



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

TECNOLOGÍAS MÓVILES Y CONTROL PARENTAL CENTINELA

Marvin Alberto Villatoro Alonzo

Enio Fabrizio Torres Noguera

Asesorado por el Ing. Edgar Santos

Guatemala, noviembre de 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

TECNOLOGÍAS MÓVILES Y CONTROL PARENTAL CENTINELA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

MARVIN ALBERTO VILLATORO ALONZO

ENIO FABRIZIO TORRES NOGUERA

ASESORADO POR EL ING. EDGAR SANTOS

AL CONFERÍRSELES EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo 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. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

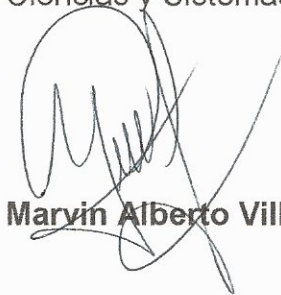
DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
EXAMINADOR	Ing. Marlon Francisco Orellana Lopez
EXAMINADOR	Ing. José Alfredo González Diaz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

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

TECNOLOGÍAS MÓVILES Y CONTROL PARENTAL CENTINELA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 10 de octubre del 2013.



Maryin Alberto Villatoro Alonzo



Enio Fabrizio Torres Noguera

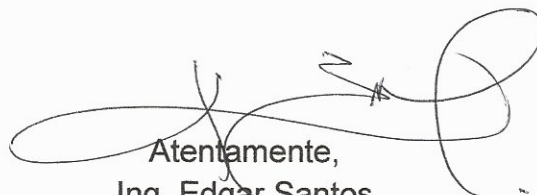
Guatemala, 13 de agosto de 2014

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

Ingeniero Pérez Türk:

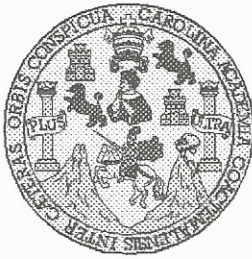
Me complace saludarle, haciendo referencia al trabajo de graduación titulado "TECNOLOGÍAS MÓVILES Y CONTROL PARENTAL CENTINELA", desarrollado por el estudiante Marvin Alberto Villatoro Alonzo con número de carné 200818939 y el estudiante Enio Fabrizio Torres Noguera con número de carné 200815519, que como asesor apruebo el contenido del mismo.

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



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

Edgar Santos
INGENIERO EN CIENCIAS Y SISTEMAS
Colegiado 5266



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

Guatemala, 3 de Septiembre de 2014

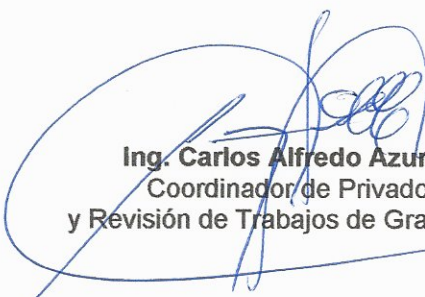
Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **ENIO FABRIZIO TORRES NOGUERA** con carné **2008-15519**, y **MARVIN ALBERTO VILLATORO ALONZO** con carné **2008-18939**, titulado: **"TECNOLOGÍAS MÓVILES Y CONTROL PARENTAL CENTINELA"**, 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 **“TECNOLOGÍAS MÓVILES Y CONTROL PARENTAL CENTINELA”**, realizado por los estudiantes **MARVIN ALBERTO VILLATORO ALONZO** y **ENIO FABRIZIO TORRES NOGUERA**, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

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



Guatemala, 14 de noviembre 2014



DTG. 660.2014

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: **TECNOLOGÍAS MÓVILES Y CONTROL PARENTAL CENTINELA**, presentado por los estudiantes universitarios: **Marvin Alberto Villatoro Alonzo y Enio Fabrizio Torres Noguera**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Ing. Murphy Olympo Paiz Recinos
Decano

Guatemala, 17 de noviembre de 2014

/gdech



ACTO QUE DEDICO A:

- Dios** Por brindarme sabiduría, conocimiento, salud y capacidad a lo largo de mi período de estudios, además de darme las oportunidades para mejorar y salir adelante.
- Mis padres** Por darme educación, ejemplo, amor y comprensión durante mi vida, enseñarme el valor de la vida y la importancia de la unidad de la familia.
- Mi esposa** Por ser un pilar en mi vida y brindarme su apoyo en los momentos más difíciles, llenándome de ánimo para no darme por vencido.
- Mi hijo** Por ser la motivación a querer seguir adelante y ser una persona de bien, porque fue la razón de un cambio positivo en mi vida.
- Hermanas** Además de ser mis hermanas son mis amigas incondicionales, brindándome consejos útiles en la vida, apoyo y unidad.

Marvin Alberto Villatoro Alonzo

ACTO QUE DEDICO A:

Dios

Por ser mi guía a lo largo de la carrera y fortalecerme en los momentos más difíciles para sobresalir y salir adelante.

Mis padres

Por darme su apoyo incondicional, amor y comprensión durante todos estos años, sin su ayuda nada sería posible, además de los buenos consejos y el ejemplo para poder ser una persona de bien.

Tíos y familiares

Por todo su cariño y apoyo, ya que son parte fundamental de mi vida.

Hermanos

Para que les ayude como un ejemplo en su vida y logren alcanzar este éxito como parte de sus vidas.

Enio Fabrizio Torres Noguera

AGRADECIMIENTOS A:

**La Universidad de San
Carlos de Guatemala**

Por ser mi casa de estudios durante todos estos años permitiéndome ser un profesional y ser alguien de futuro para esta sociedad, aportando a que Guatemala cada vez sea un mejor país.

Facultad de Ingeniería

Por ser el centro de todas las enseñanzas adquiridas durante toda mi carrera, brindándome los conocimientos, para ejercer un trabajo digno y sobre todo que disfruto al realizarlo.

Catedráticos y auxiliares

Por hacer de mí un mejor profesional, y por compartir su conocimientos y experiencias adquiridas en el entorno de la carrera.

Compañeros de clase

Por las anécdotas vividas y ayuda a superar obstáculos, como también el conocimiento que compartimos.

Amigos

Por compartir tantos buenos momentos en el período de estudio universitario haciendo de esto una experiencia inolvidable.

Marvin Alberto Villatoro Alonzo

AGRADECIMIENTOS A:

La Universidad de San Carlos de Guatemala	Por brindarme la oportunidad de ser un profesional y acogerme durante tantos años como mi casa de estudios, la experiencia vivida nunca pasará en vano.
Facultad de Ingeniería	Por ser el centro de mi aprendizaje y de las mejores experiencias vividas a lo largo de la carrera.
Catedráticos y auxiliares	Por brindarme su tiempo y esfuerzo compartiendo sus conocimientos por el bien de mi crecimiento profesional y académico.
Compañeros de clase	Por su ayuda y apoyo en los momentos difíciles, definitivamente hizo el trabajo menos arduo y más ameno.
Amigos	Por su compañía a lo largo de los años difíciles, sin duda sin ellos no hubiera sido lo mismo.
Novia	Por todo tu amor, apoyo, compañía e impulso para seguir adelante en el tramo final de mi carrera.

Enio Fabrizio Torres Noguera

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN.....	XVII
OBJETIVOS.....	XIX
INTRODUCCIÓN.....	XXI
1. TECNOLOGÍA Y NECESIDAD DE SEGURIDAD EN GUATEMALA	1
1.1. Tecnología en Guatemala	1
1.2. Historia del internet en Guatemala	2
1.3. Telecomunicaciones en Guatemala	3
1.3.1. Actualidad.....	3
1.4. Campus Tecnológico en Guatemala	6
1.4.1. ¿Qué es la incubadora de negocios?	6
1.4.2. Áreas de interés en el sector TIC	7
1.4.3. Ejes de interés para la incubadora de negocios	8
1.5. Tecnología y Control Parental	8
2. APLICACIÓN BASE	11
2.1. Sistema Operativo Android.....	11
2.1.1. Historia de Android	13
2.1.2. Versiones.....	14
2.2. Servicio de mensajes cortos.....	17
2.3. Google Voice	17
2.4. Aplicación RingyDingyDingy.....	18

2.4.1.	Sintaxis de RingyDingyDingy	18
2.4.2.	Licencia de RingyDingyDingy.....	19
2.4.3.	Arquitectura de RingyDingyDingy.....	20
3.	CASOS DE ESTUDIO Y ENTREVISTAS	21
3.1.	Edad.....	21
3.2.	Género	22
3.3.	Servicio telefónico	23
3.4.	Tipo de dispositivo.....	24
3.5.	Marca	25
3.6.	Preferencia de marca.....	26
3.7.	Sistema operativo móvil	27
3.8.	Versión de Android.....	28
3.9.	Control Parental	29
3.10.	Tipos de Control Parental.....	30
3.11.	Funcionalidad: ubicación por posicionamiento global satelital.....	31
3.12.	Funcionalidad: bloqueo remoto	32
3.13.	Funcionalidad: alerta remota.....	33
4.	PROPUESTA: APLICACIÓN CENTINELA	35
4.1.	Centinela	36
4.1.1.	<i>Splash</i>	36
4.1.2.	Clase principal (MainActivity)	41
4.1.3.	Envío de mensaje.....	44
4.1.4.	Recibir mensaje.....	46
4.1.5.	Base de datos SQLite	53
4.1.6.	Notificación.....	57
4.1.7.	Mapas.....	59

4.1.7.1.	Descargar y configurar los paquetes de Google Play Services	59
4.1.7.2.	Obtener una API Key	62
4.1.7.3.	Configurar permisos y especificaciones en el Manifest	65
4.1.7.4.	Manejo de objetos de mapas para su visualización	67
4.2.	Clientinela	73
4.2.1.	Solicitud	76
4.2.2.	Eliminar	78
4.2.3.	Alerta remota	80
4.2.4.	Geolocalización	83
4.2.5.	Bloqueo remoto de pantalla	86
CONCLUSIONES		89
RECOMENDACIONES		91
BIBLIOGRAFÍA		93
APÉNDICES		97

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Dispositivos móviles con sistema operativo Android	12
2.	Cuota de versiones de Android utilizadas	16
3.	Servicio: Google Voice	17
4.	Arquitectura de aplicación RingyDingyDingy.....	20
5.	Rango de edades.....	21
6.	Género.....	22
7.	Servicio telefónico	23
8.	Tipo de dispositivo.....	24
9.	Marca.....	25
10.	Preferencia de marca.....	26
11.	Sistema operativo móvil	27
12.	Versión de Android.....	28
13.	Control Parental	30
14.	Tipos de Control Parental.....	31
15.	Ubicación por posicionamiento global satelital.....	32
16.	Bloqueo remoto	33
17.	Alerta remota.....	34
18.	Icono Centinela, Clientinela.....	35
19.	Vista <i>Splash</i>	37
20.	<i>Splash</i> Centinela	38
21.	Modelo <i>Splash</i>	39
22.	Controlador <i>Splash</i>	40
23.	Estructura MainActivity.....	42

24.	Pantalla principal.....	42
25.	Configuración de pestañas	43
26.	Icono pestaña	44
27.	Controlador MainActivity	44
28.	Clase de envío de SMS	45
29.	Servicio de recepción de mensaje	47
30.	Validación: mensaje recibido	48
31.	Validación etiquetas: mensaje recibido.....	49
32.	Sentencia Si.....	50
33.	Sentencia No	51
34.	Sentencia gps	51
35.	Declaración recibir mensaje.....	52
36.	Diagrama entidad relación	53
37.	Importaciones de clases de base de datos.....	54
38.	Métodos de creación base de datos	55
39.	ABC de entidad, SQLite.....	56
40.	Clase notificación.....	58
41.	Descarga paquete Google Play Service	60
42.	Import Google Play Service	61
43.	Referencia Google Play Service	62
44.	Huella SHA-1	63
45.	Google Maps Android API	64
46.	API Key.....	65
47.	Metadatos mapas	66
48.	Condición OpenGL	67
49.	Vista mapa.....	67
50.	Modelo mapa	68
51.	Parámetros para modelo mapa	69
52.	Lista de selección tipo de mapa.....	70

53.	Agregar apuntador en mapa	71
54.	Modelo mapa	72
55.	Visualización de mapa	73
56.	Arquitectura Clientinela	74
57.	Declaración del Receiver	75
58.	Notificación de nueva solicitud	76
59.	Mensaje de solicitud.....	77
60.	Método del botón aceptar solicitud.....	77
61.	Vista principal, lista de Centinelas.....	78
62.	Procedimiento de eliminar un Centinela.....	79
63.	Vista principal, lista de centinelas después de eliminación.	80
64.	Método PlayRing.....	81
65.	Acción del botón detener alarma.....	82
66.	Mensaje de alerta remota.....	82
67.	Librerías para localización.....	83
68.	Métodos utilizados para localización	84
69.	Procedimiento de obtención de coordenadas	85
70.	Mensaje con coordenadas	86
71.	Procedimiento de bloqueo.....	87
72.	Actividad para solicitar permisos de bloqueo	87
73.	Pantalla de solicitud de permisos.....	88

TABLAS

I.	Estadísticas de consumo de internet en Guatemala	3
II.	Penetración de los servicio de telecomunicación	5
III.	Versiones de Android y su número de API.....	15

LISTA DE SÍMBOLOS

Símbolo	Significado
%	Porcentaje
MHz	Un megahercio, se utiliza muy frecuentemente como unidad de medida de la frecuencia de trabajo de un dispositivo de hardware, o como medida de ondas electromagnéticas en telecomunicaciones.
X	Utilizado en versionamiento de aplicaciones, la X quiere decir que puede ser reemplazado por cualquier número.

GLOSARIO

API	Interfaz de programación de aplicaciones (IPA) o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
App	Son las primeras 3 letras de la palabra en inglés <i>application</i> . Este quiere decir que es una forma de aplicación de la informática que resuelve una tarea o necesidad.
Android	Sistema operativo basado en Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, inicialmente desarrollado por Android, Inc.
C	Lenguaje de programación de medio nivel. Trata con objetos básicos como caracteres, números, así como bits y direcciones de memoria.
CMI	Abreviatura para Corporación Multi Inversiones, fue fundada por Juan Bautista Gutiérrez en Guatemala en los años 1920s, CMI actualmente abarca unas 300 compañías, emplea a cerca de 30 000 empleados y opera en 16 países.

C++	<p>Evolución del lenguaje de programación C, este cuenta con características diferentes con el manejo de variables y objetos.</p>
GNU	<p>Sistema operativo Unix-like, está formado por <i>software</i> libre. Basado en el núcleo GNU Hurd y su objetivo es ser compatible con Unix. GNU es un acrónimo recursivo de <i>GNU's Not Unix</i> (en español: GNU no es Unix).</p>
Google Maps	<p>Es un servicio gratuito de Google, que sirve para observar mapas en la web o bien de aplicación de escritorio. Ofrece mapas desplazables, fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones, estos son actualizados periódicamente, actualmente el servicio gratuito no es en tiempo real, pero tiene una opción de navegación en 3d, dando la sensación de estar ubicado en el lugar deseado.</p>
GPS	<p>Sistema Global de Navegación por Satélite (Global Position Satelital), que permite ubicar a nivel mundial un objeto, una persona y otros. Este funciona mediante triangulación, utilizando nada más 3 de 24 satélites disponibles, la precisión puede aumentar hasta centímetros.</p>
GSM	<p>Del inglés Global System for Mobile, es un sistema estándar, libre de regalías, de telefonía móvil digital.</p>

ICAITI	Sinónimo para Instituto Centroamericano de Investigación y Tecnología Industrial, se inicia en 1956 y tiene como fin el apoyo a la industria y el comercio en América Central.
Inc	Utilizado para la abreviación en inglés de <i>Incorporation</i> .
INCAP	Instituto de Nutrición de Centroamérica y Panamá, es un centro especializado en alimentación y nutrición, Institución del Sistema de la Integración Centroamericana. Fundado el 14 de septiembre de 1949, con sede en la ciudad de Guatemala.
IOS	(Siglas en inglés: iPhone Operating System). Es un sistema operativo para dispositivos móviles desarrollado por la empresa <i>Apple Inc</i> , para su línea de productos móviles como <i>iphone</i> , <i>ipad</i> y otros productos.
JAVA	Es un lenguaje de programación de propósito general, concurrente, multiplataforma, orientado a objetos y basado en clases, fue diseñado específicamente para tener pocas dependencias de implementación como fuera posible.
Kernel	Conocido como núcleo, es un software que es una parte fundamental del sistema operativo. Es el principal responsable de facilitar a los distintos

programas y accesos seguros al hardware de la computadora, es el encargado de gestionar recursos, a través de servicios de llamada al sistema.

MVC

El Modelo–Vista–Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

PC

Es la abreviatura en inglés de *personal computer*, es decir computadora personal.

PIB

Es una media utilizada en la macroeconomía, la cual indica el valor monetario de producción, ya sea de productos o servicios. Este puede aplicarse en un país o región, bajo un período de tiempo determinado.

Qwerty

El teclado QWERTY es la distribución de teclado más común. Fue diseñado y patentado por Christopher Sholes en 1868 y vendido a Remington en 1873. Su nombre proviene de las primeras seis letras de su fila superior de teclas.

S. A.	Abreviatura de Sociedad Anónima es aquella sociedad mercantil cuyos titulares lo son en una participación en el capital social a través de títulos o acciones.
SMS	Abreviatura en inglés de Short Message Service es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos entre teléfonos móviles.
SQLite	Es una base de datos ligera, ya que está contenida en una pequeña biblioteca escrita en lenguaje C.
TEC	Se refiere al Campus Tecnológico de Guatemala ubicado en la zona 4 de la ciudad de Guatemala.
TIC	Tecnologías de la Información y la Comunicación, son aquellas tecnologías que se necesitan para la gestión y transformación de la información, y muy en particular el uso de ordenadores y programas que permiten crear, modificar, almacenar, administrar, proteger y recuperar esa información.
XML	Siglas en inglés de eXtensible Markup Language, es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

RESUMEN

La inseguridad y la delincuencia en Guatemala, obliga a los padres a tener un mayor control con sus hijos, a su vez el crecimiento de la tecnología móvil y tanto en infraestructura como en dispositivos móviles, junto con el crecimiento del poder adquisitivo de la clase media, para comprar teléfonos inteligentes, se decidió crear una aplicación en el Sistema Operativo Android como parte de la solución del problema, dándole al padre una herramienta con la cual pueda tener control sobre sus hijos, pero sin perder la privacidad completamente.

Centinela es una aplicación que permite al padre obtener cierta información del hijo, entre ellas se encuentra la ubicación por medio de GPS, envío de mensaje de alerta y por último poder bloquear la pantalla del hijo en cualquier momento, todas estas solicitudes son enviadas a través de la aplicación de Centinela y son procesadas por Clientinela para ejecutar las acciones correspondientes y enviar las respuestas según amerite. Ambas aplicaciones son complementarias, por lo que tener solo una no serviría, ya que las acciones no se podrían ejecutar.

Es importante establecer que los dispositivos deben estar sincronizados para lo que el padre deberá enviar una solicitud al hijo y este aceptarla para utilizar las funciones antes mencionadas.

Una vez los dispositivos estén sincronizados el padre puede hacer uso de la función de localización, la cual envía una orden a Clientinela y este le envía las coordenadas actuales de ubicación, Centinela toma esas coordenadas y las

muestra en un mapa de Google, a la vez almacenando un historial de las ubicaciones obtenidas.

Mensaje de emergencia, esta función es muy útil para casos de suma importancia, en el cual el padre le envía un mensaje al hijo por medio de Centinela. Al recibir el mensaje no importando el estado del tono del teléfono del hijo, por ejemplo vibrador o silencioso, el teléfono empezará a sonar en el máximo volumen y al desplegar la alarma se mostrará el mensaje enviado por el padre.

Y por último teniendo el bloqueo remoto, si en algún caso el hijo pierde o le roban el teléfono, el padre mediante un botón en su aplicación bloqueará el celular del hijo, siendo necesaria la contraseña para poder ingresar a las funcionalidades del teléfono.

Por último cabe mencionar que el hijo no podrá eliminar ningún Centinela, o personas que lo monitoree, solo si el padre o Centinela desea eliminar la sincronización con el hijo o Clientinela se podrá eliminar de ambos dispositivos.

OBJETIVOS

General

Innovar en las áreas más nuevas de la carrera de Ingeniería en Ciencias y Sistemas, implementando los conocimientos adquiridos en el transcurso de la carrera, creando una herramienta de seguridad parental, por medio de una aplicación gratuita para dispositivos móviles basados en plataforma Android.

Específicos

1. Estudiar áreas de control remoto de dispositivos móviles, para conocer las necesidades de un individuo en el tema de seguridad de sus personas cercanas.
2. Desarrollar una solución para dispositivos móviles en plataforma Android, que sea funcional, intuitiva y cubra la necesidad de seguridad del ser humano. Esta herramienta debe ser amigable para los usuarios y transparente para la persona monitoreada garantizando que no perderá su privacidad.
3. Implementar utilidades de última generación, como lo es la geolocalización con Google Maps, establecer un canal de comunicación en situaciones de emergencia y por último el bloque remoto de dispositivo por razones de robo, extravío u otra eventualidad.

INTRODUCCIÓN

Siempre es necesario garantizar la seguridad en todo individuo, por lo tanto se implementó una solución en la cual se puede monitorear ciertas características en otros dispositivos, como por ejemplo localización remota y bloqueo de dispositivo remoto, por último establecer un canal de comunicación entre dispositivos en casos de emergencia.

El fin de la aplicación es poder requerir información puntual de un dispositivo enviándolo hacia la persona que está monitoreando, esta comunicación será a través de mensajes de texto, implementando una sintaxis especial dentro del mensaje para indicar las acciones que debe realizar el dispositivo con la información recibida. La aplicación es la encargada de interpretar los mensajes, mostrar y ejecutar las instrucciones solicitadas, esto haciendo una aplicación con mejor usabilidad.

Todas las instrucciones se ejecutan en segundo plano, dando transparencia al usuario que está monitoreando, pero a su vez solamente ejecutando cierto tipo de instrucciones para que la persona monitoreada no pierda su privacidad.

1. TECNOLOGÍA Y NECESIDAD DE SEGURIDAD EN GUATEMALA

En Guatemala, la seguridad es una necesidad vital pero se encuentra deficiente, ya que no se garantiza al 100 por ciento, y por medio de la tecnología es posible implementar una solución para contrarrestar esta deficiencia, siendo una herramienta en caso de emergencia.

Es de suma importancia establecer como se encuentra Guatemala a nivel tecnológico, debido que al implementar una solución de este tipo se debe definir el auge que tendrá.

1.1. Tecnología en Guatemala

La tecnología ha estado desarrollándose a pasos agigantados en los últimos años constituyéndose la misma en un factor importante en el fenómeno de la globalización, algunos países gracias a sus posibilidades económicas han hecho uso extensivo de la tecnología más que otros, por lo que han alcanzado un mejor nivel de vida para sus habitantes. Parte de este éxito se debe a que aplican los desarrollos tecnológicos a la información, cuyo acceso es vital en la toma efectiva de decisiones y acciones en cualquier campo al cual la información haga referencia, alguno de estos campos pueden ser por ejemplo: el sector agrícola, salud, educación, industria, manufactura, entre otros.

En el caso de Guatemala por varios factores tales como: económicos, sociales, culturales, educación, seguridad, prácticamente no ha participado en el desarrollo tecnológico de la humanidad, por ser un país pobre apenas cuenta

con infraestructura en comunicaciones, esto se ve reflejado en que existen comunidades que se encuentran muy aisladas del resto del país y por ende del resto del mundo. Otro factor importante a considerar es el alto grado de analfabetismo con que cuenta Guatemala y esto repercute en que el acceso a la información no sea para toda la población, como consecuencia se produce analfabetismo informático.

El grado de desarrollo tecnológico de un país influye en todos los sectores que la conforman, el sector manufacturero e industrial se ve afectado estratégicamente y competitivamente si el país no le brinda la infraestructura necesaria para desenvolverse, le resta competitividad ante un mundo cada vez más globalizado.

1.2. Historia del internet en Guatemala

En 1992 se creó el Consejo Nacional de Ciencia y Tecnología (CONCYT). La parte ejecutora de este consejo consta de un número de comisiones que trabajan en diferentes áreas de Ciencia y Tecnología. Una de estas comisiones es la Comisión de Información e Informática y el primer proyecto presentado por esta fue; el Proyecto Mayanet. De inmediato se empezó a trabajar en su diseño y recaudación de fondos, lo cual no trajo mayores inconvenientes.

Finalmente en 1995 bajo un acuerdo firmado entre el CONCYT y GUATEL quien brindó una ayuda sustancial: ofreciendo rebajas en la comunicación vía satélite, 2 años de servicio gratis a las Universidades del INCAP e ICAITI, se inició el proyecto ofreciendo todos los servicios de internet. Casi simultáneamente inicia la primera empresa comercial que fue Cybernet. Las estadísticas mostradas a continuación son hasta el 2010.

Tabla I. **Estadísticas de consumo de internet en Guatemala**

Año	Población	Usuarios	Porcentaje de penetración
2000	11 237 196	65 000	0,6%
2005	12 328 453	756 000	6,1%
2008	13 002 206	1 320 000	10,2%
2009	13 276 517	1 960 000	14,8%
2010	13 550 440	2 280 000	16,8%

Fuente: <http://www.internetworldstats.com/am/gt.htm>.

Consulta: 05 de septiembre de 2013.

1.3. Telecomunicaciones en Guatemala

Son representaciones del desarrollo que se ha venido produciendo en este país. Dado que al momento de la apertura del mercado, Guatemala era uno de los países latinoamericanos con menores índices de penetración de servicios de comunicaciones. Actualmente, es por mucho un país en el que los operadores presentes deben invertir para mantener y promover el desarrollo del mercado, con carteras de clientes cada vez más exigentes; alcanzando 119,54 por ciento de penetración del servicio móvil.

1.3.1. Actualidad

El sector de las comunicaciones es uno de los sectores que más crecimiento ha tenido durante los últimos años, a pesar de esto, el crecimiento actual se ha visto afectado por la continua crisis económica mundial, que ha reducido el poder adquisitivo de la población. El mercado más afectado ha sido el de la telefonía fija, se contrajo por primera vez en 2009 y se ha mantenido la

tendencia desde entonces. Por el otro lado se tiene que el mercado de telefonía móvil y banda ancha han mostrado un crecimiento a través de los años, este último a un ritmo más lento.

La deficiente infraestructura ha llevado a que el país tenga una de las menores densidades telefónicas de líneas fijas en la región. Esto ha llevado hacia una baja disponibilidad de banda ancha. Se espera que para el próximo año la tendencia se mantenga, inclusive la telefonía fija puede llegar a estancarse mientras que la banda ancha fija y móvil se desarrollarán constantemente.

El crecimiento del PIB *per cápita* en el futuro proporcionará un mayor poder adquisitivo para la población, estimulando la demanda de los servicios de las telecomunicaciones y TIC. Cabe mencionar que la violencia, pobreza y corrupción son obstáculos en el crecimiento de la inversión tecnológica.

Desde la venta de la empresa de telefonía del estado, que en ese tiempo era Guatel hacia la iniciativa privada, la participación del Gobierno en inversión de infraestructura ha sido nula, dejando está a manos de la iniciativa privada. Telefónica, América Móvil, Milicon, son potencias regionales y mundiales que aprovechan sus recursos y experiencia en el negocio para impulsar la inversión en Guatemala. Otro beneficio que tiene Guatemala es que se cuenta con uno de los marcos regulatorios más abiertos de la región, en todos los sectores de telecomunicaciones. Ayudando a la libre competencia.

América Móvil controla alrededor del 70 por ciento de las líneas fijas en servicio a través de su filial Claro. La telefonía móvil ha sido el mercado de telecomunicaciones más desarrollado en Guatemala durante varios trimestres y es probable que siga siendo así en los próximos años, dado el mal estado de

los servicios de línea fija. La intensa competencia entre los operadores ha ayudado a mejorar los servicios y a establecer precios más bajos.

Tabla II. **Penetración de los servicios de telecomunicación**

Penetración de los servicios de Telecomunicación	2010	2013 (e)
Fixed broadband	2,5%	10,5%
Mobile	88,3%	103,5%
Fixed-line	10,6%	13,2%

Fuente: <https://www.budde.com.au/Research/Guatemala-Telecoms-Mobile-and-Broadband.html>. Consulta: 05 de septiembre de 2013.

Puntos importantes:

- Bipolaridad en los servicios de telefonía, van desde la red moderna en la ciudad de Guatemala a la infraestructura inexistente en muchas zonas rurales. Como resultado la densidad telefónica fija sigue siendo baja. Y la telefonía móvil va hacia el alta.
- Los servicios internacionales de envío de dinero móvil, se están expandiendo con el lanzamiento de los servicios de Tigo con Western Union.
- Digicel, que tiene una licencia de funcionamiento en la banda de 900MHz, aún tiene que lanzar servicios móviles en Guatemala, aunque en 2012 se firmó un acuerdo con Claro para integrar sus redes, tanto en Guatemala como en El Salvador.
- En 2013 Telefónica firmó un acuerdo con el CMI, desinvertir una participación del 40 por ciento en su filial guatemalteca.

- El Congreso de la República aprobó una ley en que se extienden las concesiones de licencia existentes a finales de 2012 por un período de 20 años.
- A mediados de 2013 el Gobierno eligió el estándar ISDB -T de televisión digital terrestre, con miras a poner fin a las emisiones analógicas a finales de 2015.

1.4. Campus Tecnológico en Guatemala

El Campus Tecnológico es un espacio físico en la ciudad de Guatemala, donde la innovación y la tecnología puedan tener una sede para florecer a niveles de competitividad mundial.

La misión de la iniciativa es la de crear un parque tecnológico en 4 grados norte, que reúna los elementos principales que hicieron posible la revolución tecnológica, su visión es la de convertir a Guatemala en un líder tecnológico de la región y hacerla competitiva mundialmente para el 2020.

Inició en 2006 y cuenta con una infraestructura que alberga a más de 150 empresas de alta tecnología para el sector de Tecnologías de la Información y la Comunicación (TIC) que incluyen especialidades en: electrónica, telecomunicaciones, informática y lo audiovisual.

1.4.1. ¿Qué es la incubadora de negocios?

La incubadora de negocios del Campus Tecnológico es una incubadora de empresas que tiene como objetivo principal acelerar el lanzamiento y crecimiento de nuevas iniciativas de alta tecnología en el parque tecnológico, a

través de una plataforma que ofrece las herramientas y recursos necesarios para llevar un producto o servicio al mercado.

Es la organización localizada dentro del Campus Tecnológico que acelera el proceso de creación, crecimiento y consolidación de empresas y proyectos innovadores, así como la capacidad emprendedora de personas.

Dispone de recursos técnicos, económicos, infraestructura y de una red de contactos para guiar a las empresas y los emprendedores para su desarrollo competitivo.

Y finalmente, presta servicios a las empresas y emprendedores del parque tecnológico. Tales como:

- Comercialización de los productos/servicios y búsqueda de oportunidades de negocio de los miembros del Campus Tecnológico.
- Desarrollo de eventos, talleres, ruedas de negocio y capacitaciones.
- Asesoramiento y capacitación.
- Servicios de gestión de proyectos, mercadeo, asesoría legal, recurso humano y otros con red de mentoría.
- Pre-incubación, incubación y aceleración.
- Búsqueda de financiamiento.
- Redes de contactos locales e internacionales.

1.4.2. Áreas de interés en el sector TIC

Las áreas de interés para el Campus TEC dentro del sector de alta tecnología (que incluye la nanotecnología, biotecnología, robótica, tecnologías de la información y la comunicación, entre otros) son las referentes a

las tecnologías de la información y la comunicación TIC (que incluye la electrónica, telecomunicaciones, informática y lo audiovisual) y son las siguientes:

- Software
- Hardware
- Efectos especiales
- Juegos
- Aplicaciones para móviles y otras del sector de telecomunicaciones
- *Business Process Outsourcing (BPOs) – Call centers* y otros servicios
- Otros de alta tecnología

1.4.3. Ejes de interés para la incubadora de negocios

El factor económico es determinante en este tipo de actividades, en donde los inversionistas tienen que asegurar el retorno de su inversión, es por eso que se basan en 4 ejes principales para la selección de proyectos a patrocinar.

- Emprendimientos Tecnológicos (*Tech Startups*)
- Empresas en procesos de Internacionalización
- Emprendimientos Sociales (*Social Entrepreneurship*)
- Productos nuevos a patentar y comercializar

1.5. Tecnología y Control Parental

Consiste en impedir o limitar el acceso al contenido no adecuado a menores de edad, o bien el poder monitorear acciones que realizan. Esto es mediante una serie de sistemas de bloqueo y/o de controles normalmente protegidos mediante claves establecidas por los responsables legales del

menor, normalmente sus padres o los adultos responsables del uso de la correspondiente máquina.

A pesar de las limitantes que tiene Guatemala en la tecnología de punta. En los últimos años estas han decrecido, específicamente en el área de la ciudad capital, y municipios aledaños, en donde las empresas de telecomunicación han puesto un mayor énfasis e inversión.

La delincuencia y la inseguridad de la población ha tenido un gran auge en la última década en los centros urbanos del país. Abriéndose un nuevo mercado de servicios y productos como lo son la empresa que se dedica a la ubicación y rastreo de automóviles Detektor o el servicio de Telecomunicaciones de Guatemala, S. A. (Claro) Localízame.

Actualmente en el mercado de teléfonos inteligentes Android existen varias propuestas para el control parental de estos dispositivos móviles. A continuación se detallan algunas de las más utilizadas:

- Kids Place: crea una zona segura en el dispositivo para niños. El bloqueo infantil protege la información personal cuando los niños están utilizando el teléfono o tableta de los padres. Impidiendo que los niños accidentalmente descarguen aplicaciones, realicen llamadas telefónicas, envíen mensajes de texto u otras acciones que pueden costar dinero. Kids Place también incluye características de comodidad para los padres; como reinicio automático de la aplicación, útil para los niños pequeños que accidentalmente salen de una aplicación en marcha. Es gratuita.
- Kaspersky Parental Control: esta aplicación, que todavía está en versión beta, consiste en una herramienta que bloquea el acceso a otras aplicaciones instaladas mediante el uso de contraseña. También limita el

acceso a páginas con contenidos inapropiados para los niños. Es gratuita, funciona en teléfonos y tabletas y de momento se encuentra en inglés.

- Norton Family Parental Control: con ella se puede supervisar y limitar los sitios web visitados, límites de horario diarios de uso del dispositivo, programar alertas, supervisar redes sociales, otros. Dispone además de una versión de pago que todavía introduce más herramientas de control.
- Control Parental Kytetime: aporta muchísimas funciones, como el bloqueo de aplicaciones, limitaciones por horarios a ciertas aplicaciones o redes sociales, localización por *GPS* en el mapa, envío por correo electrónico de informes de actividad a los padres, control remoto desde el PC, otros. Es una de las más completas y además es gratuita.
- Qustodio Control Parental: otra aplicación con una buena diversidad de opciones, aunque la función de bloquear aplicaciones queda limitada solo a la versión de pago. Permite supervisar varios niños en varios tipos de dispositivo, teléfono, tableta y PC. Permite personalizar la navegación bloqueando páginas peligrosas, establece límites de uso, envía alertas de acciones sospechosas, otros.

Con la ayuda de la tecnología se puede ayudar en gran medida a estar más seguros, con la aplicación Centinela se busca dar al padre una mayor tranquilidad, haciendo uso de las virtudes que tienen los teléfonos inteligentes hoy en día. Proveyendo un conjunto de funciones que permitirán tener un mayor control sobre las actividades de una persona en específico.

2. APLICACIÓN BASE

Existen muchos métodos de comunicación en los diferentes sistemas operativos para teléfonos inteligentes y entre todos el que más ha perdido relevancia son los mensajes de texto también conocidos como *SMS* que es un acrónimo en inglés *short message service* o servicio de mensajes cortos. Esta aplicación reutiliza un servicio ya existente para que los diferentes dispositivos puedan comunicarse.

Como complemento se utiliza una aplicación base de código abierto, la cual brinda las herramientas para poder desarrollar la aplicación, esta es RingyDingyDingy, se puede obtener en el siguiente enlace:

- <http://ringydingydingy.com/>

Esta aplicación fue elaborada para Sistema Operativo Android, basados en el Kernel Linux para dispositivos móviles.

2.1. Sistema Operativo Android

El Sistema Operativo Android está basado en el kernel de Linux y su principal funcionamiento es para dispositivos móviles con pantallas táctiles, la mayoría de dispositivos con dicho sistema operativo cuenta en promedio con 6 botones, aunque existen dispositivos que tienen un botón adicional para búsqueda, como también existen dispositivos que cuentan con un teclado *qwerty* físico, actualmente esta funcionalidad se omitió, ya que es más económico generar un teclado por medio de software que por el contrario

invertir dinero en hardware, a continuación se muestran ejemplos de los diferentes dispositivos:

Figura 1. **Dispositivos móviles con sistema operativo Android**



Fuente: <http://www.android.com/>. Consulta: 17 de agosto de 2013.

Los dispositivos mostrados anteriormente mantienen un registro casi estandarizado de botones que pueden tener:

- Botón de encendido
- Botón de retorno
- Botón de menú
- Botón de hogar
- Botón de búsqueda
- Botón de cámara
- Botón de volumen
- Bloqueo y/o desbloqueo

2.1.1. Historia de Android

Inicialmente fue desarrollado por Android Inc. Pero contaban con respaldo económico de Google, dicha empresa fue absorbida por Google en 2005 para lanzar el sistema operativo en 2007, en la actualidad es el principal Sistema Operativo para dispositivos móviles, ya que es de código abierto. Cuenta con un alto número de fabricantes que lo utilizan y en la actualidad se encuentra entre los principales en ventas. Se ubica en primer lugar en cantidad de unidades vendidas en Estados Unidos, en el trimestre 2010 contando con un porcentaje de mercado de 43,6 por ciento, a nivel mundial del 2011 alcanzó un porcentaje de mercado del 50,9 por ciento, duplicando a su rival iOS de Apple, Inc.

Cada vez aumenta la cantidad de desarrolladores que buscan crear aplicaciones para extender sus funcionalidades como también emprendimientos. Actualmente ya sobre pasó a un millón de aplicaciones, de las cuales 2/3 son gratuitas, siendo también más económicas comparadas con el App Store de iOS, sus aplicaciones se cuentan publicadas en Google Play que es la tienda oficial de aplicaciones Android, donde también existen diferentes tiendas como por ejemplo Apps de Samsung.

Su lenguaje de programación es Java, y sus estructura se compone de aplicaciones con compilación en ejecución, utilizan un *framework* de Java orientada a objetos, cuenta también con base de datos relacional *SQLite*, y su estándar de programación está basado en MVC (modelo, vista, controlador), siendo el modelo las clases .java, las vistas, los archivo xml en las cuales se declaran los objetos y el controlador es un archivo llamado Manifest, en el cual se declaran todas las actividades y permisos de la aplicación.

El sistema operativo está compuesto aproximadamente de 12 millones de líneas de código, utilizando 3 millones en XML, 2,8 millones en lenguaje C, 2,1 millones de Java y 1,75 millones de C++.

El 5 de noviembre de 2007, un consorcio de varias compañías llamado Open Handset Alliance se inició con el fin de establecer estándares abiertos para dispositivos móviles. Su primer producto Android, fue una plataforma para dispositivos móviles construida sobre la versión 2.6 del kernel de Linux.

Open Handset Alliance lo conformaban inicialmente compañías como: Texas Instruments, Broadcom Corporation, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, Intel, LG, Marvell Technology Group, Motorola y T-Mobile. El 9 de diciembre de 2008 se anunció de 14 nuevos miembros; incluyendo a PacketVideo, ARM Holdings, Atheros Communications, Asustek, Garmin, Softbank, Sony Ericsson, Toshiba, Vodafone y ZTE.^{39 40}.

2.1.2. Versiones

Las versiones de Android reciben nombres de postres en inglés. En cada versión el postre elegido empieza por una letra distinta siguiendo el orden alfabético, actualmente la nueva versión por lanzarse utilizará la letra K y las primeras versiones fueron de pruebas, por lo cual no existieron dispositivos a la venta que utilizaran dichas versiones:

- A: Apple Pie, tarta de manzana
- B: Banana Bread, pan de plátano
- C: Cupcake, panque o cubilete
- D: Donut, rosquilla o dona
- E: Éclair, pastel francés

- F: Froyo, helado de yogur
- G: Gingerbread, pan de jengibre
- H: Honeycomb, panal de miel
- I: Ice Cream Sandwich, sándwich de helado
- J: Jelly Bean/Gummy Bear, gominola (versión actual)
- K: KitKat, por lanzarse

Android utiliza versiones de API para referenciar a las características y funciones de los dispositivos y éstas están estrechamente relacionadas con la versión de Sistema Operativo:

Tabla III. **Versiones de Android y su número de API**

Versión de Plataforma	Nivel de API	Versión de Código
Android 4.4	19	KitKat
Android 4.3	18	Jelly Bean MR2
Android 4.2, 4.2.2	17	Jelly Bean MR1
Android 4.1, 4.1.1	16	Jelly Bean
Android 4.0.3, 4.0.4	15	Ice Cream Sandwich MR1
Android 4.0, 4.0.1, 4.0.2	14	Ice Cream Sandwich
Android 3.1.x	12	Honey Comb MR2
Android 3.0.x	11	Honey Comb
Android 2.3.4, 2.3.3	10	Ginger Bread MR1
Android 2.3.2, 2.3.1, 2.3	9	Ginger Bread
Android 2.2.x	8	Froyo
Android 2.1.x	7	Eclair MR1
Android 2.0.1	6	Eclair 0.1
Android 2.0	5	Eclair

Continuación de la tabla III.

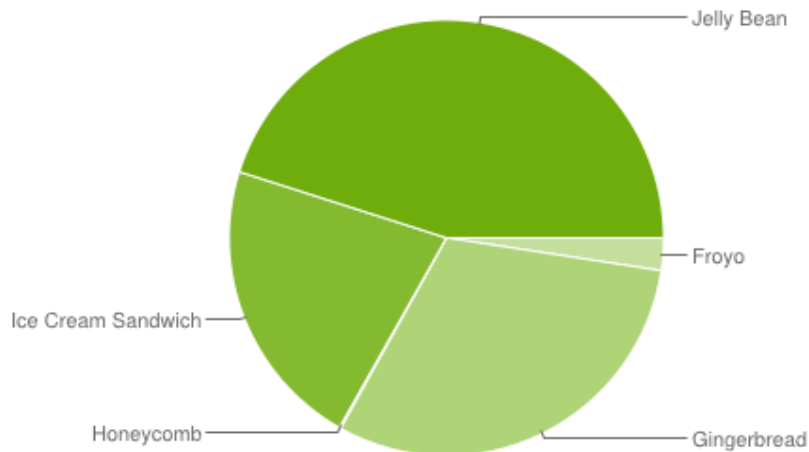
Android 1.6	4	Donut
Android 1.5	3	Cupcake
Android 1.1	2	Banana Bread
Android 1.0	1	Apple Pie

Fuente: <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>.

Consulta: 18 de agosto de 2013.

La cuota de versiones que se muestra en el diagrama son datos tomados a principios de junio de 2013:

Figura 2. **Cuota de versiones de Android utilizadas**



Fuente: http://commons.wikimedia.org/wiki/File:Android_chart.png.

Consulta: 18 de agosto de 2013.

2.2. Servicio de mensajes cortos

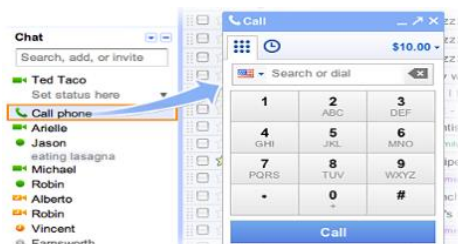
Es un servicio de mensajería para teléfonos, internet o sistemas de comunicación móvil. Utiliza protocolos con estándares de comunicación y permite el intercambio de estos en teléfonos de líneas fijas como también móviles.

Fue desarrollado en 1985 como parte del sistema global para comunicaciones móviles, con las normas de GSM, este es un método para enviar mensajes de hasta 160 caracteres.

2.3. Google Voice

Es un servicio de comunicación de Google, este es configurado por el usuario en la aplicación web base, algo parecido al servicio de correo de Google (Gmail). Este servicio permite realizar llamadas gratis de computadora hacia teléfono, computadora hacia computadora también permitiendo video llamada entre usuarios de Google Voice. Otro de sus servicios es el envío de mensajes de texto. Para instalar Google Voice se accesa al enlace: <http://www.google.com/chat/voice/>

Figura 3. Servicio: Google Voice



Fuente: <http://www.google.com/chat/voice/>. Consulta: 09 de septiembre de 2013.

2.4. Aplicación RingyDingyDingy

RingyDingyDingy es una aplicación ligera para dispositivos Android y permite realizar sonidos en su dispositivo, aunque este se encuentre en modo silencioso o vibración, también bloquear este de forma remota, entre otras funcionalidades. Puede utilizar la función de buscapersonas para que tus amigos o familiares envíen mensajes urgentes haciendo que suene el dispositivo.

RingyDingyDingy principalmente se activa mediante el envío de mensajes de texto con un formato especial en el teléfono, pero también se puede activar a través de Google Voice si tiene aplicación oficial de Google Voice de Google instalado en el dispositivo, lo que permite trabajar RingyDingyDingy en tabletas además de teléfono.

Al instalarse la aplicación genera un identificador único para el dispositivo, y dicho número es la llave por la cual se puede hacer llamados de instrucciones hacia un teléfono en específico.

2.4.1. Sintaxis de RingyDingyDingy

RingyDingyDingy utiliza los mensajes de texto con un formato especial para ejecutar las acciones, inicialmente se envía la abreviatura rdd, seguido por el código de activación único que genera automáticamente la aplicación al ser instalada y por último la sentencia o acción a ejecutar.

Por ejemplo, si su código de activación es 459753 y se desea hacer sonar un dispositivo en específico quedaría de la siguiente manera:

- rdd 459753 ring

Este texto debe ser enviado por mensaje de texto hacia un dispositivo en específico, para que este pueda ejecutar las instrucciones.

Si usted pierde su teléfono y no quiere que sus datos confidenciales caigan en manos equivocadas, puede bloquear el dispositivo de forma remota con el bloqueo de comandos. Tenga en cuenta, que el bloqueo remoto requiere como mínimo del API 8 de Android, para identificar esta API ver tabla de versiones de Sistema Operativo Android.

Como se menciona la versión mínima de Android es 2.2 (Froyo), ya que las versiones anteriores no tienen ninguna forma para bloquear el dispositivo. Además, el bloqueo remoto se encuentra desactivado por defecto, ya que requiere de una confirmación de permisos extras (administrador de dispositivos). Si desea utilizar el bloqueo remoto, asegúrese de ir a Configuración de RingyDingyDingy y vuelva a encenderlo una vez se ha instalado la aplicación.

2.4.2. Licencia de RingyDingyDingy

RingyDingyDingy es *software* libre y de código abierto bajo la licencia GNU GPLv3. El contenido de esta aplicación, así como el logotipo de RingyDingyDingy, están bajo la licencia CC BY-NC-SA 3.0 licencia, su código está alojado en *GitHub Pages*.

2.4.3. Arquitectura de RingyDingyDingy

El funcionamiento de la aplicación es como se describe en el diagrama, se envían mensajes SMS hacia el teléfono que tiene instalada la aplicación este interpreta estas mensajes y ejecuta la acción solicitada, también pueden ser mensajes de Google Voice.

Figura 4. **Arquitectura de aplicación RingyDingyDingy**



Fuente: elaboración propia.

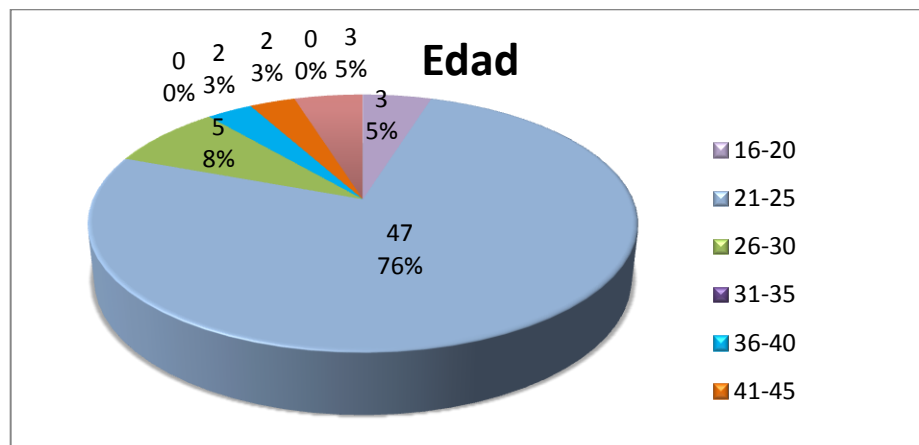
3. CASOS DE ESTUDIO Y ENTREVISTAS

Se desarrolló un estudio del posible mercado y preferencias de los usuarios acerca del control parental en dispositivos móviles. En el cual se puede afirmar que la aplicación sería exitosa según las respuestas de preferencias de los usuarios. Todo depende de cómo sea el método y monto de pago, porque existen aplicaciones gratis, otras tienen precio de \$ 0,99, y las *Premium* superan los precios de \$ 5,00. A continuación el resultado del estudio:

3.1. Edad

Se estableció diferentes rangos de edades, con el fin de establecer el mercado objetivo, el total de personas que respondieron la encuesta es de 62. Los resultados obtenidos fueron los siguientes:

Figura 5. Rango de edades



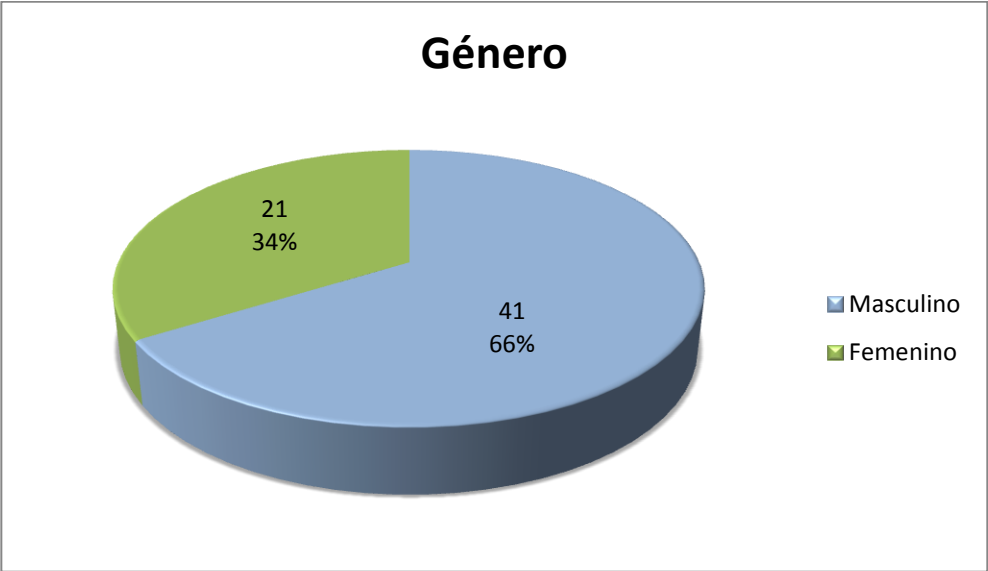
Fuente: elaboración propia.

Este resultado resulta debido a que en la actualidad los usuarios más jóvenes son los que hacen el mayor uso de teléfonos inteligentes, como también son los que conocen de mejor manera el área del mercado, esto quiere decir que conocen las tiendas de aplicaciones según su marca de teléfono, adicionalmente son los que están a la vanguardia de las tecnologías, pero se sabe que son los usuarios que menos desean invertir en compra de aplicaciones, por lo tanto podría ser un contra hacia el producto.

3.2. Género

Se tomó en cuenta el género de las personas encuestadas, esto con el fin de establecer tendencias por género, ya que se podría delimitar funcionalidades en la aplicación según el sexo de los usuarios, adicionalmente que estos datos son importantes al realizar un estudio de mercado.

Figura 6. **Género**

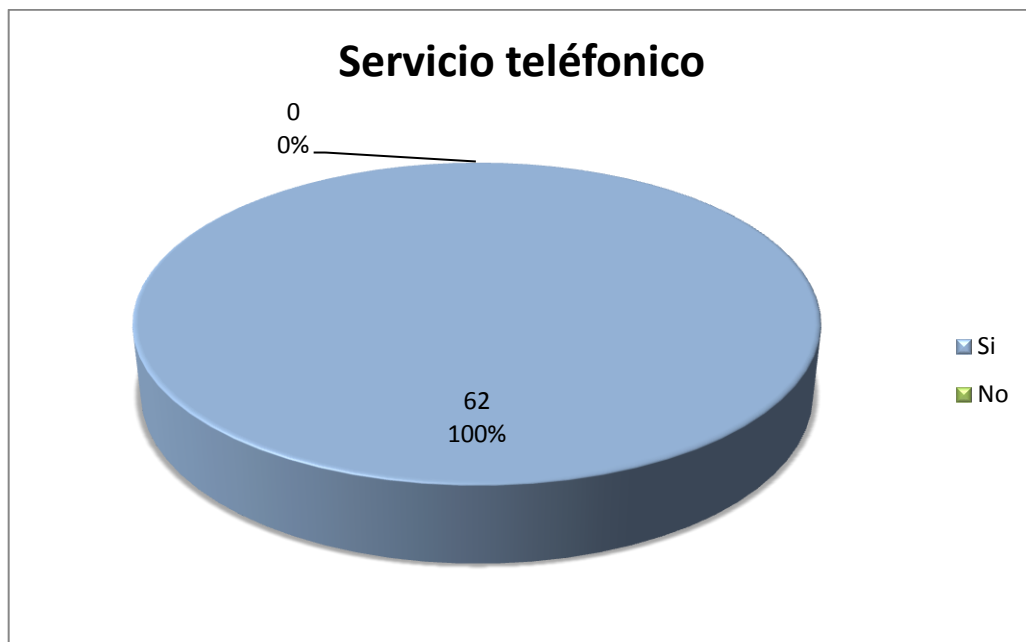


Fuente: elaboración propia.

3.3. Servicio telefónico

Es importante tomar en cuenta la cantidad de personas que en la actualidad utilizan el servicio telefónico, dado que la aplicación se basa en el segmento de dispositivos inteligentes, al ser de este tipo cuentan con el método de comunicación de mensajes de texto:

Figura 7. Servicio telefónico



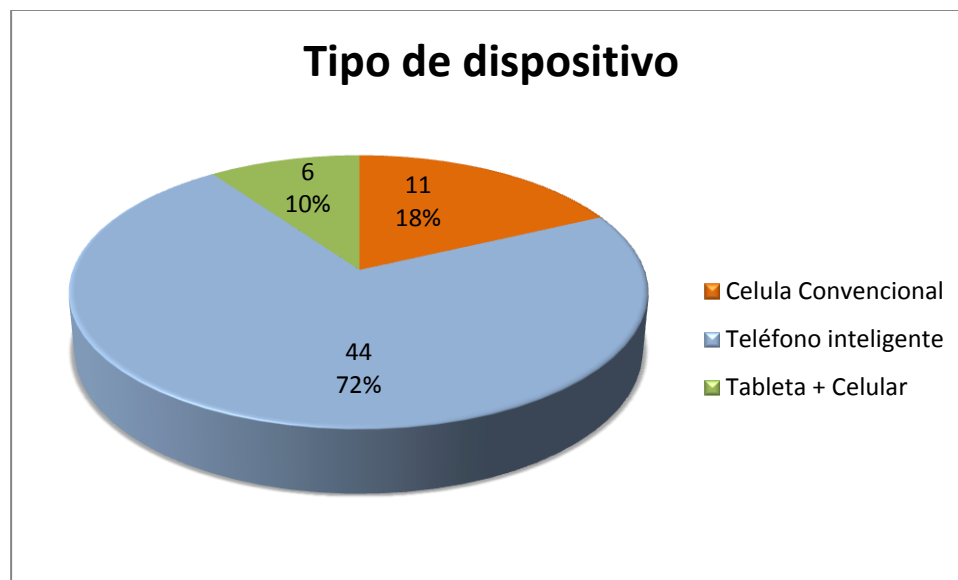
Fuente: elaboración propia.

El 100 por ciento de la muestra utiliza el servicio telefónico, ya sea líneas fijas, celulares o ambas. Es importante mencionar que estos datos fueron recopilados en el departamento de Guatemala, por lo que no aplicaría en el resto de la República. Pero se puede afirmar que en la capital el 100 por ciento de la población utiliza el servicio telefónico.

3.4. Tipo de dispositivo

El dispositivo que utilizan los usuarios en la actualidad es relevante, el mercado objetivo son los dispositivos inteligentes móviles, que cuenten con la opción de usarse como celular

Figura 8. Tipo de dispositivo



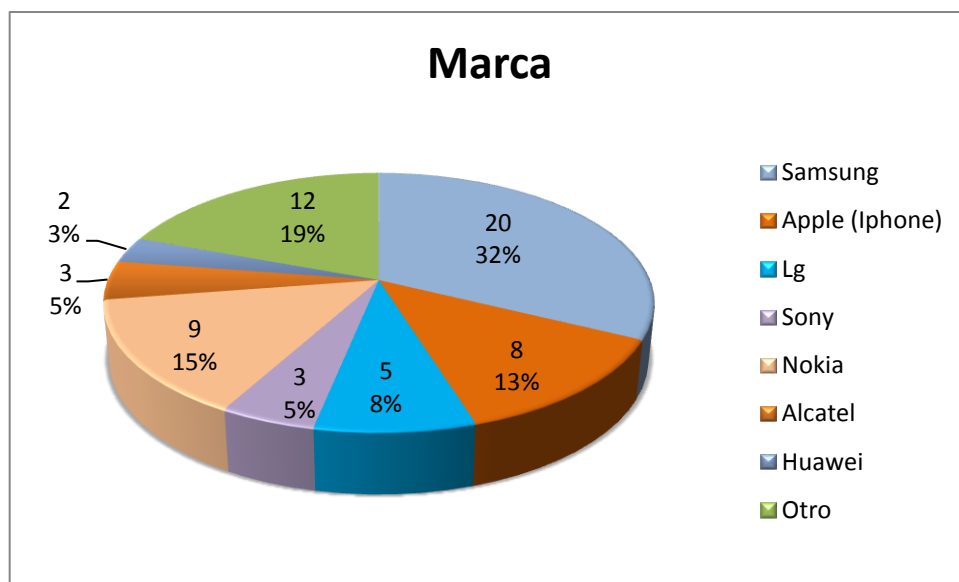
Fuente: elaboración propia.

Se puede considerar que existe un mercado amplio en Guatemala, los resultados indican que el 82 por ciento de las personas utilizan dispositivos móviles inteligentes, sumando el 72 por ciento que son teléfonos inteligentes y el otro 10 por ciento que son tabletas con opción a celular. En este resultado se están incluyendo todas las marcas de dispositivos, por lo tanto habría que descartar aquellos que no utilizan Sistema Operativo Android, debido a que la aplicación fue desarrollada para este.

3.5. Marca

Se cuestionó acerca de la marca actual del dispositivo móvil, esto con el fin de obtener información del sistema operativo preferido de los usuarios, adicionalmente se deben realizar pruebas en los diferentes dispositivos y los resultados indican la prioridad para realizar las pruebas en las diferentes marcas de dispositivos.

Figura 9. Marca



Fuente: elaboración propia.

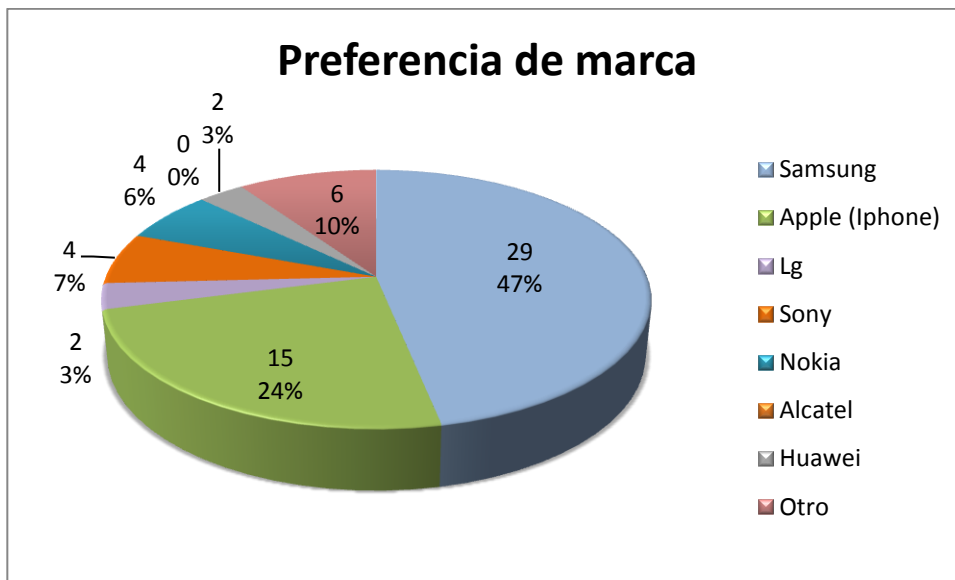
Se puede asumir que el 50 por ciento de los dispositivos utilizados actualmente cuentan con el Sistema Operativo Android, las marcas: Samsung, LG, Sony, Alcatel y Huawei, lo utilizan como software base de sus productos, la suma de sus segmentos da el resultado del 50 por ciento, en el segmento de otros se podría tener otro pequeño porcentaje, ya que no se consideraron las

marcas de Blackberry y/o HTC, esta última utiliza también Sistema Operativo Android, pero también es importante destacar que entre estas repuestas existen teléfonos celulares convencionales, y al no ser teléfonos inteligentes podrían no tener Sistema Operativo.

3.6. Preferencia de marca

En la actualidad se sabe que las marcas pioneras y que pelean en la cima son Samsung con su línea Galaxy y Apple con dispositivos Iphone, pero también existe otro tipo y variedades de marcas que cuentan con muy buenas características.

Figura 10. Preferencia de marca



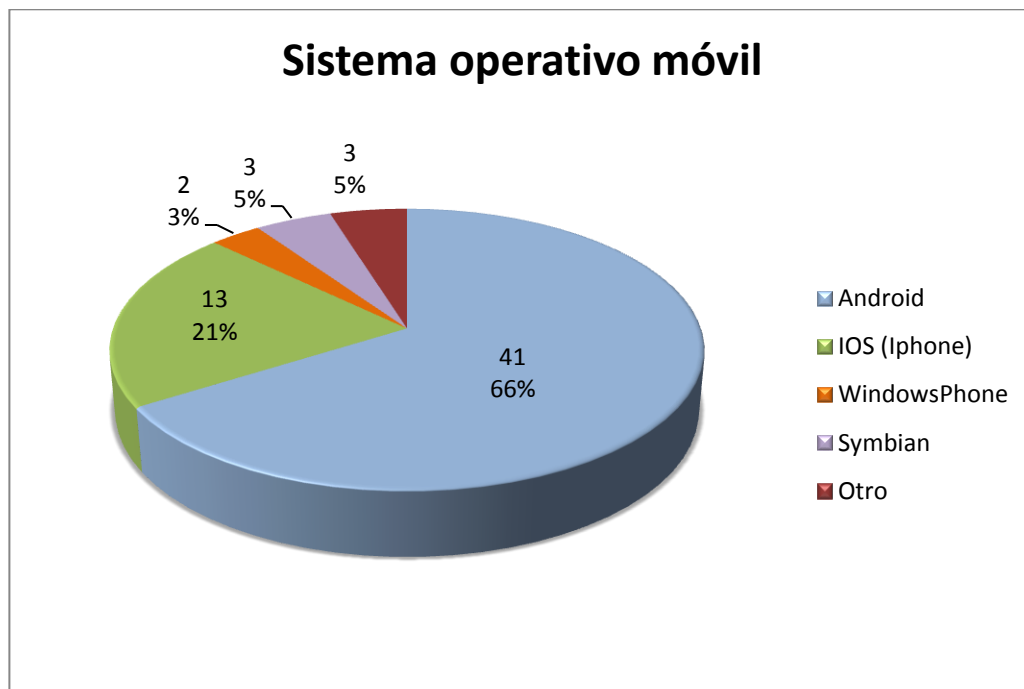
Fuente: elaboración propia.

Se puede observar que la tendencia de los usuarios es obtener dispositivos móviles con sistema operativo Android sumando un total de 66 por ciento, en segundo lugar con 24 por ciento esta Apple, en tercera posición Nokia con sus sistemas operativos Windows Phone y Symbian cuentan con el 6 por ciento.

3.7. Sistema operativo móvil

Se pudo captar información acerca de las preferencias de Sistema Operativo en los dispositivos, este resultados es uno de los más importantes, ya que la aplicación va dirigida hacia el Sistema Operativo Android, pero tampoco se puede descartar ni dejar de tomar en cuenta los demás.

Figura 11. Sistema operativo móvil



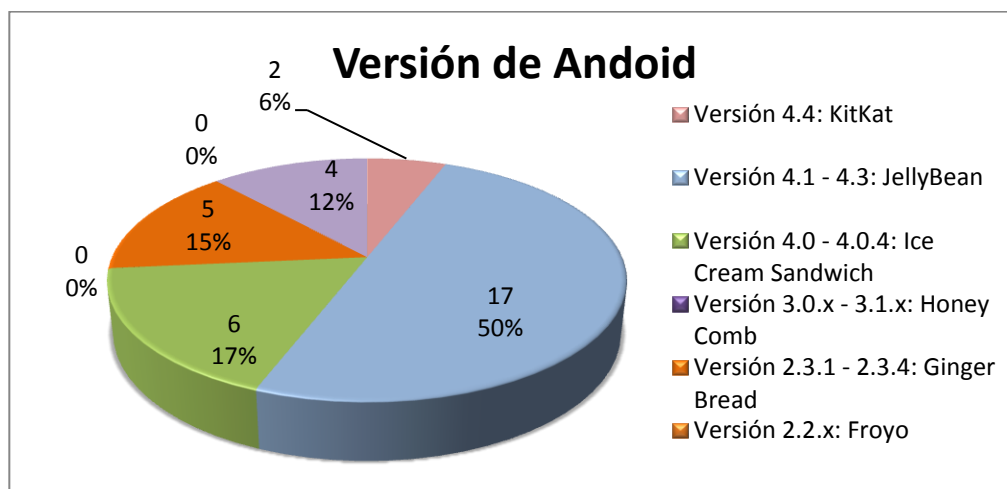
Fuente: elaboración propia.

Se logra notar claramente que la preferencia de los usuarios es Sistema Operativo Android, por lo cual si se desea construir una aplicación para los diferentes lenguajes, este debería de ser la primera opción a desarrollar, ya que en este es donde se pueden captar una mayor cantidad de clientes. De los presentados en el segundo lugar se encuentra IOS de Apple, este pueda que tenga menor cantidad de clientes y usuarios, pero su ventaja es que cuenta con el segmento de usuarios de mejor paga por aplicaciones, debe también de considerarse como principal en desarrollo.

3.8. Versión de Android

Para la cantidad total de personas, 34 utilizan dispositivo con Sistema Operativo Android, de esta cantidad se logró definir las diferentes versiones utilizadas en los dispositivos, debe de especificarse que estos datos son para personas ubicadas en Guatemala:

Figura 12. Versión de Android



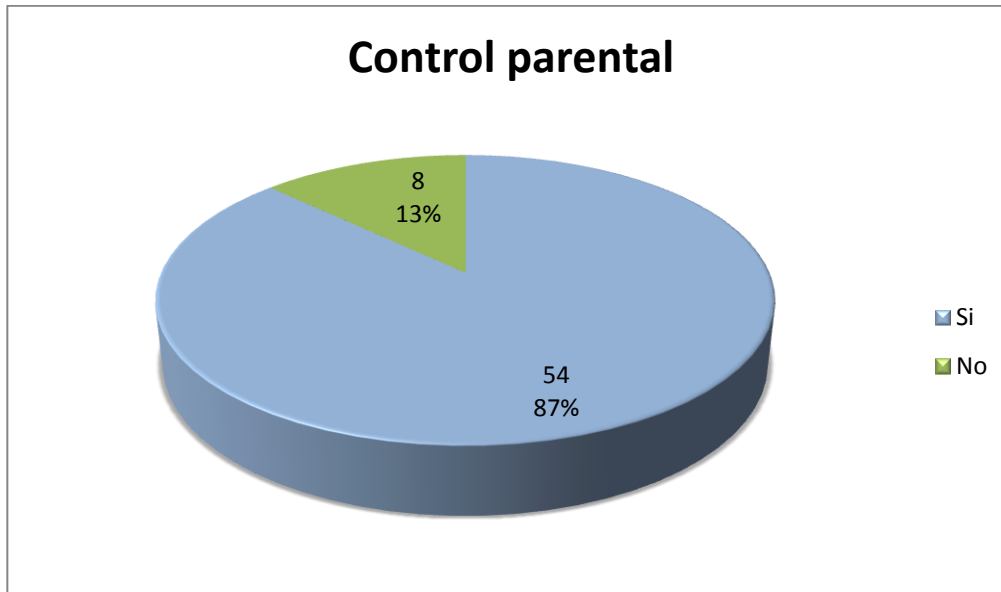
Fuente: elaboración propia.

En la actualidad la mayoría de dispositivos con Sistema Operativo Android cuentan como mínimo con la versión 2.1.x que hace referencia a Eclair, con número de API 7 de Google Android, esto reduce considerablemente el mercado de venta, ya que la aplicación el mínimo de versión requerida es el API 11 que corresponde a la versión 3.0.x (Ginger Bread). Por las diferentes funcionalidades que se implementaron si es necesaria esta versión, por lo tanto la aplicación va dirigida a un mercado de Gama media y alta, pero alrededor del mundo en general el mínimo promedio que se tiene es la versión 3.0.x Ginger Bread.

3.9. Control Parental

Consiste en impedir o limitar el acceso al manejo de dispositivos a menores de edad. Esto se realiza mediante una serie de sistemas de bloqueo, normalmente protegidos mediante claves que configuran los padres o encargados para restringir el uso de la correspondiente máquina. En el acceso de la información en internet es una de las cuestiones de control paterno más discutidas y usadas, ya que muchos de los contenidos de la red no son adecuados para los menores de edad. Los programas de navegación suelen incorporar potentes páginas de configuración, para proceder a la realización de este control de contenidos. Se elaboró una pregunta para definir si las personas consideran que el control parental es necesario en la actualidad, sabiendo la vulnerabilidad de los niños.

Figura 13. **Control Parental**

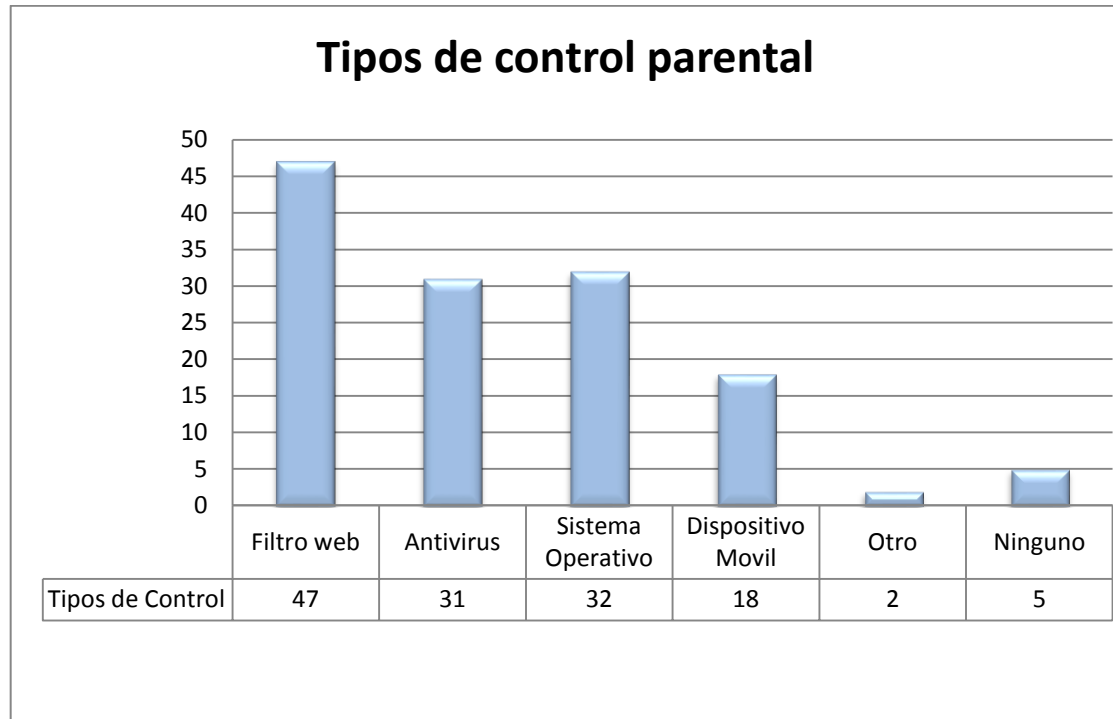


Fuente: elaboración propia.

3.10. Tipos de Control Parental

Existen diferentes tipos, entre los más conocidos son los filtros web, para restringir el accesos a determinados sitios web, en la actualidad los antivirus cuentan con un módulo de Control Parental, siendo esta la segunda opción; también se puede restringir acceso a ciertas funcionalidades por medio de los sistemas operativos, un ejemplo es Windows, donde se pueden crear usuarios con acceso limitados y si desean activar esas funcionalidades, solamente es por medio de la contraseña que configuraron los padres; en dispositivos móviles no es nuevo este tema, ha sido aplicado de diferentes formas.

Figura 14. Tipos de Control Parental



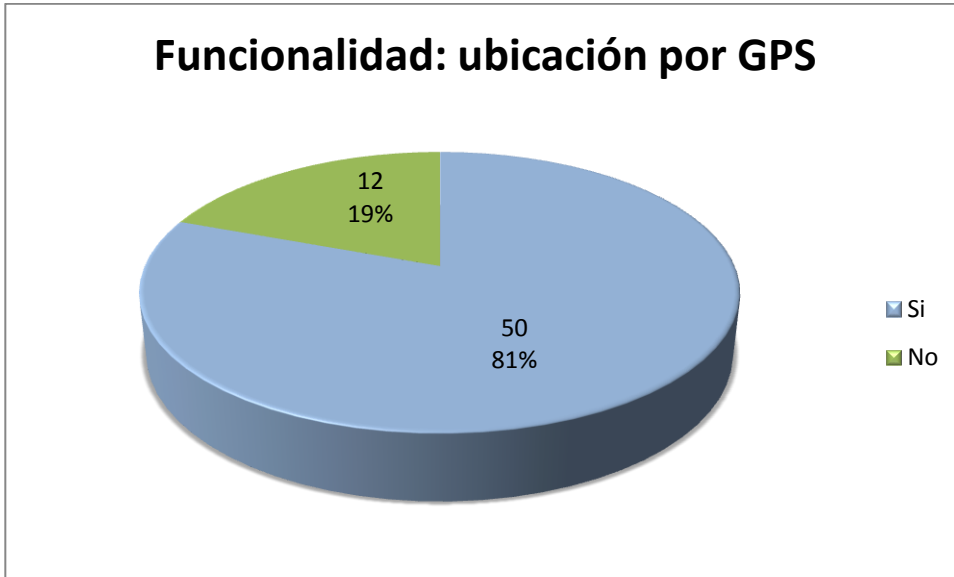
Fuente: elaboración propia.

La mayoría de los usuarios conocen el filtro web, en segundo lugar de los más conocidos esta por medio de Sistema Operativo, le sigue por antivirus, muy poca gente no conoce ninguno.

3.11. Funcionalidad: ubicación por posicionamiento global satelital

La primera funcionalidad de la aplicación es que el padre pueda rastrear la ubicación de sus hijos en cualquier momento, el hijo por ningún momento se percata cuando sucede este proceso.

Figura 15. **Ubicación por posicionamiento global satelital**



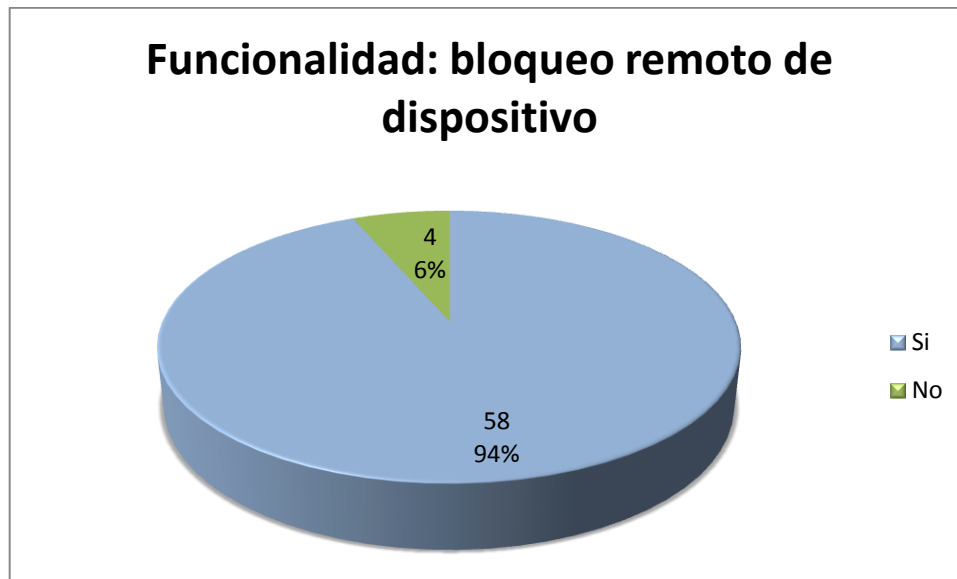
Fuente: elaboración propia.

La mayoría de los usuarios consideraron útil esta funcionalidad, haciendo útil la aplicación en esta área.

3.12. **Funcionalidad: bloqueo remoto**

Se agregó una funcionalidad de bloqueo remoto al teléfono del hijo desde el teléfono del padre, esto para evitar que la información esté en manos equivocadas en caso de robo o pérdida.

Figura 16. **Bloqueo remoto**



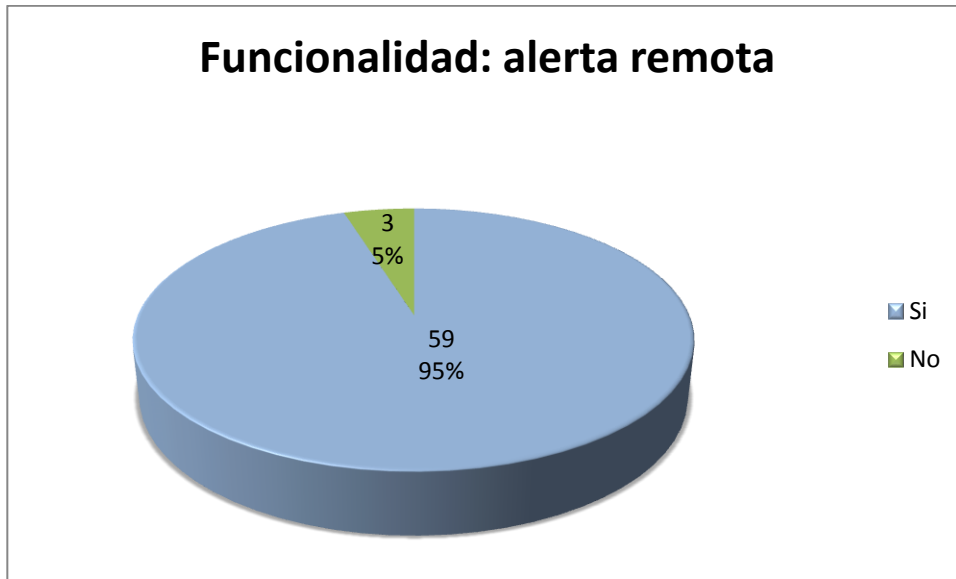
Fuente: elaboración propia.

El 94 por ciento de la población de igual manera consideró útil esta funcionalidad, esto porque la delincuencia en el país es alta, y nadie quiere que sus imágenes, contactos, videos y archivos puedan utilizarlos malintencionadamente.

3.13. Funcionalidad: alerta remota

Se generó además una funcionalidad, en la cual hay un canal de comunicación en caso de emergencia, su funcionalidad principal es que se puedan enviar mensajes a los dispositivos sincronizados y el dispositivo receptor sonará y vibrará aunque este no sea su estado seleccionado, garantizando que el usuario receptor leerá el mensaje.

Figura 17. **Alerta remota**



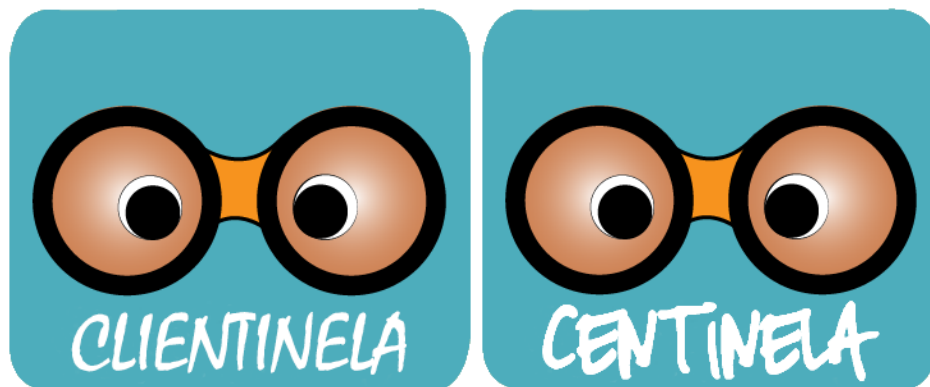
Fuente: elaboración propia.

Esta es la funcionalidad que más interesó a los usuarios, ya que es muy frecuente tener emergencias o situaciones urgentes de comunicación inmediata con los hijos, por lo cual la funcionalidad es bastante útil.

4. PROPUESTA: APLICACIÓN CENTINELA

A partir del estudio realizado, siendo este exitoso y llegando a la conclusión que existe un mercado amplio para las características de la aplicación, además del auge que hoy en día tienen las aplicaciones de dispositivos móviles, se implementó una solución que resuelve la problemática de seguridad en Guatemala en la actualidad, utilizándolas en conjunto con nuevas tecnologías de aplicaciones para dispositivos móviles, se decidió implementar una aplicación que será Cliente-Servidor, en la cual, el padre tendrá el control de ciertas características de los teléfonos inteligentes de sus hijos. Las 2 soluciones que se crearon son las siguientes:

Figura 18. **Icono Centinela, Clientinela**



Fuente: elaboración propia.

La primera será la aplicación servidor, esto quiere decir que es la aplicación que utilizará el padre, y la segunda será la aplicación de hijo(s).

4.1. Centinela

Centinela es la aplicación servidor, esta es la que podrá controlar algunas funcionalidades sobre los teléfonos que cuenten con la aplicación cliente. Su función principal es enviar instrucciones para que se ejecuten en el teléfono que cuente con la aplicación cliente, este ejecuta las acciones y le envía respuesta de nuevo al servidor, toda esta comunicación se da en una sintaxis específica para que sea interpretada por servicios en segundo plano que tiene las dos aplicaciones.

Las características que puede controlar el servidor (padre) sobre el teléfono del cliente (hijo) son las siguientes:

- Ubicación por Posicionamiento Global Satelital (GPS)
- Mensaje de Alerta Remoto
- Bloque de dispositivo

4.1.1. *Splash*

La pantalla principal de Centinela es un *splash* de inicio, este tiene una duración de 3 segundos mientras carga toda la información necesaria en la pantalla principal. Para generar el *splash* de inicio se realizó lo siguiente:

Primero se diseña la vista, siendo este un archivo llamado splash.xml, es importante mencionar que los nombres para las vistas deben ser en minúsculas, para generar la vista de *splash* se utilizó el siguiente código:

Figura 19. **Vista Splash**

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:gravity="bottom"
tools:context=".Splash" >

<ImageView
android:id="@+id/imgSplash"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_centerVertical="true"
android:src="@drawable/splashinicio" />

</RelativeLayout>
```

Fuente: elaboración propia.

Al implementar este código genera la vista que se muestra a continuación:

- Android:id= "@+id/imgSplash", indica el id del objeto imagen.
- Android:layout_width="match_parent", para establecer el ancho del objeto, en este caso se describe que sea el mismo que su contenedor padre.
- Android:layout_height="match_parent", para establecer el alto del objeto, en este caso se describe que sea el mismo que su contenedor padre.
- Android:src="@drawable/spashinicio", ruta para imagen, todas las imágenes se deben de guardar en la carpeta *drawable* del proyecto, la imagen se llama splashinicio.png.
- Android:centerVertical="true", se pide que la imagen se centre y solo se muestre en sentido vertical.

Figura 20. **Splash Centinela**



Fuente: elaboración propia.

Para el modelo, se debe generar una clase .java, esta se llamará Splash.java y lleva el siguiente código:

Figura 21. **Modelo Splash**

```
package com.example.centinelav2;

import java.util.Timer;
import java.util.TimerTask;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;

public class Splash extends Activity {
    private long splashDelay = 3000; //Delay de splash 3
segundos

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);

        TimerTask task = new TimerTask() {
            @Override
            public void run() {
                Intent mainIntent = new Intent().setClass(Splash.this,
MainActivity.class);
                startActivity(mainIntent);
                finish
            }
        };

        Timer timer = new Timer();
        timer.schedule(task, splashDelay
    }
}
```

Fuente: elaboración propia.

Esta clase debe de heredar de *activity*, además se debe importar las librerías de `java.util.Timer`, `java.util.TimerTask`, `android.os.Bundle`, `android.app.Activity`, `android.content.Intent`. En el evento *onCreate* se genera una instancia de `TimerTask`, además se genera un método llamado *run()* dentro de este se genera un *intent*, el *intent* es utilizado para pasar de una clase a otra,

y los parámetros que utiliza son: clase origen y la clase destino a la cual se quiere dirigir, el siguiente paso en el método *run* serían iniciar la actividad siguiente, y la instrucción *finish()*, sirve para destruir la clase actual, el objetivo de esta última instrucción es que no puedan regresar a la pantalla de *splash*. Fuera del método *run* se instancia un nuevo *timer*, y se programa el *timer* con un tiempo de espera de 3 segundos, cada segundo equivale a 100.

Por último se debe declarar esta clase en el archivo *Manifest.xml*, este representa el controlador de la aplicación, aquí se declaran todas las clases, permisos de aplicación, como también prioridades de los procesos. El código que se debe agregar en el Manifest es el siguiente, este código se agrega dentro de la etiqueta *<application>*:

Figura 22. **Controlador *Splash***

```
<activity
    android:name=".Splash"
    android:label="@string/app_name"
    android:theme="@android:style/Theme.NoTitleBar.Full
screen"
    android:screenOrientation="portrait">
    <intent-filter>
        <action
            android:name="android.intent.action.MAIN" />
        <category
            android:name="android.intent.category.LAUNCHER
" />
        </intent-filter>
    </activity>
```

Fuente: elaboración propia.

Se declara la clase *Splash.java*, la cual debe llevar de parámetro el *app_name*, se quitará la barra de la aplicación para que la imagen se muestre

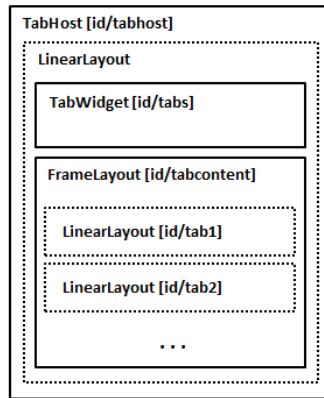
fullscreen, esto quiere decir que ocupe todo el espacio de la pantalla, la orientación de la pantalla para el *splash* será *portrait*, indicando que se mostrará en dirección vertical. Se genera un intento *filter*, configurando esta clase como la principal de la aplicación, para que sea lo primero que se muestra al ejecutarla. Se debe eliminar el código del *intent-filter* que genera por definición en la clase MainActivity, ya que la clase principal ahora es Splash.java

4.1.2. Clase principal (MainActivity)

Para la clase principal, se utilizaron pestañas para mostrar las 2 secciones principales, la primera es el listado de dispositivos sincronizados y en la segunda se pueden añadir nuevos dispositivos. Para generar la vista en el archivo main_activity.xml se implementó lo siguiente:

Se utilizó el control TabHost, este es el contenedor principal de las pestañas. Dentro se incluye varios LinearLayout, para distribuir las siguientes secciones de la aplicación, la sección de pestañas en la parte superior y la sección de contenido en la parte inferior. La lista de pestañas se mostrará mediante un elemento TabWidget y como contenedor para los controles de cada pestaña se añade FrameLayout. Por último dentro del FrameLayout se incluye el contenido de cada pestaña con un identificador único que permita posteriormente hacer referencia a ellos fácilmente.

Figura 23. **Estructura MainActivity**



Fuente: <http://www.sgoliver.net/blog/?p=2112>. Consulta: 09 de septiembre de 2013.

Si se traduce esta estructura al fichero de layout XML, se tendrá algo parecido a lo siguiente:

Figura 24. **Pantalla principal**



Fuente: elaboración propia.

Para agregar iconos en las pestañas se implementa el siguiente código en la clase MainActivity.java, este debe ser dentro del método onCreate().

Figura 25. Configuración de pestañas

```
Resources res = getResources();
pestanas = (TabHost)findViewById(android.R.id.tabhost);
pestanas.setup();

TabHost.TabSpec spec=pestanas.newTabSpec("dispositivo");
spec.setContent(R.id.pnLDispositivos);

spec.setIndicator("",res.getDrawable(R.drawable.iconodispositivo));
pestanas.addTab(spec);
spec=pestanas.newTabSpec("agregar");
spec.setContent(R.id.pnLAgregar);
spec.setIndicator("",res.getDrawable(R.drawable.iconoagregar));
pestanas.addTab(spec);

pestanas.setCurrentTab(0);
```

Fuente: elaboración propia.

El código indica que, para la primera pestaña se crea un objeto TabSpec, este tendrá la etiqueta dispositivo, se le asigna el contenido de uno de los LinearLayout que se incluye en la sección de contenido (en este caso R.id.pnLDispositivos), el icono android.R.drawable.iconodispositivos (icono importado a la aplicación, copiándolo a la carpeta drawable-hdpi). Finalmente se añade la nueva pestaña al control mediante el método addTab().

Uno de los eventos disponibles del control TabHost, pero que usualmente este no es muy utilizado, pero ayudaría a identificar que pestaña fue seleccionada, es el siguiente:

Figura 26. **Icono pestaña**

```
setOnTabChangeListener(new OnTabChangeListener() {  
    @Override  
    public void onTabChanged(String tabId) {  
        Log.i("AndroidTabsDemo", "Pulsada pestaña: " + tabId);  
    }  
});
```

Fuente: elaboración propia.

En el método `onTabChanged()` se recibe como parámetro de la etiqueta que identifica la pestaña (no `Id`), esta es la que se asigna cuando se crea su objeto `TabSpec` correspondiente.

Por último como toda clase o modelo en la aplicación, debe de declararse en el archivo `AndroidManifest`, con el siguiente código:

Figura 27. **Controlador MainActivity**

```
<activity  
    android:name=".MainActivity"  
    android:label="@string/app_name" >  
</activity>
```

Fuente: elaboración propia.

4.1.3. Envío de mensaje

Centinela se comunica con la aplicación cliente (`Clientinela`) por medio de mensajes de texto, en este se envía las instrucciones que se desea ejecutar en el teléfono cliente. La función de `Clientinela` es interpretar las solicitudes

enviadas por el padre por medio de mensajes de textos, estas ejecutan la acción correspondiente y envían respuesta de nuevo a Centinela.

Las instrucciones que envía Centinela por medio de mensaje de texto son las siguientes:

- Solicitud de sincronización
- Solicitud de coordenadas
- Bloqueo de pantalla
- Mensaje remoto
- Eliminar Centinela

Para el envío de mensaje se implementó una clase con el nombre EnviarMensaje.java, en esta se utilizará el siguiente código fuente:

Figura 28. Clase de envío de SMS

```
package com.example.centinelav2;
import android.content.Context;
import android.telephony.SmsManager;
public class EnviarMensaje{
    public static void enviarSMS (Context context, String
        NumeroTelefono, final String Mensaje, final String CajaMensaje,
        final String Error){
        try {
            SmsManager smsManager= SmsManager.getDefault();
            smsManager.sendTextMessage(NumeroTelefono, null,
                Mensaje, null, null);
            Toast.makeText(context,
                CajaMensaje,Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            Toast.makeText(context,Error,Toast.LENGTH_LONG).show();
            e.printStackTrace();}}}
```

Fuente: elaboración propia.

Se debe de importar Context, Telephony.SmsManager, Widget.Toast. La clase será pública y su método principal será *static*, esto es para que desde cualquier clase del proyecto se pueda enviar mensaje y hacer referencia al método sin hacer instancia de la clase EnviarMensaje.java, los parámetros que necesita el método son *context*, este será de tipo *context*, *String* Número Teléfono, para establecer hacia qué número de teléfono se necesita enviar el mensaje, *String* mensaje, texto que se desea enviar en el cuerpo del mensaje, *String* CajaMensaje, si el envío de mensaje fue exitoso se mostrará una notificación de confirmación, de lo contrario se mostrar un mensaje de error de envío y para eso se utilizará el parámetro *String* Error.

4.1.4. Recibir mensaje

La aplicación implementa un servicio en segundo plano para recibir las respuestas enviadas desde Clientinela, este servicio es para la lectura de mensaje de texto. La prioridad que se establece es alta, para que sea la primera aplicación que interprete los mensajes.

Si los mensajes de entrada tienen la sintaxis correcta según interprete la aplicación, ejecuta la acción y borra el mensaje, esto con la finalidad que los mensajes de la aplicación Centinela y Clientinela nunca lleguen a la bandeja de entrada ni de salida de los mensajes de texto, si el mensaje no tiene la sintaxis deseada se deja que el mensaje pase hacia la bandeja de entrada o bien hacia la aplicación que tenga la segunda prioridad para leer mensajes.

El código que se implementa para este servicio es el siguiente:

Figura 29. Servicio de recepción de mensaje

```
package com.example.centinelav2;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.widget.Toast;

public class RecibirMensaje extends BroadcastReceiver{
    Notificaciones notis;

    @Override
    public void onReceive(Context context, Intent intent) {
        //Codigo apra interpretar mensaje
    }
}
```

Fuente: elaboración propia.

Las importaciones para esta clase son: `Android.content.context`, `Android.telephony.SmsMessage`. Esta clase debe de heredar de `BroadcastReceiver`, y el método para lectura de mensajes será `onReceive`, recibiendo de parámetros el contexto y el *intent*. Y dentro del método se escribirá el código deseado para la aplicación. El servicio interpretará una sintaxis específica que es la siguiente:

- `<clientinela#ecys>` instrucción `</clientinela#ecys>`

En la instrucción puede venir lo siguiente:

- Si
- No

- gps: latitud ,longitud ; precisión

Figura 30. **Validación: mensaje recibido**

```
Bundle bundle = intent.getExtras();
SmsMessage[] ListaMensajes = null;
String mensaje = "";

if(bundle != null) {
    Object[] pdus = (Object[]) bundle.get("pdus");
    ListaMensajes = new SmsMessage[pdus.length];

    for(int i=0; i < ListaMensajes.length; i++) {
        ListaMensajes[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
        mensaje = ListaMensajes[i].getMessageBody().toString();
        String origen = ListaMensajes[i].getOriginatingAddress();

        if(mensaje.length()>37 && mensaje.contains("</")){
            //primera validacion,
            //aqui sigue el codigo de parseo
        }
    }
}
```

Fuente: elaboración propia.

Aquí se genera una instancia de objeto del tipo `SmsMessage[]`, este es un arreglo en el cual se obtendrá la cola de mensajes, y un *string* mensaje, donde se recibirá el mensaje que se interpreta. Se genera un objeto tipo `Object` en el cual se obtiene como tal la cola de mensajes, ya después se instancia la `Listamensajes` recibiendo de parámetros toda la cola de mensajes, para poder ser recorrida.

Después se genera un ciclo para recorrer `ListaMensajes`, se obtiene el mensaje en la posición `i`, como también el número de teléfono del cual fue enviado el mensaje.

Por último se procede a evaluar el mensaje, este debe cumplir con varias validaciones; la primera es que por lo menos debe tener 37 caracteres, ya que esa es la cantidad de caracteres en las etiquetas, y debe de traer </ para identificar que contiene la etiqueta de cierre. Posteriormente se proceden las siguientes validaciones:

Figura 31. **Validación etiquetas: mensaje recibido**

```
String inicioetiqueta = mensaje.substring(0, 18);
String finetiqueta = mensaje.substring(mensaje.indexOf("</"));
String tipomensaje = mensaje.substring(18, mensaje.indexOf("</"));

if (inicioetiqueta.equals("<clientinela#ecys>") &&
    finetiqueta.contains("</clientinela#ecys>")){
    this.abortBroadcast();
    ManejadorBaseDatos cdb = new
    ManejadorBaseDatos(context, "Centinela1", null, 1);
    //segunda validacion
    //codigo de instrucciones
}
```

Fuente: elaboración propia.

Si cumple con la primera validación, asegurarse que el mensaje cumple con la sintaxis correspondiente, el cual debe de traer el inicio de etiqueta <clientinela#ecys> y el cierre de etiqueta </clientinela#ecys>. Dentro de estas etiquetas vendrá la instrucción a ejecutarse, y si cumple con la validación se procede a ejecutar la sentencia `this.abortBroadcast()` con dicha sentencia se obliga al teléfono a que el mensaje se elimine y no llegue a la bandeja de entrada, además se instanciará el objeto que maneja la base de datos, en esta se encuentran los métodos para almacenar la información en las entidades según lo requerido.

A continuación se explican las tres acciones que se ejecutarán según los mensajes recibidos de Clientinela:

Figura 32. **Sentencia Si**

```
if(tipomensaje.equals("si")){
    Notificaciones.titulo="Aceptacion de Solicitud";
    Notificaciones.cuerpo="Cel: " + origen + ", ha aceptado tu
Solicitud !!!!!";
    Notificaciones.MostrarNotificacion(context);
    Toast.makeText(context, origen + ": Confirmacion Solicitud",
Toast.LENGTH_LONG).show();
    cdb.actualizarEstadoDispositivo(origen.replace("+502", ""), "A");
    cdb.insertarHistorial(origen.replace("+502", ""), "Solicitud",
"Dispositivo acepto tu solicitud");
}
```

Fuente: elaboración propia.

Si el texto dentro de las etiquetas contiene la palabra si, se procederá a mostrar una notificación, en la cual avisará al usuario que el teléfono con Clientinela al cual se le envió solicitud ha aceptado y podrá controlar las tres funciones básicas. Además, el teléfono con estado de solicitud cambia a estado activo en la base de datos y automáticamente se empieza el historial de información sobre este teléfono, por lo cual se guardan estos 2 registros en la base de datos.

Figura 33. **Sentencia No**

```
if (tipomensaje.equals("no")){
    Notificaciones.titulo="Solicitud Rechazada";
    Notificaciones.cuerpo="Cel: " + origen + ", ha rechazado tu
    Solicitud :(";
    Notificaciones.MostrarNotificacion(context);
    Toast.makeText(context, origen + ": Cancelacion Solicitud",
    Toast.LENGTH_LONG).show();
    cdb.insertarHistorial(origen.replace("+502", ""), "Solicitud",
    "Dispositivo no acepto tu solicitud");
}
```

Fuente: elaboración propia.

Si el texto dentro de las etiquetas contiene la palabra no, se procederá a mostrar una notificación, en la cual mostrará al usuario que el teléfono con Clientinela al cual se le envió solicitud ha rechazado la misma, y se guarda en el historial de información los datos de la instrucción.

Figura 34. **Sentencia gps**

```
if (tipomensaje.contains("gps:")){
    Toast.makeText(context, origen + ": Coordenadas recibidas",
    Toast.LENGTH_LONG).show();
    double Latitud =
    Double.parseDouble(tipomensaje.substring(tipomensaje.indexOf(":")+
    1, tipomensaje.indexOf(",")));
    double Longitud =
    Double.parseDouble(tipomensaje.substring(tipomensaje.indexOf(",")+
    1, tipomensaje.indexOf(";")));
    double Presicion =
    Double.parseDouble(tipomensaje.substring(tipomensaje.indexOf(";")+
    1));
    cdb.insertarHistorial(origen.replace("+502", ""), "GPS",
    "Coordenadas recibidas: Latitud:"+Latitud + ", Longitud:"+
    Longitud);
}
```

Continuación de la figura 34.

```
    Mapa.AgregarMarcador(Latitud, Longitud, 20);
```

Fuente: elaboración propia.

Si el texto dentro de las etiquetas contiene la palabra `gps`, es porque el teléfono al cual se le envió solicitud de coordenadas ha respondido con su ubicación, se obtienen los valores latitud, longitud y precisión y se procede a agregar un marcador en el mapa, la ubicación de dicha persona como también a guardar las coordenadas en el historial para ser consultadas en cualquier momento.

Por último esta clase debe de declararse en el archivo `Manifest.xml`, para poder ser usada por la aplicación, esta tiene una forma peculiar de declararse, ya que en esta parte debe de establecerse la prioridad del servicio, como también que este se ejecute en segundo plano.

Figura 35. **Declaración recibir mensaje**

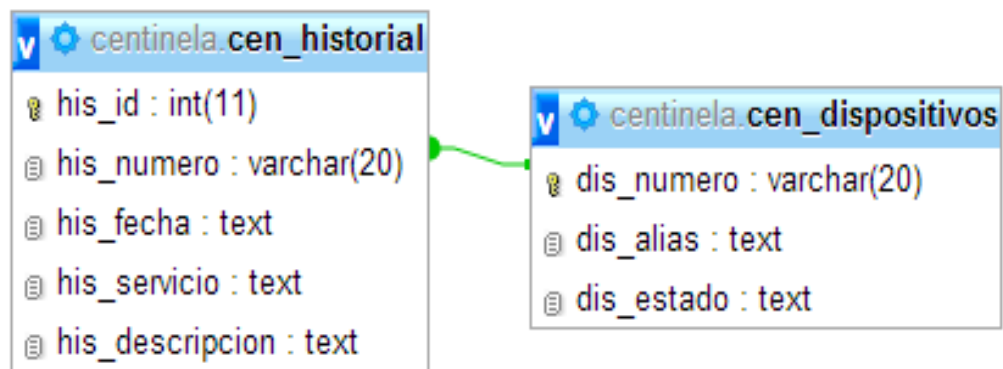
```
<receiver android:name=".RecibirMensaje"
    android:enabled="true"
    android:exported="true">
    <intent-filter android:priority="2147483645">
        <action
            android:name="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
```

Fuente: elaboración propia.

4.1.5. Base de datos SQLite

Centinela y Clientinela utilizan una base de datos SQLite para el manejo de la información, esta base de datos solamente usa dos entidades, estas sirven para almacenar el historial de acciones y el listado de dispositivos sincronizados, el diagrama entidad relación se muestra a continuación:

Figura 36. Diagrama entidad relación



Fuente: elaboración propia.

Los dispositivos Android cuentan con un sistema para administrar las bases de datos SQLite, el manejo de estas es por medio de una clase, la cual se declaró con el nombre de ManejadorBaseDatos.java, todas las sentencias para SQLite son prácticamente con la sintaxis de SQL, pero solamente cuentan con una cantidad limitada de acciones, el código inicial para esta clase es la siguiente:

Figura 37. **Importaciones de clases de base de datos**

```
package com.example.centinelav2;

import java.text.SimpleDateFormat;
import java.util.Calendar;

import android.annotation.SuppressLint;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;

public class ManejadorBaseDatos extends SQLiteOpenHelper{
    public static final String DATABASE_NAME = "MY_DATABASE";
    public static final int DATABASE_VERSION = 1;
    Context context;
    SQLiteDatabase sqlDatabase;

    String sqlCreateDispositivos = "CREATE TABLE cen_dispositivo
(" +
        "dis_numero TEXT PRIMARY KEY, " +
"dis_alias TEXT," +
        "dis_estado TEXT)";

    String sqlCreateHistorial = "CREATE TABLE cen_historial (" +
        "his_id INTEGER PRIMARY KEY AUTOINCREMENT," +
        "his_numero TEXT, " +
        "his_fecha TEXT, " +
        "his_servicio TEXT," +
        "his_descripcion TEXT)";
```

Fuente: elaboración propia.

Las importaciones de clases son necesarias, algunas se utilizan para manejo de fechas, otra para almacenamiento de datos, otros para obtener las consultas y almacenarlas en los cursores, y los básicos para la base de datos son:

- Android.database.sqlite.SQLiteDatabase
- Android.database.sqlite.SQLiteDatabase.CursorFactory
- Android.database.sqlite.SQLiteOpenHelper

La clase debe de extender de SQLiteOpenHelper y se crean variables para el nombre de la base de datos como la versión, además de los *string* para generar las entidades de la base de datos.

Figura 38. **Métodos de creación base de datos**

```

public ManejadorBaseDatos(Context context, String name, CursorFactory
factory, int version) {
    super(context, name, factory, version);
}

public void onCreate(SQLiteDatabase db) {
    db.execSQL(sqlCreateDispositivos);
    db.execSQL(sqlCreateHistorial);
}

public void onUpgrade(SQLiteDatabase db, int VersionAnterior, int
VersionNueva) {
    db.execSQL("DROP TABLE IF EXISTS cen_dispositivo");
    db.execSQL("DROP TABLE IF EXISTS cen_historial");
    db.execSQL(sqlCreateDispositivos);
    db.execSQL(sqlCreateHistorial);
}

```

Fuente: elaboración propia.

Se generan estos tres métodos al heredar del SQLiteOpenHelper el constructor en el cual se tiene los parámetros del contexto y la versión de la base de datos.

El método `onCreate` se utiliza cuando se crea la base de datos por primera vez, y este ejecutará las sentencias de creación de las entidades de la base de datos.

El método `onUpgrade`, se implementa para actualización de la versión de la aplicación, se eliminarán las tablas existentes y se volverán a crear en blanco. Es importante mencionar que si no quisieran perder los datos al actualizar la versión justo antes de eliminar las entidades, se debe sacar una copia de ellas y al volverlas a crear, para que se vuelvan a cargar los datos a partir del *backup*.

Las instrucciones del Lenguaje de Manipulación de Datos, se pueden ejecutar de 2 formas:

- `This.getWritableDatabase()`, indicando si es *insert*, *update* o *delete*, enviándole los parámetros de entidad a modificar, los valores, y en los casos que aplique las sentencia para los *where*.
- `This.getReadableDatabase.query("query")`, consulta a ejecutarse para modificación de entidad, puede ser una sentencia *select*, *update* o *delete* como se conocen en cualquier base de datos SQL.

Figura 39. **ABC de entidad, SQLite**

```
public void eliminarDispositivo(String numero){
    this.getWritableDatabase().delete("cen_dispositivo", "dis_numero="+numero,
    null);
    this.getWritableDatabase().delete("cen_historial", "his_numero="+numero,
    null);
}
```

Continuación de la figura 39.

```
public void insertarDispositivo(String numero , String alias, String estado){
    ContentValues valores = new ContentValues();
    valores.put("dis_numero", numero);
    valores.put("dis_alias", alias);
    valores.put("dis_estado", estado);
    this.getWritableDatabase().insert("cen_dispositivo", null, valores);
}

public void actualizarEstadoDispositivo(String numero , String estado){
    ContentValues valores = new ContentValues();
    valores.put("dis_estado", estado);
    this.getWritableDatabase().update("cen_dispositivo", valores,
    "dis_numero="+numero, null);
}

public Cursor consultarDispositivos(){
    String columnas[]={"dis_numero", "dis_alias", "dis_estado"};
    Cursor c = this.getReadableDatabase().query("cen_dispositivo",
    columnas, null, null, null, null, null);
    return c;
}
```

Fuente: elaboración propia.

4.1.6. Notificación

Se pueden manejar notificaciones para ser mostradas en la barra de notificaciones, para realizar estas se implementó una clase llamada Notificaciones.java, esta se declaró con métodos y atributos estáticos para poder ser accedidos desde cualquier clase de la aplicación. Los atributos son los siguientes:

- Cuerpo: este será el cuerpo de la notificación.
- Título: título de la notificación

Las notificaciones también manejan una imagen por defecto, y la acción que se desee que realice cuando le den un toque desde la barra. También se

deben importar NotificationManger, PendinIntent, BroadcastReceiver, Intent, NotificationCompat, TaskStackBuilder. La clase debe de heredar de BroadcastReceiver. El código es el siguiente:

Figura 40. Clase notificación

```
package com.example.centinelav2;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.support.v4.app.NotificationCompat;
import android.support.v4.app.TaskStackBuilder;

public class Notificaciones extends BroadcastReceiver {
    static final int IdNotificacion = 1;
    static String cuerpo="", String titulo="";
    int contador;

    public void onReceive(Context context, Intent intent) {
        MostrarNotificacion(context);
    }
    public static void MostrarNotificacion(Context context){
        NotificationCompat.Builder mBuilder = new
        NotificationCompat.Builder(context).setSmallIcon(R.drawable.inic
        io).setContentTitle(titulo).setContentText(cuerpo);
        Intent resultIntent = new Intent(context, MainActivity.class);
        TaskStackBuilder stackBuilder =
        TaskStackBuilder.create(context);
        stackBuilder.addParentStack(MainActivity.class);
        stackBuilder.addNextIntent(resultIntent);
        PendingIntent resultPendingIntent =
        stackBuilder.getPendingIntent(0,
        PendingIntent.FLAG_UPDATE_CURRENT);
        mBuilder.setContentIntent(resultPendingIntent);
        NotificationManager mNotificationManager = (NotificationManager)
        context.getSystemService(Context.NOTIFICATION_SERVICE);
        mNotificationManager.notify(IdNotificacion, mBuilder.build());
    }
}
```

Fuente: elaboración propia.

4.1.7. Mapas

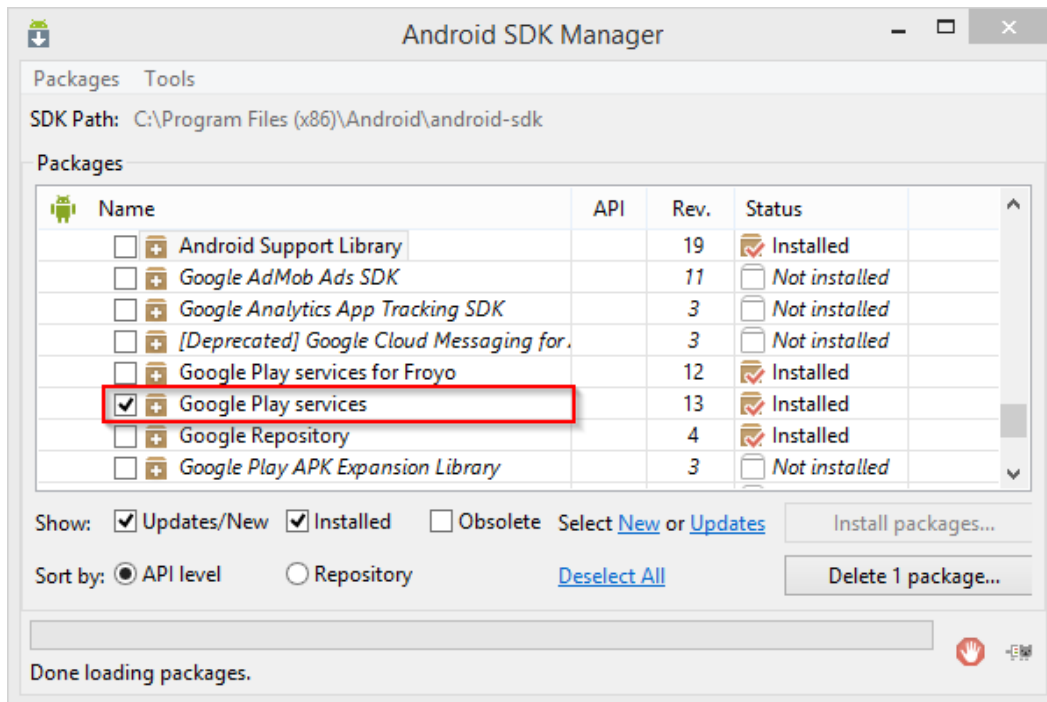
La función de localización es prácticamente la principal entre Centinela y Clientinela, como se mencionó anteriormente, Centinela le envía un mensaje de solicitud de coordenadas a Clientinela, este interpreta el mensaje, obtiene su ubicación por medio del *GPS* incluido en el teléfono, o bien por medio de la red del proveedor del servicio móvil, y dichos datos los envía de vuelta, Centinela se encarga de interpretar la respuesta y mostrarla en un mapa, utilizando el servicio de Google Maps. Para lograr que Centinela pueda interpretar coordenadas y mostrarlas en el mapa se realizan en general lo siguiente:

- Descargar y configurar los paquetes de Google Play Services
- Obtener una *API key*
- Configurar permisos y especificaciones en Manifest
- Manejo de objetos de mapas para su visualización

4.1.7.1. Descargar y configurar los paquetes de Google Play Services

Al descargar el SDK de *Android*, se ejecuta el *SDK Manager* para descargar el paquete extra de Google Play Services, a continuación se muestra la ubicación de este extra, el siguiente paso es descargarlo e instalarlo, de preferencia descargar la revisión actual, en este caso se descargó la revisión 13:

Figura 41. Descarga paquete Google Play Service



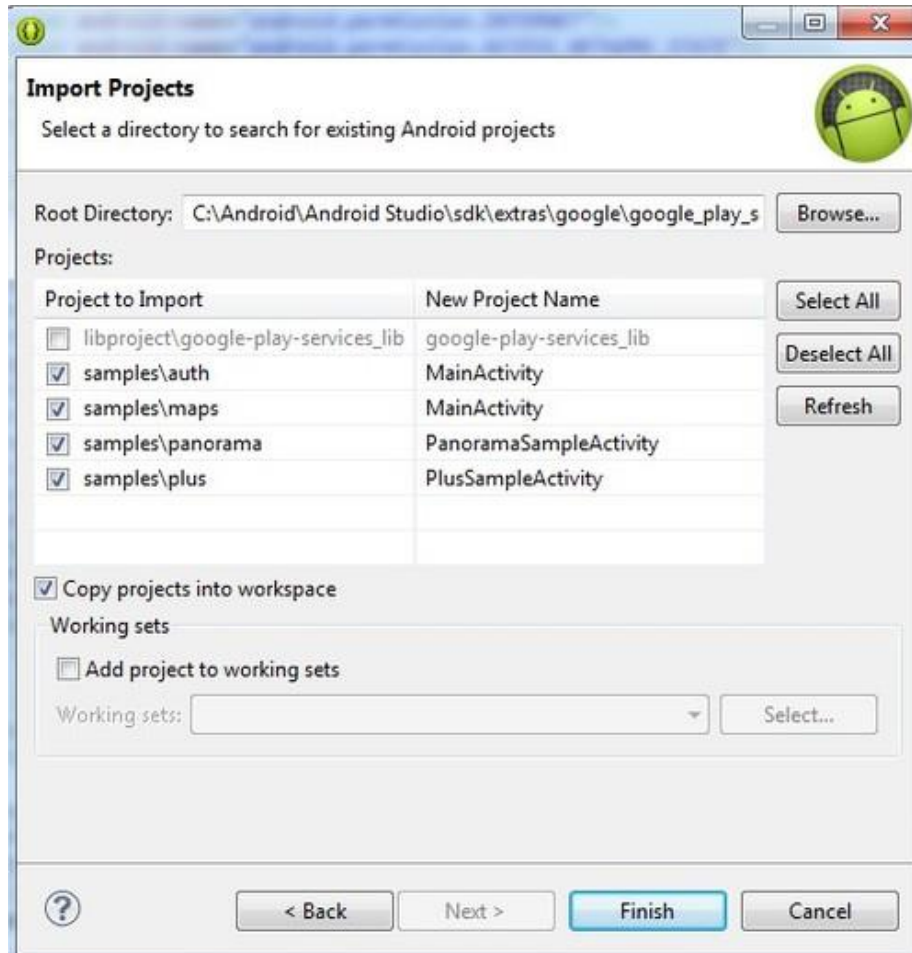
Fuente: <http://www.sgoliver.net/blog/wp-content/uploads/2013/08/SDK-Manager-Google-Play-Services-2.png>. Consulta: 15 de septiembre de 2014.

Después de descargado se puede ubicar este apk en la ruta:

- <android-sdk>/extras/google/google_play_services/

El siguiente paso es importar esta carpeta como una librería. Para realizar este paso se abre Eclipse (IDE de Java para desarrollo móvil), sobre la barra de tareas pulsar *File*, ubicar *Import*, después *Android* y por último presionar la opción *Existing Android Code into Workspace*, se busca la ruta antes mencionada y presionar *Finish*.

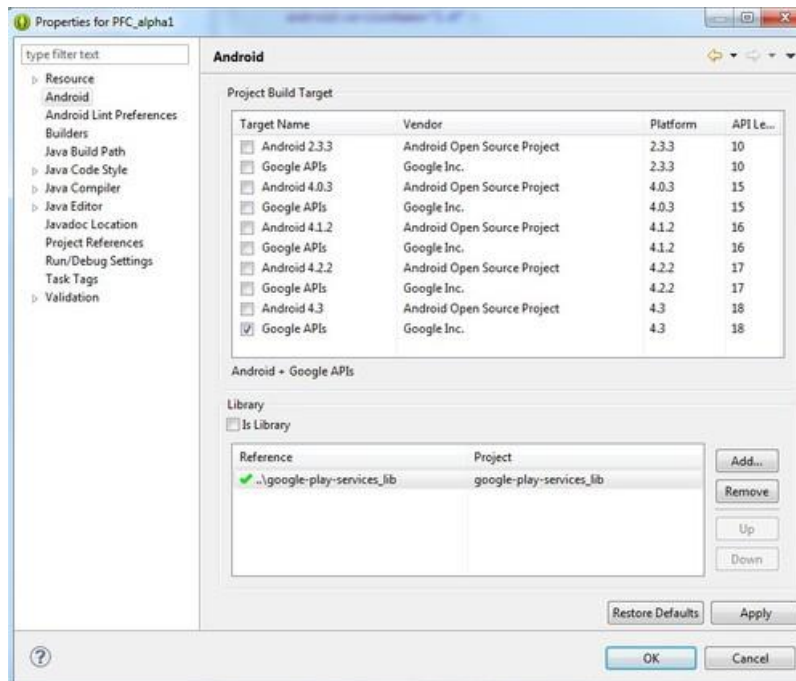
Figura 42. Import Google Play Service



Fuente: http://media.tumblr.com/9af98d17df74ddc774d16a778849b33c/tumblr_inline_mu78mxBv771qfbwj9.jpg. Consulta: 15 de septiembre de 2013.

Ahora se referencia el proyecto para que utilice esta librería. Clic derecho sobre el proyecto, opción *Properties*, después *Android* y en el apartado *Library* se añade la librería.

Figura 43. Referencia Google Play Service



Fuente: http://media.tumblr.com/25aadb1a237157672590c27687d052b9/tumblr_inline_mu78r1UZoU1qfbwj9.jpg, Consulta: 15 de septiembre de 2014.

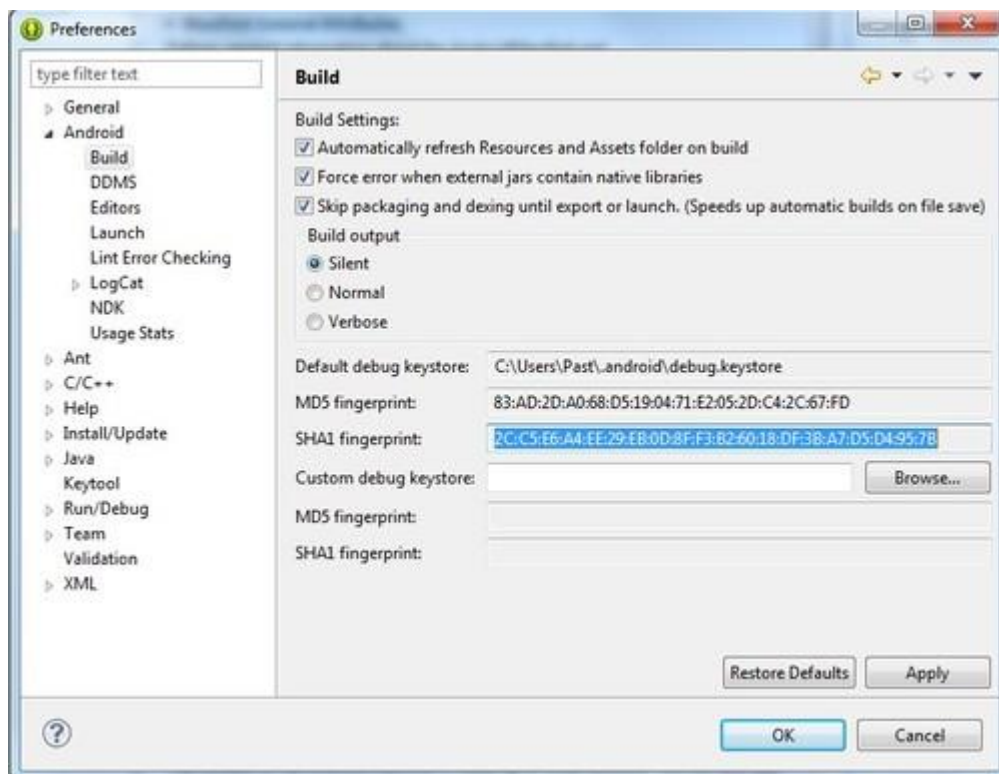
Se selecciona la opción de Google APIs. De esta manera ya se tiene descargado, instalado y configurado el proyecto para que utilice Google Play Services.

4.1.7.2. Obtener una API Key

El API Key es un identificador que genera Google para que sea utilizada única y exclusivamente por la aplicación. Para generar este identificador es necesario obtener la huella SHA-1 de la aplicación.

Para saber la huella SHA-1 de la aplicación se escoge en la barra de tareas Eclipse la opción Window, Preferences, Android y por último Build. Desplegando lo siguiente:

Figura 44. **Huella SHA-1**



Fuente: http://media.tumblr.com/c087630a542749fa0a90880188773410/tumblr_inline_mu8pvz09M1qfbwj9.jpg. Consulta: 15 de septiembre de 2014.

La huella es la parte marcada en la imagen anterior, este dato sirve para configurar la aplicación y sea pública.

El siguiente paso es configurar la aplicación en el gestor de aplicaciones de Google ubicado en el siguiente enlace:

- <https://code.google.com/apis/console/>.

Para esta consola es necesario registrarse como usuario utilizando una cuenta de correo personal de Gmail.

Ahora se va a crear una nueva aplicación en este caso la aplicación tiene el nombre de Centinela. Después de creado se va al apartado de Services y se activa las siguientes 2 API's:

Figura 45. **Google Maps Android API**



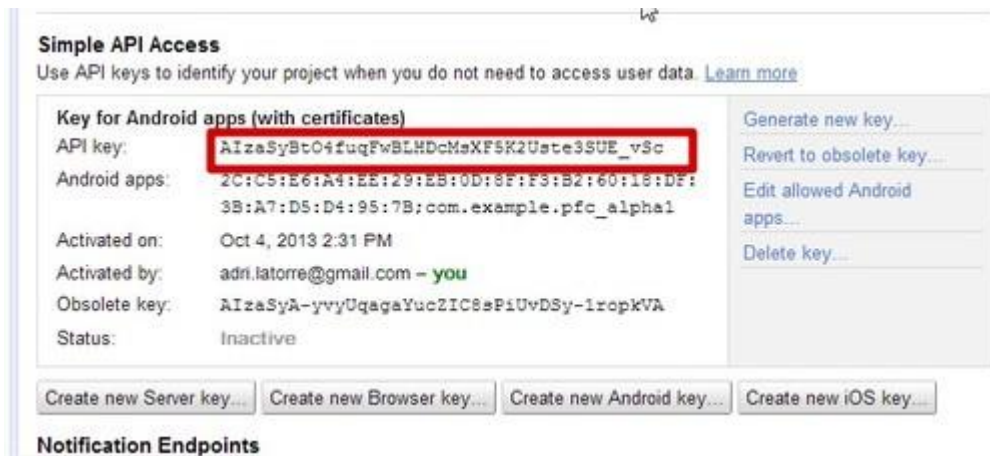
Fuente: http://media.tumblr.com/3748d7db1bf199034f3334dafd3919776/tumblr_inline_mu8q64Uh1j1qfbwj9.jpg. Consulta: 15 de septiembre de 2014.

Ahora se dirige a la opción de API Access y se presiona el botón *Create New Android Key*. En el cuadro de texto se escribe la huella SHA-1 el símbolo; y el nombre del paquete en la aplicación, para este caso:

- 2C:C5:E6:A4:EE:29:EB:0D:8F:F3:B2:60:18:DF:3B:A7:D5:D4:95:7B;com.example.pfc_alpha1

Ahora se encuentra la API Key de Google para configurarla en la aplicación.

Figura 46. **API Key**



Fuente: http://media.tumblr.com/0954186125ed992ece42f997d5164e21/tumblr_inline_mu8qcprhfM1qfbwj9.jpg. Citado 15 de septiembre de 2014.

4.1.7.3. **Configurar permisos y especificaciones en el Manifest**

El paso que corresponde es añadir a la aplicación todos los permisos, configuraciones y API Key, los permisos se agregan en el archivo Manifest.xml:

- android.permission.INTERNET: para descarga de los mapas en los servidores.
- android.permission.ACCESS_NETWORK_STATE: para la comprobación y estado de la conexión en la descarga de datos.

- com.google.android.providers.gsf.permission.READ_GSERVICES: acceso a los servicios de Google basados en la web.
- android.permission.WRITE_EXTERNAL_STORAGE: almacenamiento de datos en cache hacia la tarjeta SD o micro SD.

Adicionalmente si en la aplicación se obtendrán coordenadas de posicionamiento (GPS) se deben de agregar estos permisos:

- Android.permission.ACCESS_COARSE_LOCATION: para utilización de WiFi, datos móviles o ambos para determinar la localización.
- Android.permission.ACCESS_FINE_LOCATION: permite utilizar el *GPS* para determinar la localización con precisión.

Por último agregar el siguiente bloque de código justo antes de que finalice la etiqueta `</application>`:

Figura 47. **Metadatos mapas**

```

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

```

Fuente: elaboración propia.

Si no tienes instalado OpenGL ES v2, el mapa no aparecerá y por tanto también es recomendable añadir la siguiente condición debajo de los permisos antes mencionados:

Figura 48. **Condición OpenGL**

```
<uses-feature  
    android:glEsVersion="0x00020000"  
    android:required="true"/>
```

Fuente: elaboración propia.

4.1.7.4. Manejo de objetos de mapas para su visualización

Como primer paso se crea una vista llamada mapa.xml, agregando el fragmento para visualizar los mapas, quedando de la siguiente manera:

Figura 49. **Vista mapa**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="@drawable/fondo"  
    tools:context=".Mapa" >  
  
    <fragment  
        android:id="@+id/map"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_alignParentBottom="true"  
        android:layout_alignParentLeft="true"  
        android:layout_alignParentTop="true"  
        class="com.google.android.gms.maps.SupportMapFragment" />  
</RelativeLayout>
```

Fuente: elaboración propia.

Al agregar una vista se debe de agregar su clase o modelo el cual manejará los objetos, esta tiene el nombre de Mapa.java. Se agregaron varias

funcionalidades a la aplicación, implementando varios métodos, pero el principal para renderizar los mapas, se realiza lo siguiente:

Figura 50. **Modelo mapa**

```
package com.example.centinelav2;

import android.app.ActionBar;
import android.app.ActionBar.OnNavigationListener;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.Spinner;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.Circle;
import com.google.android.gms.maps.model.CircleOptions;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class Mapa extends FragmentActivity {
    public static LatLng inicio = new LatLng(15.740900, -90.2);
    public static LatLng dispositivo;
    public static SupportMapFragment mapFrag;
    private static GoogleMap map;
    static String Alias;
    static String Numero;
    Spinner cmbTipo;
    Button cmdActualizar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mapa);
    }
}
```

Fuente: elaboración propia.

Se realizan las importaciones de librerías, botones, listas de selección, colores, figuras, arreglos de memoria y otros. La clase debe de heredar de *FragmentActivity*. Y se crean objetos estáticos y públicos que son la latitud inicial (para la posición inicial del mapa), posición del dispositivo, mapa, alias, número de teléfono, tipo de mapa y botón actualizar.

Dentro del método *onCreate* agregar lo siguiente:

Figura 51. **Parámetros para modelo mapa**

```
mapFrag = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
map = mapFrag.getMap();

//Obtengo los parametros
Bundle bundle = getIntent().getExtras();
Alias = bundle.getString("parAlias");
Numero = bundle.getString("parNumero").toString().trim();

//Establezco el appname con el alias
this.setTitle("Ubicación");

//Habilito el boton atras
ActionBar actionBar = getActionBar();
actionBar.setDisplayHomeAsUpEnabled(true);
actionBar.setSubtitle(Numero);
```

Fuente: elaboración propia.

Esta es la configuración principal de cómo se visualizará el mapa, estableciendo el título de la pantalla ubicación, subtítulo el número de teléfono de la persona monitoreada.

Después se procede a configurar la lista de selección del tipo de mapa y las coordenadas para centrar el mapa, esta última se basa en la variable inicio,

para que el mapa se centre en el país de Guatemala, de igual manera el código se implementa siempre dentro del método *onCreate*, :

Figura 52. **Lista de selección tipo de mapa**

```
actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_LIST);
    ArrayAdapter adapter1 =
ArrayAdapter.createFromResource(getBaseContext(), R.array.mapTipos,
    android.R.layout.simple_spinner_item);
adapter1.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

    actionBar.setListNavigationCallbacks(adapter1, new
    OnNavigationListener() {
        @Override
    public boolean onNavigationItemSelected(int itemPosition, long
        itemId)
        {
            switch (itemPosition) {
                case 0:
                    map.setMapType(GoogleMap.MAP_TYPE_NORMAL);
                    break;
                case 1:
                    map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
                    break;
                case 2:
                    map.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
                    break;
                case 3:
                    map.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
                    break;
                default:
                    break;
            }
            return true;
        }
    });

    //Centro el mapa en las coordenadas
    map.moveCamera(CameraUpdateFactory.newLatLngZoom(inicio, 20));
    map.animateCamera(CameraUpdateFactory.zoomTo(7), 2000, null);
```

Fuente: elaboración propia.

Para agregar un apuntador en el mapa se generó el siguiente método, para el cual se necesita la latitud, longitud y precisión de las coordenadas.

Figura 53. **Agregar apuntador en mapa**

```
public static void AgregarMarcador(double latitud, double longitud, double
presicion){
    inicio = new LatLng(latitud, longitud);//Creo la coordenada
    map.addMarker(new MarkerOptions().position(inicio).title(Alias)
        .snippet("Ubicación: "+Alias+"("+Numero+")"
        ).icon(BitmapDescriptorFactory.fromResource(R.drawable.iconog
ps)));

    ////////////////Agrego Circulo
    CircleOptions opcionesCirculo = new CircleOptions().center(
        new LatLng(latitud, longitud)).radius(presicion);
    Circle circulo = map.addCircle(opcionesCirculo);
    circulo.setFill(Color.TRANSPARENT);
    circulo.setStrokeColor(Color.BLUE);
    circulo.setStrokeWidth(2f);
}
```

Fuente: elaboración propia.

A continuación se encuentran los métodos para los controles utilizados en el *Action Bar*, que son el botón atrás, selección en la lista de selección del tipo de mapa y el botón actualizar coordenadas.

Figura 54. **Modelo mapa**

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu items for use in the action bar
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_mapa, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            Intent intent = new Intent(Mapa.this, MenuDispositivos.class);
            intent.putExtra("parAlias", Alias);
            intent.putExtra("parNumero", Numero);
            startActivity(intent);
            finish();
            return true;
        case R.id.menRefrescar:
            EnviarMensaje.enviarSMS(getBaseContext(), Numero,
                "<centinela#ecys>gps</centinela#ecys>", "Solicitud de
                Coordenadas Enviada!!!!", "Fallo de Solicitud de
                Coordenadas");
            ManejadorBaseDatos cdb = new
            ManejadorBaseDatos(getBaseContext(),"Centinela1", null,1);
            cdb.insertarHistorial(Numero, "GPS", "Se envio solicitud de
                actualizar coordenadas");
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

Fuente: elaboración propia.

Una vez implementado todo el código se tiene interactividad con los mapas de Google, obtención de coordenadas de dispositivos sincronizados, visualización y navegación entre los diferentes tipos de vistas que provee Google Play Services, como también actualización de coordenadas (punteros), teniendo como resultado lo que se muestra a continuación:

Figura 55. Visualización de mapa



Fuente: elaboración propia.

4.2. Clientinela

Clientinela es la aplicación cliente que estará instalada en el teléfono del hijo. Su función es interpretar las solicitudes realizadas por el padre y ejecutar la acción correspondiente. Las funciones que Clientinela realiza son las siguientes:

- Enviar coordenadas de ubicación
- Mostrar alerta

- Bloqueo de pantalla
- Aceptar solicitud
- Eliminar Centinela

La aplicación está conformada por una actividad principal MainActivity.java, esta clase que hereda de la clase Activity.java, muestra en la pantalla principal la lista de los centinelas que tiene asociados; en el contexto de la aplicación se llama; Centinelas a los padres o supervisores de las actividades realizadas por los menores en su dispositivo móvil, así como las solicitudes pendientes de aceptar o rechazar.

La comunicación entre la aplicación padre Centinela y la aplicación hijo Clientinela se realiza a través de mensajes de texto. Manteniendo la arquitectura de la aplicación base. Con el cambio de que la comunicación es en 2 vías.

Figura 56. **Arquitectura Clientinela**



Fuente: elaboración propia.

La clase principal detrás de toda esta arquitectura de comunicación es `RecibirMensaje.java`, la cual hereda de la clase nativa `BroadcastReceiver.java`, en esta clase se leen todos los mensajes que recibe el teléfono y es la encargada de filtrar los mensajes que son exclusivos de la aplicación. Todo esto lo realiza sin que el usuario se dé cuenta de que recibió un mensaje de texto. Para lograr esto se establece una prioridad de lectura de mensajes de texto a la aplicación en el *Manifest* de la aplicación la cual es de: 2147483645.

Figura 57. **Declaración del Receiver**

```
<receiver android:name=".RecibirMensaje"
    android:enabled="true"
    android:exported="true">
    <intent-filter android:priority="2147483645">
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
```

Fuente: elaboración propia.

`RecibirMensaje.java` lee el mensaje recibido, si contiene la sintaxis esperada lo ejecuta, caso contrario lo deja pasar y se va directo a la bandeja de entrada de los mensajes de texto del teléfono. La sintaxis para interpretación de mensajes es la siguiente:

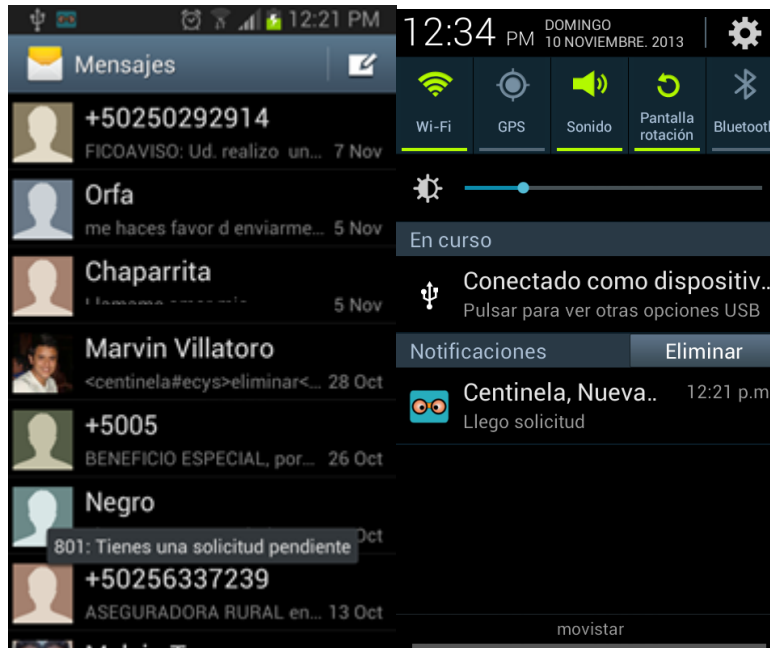
- `<centinela#ecys> [accion] </centinela#ecys>`

Las acciones que se pueden recibir son las siguientes:

4.2.1. Solicitud

Esta acción es una solicitud del padre al hijo para ser el centinela, por cuestiones de seguridad, la aplicación hijo debe aceptar quien lo va a controlar. Al recibir el mensaje de solicitud, se muestra una notificación.

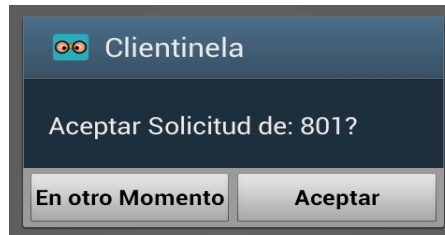
Figura 58. Notificación de nueva solicitud



Fuente: elaboración propia.

Ahora el usuario deberá abrir la notificación. Para aceptar la solicitud al crear la notificación. Esto se realiza al llamar la actividad `ActividadAceptarSolicitud.java`, esta actividad levanta un `alertDialog`, con los botones de aceptar la solicitud o hacerlo en otro momento.

Figura 59. **Mensaje de solicitud**



Fuente: Elaboración propia.

Al aceptar la solicitud se le envía un mensaje al teléfono solicitante, confirmando el vínculo entre los 2 teléfonos móviles. Luego de esto se almacena la información en la base de datos de la aplicación a través de la clase `ClientinelaDatabase.java`, mediante esta clase se realiza todo lo relacionado a la base de datos; agregar centinelas, darles de baja o actualizar la información de los mismos.

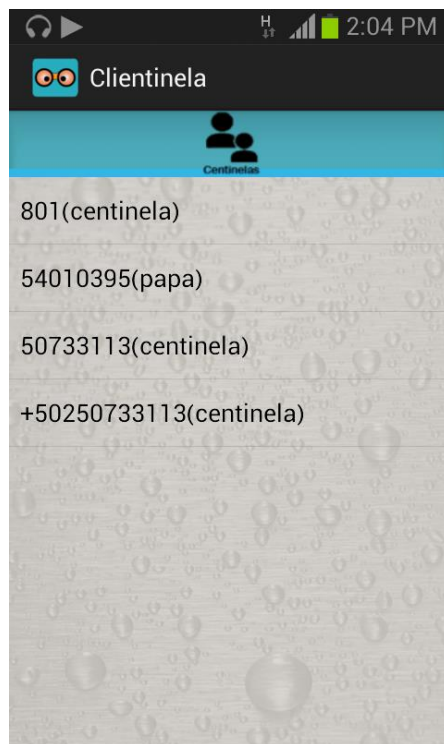
Figura 60. **Método del botón aceptar solicitud**

```
public void onClick(DialogInterface dialog, int id) {
    EnviarMensaje.enviarSMS(getBaseContext(), ntel,
        "<clientinela#ecys>si</clientinela#ecys>", "Enviado", "Error");
    ClientinelaDatabase cdb = new
    ClientinelaDatabase(getBaseContext(), "DBCliente", null, 2);
    cdb.abrir();
    cdb.insertarCentinela(ntel, "centinela", "A");
    cdb.cerrar();
    finish();
}
```

Fuente: elaboración propia.

Al finalizar se puede ir a la vista principal de la aplicación y ver que ya se tiene agregado al nuevo número en la lista de centinelas.

Figura 61. **Vista principal, lista de Centinelas**



Fuente: elaboración propia.

4.2.2. Eliminar

Clientinela realmente no puede realizar la acción de eliminar un centinela de manera arbitraria, como se explicó anteriormente solo realiza acciones demandadas por la aplicación padre, entonces para poder eliminar un centinela, este debe eliminar primero al cliente de su lista de números controlados, al hacer esto envía un mensaje con la siguiente sintaxis:

- <centinela#ecys>eliminar</centinela#ecys>

Al recibir este mensaje Clientinela ejecuta la siguiente acción:

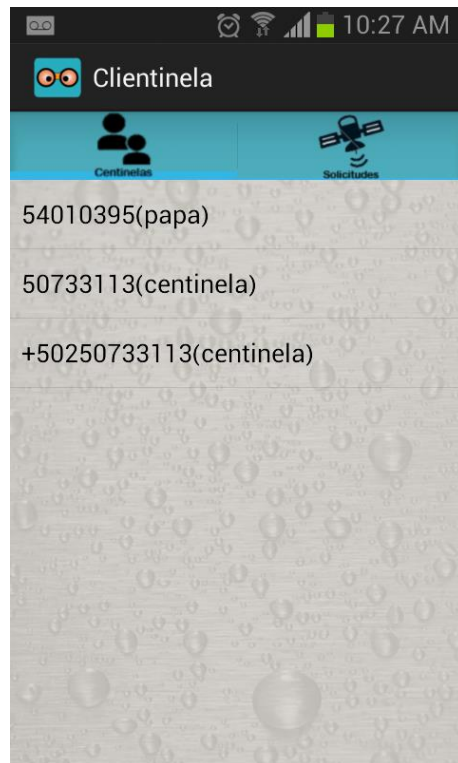
Figura 62. **Procedimiento de eliminar un Centinela**

```
if(tipomensaje.contains("eliminar")){  
  
    ClientinelaDatabase cdb = new  
    ClientinelaDatabase(context, "DBCliente", null, 2);  
  
        cdb.abrir();  
  
    cdb.eliminarCentinela(origen.replace("+502", ""));  
}
```

Fuente: elaboración propia.

Como se puede observar, se elimina directamente de la base de datos el centinela, por lo que ya no podrá ejercer ninguna de las acciones permitidas por la aplicación al cliente en cuestión. Al obtener el número origen se agrega el código del área en este caso +502, por tal razón se elimina y se guarda el número integro, por tal motivo para eliminarlo hay que realizar el mismo procedimiento.

Figura 63. Vista principal, lista de centinelas después de eliminación



Fuente: elaboración propia.

4.2.3. Alerta remota

Es una actividad muy útil en casos de emergencia. Tiene la particularidad que no importando el estado del teléfono móvil, vibrador, silencio, o simplemente en volumen bajo, al recibir una acción de alerta remota los controles de volumen se van al máximo y harán sonar la melodía predeterminada por el usuario como su tono principal del dispositivo móvil. La sintaxis para una alerta remota es la siguiente

- `<centinela#ecys> emg:{mensaje} </centinela#ecys>`

Al recibir este mensaje la clase `RecibirMensaje.java`, ejecuta la acción asociada a esta solicitud, la clase utilizada para activar la alarma remota es `AlertaRemota.java` mediante el método `PlayRing()` se obtiene la melodía y se activa el volumen máximo del teléfono.

Figura 64. Método `PlayRing`

```
public static void PlayRing(Context mContext){
    AudioManager = (AudioManager)
    mContext.getSystemService(Context.AUDIO_SERVICE);
    oldMode = AudioManager.getRingerMode();
    oldVolume = AudioManager.getStreamVolume(AudioManager.STREAM_RING);
    AudioManager.setRingerMode(AudioManager.RINGER_MODE_NORMAL);
    AudioManager.setStreamVolume(AudioManager.STREAM_RING,
    AudioManager.getStreamMaxVolume(AudioManager.STREAM_RING),
    AudioManager.FLAG_REMOVE_SOUND_AND_VIBRATE);
    String preferredRingtone =
    PreferenceManager.getDefaultSharedPreferences(mContext).
    getString("ringtone", "content://settings/system/ringtone");
    ringtone = RingtoneManager.getRingtone(mContext,
    Uri.parse(preferredRingtone));
    ringtone.play();
}
```

Fuente: elaboración propia.

Mediante el `AudioManager` se obtienen todas las propiedades del audio, y mediante el método `play()`, de la clase `Ringtone`, se inicia la ejecución del tono de llamada.

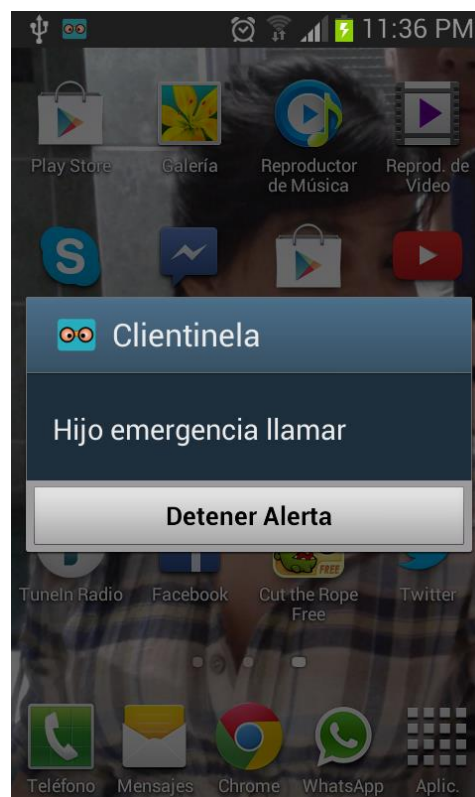
Al abrir la notificación se activará una actividad, `ActividadAlarma.java`, esta actividad muestra un diálogo con el mensaje enviado por el centinela y un botón que detiene la alarma mediante el método `AlertaReomota.stopRinging()`.

Figura 65. **Acción del botón detener alarma**

```
public void onClick(DialogInterface dialog, int id) {  
    AlertaRemota.stopRinging();  
    finish();  
}
```

Fuente: elaboración propia.

Figura 66. **Mensaje de alerta remota**



Fuente: elaboración propia.

4.2.4. Geolocalización

La localización es quizás la función más atractiva de esta aplicación. Esta es utilizada por el padre o centinela para saber la ubicación del hijo o cliente, en un momento específico. La aplicación Clientinela obtiene las coordenadas de la ubicación de donde se encuentre el teléfono en el momento de recibir el mensaje. La sintaxis del mensaje para esta acción es la siguiente:

- `<centinela#ecys>gps</centinela#ecys>`

La clase utilizada para realizar esta acción es *LocationGPS.java*, esta implementa un *LocationListener* las librerías utilizadas para realizar estas operaciones son las siguientes:

Figura 67. Librerías para localización

```
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
```

Fuente: elaboración propia.

El método utilizado para acceder a la localización es *getLocation()* en este método se obtiene las coordenadas de localización, se utilizan 2 vías de acceso a la obtención de estas coordenadas, la primera es por medio del *GPS*, que ya trae el dispositivo móvil, en caso de que esté apagado, se procede a utilizar la red de datos para obtenerlas, a diferencia entre estas 2 es la exactitud de la ubicación, en caso contrario que el dispositivo móvil no cuente con red de datos, será imposible de obtener esta información.

Lo primero que se hace es establecer un *LocationManager*. Este es el utilizado para obtener las coordenadas por medio de un proveedor, para este caso el *GPS_PROVIDER* o el *NETWORK_PROVIDER*. Ya establecido el manejador de localización, se verifica si el GPS está habilitado, en caso verdadero se procede a obtener las coordenadas, dado que es más exacto obtener por medio de GPS y entonces no se obtienen por la red de datos, cabe que hay una excepción, en caso el GPS en el momento que son requeridas las coordenadas no tiene, se procede a obtenerlas por la vía alterna de la red de datos. Se recuerda que el GPS necesita un tiempo prudente para poder obtener los datos.

Figura 68. **Métodos utilizados para localización**

```
locationManager=  
(LocationManager)mContext.getSystemService(LOCATION_SERVICE);  
  
// verificando el estado del GPS activado o no  
isGPSEnabled=  
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);  
  
// verificando el estado de la red de datos esta activada o no  
isNetworkEnabled=  
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);  
  
//Si ninguna esta activa se retona cero en las coordenadas  
if (!isGPSEnabled && !isNetworkEnabled) {  
    latitude =0;  
    longitude = 0;  
    accuracy = 0;  
}
```

Fuente: elaboración propia.

Antes de obtener las coordenadas es necesario actualizarlas, ya que se almacenan, las últimas coordenadas consultadas mediante el método `locationManager.requestLocationUpdates`, con este se actualiza la localización y luego se obtienen las coordenadas `locationManager .getLastKnownLocation`, las coordenadas se almacenan mediante dos atributos, latitud y longitud, pertenecientes a la clase *Location*.

Figura 69. **Procedimiento de obtención de coordenadas**

```
//Se actualizan las coordenadas utilizando el GPS
locationManager.requestLocationUpdates(
    locationManager.GPS_PROVIDER,
    MIN_TIME_BW_UPDATES,
    MIN_DISTANCE_CHANGE_FOR_UPDATES, this);

//Se obtienen las coordenadas
location =
locationManager.getLastKnownLocation(locationManager.GPS_PROVIDER);

// Se actualizan las coordenadas utilizando la red de datos.
locationManager.requestLocationUpdates(
    locationManager.NETWORK_PROVIDER,
    MIN_TIME_BW_UPDATES,
    MIN_DISTANCE_CHANGE_FOR_UPDATES, this);

// Se obtienen las coordenadas.
location =
locationManager.getLastKnownLocation(locationManager.NETWORK_PROVIDER);
```

Fuente: elaboración propia.

Una vez obtenida la localización, se separan los datos por latitud, longitud y exactitud de las coordenadas, ya que como se mencionó el área de ubicación es más exacta con el GPS. Finalizada la obtención de los atributos se le envía un mensaje de texto al padre o centinela para que lo visualice por medio de Google Maps, la sintaxis del mensaje enviado es la siguiente:

- `<clientinela#ecys>gps:latitud,longitud;exactitud </clientinela#ecys>`

Figura 70. **Mensaje con coordenadas**

```
//Obtener Latitud
longitud = location.getLongitude();
//Obtener Longitud
latitud = location.getLatitude();
//Obtener Exactitud
accuracy = location.getAccuracy();

coordenadas ="gps:"+ String.valueOf(latitud) + "," +
String.valueOf(longitud)+";"+String.valueOf(accuracy
);
```

Fuente: elaboración propia.

4.2.5. **Bloqueo remoto de pantalla**

Es una función muy útil para bloquear el teléfono en caso de pérdida o robo, y así evitar que alguien pueda tener acceso a la información personal, siempre y cuando este protegida con una contraseña. El padre o centinela enviará un mensaje con la sintaxis que se muestra a continuación, para pedir el bloqueo.

- `<centinela#ecys>bloquee</centinela#ecys>`

Al recibir este mensaje dentro de la clase *RecibirMensaje.java* se ejecuta la actividad *BloqueoRemoto.java*, esta actúa como un usuario administrador para bloquear la pantalla, es necesario que la aplicación tenga permisos de administrador, para que esta función pueda hacerse efectiva. En caso contrario no se podrá utilizar. Si la aplicación *Clientinela* no cuenta con los permisos, la

actividad *BloqueoRemoto.java* inicia una actividad que permita dar esos permisos, con la aceptación del usuario.

Figura 71. **Procedimiento de bloqueo**

```
//Manejador de politicas, se encarga del bloqueo y otras funciones.  
private DevicePolicyManager policyManager;  
  
//Utilizado como administrador para las funciones del PolicyManager  
private ComponentName deviceAdmin;  
  
//Se verifica que nuestro Componente este activo y tenga los permisos  
correspondientes.  
if(policyManager.isAdminActive(deviceAdmin)) {  
    policyManager.lockNow();  
    finish();  
}
```

Fuente: elaboración propia.

El método *lockNow()* de DevicePoliceManager, bloquea la pantalla.

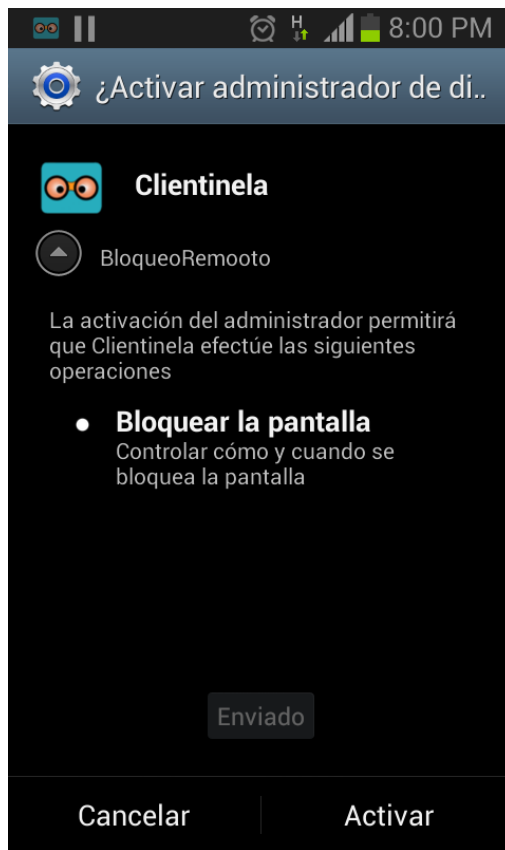
Figura 72. **Actividad para solicitar permisos de bloqueo**

```
//Sino se tiene nuestro deviceAdmin inicia la actividad que permite  
que la aplicación tenga los permisos para ejecutar el bloqueo.  
else{  
    Intent intent = new  
    Intent(DevicePolicyManager.ACTION_ADD_DEVICE_ADMIN);  
    intent.putExtra(DevicePolicyManager.EXTRA_DEVICE_ADMIN,  
        deviceAdmin);  
    intent.putExtra(DevicePolicyManager.EXTRA_ADD_EXPLANATION,  
        "BloqueoRemoto");  
    startActivityForResult(intent, REQUEST_CODE_ENABLE_ADMIN);  
}
```

Fuente: elaboración propia.

Mediante esta actividad se dan los permisos correspondientes a la aplicación para poder aplicar el bloqueo remoto.

Figura 73. **Pantalla de solicitud de permisos**



Fuente: elaboración propia.

CONCLUSIONES

1. El control parental a través de los dispositivos móviles es el menos utilizado y conocido en Guatemala, pero es de gran utilidad debido a la portabilidad y facilidad de uso de estos, con base en la necesidad de seguridad en la actualidad.
2. Las funciones de localización y mensaje de emergencia fueron las preferidas por las personas entrevistadas.
3. La localización solo es posible realizarse si se cuenta con acceso a la red móvil de datos.
4. Se pueden obtener las coordenadas por medio del GPS, siendo esta la más precisa, pero si en un caso estuviese apagado el servicio de GPS se obtienen las coordenadas por medio de la red del proveedor de servicio móvil, pero a diferencia del otro, este tiene un margen de error más grande.

RECOMENDACIONES

1. Investigar el mercado de IOS y Windows Phone para estudiar la factibilidad de implementar este tipo de aplicaciones en otras plataformas.
2. Indagar otras formas de obtener las coordenadas de ubicación en el caso de que no se cuenta con red de datos móviles.
3. Algo importante de mencionar es que, dependiendo la ubicación donde se desee implementar este tipo de tecnología, se debe investigar sobre las leyes de privacidad, ya que estas varían en cada lugar o región, y la aplicación podría estar violando alguna ley en la cual el individuo pierda su libertad y causar demandas hacia las personas que son monitoreadas como a la empresa dueña de la aplicación.
4. Se debe definir el mercado objetivo de la aplicación, ya que ciertas versiones de Sistema Operativo Android y modelos de teléfonos se encuentran discontinuados, y se menciona esto porque hay ciertas funcionalidades que no se pueden implementar por querer abarcar que la aplicación sea compatible con todas las versiones.
5. Desarrollar la unificación de la aplicación servidor con la aplicación cliente, garantizando escalabilidad en la aplicación.
6. Implementar el mensaje de emergencia en las dos vías, es decir de la aplicación del padre hacia la del hijo y a la inversa.

BIBLIOGRAFÍA

1. Android Open Source Project. *Notifications*. [en línea]. Actualizado 22 de septiembre de 2014. [ref. 02 de octubre 2013]. Disponible en web: <<https://developer.android.com/guide/topics/ui/notifiers/notifications.html>>
2. AZZOLA, Francesco. *Android ListView Tutorial and basic example*. [en línea]. Actualizado junio 13 del 2013. [ref. 06 de septiembre 2013]. Disponible en web: <<http://www.javacodegeeks.com/2013/06/android-listview-tutorial-and-basic-example.html>>
3. Edu 4 Android. *Utilizar la Barra de Acción*. [ref. 04 de octubre 2013]. Disponible en web: <<http://www.edu4android.com/es/guide/topics/ui/actionbar.html>>
4. GÓMEZ, Salvador. *Bases de Datos en Android (I): Primeros pasos*. [en línea]. Actualizado 31 de enero de 2011. [ref. 04 de octubre 2013]. Disponible en web: <<http://www.sgoliver.net/blog/?p=1611>>
5. _____. *Interfaz de usuario en Android: Pestañas Tabs*. [en línea]. Actualizado 10 de septiembre de 2011. [ref. 02 de octubre 2013]. Disponible en web: <<http://www.sgoliver.net/blog/?p=2112>>

6. GONZÁLEZ, Hugo. *Construyendo una splash screen en Android*. [en línea]. Actualizado 09 de enero de 2013. [ref. 05 octubre 2013]. Disponible en web:
<<http://androidsensei.net/construyendo-una-splashscreen-en-android/>>
7. _____. *Enviar, recibir y procesar SMS en la aplicación android, si así como hacen en WhatsApp y Line*. [en línea]. Actualizado 10 de febrero de 2013. [ref. 04 de octubre 2013]. Disponible en web:
<<http://androidsensei.net/Enviar-recibir-y-procesar-sms-en-nuestra-aplicacion-android-si-asi-como-hacen-en-whatsapp-y-line/>>
8. Internet World Stats. *Guatemala internet usage, broadband and telecommunications reports*. [en línea]. Actualizado 15 de marzo 2014. [ref. 05 de septiembre 2013]. Disponible en web:
<<http://www.internetworldstats.com/am/gt.htm>>
9. LANCASTER, Henry. *Guatemala - Telecoms, IP Networks and Digital Media*. [en línea]. Actualización 17 abril 2014 [ref. 08 de septiembre 2013]. Disponible en web:
<<https://www.budde.com.au/Research/Guatemala-Telecoms-Mobile-and-Broadband.html>>
10. LARA, Antonio Luis. *Android TableLayout*. [en línea]. Actualizado 12 de septiembre de 2012. [ref. 02 de octubre 2013]. Disponible en web:
< <http://blog.saducelabs.com/android/androidi-tablelayout/> >

11. RODRÍGUEZ, Aurora. *Creando notificaciones en la Status Bar de Android*. [en línea]. Actualizado 27 de septiembre de 2011. [ref. 04 de octubre 2013]. Disponible en web: <<http://androideity.com/2011/09/27/creando-notificaciones-en-la-status-bar/>>
12. VOGUEL, Lars. *Android Location API tutorial*. [en línea]. Actualizado 20 de agosto de 2012. [ref. 10 octubre 2013]. Disponible en web: <<http://www.vogella.com/articles/AndroidLocationAPI/article.html>>
13. _____. *Android SQLite database and content provider - tutorial*. [en línea]. Actualizado 08 de agosto de 2013. [ref. 05 octubre 2013]. Disponible en web: <<http://www.vogella.com/articles/AndroidSQLite/article.html>>

APÉNDICES

Se realizó una encuesta a través del servicio de encuestas gratis de *Google Drive*, teniendo una muestra de 62 encuestados, el enlace temporal para la encuesta es:

- https://docs.google.com/forms/d/1v_y8gPcKGWwNvdJD_rpLI5uJBSWc49IXAwCDyesfHA0/viewform

Los resultados se describen en el capítulo 3, obteniendo resultados claros acerca de la tecnología Android y Control Parental.

1. Edad *

¿En qué rango de edad se encuentra?

Mark only one oval.

- 10 - 15 años
- 16 - 20 años
- 21 - 25 años
- 26 - 30 años
- 31 - 35 años
- 36 - 40 años
- 41 - 45 años
- 46 - 50 años
- Mas de 50 años

2. Sexo *

¿Cuál es su sexo?

Mark only one oval.

- Masculino
- Femenino

3. Servicio Telefónico *

¿Utiliza teléfono celular en la actualidad? Si su respuesta es "SI" continuar con las demás preguntas, de lo contrario Ir a pregunta No. 6

Mark only one oval.

- SI
- No

4. Tipo de Celular

¿Qué tipo de celular utiliza actualmente?

Mark only one oval.

- Celular Convencional
- Teléfono Inteligente
- Tableta + celular

5. Marca *

¿Cuál es la marca de su celular?

Mark only one oval.

- Samsung
- Iphone
- Lg
- Sony
- Nokia
- Alcatel
- Huawei
- Otro

6. Preferencia de Marca *

¿Si tuviera que comprar celular, que marca sería la de su elección?

Mark only one oval.

- Samsung
- Iphone
- Lg
- Sony
- Nokia
- Alcatel
- Huawei
- Otro

7. Sistema Operativo Móvil *

¿Qué Sistema Operativo para dispositivos móviles prefiere?

Mark only one oval.

- Android
- IOS (Iphone)
- WindowsPhone
- Symbian
- Otro

8. Versión Android

Si su teléfono utiliza Sistema Operativo Android, ¿Con qué versión cuenta? Para ver la versión en su dispositivo ir a: Menú de aplicaciones -> Ajustes -> Acerca del teléfono (Por lo general es la última opción) -> Versión de Android. De lo contrario continuar con la siguiente pregunta.

Mark only one oval.

- Versión 4.4: KitKat
- Versión 4.1 - 4.3: JellyBean
- Versión 4.0 - 4.0.4: Ice Cream Sandwich
- Versión 3.0.x - 3.1.x: Honey Comb
- Versión 2.3.1 - 2.3.4: Ginger Bread
- Versión 2.2.x: Froyo
- Versión 2.1.x: Eclair
- Versión 1.6: Donut
- Versión 1.5: Cupcake
- Versión 1.1: Banana Bread
- Versión 1.0: Apple Pie

9. Control parental *

El control parental consiste en impedir o limitar el acceso al manejo de dispositivos a menores de edad. Esto se realiza mediante una serie de sistemas de bloqueo, normalmente protegidos mediante claves, que configuran normalmente sus padres, para restringir el uso de la correspondiente máquina. En el acceso de la información en Internet es una de las cuestiones de control paterno más discutidas y usadas ya que muchos de los contenidos de la red no son adecuados para los menores de edad. Los programas de navegación suelen incorporar potentes páginas de configuración para proceder a la realización de este control de contenidos. ¿Creó usted que el control parental es necesario por la vulnerabilidad de los niños en la actualidad?

Mark only one oval.

- Sí
- No

10. Tipos de Control Parental *

¿Qué tipos de control parental a escuchado o utilizado?

Check all that apply.

- Filtro a sitios web
- Seguridad por Antivirus
- Control Parental de Sistema Operativo (por permisos de usuario)
- Control parental en dispositivos móviles
- Otro
- Ninguno

11. Funcionalidad 1: Ubicación por GPS *

¿Cree que es útil una aplicación en su teléfono Inteligente, para obtener la ubicación de sus personas allegadas en cualquier momento?

Mark only one oval.

- Sí
- No

12. Funcionalidad 2: Bloqueo remoto *

¿Utilizaría usted una aplicación, en la cual pueda bloquear el teléfono de sus personas allegadas, en caso de robo o pérdida?

Mark only one oval.

- Sí
- No

13. Funcionalidad 3: Mensaje de emergencia *

¿Le sería útil una funcionalidad en su teléfono inteligente, en la cual le pueda enviar mensajes a sus personas allegadas y estos dispositivos suenen y vibren aunque estén en estado de Vibrador o Silencio? Esto con el fin de enviar mensajes de emergencia

Mark only one oval.

- Sí
- No