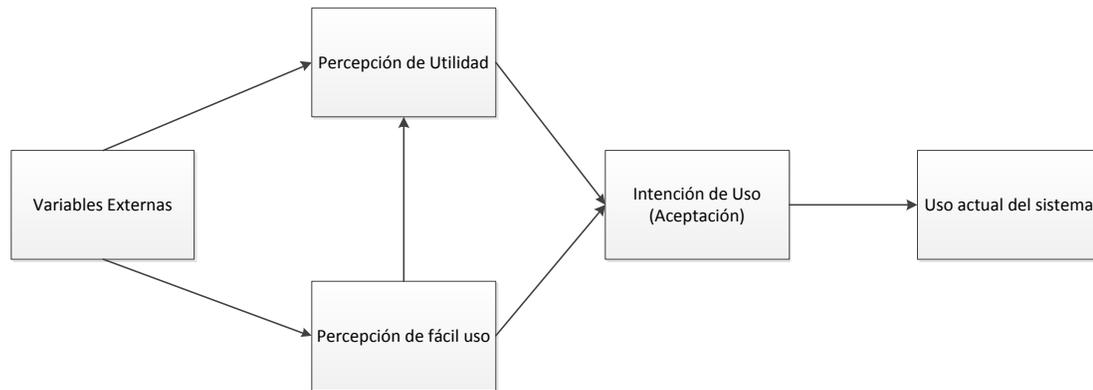
Figura 4. **Formulación de nueva relación en TAM (Davis, 1993)**



Fuente: elaboración propia.

Sin embargo la facilidad de uso percibida se encontró que tiene un efecto pequeño pero significativo sobre la intención de comportamiento, que luego desaparece con el tiempo. Pero el hallazgo principal fue que tanto la utilidad y la facilidad de uso percibidas tienen una influencia directa en la intención de comportamiento, lo que elimina la necesidad de una actitud constructiva del modelo que se muestra en la figura 4. El modelo resultante es mostrado en la figura 6.

Figura 5. **Versión final de TAM (Venkatesh & Davis, 1996)**



Fuente: elaboración propia.

2.1.1. Concepto

El Modelo de Aceptación Tecnológica (TAM) modela la forma en que los usuarios logran aceptar y utilizan una tecnología. El modelo sugiere que cuando a los usuarios se les presenta una nueva tecnología, una serie de factores que influyen en su decisión sobre cómo y cuándo la van a utilizar. Los factores son:

- Utilidad percibida: es el grado en que una persona cree que usando un sistema, puede mejorar su desempeño en un trabajo específico.
- Facilidad de uso percibida: es el grado en que una persona cree que usando un sistema hará menos esfuerzo para realizar sus tareas.

2.1.2. Definición de variables de TAM

El Modelo de Aceptación Tecnológica TAM posee variables, las cuales son de utilidad para la realización de un estudio cualitativo de las preferencias, utilidades y necesidades del usuario hacia el nuevo producto.

Tabla I. Definición de variables de TAM

Variable	Definición
Percepción de fácil uso	Sentimiento por parte del usuario que usar la herramienta será una actividad simple y sin requerimiento de esfuerzo físico y mental.
Percepción de utilidad	Sentimiento por parte del usuario en el cual evalúa si el uso de la herramienta lo lleva a un comportamiento mejorado o más eficiente.
Variables externas	Características objetivas del diseño, formación, eficacia en el uso de los ordenadores, implicación de los usuarios en el diseño, naturaleza del proceso de implementación.
Intención de uso	Voluntad del usuario para adoptar el uso de la herramienta para el desarrollo de sus actividades.
Uso actual	Uso de la herramienta posterior a ser aceptada y que ayuda en el desarrollo de las actividades, dándole un comportamiento más eficiente, siendo una actividad que no requiere mayor esfuerzo físico y mental.

Fuente: elaboración propia.

2.1.3. Limitaciones de TAM

El modelo está enfocado en predecir el uso la tecnología pero no en el incremento del rendimiento del usuario. No existe una relación positiva entre uso y rendimiento.

La intención de comportamiento del modelo predice el uso real, sin embargo la percepción de fácil uso y la percepción de utilidad no son consistentemente buenos predictores del uso real.

Las variables TAM han sido medidas en grupos homogéneos para la mayor parte de la investigación. Cuando se ha medido el modelo, se ha hecho en grupos de estudiantes o de trabajadores. Los ambientes reales tienden a ser heterogéneos, por lo que no se puede generalizar los resultados obtenidos.

2.2. Teoría y la relación con la tecnología escogida

Las variables del Modelo de Aceptación Tecnológica TAM, se pueden aplicar al estudio de la interacción de los usuarios con la aplicación Ctrl – Taxi, para determinar puntos claves para la aceptación del producto de software.

2.2.1. Percepción de fácil uso

La forma tradicional de requerir un servicio de taxi y hacer el pago respectivo ha sido por medio de teléfono y efectivo o hacer una parada en la calle corriendo el riesgo de ser víctima de un atraco, esto viene a ser reemplazado por un dispositivo tecnológico siendo este un teléfono inteligente o una tableta con sistema operativo Android.

2.2.2. Percepción de utilidad

Reemplazo de la manera tradicional de requerir un servicio de taxi, dando una seguridad mayor y descartando el uso de efectivo. Existe un mejor control de todo el proceso por parte del cliente y por parte del chofer, ya que las peticiones de servicio se hacen en línea y el taxista puede hacer entrega de un recibo de pago al usuario al haber finalizado el servicio.

2.2.3. Variables externas

Las variables externas son todas aquellas ajenas al sistema, pero tienen un impacto de forma directa con la aplicación, por ello pueden afectar el uso y la aceptación del software por parte del usuario.

2.2.3.1. El internet

En una situación donde la persona no dispone de internet no podrá intercambiar información con la aplicación; el ingreso económico, si una persona no tiene los suficientes recursos para adquirir un *smartphone* o una tableta no podrá hacer uso de la aplicación ya que únicamente funciona con dispositivos con sistema operativo Android; la edad es un factor importante ya que las personas con edad avanzada tienden a utilizar dispositivos móviles con funciones básicas (llamar, agenda y mensajes de texto), por lo que es posible que no se adapten o no puedan utilizar la aplicación y sigan utilizando la forma tradicional de solicitar un taxi.

El nivel educacional también es un factor influyente, mientras mayor sea el grado académico mayor será la probabilidad de que utilicen la aplicación.

2.2.4. Intención de uso (aceptación)

El usuario utilizará la herramienta al darse cuenta de las ventajas que esta le proporciona, ventajas como el ahorro de tiempo, seguridad, control, no manejo de efectivo, manipulación de un mapa y otros, lo cual hará que la recomiende y más usuarios se vean beneficiados con la aplicación.

2.2.5. Uso actual

El usuario al momento de solicitar un taxi enviará la posición exacta donde desea iniciar el servicio. Al momento de hacer el pago por el servicio recibido se imprimirá el comprobante, este se efectuará con cualquiera de las dos formas de pago, efectivo o tarjeta de crédito. Los puntos de inicio de servicio se guardarán y estarán disponibles para un uso más rápido en el futuro, podrán registrarse puntos de inicio de servicio favoritos, los cuales el usuario frecuenta y desea tener un acceso inmediato para solicitar el servicio.

3. IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN QUE LA APLICACIÓN REALIZARÁ

3.1. Antecedentes

La forma tradicional de solicitar un servicio de taxi involucra llamar por teléfono o parar un taxi en la calle, en esta última se corre el riesgo de sufrir un atraco por parte del taxista al cual se conoce y no tiene una empresa que lo respalde. También se tiene el inconveniente que el cliente debe saber exactamente donde está parado para indicarle al taxi donde recogerlo, lo cual en ciertas situaciones puede ser complicado.

La forma de pago debe ser en efectivo ya que los taxistas no cuentan con un sistema que les permita recibir pagos con tarjetas de crédito o débito, lo cual conlleva otro riesgo que es el transportar efectivo. Al efectuar dicho pago el cliente no recibe ningún tipo de comprobante, el cual puede ser útil para notificar viáticos.

Dada esta necesidad, una nueva aplicación fue desarrollada por dos egresados de Harvard en conjunto con un ingeniero comercial chileno, promete simplificar y modernizar la acción de pedir un taxi. El nombre de la aplicación es SaferTaxi.

Desde un *smartphone* o cualquier navegador web, SaferTaxi permite a través de su interfaz solicitar un taxi las principales ciudades de Argentina, Chile y Brasil, identificando al conductor más cercano a la zona, e incluso, convertirlo en una experiencia social, donde los pasajeros pueden calificar el servicio de la

unidad así como interactuar con los comentarios de otros usuarios, creando una vivencia integral y marcando a su vez una tendencia interesante, en un rubro de servicios donde era difícil concebir la intervención de tecnologías móviles bajo esta perspectiva.

La aplicación fue realizada en el 2009 y ha crecido de manera importante y exponencial. Para el 2015 se estima invertir 5 millones de dólares en Argentina, debido al éxito obtenido. De igual forma, tanto en Europa como en América Latina se han desarrollado aplicaciones similares con el mismo éxito.

En Guatemala, no se ha desarrollado alguna aplicación para la gestión de taxis, todo el proceso se realiza de forma tradicional (por medio de llamada telefónica), siendo los usuarios propensos a los riesgos mencionados.

3.2. Mercado objetivo (usuarios de la aplicación)

El mercado objetivo son aquellos usuarios a los cuales está enfocada la aplicación, se deben tomar en cuenta diversos factores del entorno en el que se desenvuelve el usuario, los cuales determinan sus necesidades.

Tabla II. **Mercado objetivo**

Base de segmentación	Segmentos del mercado
Geográfica	
Región	Metropolitano y departamental.
Urbana - rural	Urbano
Clima	Indiferente

Continuación de la tabla II.

Beneficiarios Directos	Personas en el territorio guatemalteco
Beneficiarios Indirectos	Turistas que se encuentran en el territorio guatemalteco
Demografía	
Edad	Mayores de 14 años
Sexo	Masculino, femenino
Ingresos	Indiferente
Educación	Conocimientos mínimos: <ul style="list-style-type: none"> • Leer • Escribir • Conocimiento básico en sistema operativo Android • Conocimiento en ubicaciones en mapas.
Ocupación	Indiferente
Religión	Indiferente
Otros	
Industria	Usuarios de taxi Empresa proveedora de taxis Chofer de taxi

Fuente: elaboración propia.

3.3. **Benchmarking de la aplicación**

El *benchmarking* es una herramienta estratégica para indagar, identificar, contrastar y aprender de las mejores prácticas de otras organizaciones,

analizando el conjunto de factores que incurren en el éxito éstas, aprender de sus logros y aplicarlos a los procesos propios para mejorarlos.

La finalidad del *benchmarking* es producir un impacto positivo de reflexión, motivación y transformación en la organización propia al compararse con los mejores del mercado.

Existen 5 tipos de *benchmarking* (interno, competitivo primario, competitivo secundario, cooperativos y genéricos), para el objeto de la presente investigación se aplicó *benchmarking* competitivo secundario, en el cual se recopila información de la competencia por vías indirectas, como internet, publicaciones, catálogos, estudios directo de sus productos, entre otros.

El *benchmarking* es importante por el impacto (mejoras y cambios) que puede tener sobre los procesos de una organización, en base a comparaciones.

3.4. *Benchmarking* aplicado a Ctrl - Taxi

La herramienta *benchmarking* se puede utilizar para indagar las fortalezas y debilidades de Ctrl – Taxi comparado con otras aplicaciones similares existentes en el mercado, el estudio puede tomarse como base para el mejoramiento de la aplicación.

3.4.1. Organizaciones líderes

En la actualidad existen aplicaciones con objetivos similares a Ctrl – Taxi, de las cuales se pueden evaluar sus características principales para la realización del estudio de *benchmarking*. Las principales aplicaciones son:

- SaferTaxi: el usuario pide el taxi, tras lo cual el sistema hace una búsqueda de los taxistas que están más cerca, primero buscando en un radio de un kilómetro y tras 20 segundos, va ampliándolo un kilómetro más. Al encontrar un taxista disponible, le manda su perfil a quien pidió el taxi (incluyendo nombre, una foto, patente del auto y calificación de otros usuarios). La persona puede aceptar al taxista o rechazarlo. Si lo acepta, un mapa le indica por dónde viene el auto, a través de un GPS. Si lo rechaza, le muestra otra sugerencia (4).
- EasyTaxi: esta aplicación permite que el usuario consiga un taxi y, al mismo tiempo, obtenga la información completa de la unidad, incluyendo el nombre, foto y número de teléfono del conductor; así como la placa y el modelo del vehículo. El usuario también puede ver en tiempo real la posición del taxi mientras espera. El servicio no representa un costo adicional para el cliente (5).
- JoinUp Taxi: no sólo permite pedir un taxi de forma individual, para el mismo momento o con una antelación de hasta una semana, sino que intenta fomentar la sostenibilidad introduciendo la posibilidad de compartir el taxi con otra persona cuyo destino sea el mismo. Si se decide compartirlo, la aplicación buscará compañeros en un radio de tres manzanas y con un margen de 30 minutos. Se sale desde el mismo punto y se parte hacia el mismo destino, por lo que si el sistema encuentra coincidencias se puede ver la ubicación y unirse. Si no hay nadie que haya solicitado la misma carrera, se puede pedir el taxi para una persona individual o indicar que se deja la puerta abierta para que otras personas puedan unirse (6).

3.4.2. Problemas de las organizaciones líderes

Las organizaciones líderes en el mercado no son completas, ya que carecen de algunas características que pueden ser de utilidad para la usabilidad del usuario.

Tabla III. Problemas de las organizaciones líderes

SaferTaxi	EasyTaxi	JoinUp Taxi
<ul style="list-style-type: none">• Falta un cálculo aproximado de cuanto sale el viaje.• Un taxista no se puede registrar como cliente para pedir un taxi.• Inseguridad en los datos del cliente.• El taxista no puede calificar al pasajero.• El taxista no puede cancelar mensajes falsos.	<ul style="list-style-type: none">• Al actualizar la aplicación del cliente, se interpone la interface de taxista.• En algunas ocasiones no reconoce el GPS.• No guarda historial.	<ul style="list-style-type: none">• No se han registrado problemas.

Fuente: elaboración propia.

3.4.3. Mejores prácticas de las organizaciones líderes

Las organizaciones líderes poseen características que las distinguen y las hacen aplicaciones eficientes y populares, dichas características que aumentan la usabilidad y cubren las necesidades del usuario.

Tabla IV. **Mejores prácticas de organizaciones líderes**

SaferTaxi	EasyTaxi	JoinUp Taxi
<ul style="list-style-type: none"> • Aplicación para cliente y taxista. • Gratuita. • Localización automática por medio de GPS. • Monitoreo del taxi en tiempo real. • Historial de viajes. • Puntos por lealtad. 	<ul style="list-style-type: none"> • Es seguro, dado a que todos los taxistas son registrados. • Visualización de la ubicación del taxista en tiempo real. • Es gratuito, no hay cobros para instalar y usar. 	<ul style="list-style-type: none"> • Descarga y uso gratuito. • Opción de compartir el trayecto con otra persona. • Reservación de taxista hasta con una semana de antelación. • Visualización del ahorro económico y de CO2 al compartir un taxi.

Fuente: elaboración propia.

3.5. Comparación entre aplicaciones

Comparación de Ctrl-Taxi respecto a otras aplicaciones similares ya disponibles en el mercado internacional, en el cuadro siguiente se pueden observar los aspectos que se cubren con las diferentes aplicaciones con lo que también se aprecian las deficiencias, que para el caso de Ctrl-Taxi se convierten en oportunidades.

Tabla V. **Comparación entre aplicaciones**

Aplicación Android	Ubicación del Cliente	Pago tarjeta crédito o débito	Impresión de recibo	Historial	Favoritos	Mapas	Planificación servicio	Info Taxista	Compartir Taxi
Ctrl - Taxi	10	10	10	10	10	10	10	--	--
Safer Taxi	10	--	--	--	--	10	--	10	--
Easy Taxi	10	--	--	--	--	--	--	10	--
Join Up Taxi	10	10	--	10	--	10	--	10	10

Fuente: elaboración propia.

3.5.1. Escala de calificación

Para la realización de la comparación entre las diversas aplicaciones se utilizó una escala numérica ascendente, para realizar una comparación cuantitativa de las características de las diferentes aplicaciones.

Tabla VI. **Comparativa entre aplicaciones del mercado**

Muy malo	Normal	Muy bueno	Aspecto inexistente
0	5	10	----

Fuente: elaboración propia.

4. DISEÑO DE LA APLICACIÓN BAJO LA NECESIDAD IDENTIFICADA

4.1. Prototipo

Ctrl-Taxi es una aplicación enfocada al mercado de taxis. El cliente podrá pedir un taxi desde su dispositivo móvil Android y su ubicación será rastreada automáticamente por medio de GPS. El piloto del taxi estará la posibilidad de aceptar y rechazar una solicitud, marcar la ubicación del inicio y fin de una ruta, calcular automáticamente el costo del recorrido e imprimir un recibo de pago.

Dadas las especificaciones anteriores se puede determinar que la aplicación tendrá dos versiones, una para el cliente y otra para el piloto. Cada una con características y funciones diferentes.

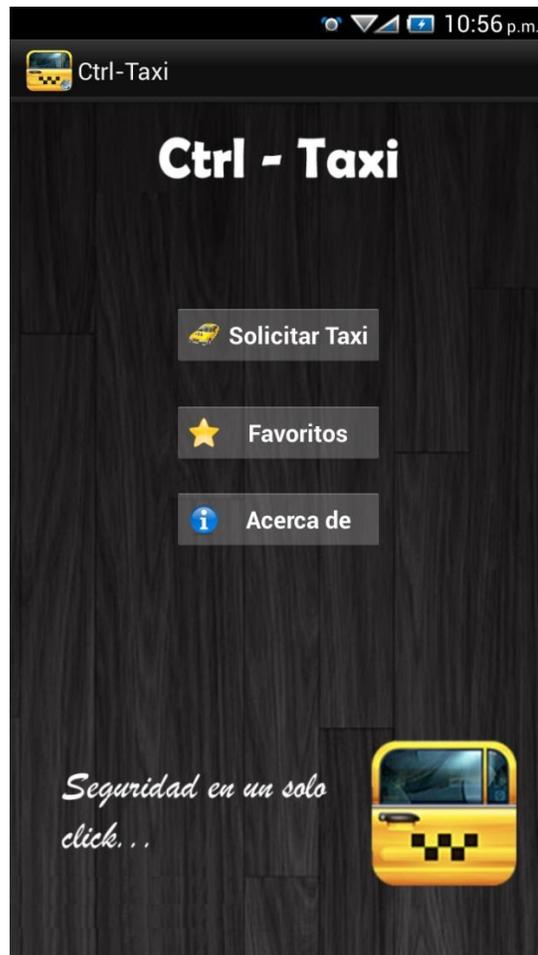
4.1.1. Aplicación cliente

Esta aplicación está enfocada al cliente, la puede descargar e instalar en cualquier dispositivo móvil que utilice sistema operativo Android, posteriormente podrá utilizarla para toda la gestión de petición de taxis.

La aplicación del cliente tendrá las siguientes opciones:

- Registro
- Solicitar taxi
- Favoritos
- Acerca de

Figura 6. Menú principal de la aplicación cliente



Fuente: elaboración propia.

Registro: cuando el cliente instale la aplicación, deberá registrarse antes de iniciar. Los datos que debe ingresar son:

- Nombre de usuario
- Correo electrónico
- Teléfono

Figura 7. **Formulario de registro para el cliente**



The image shows a registration form for a taxi application. The form is titled "Ctrl - Taxi" and features a yellow taxi icon. It contains three input fields for "Usuario:", "Correo Electrónico:", and "Teléfono:". At the bottom, there are two buttons labeled "Registrar" and "Cancelar".

Fuente: elaboración propia.

Solicitar taxi: el cliente podrá solicitar un taxi por medio de la aplicación, al inicio se abrirá con su ubicación actual, la cual será detectada por medio de GPS. Si el cliente lo desea, puede cambiar el marcador para señalar otra ubicación origen.

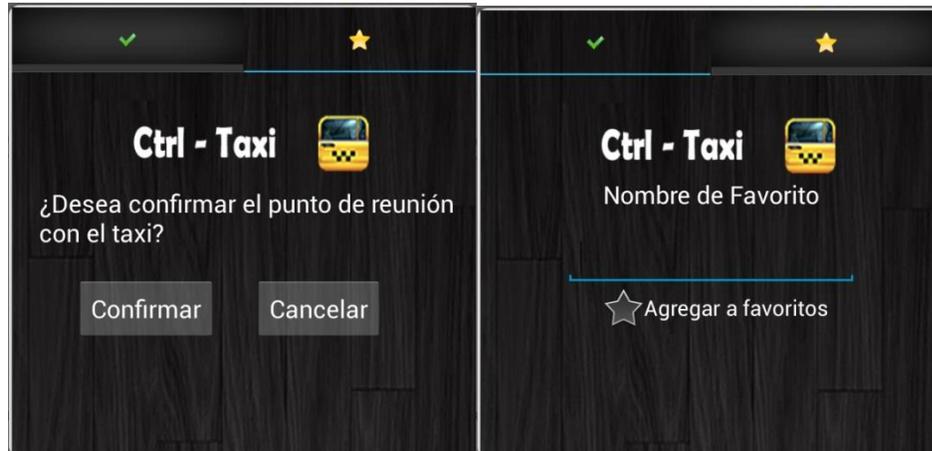
Figura 8. **Mapa que muestra la ubicación para la solicitud**



Fuente: elaboración propia.

Para confirmar la ubicación existirá un cuadro de diálogo, el cual dará la opción de confirmar la petición y de agregar la ubicación a favoritos.

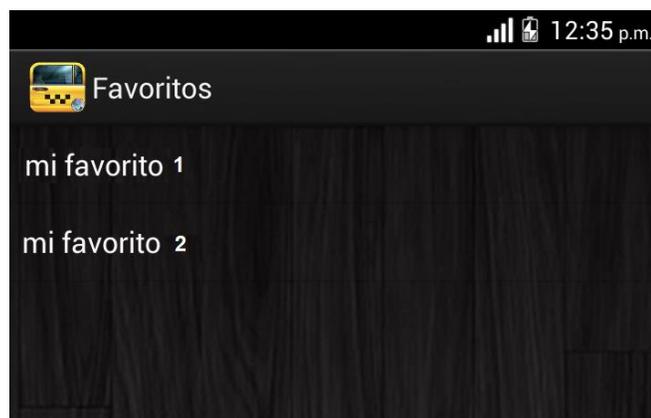
Figura 9. **Confirmación de petición y agregar ubicación a favoritos**



Fuente: elaboración propia.

Favoritos: el cliente podrá ingresar a la lista de favoritos que ha guardado, y escoger uno. Al seleccionarlo se abrirá el mapa con la ubicación y tendrá la opción de realizar una petición.

Figura 10. **Lista de favoritos**



Fuente: elaboración propia.

Acerca de: mostrará un cuadro de diálogo con la información de los creadores de la aplicación.

Figura 11. **Creadores de la aplicación**



Fuente: elaboración propia.

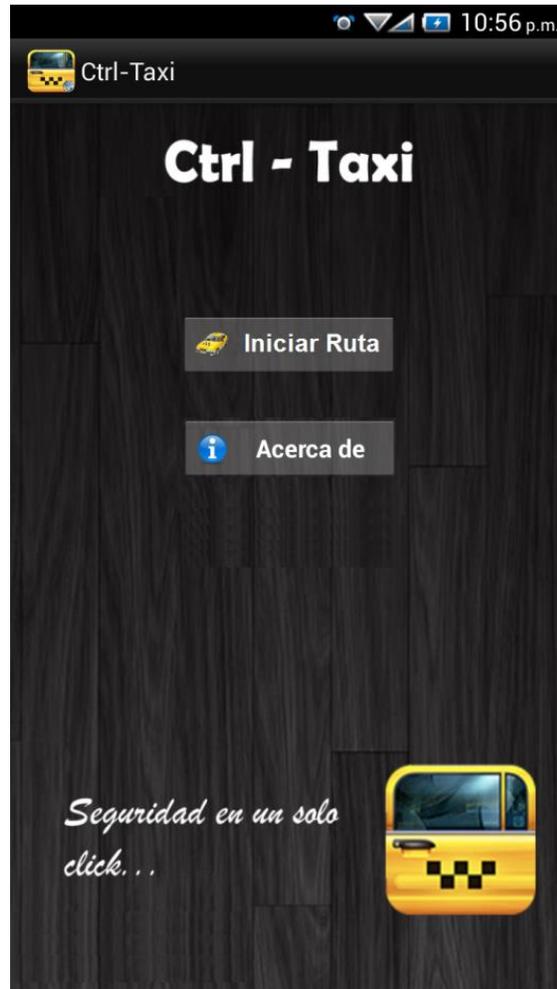
4.1.2. Aplicación piloto de taxi

Es la aplicación orientada al piloto, con la cual el taxista recibirá notificaciones a su dispositivo móvil indicando que le fue asignado un recorrido, el taxista decidirá si puede tomar el recorrido por medio de la aplicación.

El piloto de taxi tendrá las siguientes opciones:

- Marcar inicio y fin de la ruta
- Aceptar y rechazar una petición
- Imprimir recibo
- Acerca de

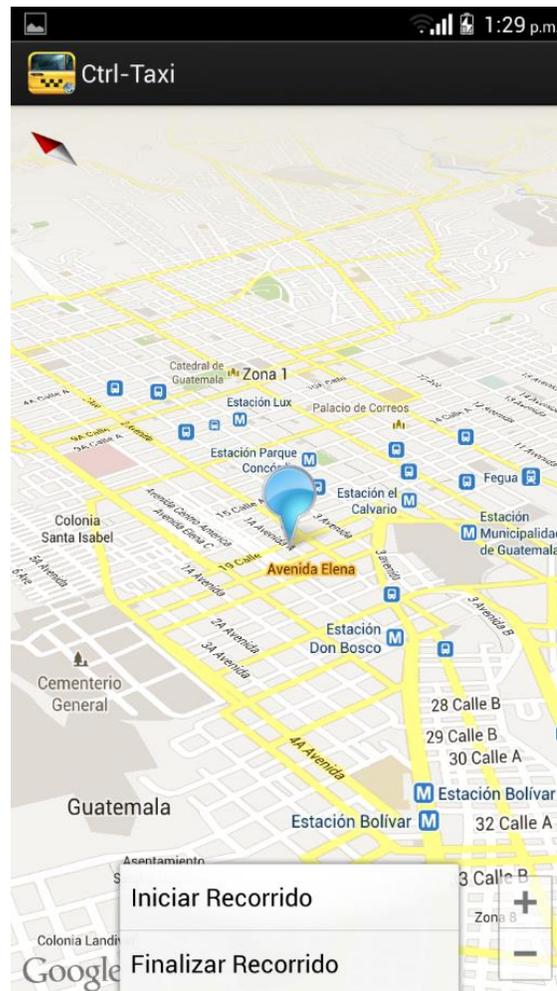
Figura 12. Menú principal del piloto de taxi



Fuente: elaboración propia.

Marcar inicio y fin de una ruta: la aplicación tomará la posición actual, el piloto debe indicar si es el inicio o el fin del recorrido.

Figura 13. Marcar inicio y fin de una ruta



Fuente: elaboración propia.

Al finalizar el recorrido aparecerá una ventana indicando el costo del recorrido, con base en la distancia, y si desea imprimir el recibo de pago.

Figura 14. **Costo total de un recorrido**



Fuente: elaboración propia.

Aceptar y rechazar una petición: le llegará un mensaje al piloto de taxi, y podrá elegir si acepta o no la ruta del taxi.

Figura 15. **Aceptar o rechazar la petición de un cliente**



Fuente: elaboración propia.

4.2. Validaciones y formularios

Los formularios son utilizados como medio para que el cliente pueda remitir información hacia el servidor, creando de esta forma un sistema completo de iteración.

Android dispone de elementos que permiten crear formularios, los cuales permiten al usuario interactuar con la aplicación.

- Cliente
 - Existirá una versión de la aplicación para el cliente.
 - Se utilizará GPS para las ubicaciones en los mapas al momento que se solicite un taxi.
 - Se deberá registrar para poder ingresar a la aplicación.
 - Posibilidad de guardar ubicaciones deseadas en una lista de favoritos.
 - Cualquier ubicación de la lista de favoritos debe poder seleccionarse y mandar solicitud.
 - Cuando la solicitud sea aceptada por un piloto, deberá llegar al cliente un mensaje de confirmación.

- Piloto
 - La aplicación tendrá una versión para el piloto de taxi.
 - Podrá existir un número infinito de taxis, por lo que la aplicación no tendrá restricciones en el número de usuarios.
 - El piloto deberá registrarse para poder utilizar la aplicación.
 - El piloto podrá indicar el punto de inicio y punto final de un recorrido, por medio de GPS.

- La aplicación deberá calcular el costo del recorrido e imprimir un recibo para constancia de pago.
- El piloto tendrá la opción de aceptar o rechazar un recorrido.
- Formularios
 - Registro: los usuarios, tanto cliente como piloto, deberán registrarse y proporcionar los siguientes datos:
 - Cliente
 - ✓ Nombre de usuario
 - ✓ Correo electrónico
 - ✓ Teléfono
 - Piloto
 - ✓ Usuario
 - ✓ Número de identificación
 - Favoritos: para almacenar las ubicaciones favoritas deberá proporcionar los siguientes datos:
 - Nombre del favorito

4.3. Diseño intuitivo y usabilidad

Parte del éxito de una aplicación es su facilidad de uso. Las aplicaciones móviles tienen la ventaja de poseer una interfaz táctil, lo cual contribuye a su usabilidad y diseño por la manera natural de interactuar con el dispositivo.

La usabilidad debe abordar tres dificultades clave:

- La pantalla de tamaño limitada que posee el dispositivo móvil, se debe distribuir los elementos necesarios sin sobrecargar el espacio.

- La falta de herramientas de software específicas, las aplicaciones móviles no poseen software adecuado para evaluaciones de usabilidad. Existen algunos que poseen limitantes como grabar el manejo e interacción con el dispositivo físico.
- Dificultades adicionales derivadas de un contexto móvil, el usuario y el teléfono se encuentran en movimiento constante, esto significa que el entorno es cambiante. Otra dificultad es que las tareas del usuario se encuentran sujetas a interrupciones, como pérdida de cobertura, llamadas entrantes, baja batería o distracciones.

4.3.1. Herramientas utilizadas en la evaluación de usabilidad

Un factor importante para el éxito de una aplicación móvil es la usabilidad, ya que debido a fallos de este tipo, como gestos poco intuitivos y mala administración del diseño, muchos usuarios desechan algunas aplicaciones y optan por aplicaciones más eficientes.

4.3.1.1. Evaluaciones con expertos

Los expertos realizan una evaluación, en la cual aplican una lista con criterios y conocimientos sobre la experiencia de los usuarios.

Existe un conjunto de heurísticas definidas según el tipo de dispositivo (teléfono inteligente, *tablet* y otros) o según la plataforma (sistema operativo).

4.3.1.2. Evaluaciones heurísticas

Es una técnica basada en la inspección, en la cual los especialistas examinan la interfaz de usuario y la confrontan con un conjunto de criterios heurísticos seleccionados de manera específica y minuciosa.

Un ejemplo de heurísticas a seguir son las siguientes.

Tabla VII. **Evaluaciones heurísticas**

Heurística Móvil	Descripción
Heurística 1	Visibilidad del estado del sistema y perdida/búsqueda del dispositivo móvil
Heurística 2	Correspondencia entre el sistema y el mundo real
Heurística 3	Consistencia y mapeo
Heurística 4	Buena ergonomía y diseño minimalista
Heurística 5	Facilidad de entrada, lectura y vista de pantalla
Heurística 6	Flexibilidad, eficiencia de uso y personalización
Heurística 7	Estética, privacidad y convencionalismos sociales
Heurística 8	Gestión realista de errores

Fuente: elaboración propia.

4.3.1.3. Recorridos cognitivos

El especialista recorre un escenario de tareas determinado tal como lo haría un usuario potencial, es decir que asume el papel de usuario. La ventaja de realizar estas aplicaciones es que los expertos no sólo empatizan con los usuarios finales de la aplicación, sino también pueden contribuir a mejorar dicha

evaluación en lo que se refiere a la adquisición de datos tanto cuantitativos como cualitativos.

4.3.1.4. Test de usabilidad

Es un proceso que permite conocer a los usuarios lo usable que es una aplicación, observando el test y monitoreándolo mientras lo utilizan. Los test de usabilidad intentan recaudar la mayor información de la interacción del usuario con la aplicación y el dispositivo móvil, monitoreando sus reacciones.

Los test de usabilidad para dispositivos móviles siguen las mismas métricas que los test convencionales para aplicaciones de escritorio, pero poseen dificultades adicionales derivadas de sus particularidades: limitaciones físicas, la pantalla pequeña con la que se interactúa, el factor movilidad y otros.

4.3.1.5. Equipos montados en la cabeza

Existen dispositivos que se utilizan en la cabeza, para monitorear las reacciones del usuario frente a una aplicación.

Uno de estos dispositivos son gafas, que incorporan tecnología de seguimiento de ojos, utilizando tanto la cámara como luces infrarrojas. Las gafas obtienen la información y la transmiten mediante *wifi* o *bluetooth*. Este método simula un entorno más natural, pero debido a que el usuario no está centrado en ver únicamente el dispositivo móvil, se pierde mucha información.

Figura 16. **Gafas comerciales para test de dispositivos móviles**



Fuente: <http://www.eyetracking-glasses.com/>. Consulta: 2 de mayo de 2013.

Figura 17. **Captura de información mediante gafas**



Fuente: <http://www.eyetracking-glasses.com/>. Consulta: 2 de mayo de 2013.

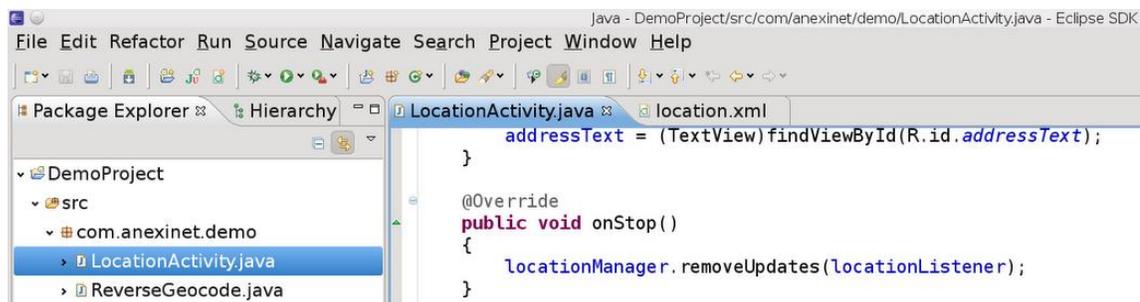
5. DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN

5.1. Software

Son todas las herramientas intangibles que se utilizaron para el desarrollo de la aplicación. Las diversas herramientas de software utilizadas poseen licenciamiento libre.

- ADT: el paquete ADT contiene diversas herramientas que se utilizan en el desarrollo de aplicaciones Android. El paquete incluye lo siguiente:
 - Eclipse Mobile: IDE para desarrolladores móviles, posee la capacidad de integrarse con el SDK Android y el agregado Manager SDK.

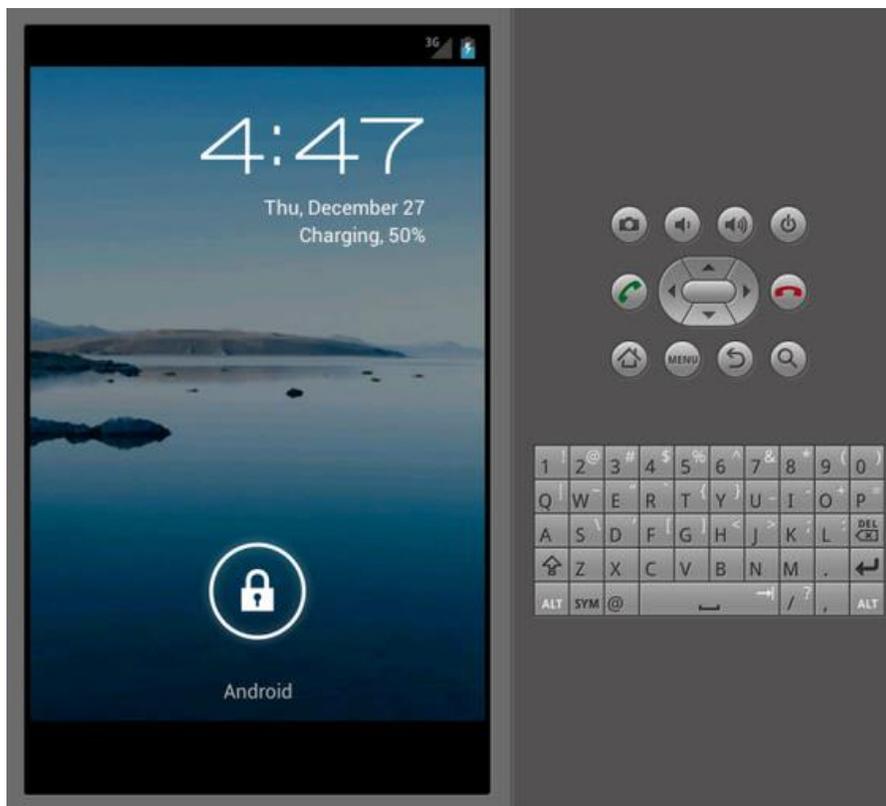
Figura 18. IDE Eclipse



Fuente: elaboración propia.

- SDK Android: proporciona las bibliotecas API y herramientas de desarrollo necesarias para crear, probar y depurar aplicaciones para Android, con la ayuda de un emulador o un dispositivo físico que funcione con el sistema operativo Android.

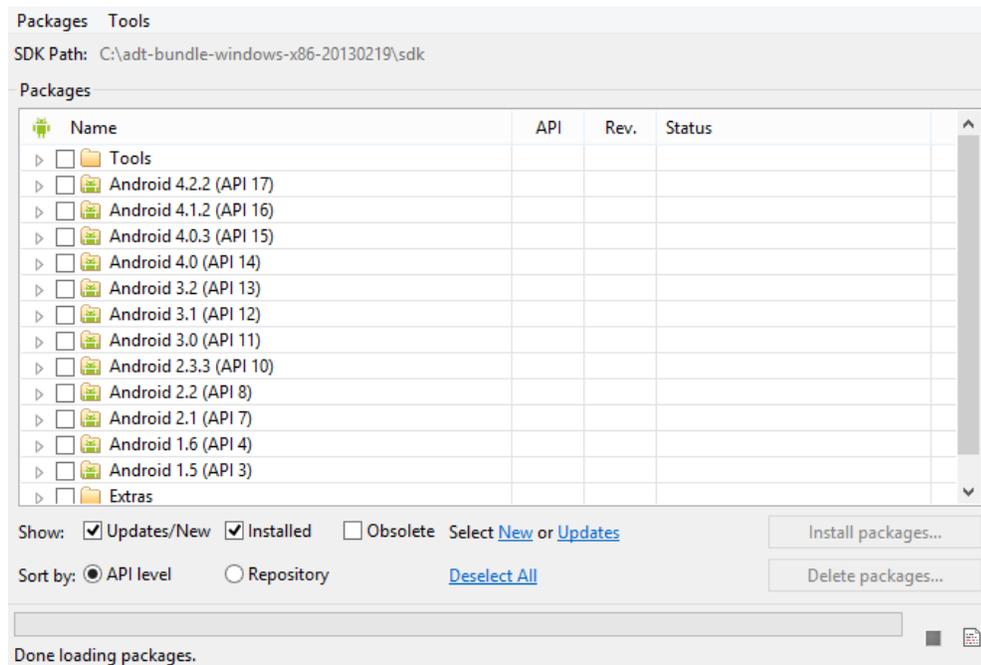
Figura 19. **Emulador SDK de Android**



Fuente: elaboración propia.

- *SDK Manager*: interfaz para la gestión de los diversos SDK disponibles, desde la versión 1.5 hasta la versión actual, posee la opción de instalar librerías de soporte y permite actualizar los SDK cuando existen actualizaciones lanzadas al mercado.

Figura 20. Manejador de SDK



Fuente: elaboración propia.

5.2. Instalación

Para el proceso de instalación del paquete ADT de Android se debe seguir una serie de pasos dependiendo el sistema operativo utilizado. Para el sistema operativo Windows 7 se debe realizar en el orden que se describe a continuación.

- Ingresar al *link* de descarga
<http://developer.android.com/intl/es/sdk/index.html>
- Clic en Descargar el SDK (*Download the SDK*).

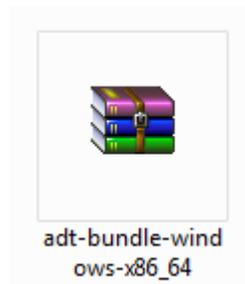
Figura 21. **Descarga el SDK Android**



Fuente: elaboración propia.

- Se descarga un archivo .zip el cual se descomprime y se copia en cualquier localidad.

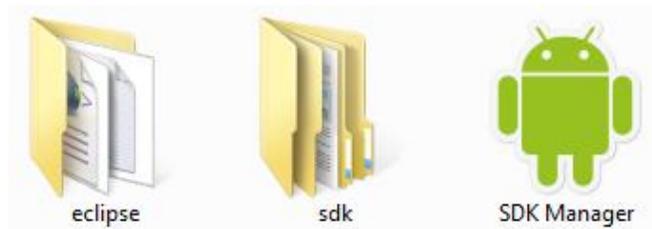
Figura 22. **ADT-Bundle.zip**



Fuente: elaboración propia.

- En la localidad donde se descomprime, estarán presentes 3 elementos
 - Eclipse, IDE de desarrollo
 - SDK, herramientas para desarrollo sobre Android
 - *SDK Manager*, gestor de APIs Android que son indispensables para el desarrollo

Figura 23. **Herramientas desarrollo Android**



Fuente: elaboración propia.

5.3. Desarrollo

Para iniciar el proceso de desarrollo se debe crear un nuevo proyecto móvil en Eclipse, el cual utiliza una versión SDK previamente configurada y crea una actividad sobre la cual se manejarán los componentes Android.

- Abrir el IDE de desarrollo, Eclipse ->Eclipse.exe

Figura 24. IDE Eclipse para desarrollo Android



Fuente: elaboración propia.

- Crear un nuevo proyecto Archivo -> Nuevo -> Proyecto aplicación Android

Figura 25. Nuevo proyecto Android

New Android Application
Creates a new Android Application 

Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

Compile With:

Theme:

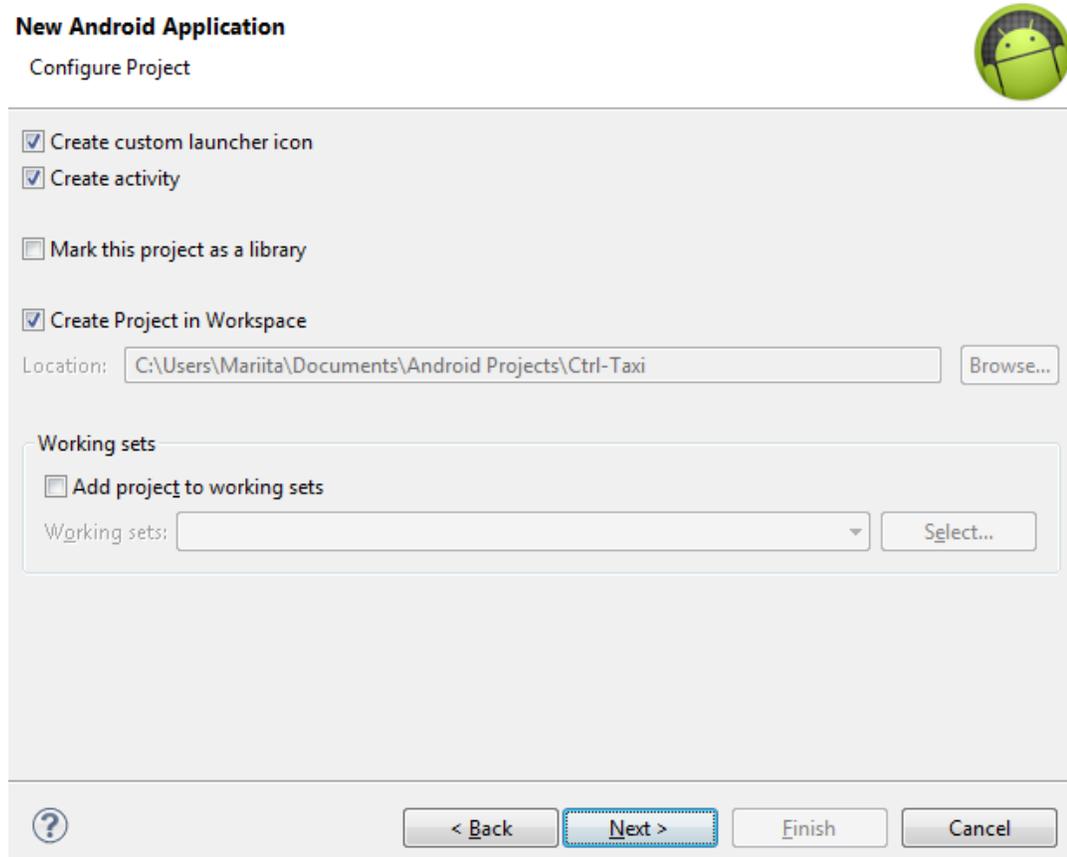
 Choose the highest API level that the application is known to work with. This attribute informs the system that you have tested against the target version and the system should not enable any compatibility behaviors to maintain your app's forward-compatibility with the target version. The application is still able to run on older versions (down to minSdkVersion). Your application may look dated if you are not targeting the current



Fuente: elaboración propia.

- Configuración del proyecto

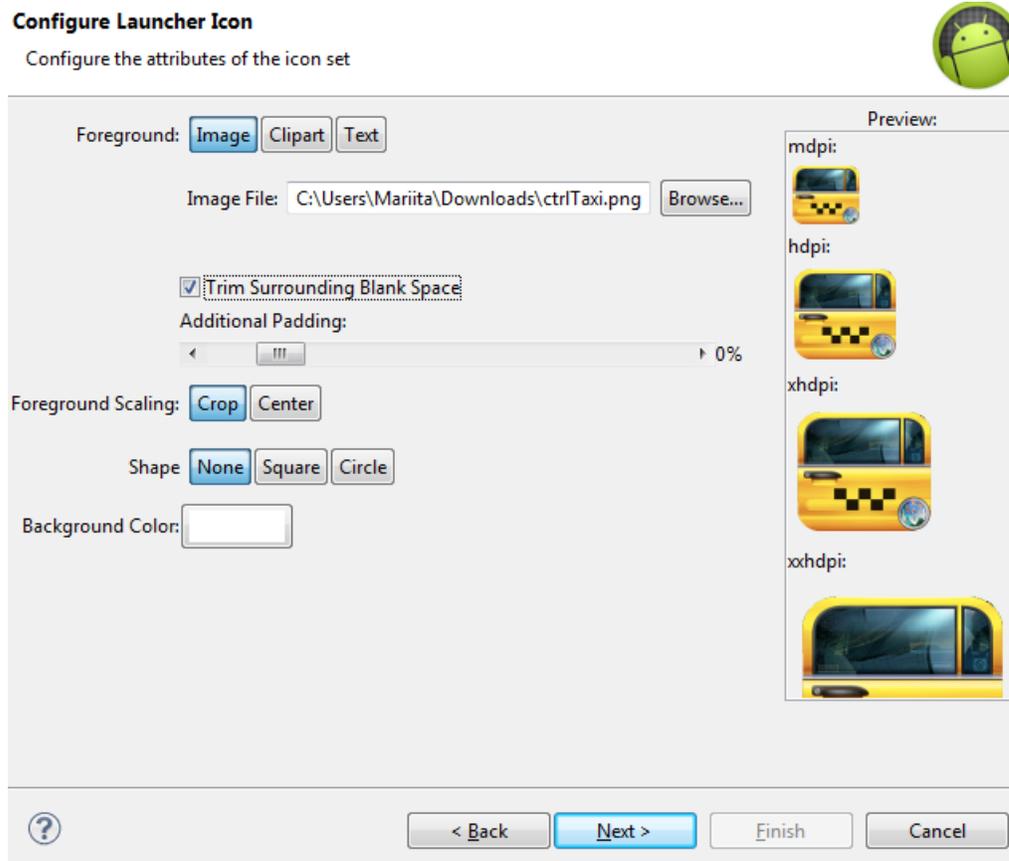
Figura 26. **Configuración del Proyecto Android**



Fuente: elaboración propia.

- Selección de la imagen distintiva para la aplicación Android utilizada para mostrar al usuario como icono de la aplicación.

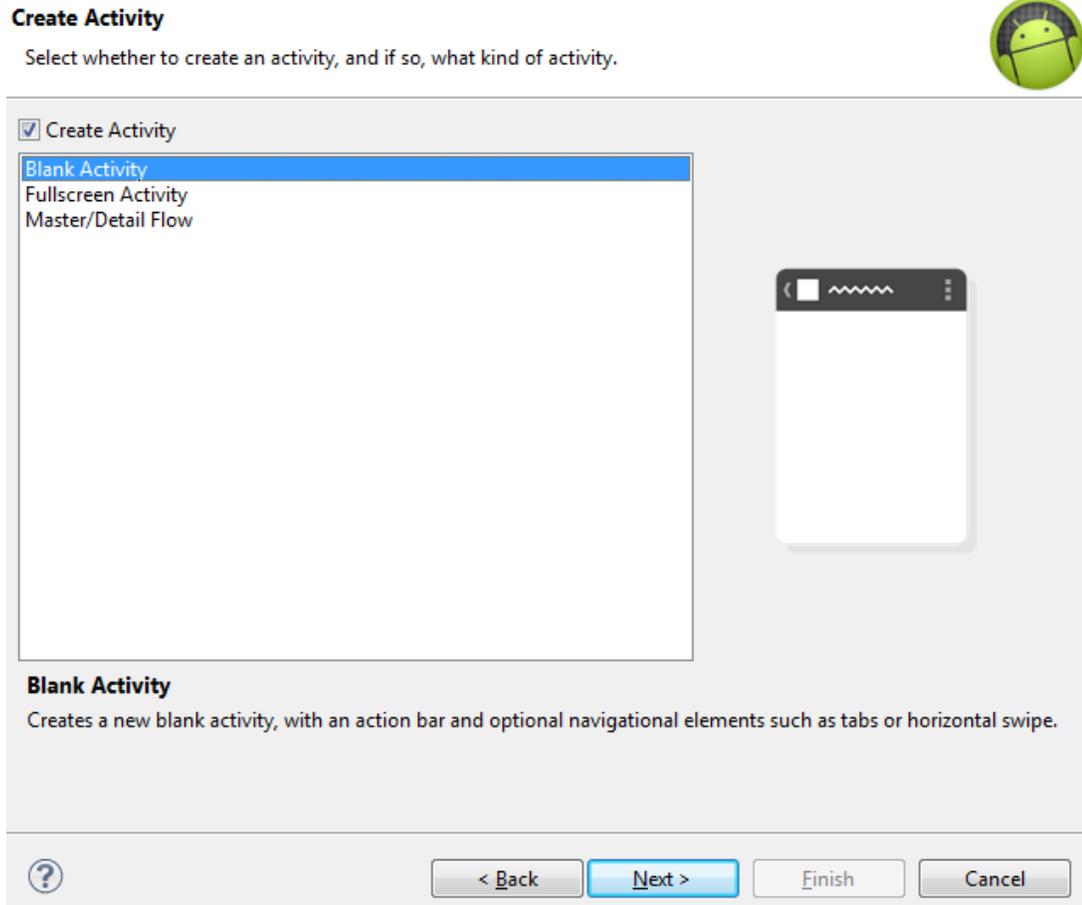
Figura 27. **Imagen identificativa para la aplicación Android**



Fuente: elaboración propia.

- Seleccionar el tipo de actividad principal de la aplicación.

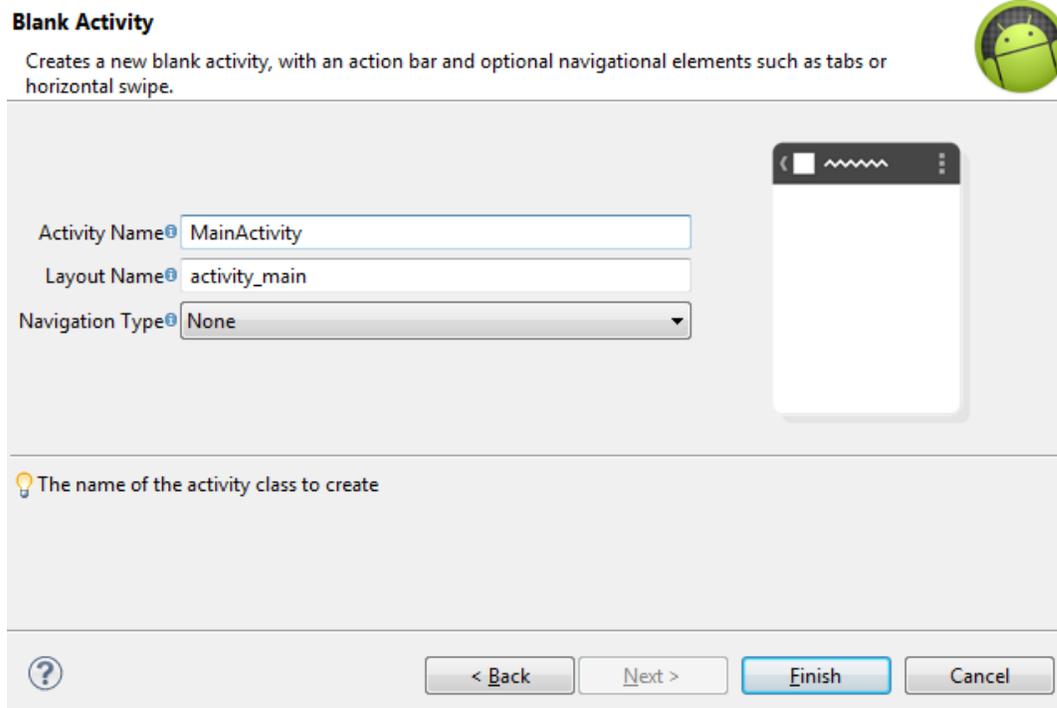
Figura 28. Creación de actividad principal



Fuente: elaboración propia.

- Especificación del nombre de la actividad principal. Esta será la pantalla inicial de la aplicación.

Figura 29. **Nombre actividad principal**



Fuente: elaboración propia.

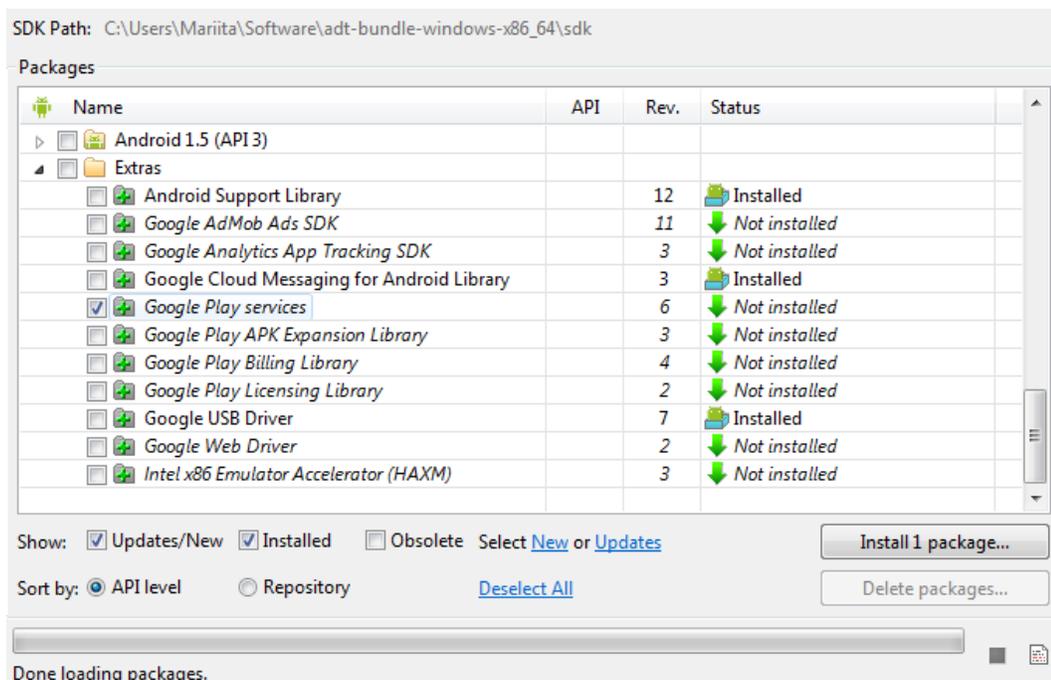
5.4. Google Maps

Es un servicio de mapas que proporciona Google de forma gratuita, publicado desde 6 de octubre de 2005. Proporciona fotografías satelitales e imágenes desplazables de mapas de todo el mundo.

Para la utilización de este servicio se deben seguir los siguientes pasos:

- Ya con la aplicación creada, se procede a la implementación de Google maps en la aplicación Android. Ingresar al SDK *manager*, Ventana -> SDK *Manager*, y seleccionar Google *Play Services*, iniciar la descarga e instalar.

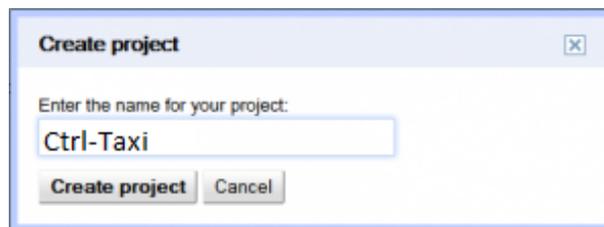
Figura 30. Descarga de servicios de Google *Play*



Fuente: elaboración propia.

- Al terminar de instalar *Google Play Service* se crea el siguiente directorio:
<carpeta_sdk_android>/extras/google/google_play_services
- Obtener una clave llamada *API key* que hace posible el uso de los servicios de mapas de Google en la aplicación. Acceder a la página <https://code.google.com/apis/console/>
- Pulsar sobre Crear, con lo cual se crea un nuevo proyecto especificando un nombre descriptivo para el proyecto.

Figura 31. **Creación de nuevo proyecto en la consola de APIs de Google**



Fuente: elaboración propia.

- Acceder a la opción Servicios, activar Google Maps Android API v2.

Figura 32. **Activación de API de Google maps**



Fuente: elaboración propia.

- Con este servicio activado, ir a la opción API Access, donde está la API Key que es la que permite usar el servicio de Google maps desde la aplicación.

Figura 33. **Obtención de API Key**

```
Key for Android apps (with certificates)
API key: AIzaSyBvLPCF6V5-Uoej6Te-6Vewo9NjxcU_adM
Android apps: 35:19:A6:6B:2C:FD:D0:D5:A4:A9:86:77:8B:7F:35:25:01:4B:F1:BE;ctrl.taxi.ctrl_taxi
Activated on: Mar 30, 2013 6:49 PM
Activated by: leonelrojasch@gmail.com - you
```

Fuente: elaboración propia.

- Ir a la opción Crear nueva llave Android, con esto aparece un diálogo donde se debe especificar la huella digital (SHA1) y el paquete java de la aplicación que se está desarrollando.

Toda aplicación Android debe estar firmada para ejecutarse en un dispositivo, ya sea un dispositivo físico o un dispositivo emulado, durante el proceso de desarrollo también es necesaria la firma pero es realizada automáticamente con el IDE el cual es un Certificado de pruebas. El certificado está almacenado en la ubicación que se puede obtener desde el IDE de desarrollo *Window->Preferences->Android->Build* con la extensión *.keystore*.

- Desde la consola de comandos (CMD) ejecutar el siguiente comando, estando actualmente en la ubicación donde se encuentra el *.keystore*.

```
C:\Program Files\Java\jdk1.7.0_07\bin\keytool.exe -list -v -keystore  
debug.keystore -alias androiddebugkey -storepass android -keypass  
android.
```

La ubicación que se especifica en el comando de consola, es la ubicación del JDK, se debe sustituir por la ruta del JDK del ambiente de desarrollo.

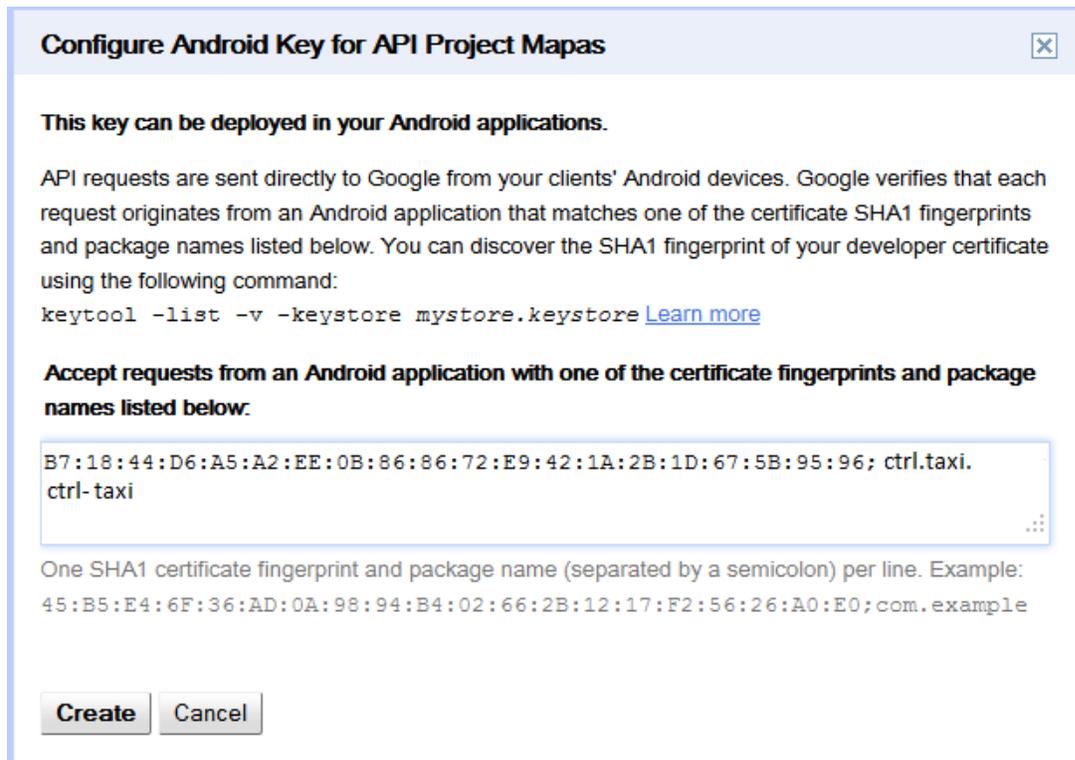
Figura 34. Generación de huella digital de ambiente de desarrollo

```
Nombre de Alias: androiddebugkey
Fecha de Creación: 30-nov-2011
Tipo de Entrada: PrivateKeyEntry
Longitud de la Cadena de Certificado: 1
Certificado[1]:
Propietario: CN=Android Debug, O=Android, C=US
Emisor: CN=Android Debug, O=Android, C=US
Número de serie: 4ed679f4
Válido desde: Wed Nov 30 19:46:12 CET 2011 hasta: Fri Nov 22 19:46:12 CET 2041
Huellas digitales del Certificado:
MD5: 5A:F0:7B:4F:75:88:BE:3C:B3:F0:52:0C:FA:94:4F:E6
SHA1: B7:18:44:D6:A5:A2:EE:0B:86:86:72:E9:42:1A:2B:1D:67:5B:95:96
SHA256: 50:EF:2E:51:0D:19:69:60:E3:2D:AE:4C:AE:EF:44:39:84:24:47:CA:C3:
71:D4:C8:9A:F2:8D:BC:97:09:2A:E9
Nombre del Algoritmo de Firma: SHA1withRSA
Versión: 3
```

Fuente: elaboración propia.

- Este dato se copia, en la ventana de configuración de API key se pega junto con el nombre del paquete de la aplicación separado por punto y coma.

Figura 35. Configuración de nueva API Key



Fuente: elaboración propia.

- Clic en crear para que los servicios de Google maps queden disponibles para el ambiente de desarrollo. El valor correspondiente a API key será utilizado para el desarrollo de la aplicación Android.
- Ya concluidos todos los preparativos para empezar el desarrollo, en el archivo *AndroidManifest.xml* se debe agregar la API key generada por los servicios de Google.

Figura 36. **Android manifest API key**

```
<application>
  <meta-data android:name="com.google.android.maps.v2.API_KEY"
    |         |         |         android:value="api_key"/>
  |         |         |
</application>
```

Fuente: elaboración propia.

Permisos necesarios para el acceso a Google maps desde la aplicación.

Figura 37. **Permisos para acceso Google maps**

```
<permission
  | android:name="ctrl.taxi.android.mapasapi2.permission.MAPS_RECEIVE"
  | android:protectionLevel="signature"/>
<uses-permission android:name="ctrl.taxi.android.mapasapi2.permission.MAPS_RECEIVE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<uses-feature android:glEsVersion="0x00020000" android:required="true"/>
```

Fuente: elaboración propia.

- Añadir un proyecto externo con la opción de eclipse Archivo->Importar...->Código existente Android al área de trabajo, el proyecto a importar está localizado en la ruta:
<carpeta-sdk-android>/extras/google/google_play_services/libproject/google-play-services_lib.

- Ya está lista toda la configuración, es hora de crear la vista y lógica del mapa, para la vista se hace uso de un archivo xml.

Figura 39. **Creación de vista de mapa**

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    class="com.google.android.gms.maps.SupportMapFragment"/>
```

Fuente: elaboración propia.

Y una clase java que hereda de fragmentActivity en la cual se establece la vista taxirequest que es el nombre del archivo xml.

Figura 40. **Actividad controladora del mapa**

```
public class TaxiRequest extends android.support.v4.app.FragmentActivity
    implements LocationListener{

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.taxirequest);
    }
}
```

Fuente: elaboración propia.

- Para agregar la funcionalidad e interacción con el mapa, se pueden implementar escuchadores de acciones como una pulsación larga en el mapa:

Figura 41. **Evento pulsación larga**

```
mapa.setOnMapLongClickListener(new OnMapLongClickListener() {  
    public void onMapLongClick(LatLng point) {  
        Projection proj = mapa.getProjection();  
        Point coord = proj.toScreenLocation(point);  
  
        Toast.makeText(  
            MainActivity.this,  
            "Click Largo\n" +  
            "Lat: " + point.latitude + "\n" +  
            "Lng: " + point.longitude + "\n" +  
            "X: " + coord.x + " - Y: " + coord.y,  
            Toast.LENGTH_SHORT).show();  
    }  
});
```

Fuente: elaboración propia.

Agregar marcadores para resaltar puntos importantes o de interés en el mapa.

Figura 42. **Incorporación de marcador al mapa**

```
private void mostrarMarcador(double lat, double lng)
{
    mapa.addMarker(new MarkerOptions()
        .position(new LatLng(lat, lng))
        .title("Ubicacion de interes Ctrl-Taxi"));
}
```

Fuente: elaboración propia.

Acciones para la ventana de información de los marcadores.

Figura 43. **Ventana informativa de marcadores en el mapa**

```
private void setInfoWindowClickListener() {
    this.map.setOnInfoWindowClickListener(
        new OnInfoWindowClickListener() {
            public void onInfoWindowClick(Marker marker) {
            }
        }
    );
}
```

Fuente: elaboración propia.

Centrar la cámara en la ubicación actual

Figura 44. **Centrar cámara en posición actual**

```
Location location = locationManager.getLastKnownLocation(provider);
LatLng actualPosition = new LatLng(location.getLatitude(),
    location.getLongitude());
CameraPosition camPos1 = new CameraPosition.Builder()
    .target(actualPosition)
    .zoom(10)
    .bearing(45) //Orientacion, 0 orientado al norte
    .tilt(70) //inclinacion, inclinado 70 grados
    .build();
CameraUpdateFactory.newCameraPosition(camPos1);
```

Fuente: elaboración propia.

Tipos de mapa que se pueden seleccionar.

Figura 45. **Tipos de mapa disponibles**

```
map.setMapType(GoogleMap.MAP_TYPE_NORMAL);
map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
map.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
map.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
```

Fuente: elaboración propia.

5.5. Google Cloud Messaging

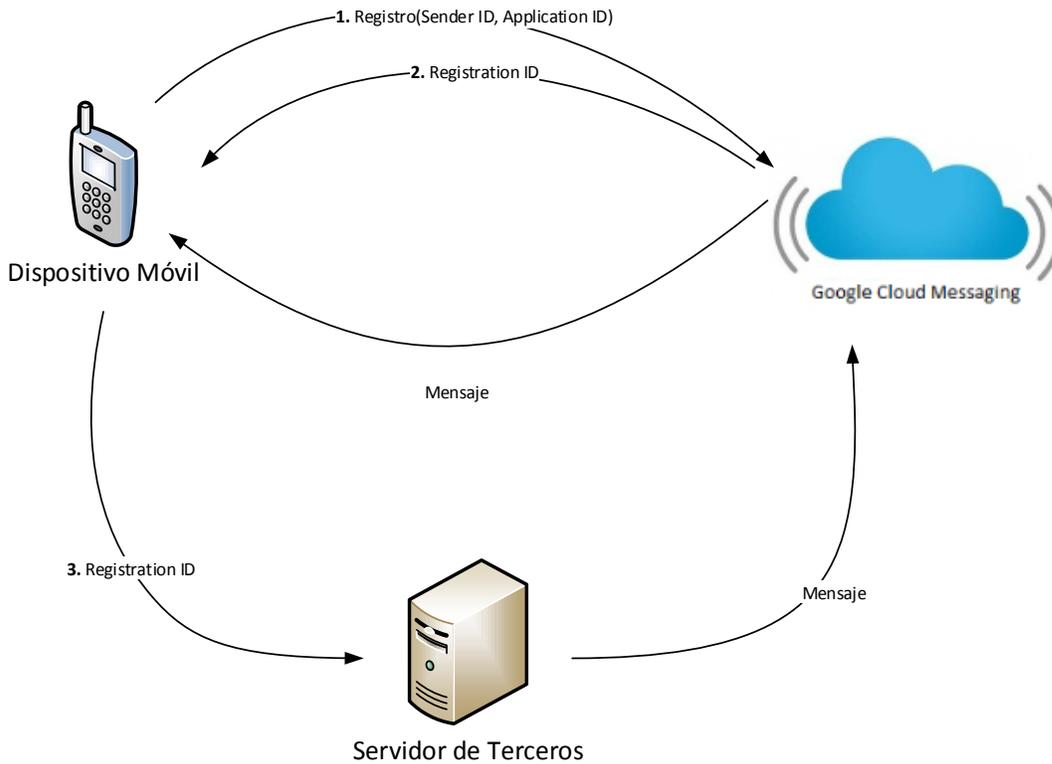
Es un servicio gratuito que Google ofrece para envío de información, entre un servidor y un dispositivo Android. Google *Cloud Messaging* (GCM) se encarga del control de cola de mensajes y de la transmisión a los dispositivos.

5.5.1. Flujo de transmisión

Google *Cloud Messaging* para lograr una comunicación efectiva entre el dispositivo móvil y el servidor, realiza una serie de pasos, la trayectoria del proceso se describe a continuación:

- Se registra la aplicación, utilizando un *Intent* de registro, el cual incluye en *Sender ID* y el *Application ID*.
- El servidor Google *Cloud Messaging* realiza un *Intent* con el cual devuelve el *Registration ID*.
- La aplicación manda el *Registration ID* al servidor propio.
- El servidor manda en mensaje que será recibido por el dispositivo a Google *Cloud Messaging*, el cual se encarga de administrarlo.
- Google *Cloud Messanging* envía el mensaje a la aplicación.
- El dispositivo se encarga de mandar el mensaje a la aplicación mediante un *Intent*. El mensaje se procesa aún si la aplicación no estuviese ejecutándose.

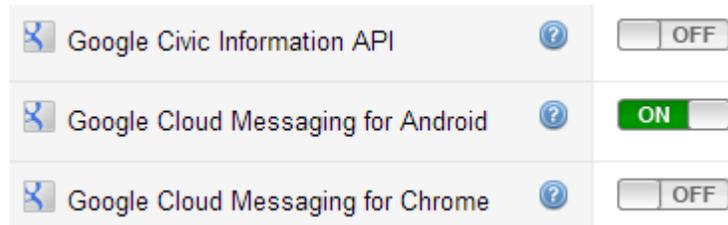
Figura 46. Flujo de transmisión GCM



Fuente: elaboración propia.

Para la integración de GCM en el proyecto Ctrl-Taxi, acceder a <https://code.google.com/apis/console/> donde ya está registrado el proyecto con el API de mapas activado. Para la activación de GCM se usa el mismo proceso, activar el servicio de Google *Cloud Messaging* for Android.

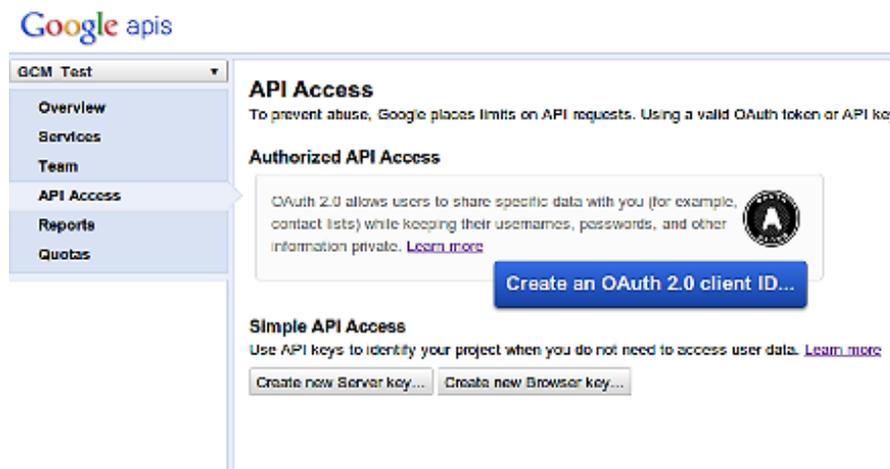
Figura 47. **Activación servicio GCM**



Fuente: elaboración propia.

En la opción API access es muestra una pantalla siguiente:

Figura 48. **Creación de llave de servidor**



Fuente: elaboración propia.

En esta pantalla seleccionar Crear una nueva llave de servidor (*Create new Server key*), con una llave de servidor se tiene la ventaja de manejar listas de direcciones IP permitidas, un diálogo para especificar las IP permitidas es mostrado justo al hacer clic en crear la nueva llave de servidor, para este caso continuar con el proceso haciendo clic en Crear (*Create*).

Figura 49. **Configuración de llave de servidor para el proyecto**

Configure Server Key for My Project

This key should be kept secret on your server.

Every API request is generated by software running on a machine that you control. Per-user limits will be enforced using the address found in each request's `userIp` parameter, (if specified). If the `userIp` parameter is missing, your machine's IP address will be used instead. [Learn more](#)

Accept requests from these server IP addresses:

Example: 192.168.12.0/23. One IP address or subnet per line.

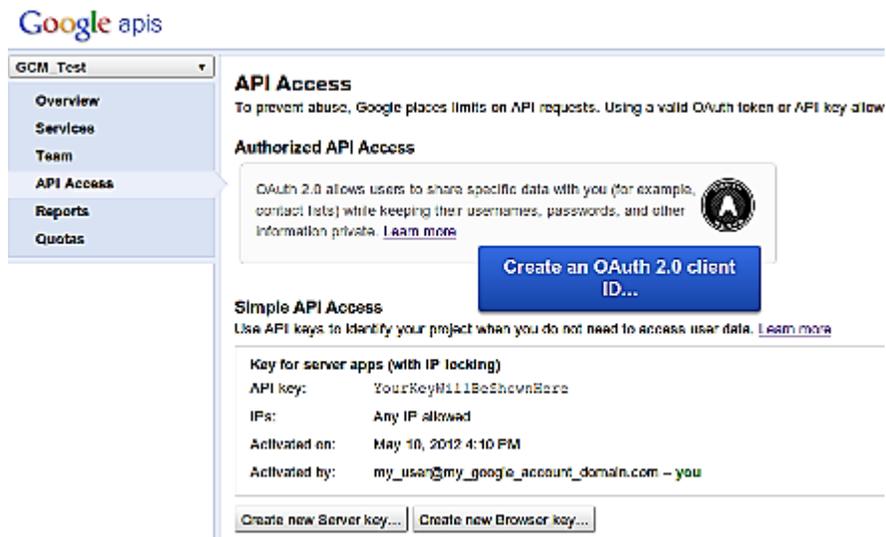
Create **Cancel**

Fuente: elaboración propia.

Al crear la llave se tendrán valores de interés en la pantalla como:

- API key
- IPs permitidas
- Fecha de activación
- Activada por (usuario que crea el acceso)

Figura 50. Llave de servidor creada



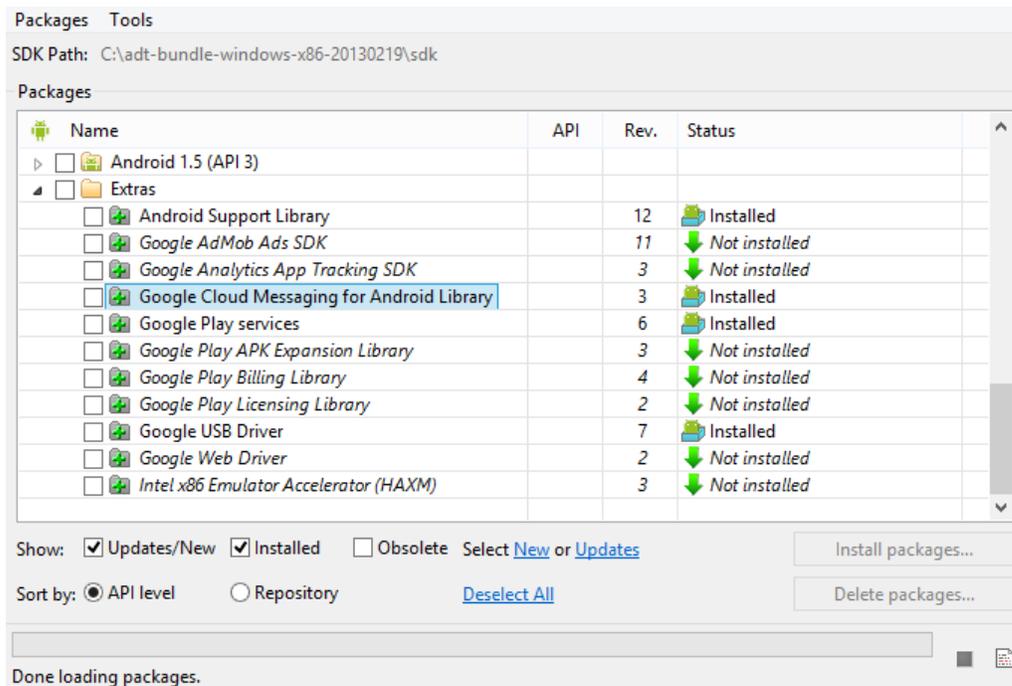
Fuente: elaboración propia.

Toda la configuración del lado del servicio de Google ya está realizada.

5.5.2. Aplicación Android lado del cliente

Descargar las librerías de apoyo para el manejo de GCM en la aplicación *Android*. Para este efecto en el gestor de SDK dirigirse a Extras -> Google *Cloud Messaging for Android Library*.

Figura 51. Descarga de servicios de GCM para la aplicación Android



Fuente: elaboración propia.

La instalación de estas librerías de apoyo crea un directorio en el la ubicación raíz del SDK /extras/google/ en este directorio se crean los siguientes subdirectorios:

- *gcm-client*
- *gcm-server*
- *samples/gcm-demo-client*
- *samples/gcm-demo-server*
- *samples/gcm-demo-appengine*

Serán usados para la implementación de GCM. En el directorio *gcm-client/dist* se encuentra el archivo *gcm.jar* el cual debe ser agregado al *classpath* del proyecto del lado del cliente.

Gcm requiere de Android 2.2 o más actual, para delimitar las versiones de Android que pueden usar esta aplicación, agregar en el archivo *Android-Manifest*.

Figura 52. **Versión mínima Android SDK**

```
<uses-sdk android:minSdkVersion="8" android:targetSdkVersion="xx"/>
```

Fuente: elaboración propia.

Permisos para la recepción de mensajes GCM.

Figura 53. **Permisos recepción de mensajes GCM**

```
<permission android:name="my_app_package.permission.C2D_MESSAGE"  
    android:protectionLevel="signature" />  
<uses-permission android:name="my_app_package.permission.C2D_MESSAGE" />
```

Fuente: elaboración propia.

La palabra *my_app_package* debe ser sustituida por el nombre del paquete de la aplicación tal y como se define en el archivo *Android-Manifest*, si esto no se cumple la aplicación no funcionará.

Permisos para la aplicación:

Figura 54. **Permisos Android-*Manifest* GCM**

```
<!-- Aplicacion recibe mensajes GCM. -->
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
<!-- GCM se conecta a los servicios de Google. -->
<uses-permission android:name="android.permission.INTERNET" />
<!-- GCM requiere una cuenta de Google. -->
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<!-- Mantiene el procesador activo cuando un mensaje ha sido recibido -->
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

Fuente: elaboración propia.

En el mismo archivo *Android-Manifest* agregar el siguiente *Broadcast receiver*, dentro de los *tags application*.

Figura 55. **Configuración Android-*Manifest Broadcast Receiver***

```
<receiver android:name="com.google.android.gcm.GCMBroadcastReceiver"
  android:permission="com.google.android.c2dm.permission.SEND" >
  <intent-filter>
    <action android:name="com.google.android.c2dm.intent.RECEIVE" />
    <action android:name="com.google.android.c2dm.intent.REGISTRATION" />
    <category android:name="my_app_package" />
  </intent-filter>
</receiver>
```

Fuente: elaboración propia.

La palabra *my_app_package* debe ser sustituida por el nombre del paquete de la aplicación tal y como se define en el archivo *Android-Manifest*, si esto no se cumple la aplicación no funcionará.

La función del *broadcast receiver* es manejar 2 intentos que pueden ser mandados por GCM, el intento de recibir y el intento de registrar, estos deben ser definidos en el archivo *Android-Manifest* en lugar de programáticamente para que la aplicación pueda recibir alguno de estos dos intentos incluso si la aplicación no se está ejecutando.

Agregar el siguiente servicio en el *Android-Manifest*, este servicio será llamado por el *GCMBroadcastReceiver*, el cual está contenido en las librerías de GCM.

Figura 56. **Configuración servicio *GCMIntentService***

```
<service android:name=".GCMIntentService" />
```

Fuente: elaboración propia.

Escribir la clase *GCMIntentService* la cual debe sobrescribir los métodos:

- *onRegistered*
- *onUnregistered*
- *onMessage*
- *onError*
- *onRecoverableError*

Los métodos antes nombrados están ejecutándose en el hilo de *intent service*, por lo que son libres de hacer llamadas a la red sin correr el riesgo de bloquear el hilo gráfico de la aplicación.

Figura 57. Clase *GCMIntentService*, sobrescribiendo métodos

```
public class GCMIntentService extends com.google.android.gcm.GCMBaseIntentService{

    private static final String TAG = "Ctrl-Taxi";

    public GCMIntentService() {
        super(CommonUtilities.SENDER_ID);
    }

    @Override
    protected void onRegistered(Context context, String registrationId) {
        Log.i(TAG, "Device registered: regId = " + registrationId);
        //acciones del registro del dispositivo
    }

    @Override
    protected void onUnregistered(Context context, String registrationId) {
        Log.i(TAG, "Device unregistered");
        //acciones para el desregistro del dispositivo
    }

    @Override
    protected void onMessage(Context context, Intent intent) {
        //acciones para la recepcion de un mensaje
    }

    @Override
    protected void onDeletedMessages(Context context, int total) {
        Log.i(TAG, "Received deleted messages notification");
        //acciones para la recepcion de mensajes eliminados
    }

    @Override
    public void onError(Context context, String errorId) {
        Log.i(TAG, "Received error: " + errorId);
        //acciones para la recepcion de un mensaje de error
    }

    @Override
    protected boolean onRecoverableError(Context context, String errorId) {
        Log.i(TAG, "Received recoverable error: " + errorId);
        //acciones para la recepcion de un error recuperable
    }
}
```

Fuente: elaboración propia.

En la actividad principal de la aplicación, hacer las verificaciones y acciones correspondientes al registro del dispositivo.

Figura 58. **Actividad principal chequeo registro GCM**

```
GCMRegistrar.checkDevice(this);
GCMRegistrar.checkManifest(this);
final String regId = GCMRegistrar.getRegistrationId(this);
if (regId.equals("")) {
    GCMRegistrar.register(this, SENDER_ID);
} else {
    Log.v(TAG, "Dispositivo ya registrado");
}
```

Fuente: elaboración propia.

La palabra *SENDER_ID* debe sustituirse por el número que aparece en la dirección URL cuando se ingresa a la consola de APIs, es el número al final de la URL: <https://code.google.com/apis/console/#project:4815162342>

El método *checkDevice()* verifica que el dispositivo soporta GCM y lanza una excepción en caso contrario, el método *checkManifest()* verifica que el archivo *Android-Manifest* de la aplicación cumpla con todos los requerimientos de GCM cubiertos en esta guía, *GCMRegistrar.getRegistrationId(this)* se encarga de obtener el id de registro del dispositivo en caso ya está registrado ahora bien si el dispositivo no está registrado retorna una cadena vacía.

Para mandar un mensaje desde el dispositivo móvil al servidor se debe hacer uso de un hilo auxiliar sobre el cual se hará la petición al servidor con el siguiente método:

Figura 59. Envío de petición al servidor

```
public static void post(String endpoint, Map<String, String> params) throws IOException {
    URL url;
    try {
        url = new URL(endpoint);
    } catch (MalformedURLException e) {
        throw new IllegalArgumentException("URL invalida: " + endpoint);
    }
    StringBuilder bodyBuilder = new StringBuilder();
    Iterator<Entry<String, String>> iterator = params.entrySet().iterator();
    // construir el contenido del POST con los parametros 'params'
    while (iterator.hasNext()) {
        Entry<String, String> param = iterator.next();
        bodyBuilder.append(param.getKey()).append('=')
            .append(param.getValue());
        if (iterator.hasNext()) {
            bodyBuilder.append('&');
        }
    }
    String body = bodyBuilder.toString();
    Log.v(CommonUtilities.TAG, "Enviando '" + body + "' a '" + url);
    byte[] bytes = body.getBytes();
    HttpURLConnection conn = null;
    try {
        conn = (HttpURLConnection) url.openConnection();
        conn.setDoOutput(true);
        conn.setUseCaches(false);
        conn.setFixedLengthStreamingMode(bytes.length);
        conn.setRequestMethod("POST");
        conn.setRequestProperty("Content-Type", "application/x-www-form-urlencoded;
        charset=UTF-8");

        // envia la petición
        OutputStream out = conn.getOutputStream();
        out.write(bytes);
        out.close();
        // maneja la respuesta
        int status = conn.getResponseCode();
        if (status != 200) {
            throw new IOException("Envío fallido con código de error " + status);
        }
    } finally {
        if (conn != null) {
            conn.disconnect();
        }
    }
}
}
```

Fuente: elaboración propia.

5.5.3. Aplicación del lado del servidor

En el directorio *gcm-server/dist* se encuentra el archivo *gcm-server.jar* el cual debe ser agregado al *classpath* del proyecto del servidor.

Se debe crear un *servlet* que pueda ser usado por la aplicación Android para enviar el ID de registro recibido por GCM, este ID será almacenado en una base de datos para un uso posterior, el ID del dispositivo podrá ser asociado con otros datos como el nombre o correo electrónico.

Código para mandar un mensaje a un dispositivo con la aplicación instalada.

Figura 60. Envío de mensaje a dispositivo Android desde servidor

```
List<String> devices = new ArrayList<String>();
devices.add(ID_DISPOSITIVO);
Sender sender = new Sender(API_KEY);
Message msg = new Message.Builder()
    .collapseKey("1")
    .timeToLive(3)
    .delayWhileIdle(true)
    .addData("NOMBRE_PARAMETRO", "VALOR_DEL_PARAMETRO")
    .build();
try {
    MulticastResult result = sender.send(msg, devices, 5);
    for (Result r : result.getResults()) {
        if (r.getMessageId() != null) {
            String canonicalRegId = r.getCanonicalRegistrationId();
            if (canonicalRegId != null) {
                System.out.println("El mismo dispositivo tiene mas de un ID de registro, "+
                    "cual id es el correcto?");
            }
        } else {
            String error = r.getErrorCodeName();
            if (error.equals(Constants.ERROR_NOT_REGISTERED)) {
                System.out.println("No se encuentra el dispositivo con este ID, "+
                    "la aplicacion fue removida del dispositivo");
            }
        }
    }
} catch (IOException e) {
    e.printStackTrace();
}
```

Fuente: elaboración propia.

Con este código se puede mandar un mensaje a varios dispositivos a la vez, se pueden mandar datos de interés como parámetros a los cuales se les pone un nombre y un valor. Del lado del cliente los mensajes que se envían desde el servidor activan el método *onMessage* de la clase *GCMIntentService* entonces en ese método se deben poner las acciones específicas para interpretar los mensajes de manera correcta.

Ejemplo de las acciones para ejecutar en el método *onMessage*:

Figura 61. **Recepción de mensajes en aplicación Android**

```
@Override
protected void onMessage(Context context, Intent intent) {
    String action = intent.getAction();
    String valorParametroRecibido="";
    if ("com.google.android.c2dm.intent.RECEIVE".equals(action)) {
        .....
        valorParametroRecibido = intent.getStringExtra("NOMBRE_PARAMETRO");
    }

    Log.i(TAG, "Ctrl-Taxi, mensaje recibido");
}
```

Fuente: elaboración propia.

Se obtienen los valores de parámetros enviados desde el servidor, ya con estos valores se pueden realizar diversas acciones, según el caso en cuestión.

CONCLUSIONES

1. Se dio a conocer la historia de los dispositivos móviles, desde paginadores hasta sistemas de televisión, evaluando su desarrollo, ventajas, capacidades actuales, métodos de interacción.
2. Se presentó un prototipo que cubre los requerimientos planteados con diferentes características y capacidades para interactuar con la aplicación, se detectaron las validaciones y formularios necesarios enfocados a tener un diseño intuitivo y una usabilidad alta.
3. El análisis de las variables de aceptación tecnológica reveló que la aplicación tendrá un impacto positivo en la sociedad urbana.
4. Todas las aplicaciones similares existentes en el mercado poseen debilidades que otra complementa, su grado de aceptación ante otras sociedades ha sido a gran escala, solventando las deficiencias de los competidores el impacto y la aceptación será a mayor escala.
5. Se detalló el desarrollo de una plataforma para la gestión de taxis en el territorio guatemalteco, dando una guía con imágenes y descripciones de los pasos con lo que se puede desarrollar la plataforma tomando como ejemplo el tutorial y código proporcionado.

RECOMENDACIONES

1. Para la realización de pruebas con la aplicación móvil, hacer uso de un dispositivo físico que funcione con sistema operativo Android en lugar de un dispositivo virtual en el emulador provisto por Google o una máquina virtual, ya que se hace uso de hardware con el que no cuenta una computadora, como el GPS.
2. Contar con dos dispositivos móviles físicos con versión de Android 4.1 o posterior, ya que se requiere hacer pruebas del lado del cliente y del lado del piloto al mismo tiempo, lo cual no puede hacerse con un solo dispositivo.
3. Analizar detalladamente el ciclo de vida de los mensajes transmitidos por medio de GCM (*Google Cloud Messaging*) y sus capacidades, con el objetivo de tener presente las bondades y limitaciones de este servicio.

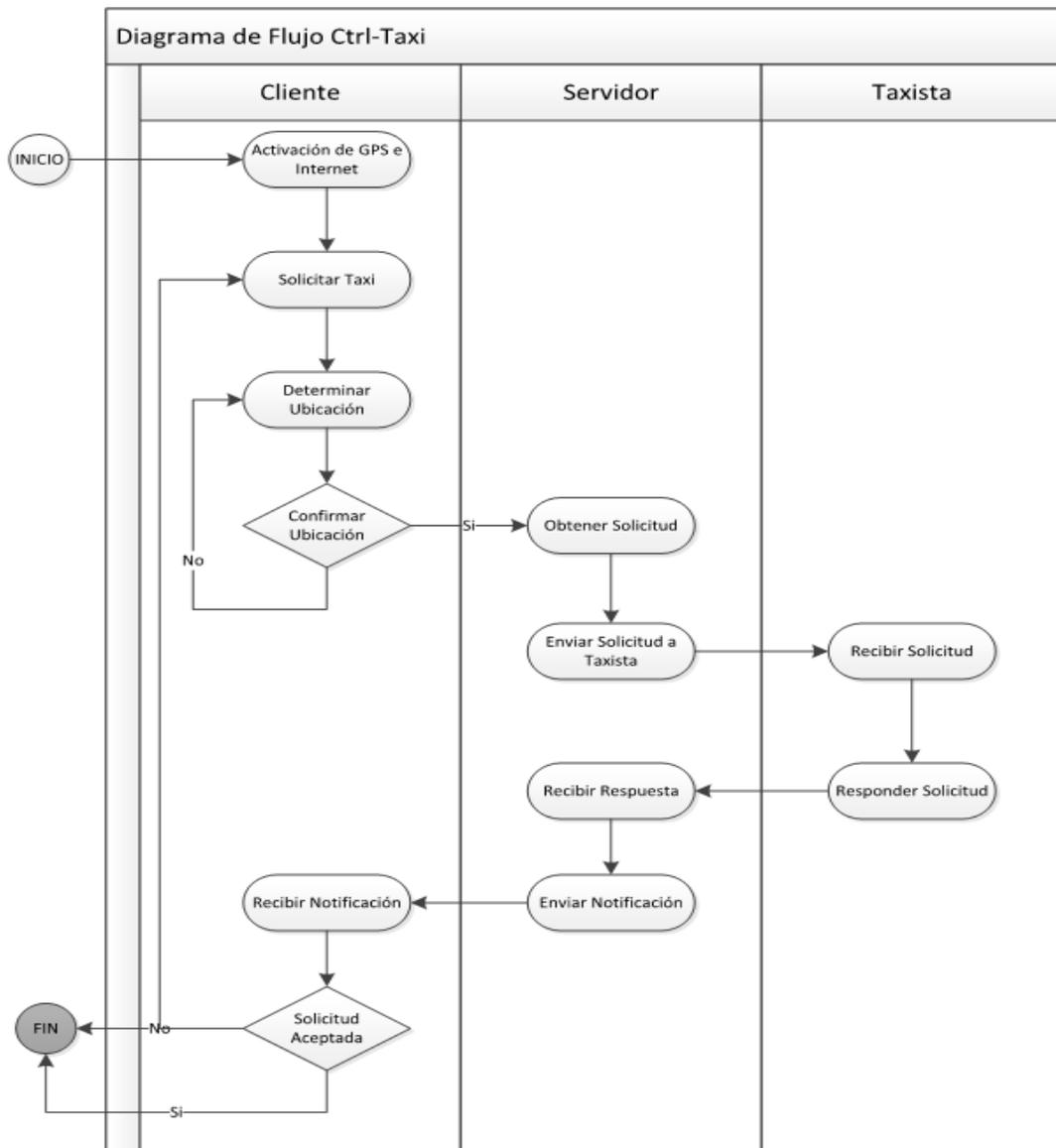
BIBLIOGRAFÍA

1. *Android developer*. [en línea] [ref. 18 de abril de 2013]. <<http://developer.android.com/intl/es/google/gcm/gs.html>>. [Consulta: 03 de abril de 2013].
2. Anónimo. *El Mundo*. [en línea] [ref. 13 de febrero de 2013]. <<http://www.elmundo.com.ve/Noticias/Mundo-Corporativo/Lanzamientos/Easy-Taxi-lanza-aplicacion-para-pedir-taxi-con-sol.aspx>>. [Consulta: 03 de mayo de 2013].
3. Anónimo. *Techbuzzes*. [en línea] [ref. 21 de marzo de 2013]. <<http://techbuzzes.com/2013/03/sony-xperia-tablet-z/>>. [Consulta: 03 de mayo de 2013].
4. Anónimo. *Xataka Movil*. [en línea] [ref. 18 de marzo de 2013]. <<http://www.xatakamovil.com/espaciomasmovil/pedir-un-taxi-desde-el-movil-aplicaciones-gratuitas>>. [Consulta: 03 de mayo de 2013].
5. BAZ ALONSO Arturo, FERREIRA ARTIME Irene, ÁLVAREZ RODRÍGUEZ María. *Dispositivos móviles*. [en línea] <<http://156.35.151.9/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf>>. [Consulta: 3 de abril de 2013].

6. HOLDEN Richard. *Science direct*. [en línea] [ref. febrero de 2010]. <<http://www.sciencedirect.com/science/article/pii/S1532046409000963>>. [Consulta: 03 de mayo de 2013].
7. *La razón*. [en línea] [ref. 18 de mayo de 2012]. <<http://www.larazonсанluis.com/sitio/nota.php?id=111259>>. [Consulta: 1 de mayo de 2013].
8. NEIRA Javier. *Emol*. [en línea] [ref. 12 de mayo de 2012]. <<http://www.emol.com/noticias/tecnologia/2012/05/09/539729/safe-r-taxi-aplicacion-que-moderniza-la-accion-de-pedir-un-radiotaxi.html>>. [Consulta: 03 de mayo de 2013].
9. SAAVEDRA NILO Esteban. *Slideshare*. [en línea] [ref. 10 de abril de 2012]. <<http://www.slideshare.net/esaaveni/benchmarking-que-es-paraquesirveycomohaceruno>>. [Consulta: 03 de mayo de 2013].
10. *Sprouts*. [en línea] [ref. 2009]. <<http://sprouts.aisnet.org/785/1/TAMReview.pdf>>. [Consulta: 03 de mayo de 2013].
11. VVENKATESG. *Venkat* [en línea] [ref. junio de 2012]. <http://www.vvenkatesh.com/it/organizations/theoretical_models.asp>. [Consulta: 03 de mayo de 2013].

APÉNDICE

Figura 62. Proceso de petición de un taxi



Fuente: elaboración propia.

