



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

**IMPLEMENTACIÓN DE TECNOLOGÍAS WEB Y MÓVILES PARA LA APLICACIÓN DE REALIDAD
AUMENTADA Y GEOLOCALIZACIÓN DE LUGARES TURÍSTICOS EN GUATEMALA**

Erick Wilfredo Díaz Saborío
Emanuel Alejandro Cifuentes Carrillo
Asesorado por el Ing. Edgar Estuardo Santos Sutuj

Guatemala, septiembre de 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**IMPLEMENTACIÓN DE TECNOLOGÍAS WEB Y MÓVILES PARA LA APLICACIÓN DE REALIDAD
AUMENTADA Y GEOLOCALIZACIÓN DE LUGARES TURÍSTICOS EN GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

ERICK WILFREDO DÍAZ SABORÍO

EMANUEL ALEJANDRO CIFUENTES CARRILLO

ASESORADO POR EL ING. EDGAR ESTUARDO SANTOS SUTUJ

AL CONFERÍRSELES EL TÍTULO DE

INGENIERIOS EN CIENCIAS Y SISTEMAS

GUATEMALA, SEPTIEMBRE DE 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Ludwing Federico Altán Sac
EXAMINADOR	Ing. Edgar Estuardo Santo Sutuj
EXAMINADOR	Ing. José Ricardo Morales Prado
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

Erick Wilfredo Díaz Saborío

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
EXAMINADOR	Ing. Marlon Francisco Orellana López
SECRETARIA	Inga. Lesbia Magalí Herrera López

Emanuel Alejandro Cifuentes Carrillo

HONORABLE TRIBUNAL EXAMINADOR

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

**IMPLEMENTACIÓN DE TECNOLOGÍAS WEB Y MÓVILES PARA LA APLICACIÓN DE REALIDAD
AUMENTADA Y GEOLOCALIZACIÓN DE LUGARES TURÍSTICOS EN GUATEMALA**

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



Emanuel Alejandro Cifuentes Carrillo



Erick Wilfredo Díaz Saborío

Guatemala, 06 de abril de 2015.

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 "IMPLEMENTACION DE TECNOLOGIAS WEB Y MÓVILES PARA LA APLICACIÓN DE REALIDAD AUMENTADA Y GEOLOCALIZACION DE LUGARES TURISTICOS EN GUATEMALA", desarrollado por el estudiante Emanuel Alejandro Cifuentes Carrillo con número de carné 200714604 y el estudiante Erick Wilfredo Díaz Saborío con número de carné 200818835, 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, 29 de Abril del 2015

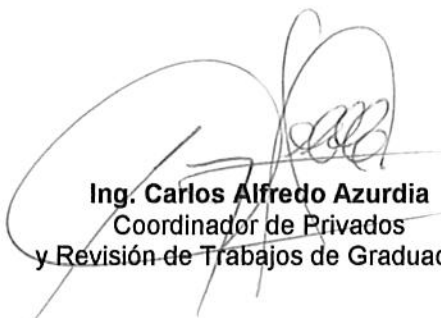
Ingeniero
Marlon Antonio Pérez Türk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **EMANUEL ALEJANDRO CIFUENTES CARRILLO** con carné 2007-14604, y **ERICK WILFREDO DIAZ SABORIO** con carné 2008-18835, titulado: **"IMPLEMENTACIÓN DE TECNOLOGÍAS WEB Y MÓVILES PARA LA APLICACIÓN DE REALIDAD AUMENTADA Y GEOLOCALIZACIÓN DE LUGARES TURÍSTICOS EN GUATEMALA"**, 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 **"IMPLEMENTACIÓN DE TECNOLOGIAS WEB Y MÓVILES PARA LA APLICACIÓN DE REALIDAD AUMENTADA Y GEOLOCALIZACIÓN DE LUGARES TURÍSTICOS EN GUATEMALA"**, realizado por el estudiante **ERICK WILFREDO DIAZ SABORIO**, 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, 03 de Septiembre de 2015



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **IMPLEMENTACIÓN DEL PROGRAMA DE TECNOLOGÍAS WEB Y MÓVILES PARA LA APLICACIÓN DE REALIDAD AUMENTADA Y GEOLOCALIZACIÓN DE LUGARES TURÍSTICOS EN GUATEMALA**, presentado por los estudiantes universitarios: **Erick Wilfredo Díaz Saborio y Emanuel Alejandro Cifuentes Carrillo**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Pedro Antonio Aguilar Polanco
Decano



Guatemala, septiembre de 2015

/cc

ACTO QUE DEDICO A:

Dios

Por ser una importante influencia en mi carrera.

Mis padres

Wilfredo Díaz Lima y Flor María Saborío Mora, gracias por su apoyo incondicional, por todas sus sabias enseñanzas he logrado conseguir mis metas.

Mi hermano

Gonzalo Díaz, por todas las experiencias y apoyo a lo largo de nuestras vidas.

Erick Wilfredo Díaz Saborío

ACTO QUE DEDICO A:

Mis padres

Luis Cifuentes y Arlette Carrillo, por ser una constante motivación a ser la mejor persona que puedo ser.

Emanuel Alejandro Cifuentes Carrillo

AGRADECIMIENTOS A:

**Universidad de San
Carlos de Guatemala**

Por convertirse en mi segundo hogar y por tener la oportunidad de formarme como un profesional al servicio de la comunidad.

Facultad de Ingeniería

Por brindarme las herramientas y el conocimiento de esta profesión.

**Mis amigos de
la Facultad**

Por todas las experiencias aprendidas tanto de vida como de la profesión, por los éxitos, los desvelos, por el apoyo a lo largo de la carrera.

**Mi compañero en el
trabajo de graduación**

Emanuel Cifuentes, por el esfuerzo y dedicación para concluir este proyecto.

Mis padres

Wilfredo Díaz y Flor María Saborío, por todo el apoyo que han dado, por todas sus enseñanzas.

Erick Wilfredo Díaz Saborío

AGRADECIMIENTOS A:

**Universidad de San
Carlos de Guatemala**

Por mantenerse firme en sus ideales y transmitirlos a todos los estudiantes, que son el futuro de esta nación.

Facultad de Ingeniería

Por exigir siempre mi mayor esfuerzo y no dejarme caer en la mediocridad, manteniendo el nivel del que me trae orgullo ser partícipe.

Mis amigos de la Facultad

Por hacer de mi tiempo en la universidad una experiencia agradable y enriquecedora, y compartirme su conocimiento.

Mis padres

Luis Cifuentes y Arlette Carrillo, por ser esas personas excelentes y admirables que me trajeron hasta este momento.

Emanuel Alejandro Cifuentes Carrillo

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
GLOSARIO	XI
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO EN GUATEMALA.....	1
1.1. Impacto de la tecnología móvil en Guatemala.....	1
1.1.1. Sistema operativo Android.....	3
1.1.2. Historia de Android	3
1.1.3. Arquitectura de Android	4
1.1.3.1. Núcleo Linux	4
1.2. Realidad Aumentada	5
1.2.1. Motor de Realidad Aumentada	5
1.2.1.1. Motor de Realidad Aumentada Mixare.....	6
1.2.1.1.1. Arquitectura de Mixare.....	6
1.3. Geolocalización	6
1.4. Teorías que respaldan el trabajo de graduación	7
1.4.1. Objetos fronterizos.....	8
1.4.2. Riqueza de los medios	10
1.4.3. Capital social	13
2. IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN PLANTEADA	15
2.1. Antecedentes.....	15

2.2.	Descripción del problema.....	15
2.3.	Benchmark de la aplicación.....	18
2.4.	Waze	18
2.5.	Junaio.....	20
3.	DOCUMENTACIÓN BASE PARA EL DESARROLLO DE LA APLICACIÓN	21
3.1.	Documentación base para el desarrollo del sitio web	21
3.1.1.	Tecnologías utilizadas.....	21
3.1.1.1.	PHP	21
3.1.1.2.	Zend Framework 2.....	22
3.1.1.3.	Google Static Maps API.....	23
3.1.1.4.	Bootstrap	23
3.1.1.5.	MySQL.....	23
3.1.2.	Dependencias	23
3.1.2.1.	Servidor web.....	24
3.1.2.2.	PHP	24
3.1.2.3.	MySQL.....	25
3.1.3.	Arquitectura de la aplicación web.....	25
3.1.3.1.	Modelo.....	25
3.1.3.2.	Vista.....	26
3.1.3.3.	Controlador.....	26
3.1.4.	Construcción de una aplicación utilizando Zend Framework	26
3.1.4.1.	Aplicación esqueleto.....	26
3.1.4.2.	Obteniendo el Zend Framework	28
3.1.4.3.	Configuración de Apache	28
3.1.4.4.	Estructura de directorios.....	31

	3.1.4.4.1.	Configuración del módulo.....	33
	3.1.4.4.2.	Rutas	34
	3.1.4.4.3.	Carga e inicialización del módulo	36
	3.1.4.4.4.	Registrando el módulo..	37
	3.1.4.4.5.	Modelo para los puntos.....	38
	3.1.4.4.6.	Creando el controlador.....	40
	3.1.4.4.7.	Función coordDistance.....	41
	3.1.4.4.8.	Función addPointAction	41
3.2.		Documentación base para el desarrollo de la aplicación móvil .	42
3.2.1.		Requerimientos para el desarrollo	43
	3.2.1.1.	Android SDK	43
	3.2.1.2.	Android Development Tools (ADT)	44
	3.2.1.3.	Instalación y configuración.....	44
	3.2.1.3.1.	Descarga e instalación del Android SDK.....	44
	3.2.1.3.2.	Android SDK Manager..	45
	3.2.1.3.3.	Crear un nuevo proyecto en Eclipse	45
	3.2.1.4.	Ambiente de real pruebas	46
3.2.2.		Desarrollo de aplicaciones para Android	47
	3.2.2.1.	Conceptos generales	47
	3.2.2.1.1.	Actividad	48

3.2.2.1.2.	Servicios.....	48
3.2.2.1.3.	Intención (<i>Intent</i>)	48
3.2.2.1.4.	Receptor de transmisión (<i>Broadcast Receiver</i>)	48
3.2.2.1.5.	Proveedor de contenidos (<i>Content Provider</i>).....	49
3.2.2.1.6.	Android Manifest	49
3.2.2.1.7.	Ciclo de vida de los componentes.....	49
3.2.3.	Desarrollo de la aplicación como solución al trabajo de investigación.....	50
3.2.3.1.	Arquitectura	50
3.2.3.2.	Componentes de la aplicación.....	52
3.2.3.2.1.	Integración con Mixare	52
3.2.3.2.2.	Integrando Mixare en Eclipse.....	52
3.2.3.2.3.	Actividades.....	55
3.2.3.2.4.	Permisos en el Android Manifest	55
3.2.3.2.5.	Linear <i>layout</i>	56
3.2.3.2.6.	<i>Text View</i>	56
3.2.3.2.7.	<i>Edit Text</i>	56
3.2.3.2.8.	<i>Button</i>	57
3.2.3.2.9.	Alert Dialog.....	57
3.2.3.2.10.	Progress Dialog.....	58
3.2.3.2.11.	JSON Parser	59

	3.2.3.2.12. Strings	59
	3.2.3.3. Integración con el API de elevación de Google	60
	3.2.3.4. Integración con Google Maps v2	61
	3.2.3.5. Integración con el API de rutas de Google	63
4.	MANUAL DE USO DE LA APLICACIÓN.....	65
4.1.	Aplicación Web	65
4.1.1.	Creando usuarios	65
4.1.2.	Autenticando los usuarios.....	67
4.1.3.	Puntos del usuario	68
4.1.4.	Agregando puntos	70
4.1.5.	Consultando puntos públicos.....	71
4.2.	Aplicación móvil.....	71
4.2.1.	Autenticando usuarios	72
4.2.2.	Creación de usuarios.....	72
4.2.3.	Agregar puntos	73
4.2.4.	Modo Realidad Aumentada	74
4.2.5.	Menú.....	75
4.2.6.	Mapa.....	76
	CONCLUSIONES	79
	RECOMENDACIONES.....	81
	BIBLIOGRAFÍA.....	83
	APÉNDICES	85

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Resultados de la encuesta, plataformas móviles	2
2.	Arquitectura Android.....	4
3.	Funcionamiento de Mixare	6
4.	Línea de tiempo en la que una aplicación obtiene la localización	7
5.	Porcentaje de personas que utilizan aplicaciones para localizar sitios de interés.....	16
6.	Porcentaje de personas interesadas en visualizar lugares turísticos con una aplicación de realidad aumentada	17
7.	Porcentaje de personas interesadas en visualizar y agregar lugares turísticos con una aplicación móvil	17
8.	Crecimiento de la telefonía fija y móvil del 2004 al primer semestre 2011	18
9.	Github.....	27
10.	Archivo httpd.conf 1.....	29
11.	Archivo httpd-conf 2.....	29
12.	Archivo httpd.conf 3.....	30
13.	Archivo hosts	30
14.	Zend Framework 2	31
15.	Estructura de directorios.....	31
16.	Archivo module.config.php I	33
17.	Archivo module.config.php II	34
18.	Rutas en archivo module.config.php	35
19.	Archivo Module.php.....	36

20.	Archivo aplicacion.config.php	37
21.	Archivo Point.php	38
22.	Funciones en el archivo Point.php	39
23.	Archivo SiteController.php	40
24.	Función coordDistance en archivo SiteController.php	41
25.	Función addPointAction en archivo SiteController.php	42
26.	Android SDK Manager	45
27.	New Android Application	46
28.	Android Device Chooser	47
29.	Ciclo de vida de una Actividad	50
30.	Arquitectura del sistema.....	51
31.	Funcionalidad Mixare	52
32.	Propiedades del proyecto 1	53
33.	Propiedades del proyecto 2	54
34.	Permisos en el archivo Android Manifest	56
35.	Layout	58
36.	Resultados en formato JSON el API	61
37.	Archivo XML de configuración de la interfaz web	62
38.	Creación de usuarios	66
39.	Error al crear usuarios.....	66
40.	Autenticación de usuarios	67
41.	Error al iniciar sesión.....	68
42.	Mis puntos.....	69
43.	Punto sobre el mapa.....	69
44.	Agregar punto	70
45.	Puntos públicos.....	71
46.	Autenticación de usuario	72
47.	Registro de usuarios	73
48.	Agregar puntos	74

49.	Modo Realidad Aumentada	75
50.	Menú de usuario.....	76
51.	Mapa	77

GLOSARIO

API	Conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
Dalvik	Máquina virtual que utiliza la plataforma para dispositivos móviles Android.
Geolocalización	Técnica de posicionamiento espacial de una entidad en una localización geográfica única y bien definida en un sistema de coordenadas y datum específicos.
Github	Es una forja para alojar proyectos utilizando el sistema de control de versiones Git.
JavaScript	Lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.
JSON	Acrónimo de JavaScript Object Notation. Es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Kernel	Software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado. Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora.
Linux	Combinación del núcleo o Kernel libre similar a Unix denominado Linux con el sistema GNU.
Máquina virtual	Software que simula a una computadora y puede ejecutar programas como si fuese una computadora real.
SDK	Conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto.
Parser	Programa informático capaz de analizar y ejecutar otros programas. Se diferencia de los compiladores o de los ensambladores porque traducen un programa desde su descripción en un lenguaje de programación al código de máquina del sistema. El intérprete solo realiza la traducción a medida que sea necesaria, típicamente, instrucción por instrucción, y normalmente no guardan el resultado de dicha traducción.

Realidad aumentada Se define una visión a través de un dispositivo tecnológico, directa o indirecta, de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real.

RESUMEN

A medida que la tecnología avanza se vuelve cada vez más barata y accesible. Los dispositivos se producen en masa y su costo y tamaño ha disminuido grandemente en los últimos 20 años. La infraestructura que soporta los servicios que los aparatos acceden se extiende cada vez más, alcanzando las zonas más remotas del planeta. La cantidad de personas que tienen acceso a las nuevas tecnologías aumenta exponencialmente.

La Universidad de San Carlos de Guatemala, como única universidad estatal, pretende poner a disposición de todas las personas estas nuevas tecnologías, buscando el progreso a través del fácil acceso a la información. De parte de la Facultad de Ingeniería se creó el grupo de Nuevas Tecnologías en un intento de promover la innovación y distribución del conocimiento, específicamente en la Escuela de Ciencias y Sistemas. Se busca explotar el rápido crecimiento de las ciencias del área de ingeniería para solucionar problemas comunes a un grupo o todos los estudiantes, y población en general.

La siguiente es una guía técnica para aplicar los conceptos de geolocalización y realidad aumentada en un programa para dispositivos móviles. El documento se divide en cinco capítulos. El primer capítulo es una descripción de los conceptos que se estarán tratando y la razón para incluirlos en la solución. El segundo capítulo identifica el problema y presenta una solución a este. En el tercer capítulo es una introducción a las tecnologías que se utilizarán en el desarrollo del presente trabajo de graduación. El penúltimo capítulo describe de manera general el procedimiento que permitió crear la

solución de software que resuelve el problema. Por último, se describe el uso corriente de la aplicación, las opciones y servicios que provee.

OBJETIVOS

General

Proveer una solución de software diseñada para dispositivos móviles que permita a los usuarios compartir información acerca de su ubicación, definir puntos importantes en cualquier parte del mundo y utilizar su dispositivo para consultar los puntos de interés de otros usuarios para obtener información relevante, basada en su posición sobre el globo terráqueo.

Específicos

1. Generar un documento que sea de utilidad a otros estudiantes para aplicar los conceptos de realidad aumentada y geolocalización en una aplicación para dispositivos móviles que utilizan el sistema operativo Android.
2. Presentar una guía para el desarrollo de una aplicación utilizando el Zend Framework, que además de consultar y agregar puntos provea el medio de almacenamiento de varios que necesita la aplicación móvil.

INTRODUCCIÓN

Desde los inicios de la era digital la tecnología ha progresado a pasos agigantados. Cada año las computadoras duplican su capacidad de procesamiento. Los dispositivos móviles son cada vez más pequeños, livianos y baratos. Conceptos como el internet y redes sociales se van involucrando en más áreas de la ciencia y las relaciones humanas. El software crece para proveer más funcionalidad y mejores servicios, para ser más fácil de usar. Con la globalización es más sencillo compartir y obtener información de personas en todo el mundo en segundos.

Uno de los campos en donde se puede obtener gran provecho del progreso de la tecnología es en la educación. Los países más desarrollados han sabido implementar la infraestructura y servicios que les permitan a sus habitantes el fácil acceso a la información. En Guatemala ha sido lenta la adopción de estas prácticas y es fundamental el acelerar el proceso para explotar al máximo el crecimiento de la tecnología y beneficiar a la mayor cantidad de personas posible.

En el presente trabajo de graduación se relacionan dos conceptos fundamentales: geolocalización y Realidad Aumentada. Se explica la relación entre los dos conceptos y la forma de aplicarlos en un programa para dispositivos móviles. El documento plantea un problema, la solución propuesta y la forma de construir la solución. Con este trabajo se busca instruir a un público general en el uso de las tecnologías antes mencionadas y mostrar cómo se pueden aplicar para enriquecer la experiencia de un usuario de dispositivos móviles.

1. ESTUDIO DE LA TECNOLOGÍA Y SU IMPACTO EN GUATEMALA

Es este capítulo se hace un análisis del impacto y el crecimiento de la tecnología móvil en Guatemala, específicamente en los dispositivos móviles. Uno de ellos el sistema operativo Android y el impacto que tiene la sociedad guatemalteca.

Se presenta un estudio de las teorías utilizadas en la investigación de sistemas de información y cómo estas se adaptan a la estructura de investigación y la aplicación de campo.

1.1. Impacto de la tecnología móvil en Guatemala

Los dispositivos móviles inteligentes gracias a sus múltiples capacidades tanto de hardware como software, permiten un sin fin de aplicaciones prácticas a los usuarios de los dispositivos, pues permiten una buena capacidad de procesamiento y acceso a internet. Lo anterior ha provocado un incremento en la venta de estos dispositivos y han abierto el mercado de aplicaciones. Las mismas satisfacen diversas necesidades que han ido surgiendo a medida que han evolucionado los teléfonos inteligentes. Esta constante innovación y gran demanda de tecnologías ha provocado un gran avance y que la tendencia sea a las empresas que busquen ofrecer sus servicios en aplicaciones para teléfonos inteligentes.

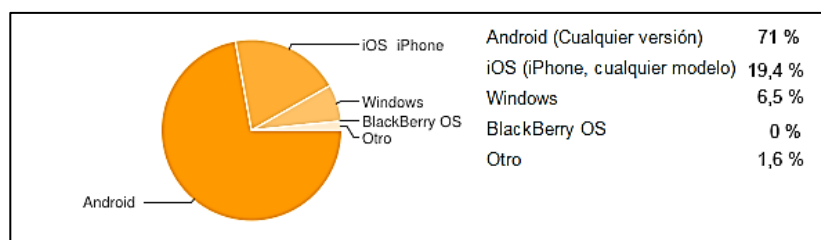
Existe una investigación llamada Invasión Mobile 2014 Latinoamérica publicada por GuiaLocal, una de las guías más grandes de comercios en

Iberoamérica. Este estudio demuestra que en Latinoamérica el uso de dispositivos móviles creció en un 61 % en el primer semestre del 2014. En este estudio se hizo un *ranking* de los 10 países con mayor crecimiento en el uso de estos dispositivos. Guatemala ocupa la tercera posición con un incremento del 108 % en celulares esto es en comparación con el primer semestre del 2013.

Otro estudio de GuiaLocal demuestra que el sistema operativo para dispositivos móviles más utilizado en la región de Latinoamérica es Android. El porcentaje, en el mercado de Guatemala, para los dispositivos móviles con Android es del 60,42 % este estudio fue realizado en el 2013. El análisis de los datos de este estudio se realizó en los países en los que GuiaLocal ha tenido presencia durante todo el 2013.

Se encuestaron personas que cuentan con dispositivos inteligentes con el objetivo de analizar una muestra de la población y determinar tanto la aceptación del proyecto como el sistema operativo para dispositivos móviles más popular. Los resultados que se obtuvieron concuerdan con los resultados del estudio de GuiaLocal. Del grupo de personas que respondieron la encuesta el 71 % utiliza dispositivos móviles con sistema operativo Android siendo este al más popular.

Figura 1. **Resultados de la encuesta, plataformas móviles**



Fuente: elaboración propia, con programa Google Forms.

1.1.1. Sistema operativo Android

Android es una de las plataformas de más rápido crecimiento, una de las más utilizadas en Latinoamérica. Android es un sistema operativo basado en Linux, depende de este para los servicios base del sistema. Junto con el núcleo Linux esta Dalvik una máquina virtual de Java que fue creada para ser utilizada en estos dispositivos, ya que no soportarían la máquina virtual estándar de Java, esta se ejecuta sobre el núcleo del sistema. Este sistema operativo está basado principalmente para dispositivos con pantalla táctil.

Android provee todas las herramientas necesarias para desarrollar, una plataforma para crear aplicaciones y un Marketplace para distribuir las aplicaciones.

1.1.2. Historia de Android

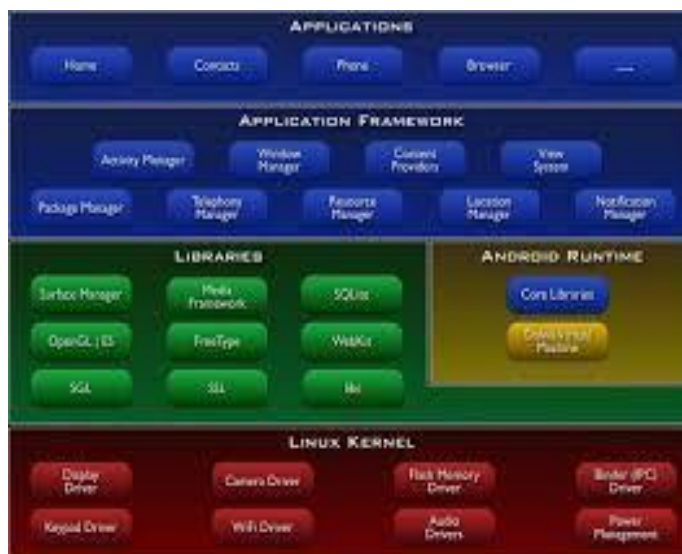
La compañía Android, Inc. fundada en Palo Alto, California en octubre de 2003 por Andy Rubin, Rich Miner, Nick Sears y Chris White. Esta compañía, durante sus inicios desarrollo Android hasta que en junio de 2005, fue adquirida por Google.

El 5 de noviembre de 2007 un consorcio de varias compañías llamada Open Handset Alliance presentó Android; un sistema operativo para móviles. Su modelo de desarrollo es de software libre construido sobre la versión 2,6 de Linux.

1.1.3. Arquitectura de Android

Es importante conocer la estructura del sistema operativo Android. Esta estructura es conocida como la arquitectura, cuyo núcleo es Linux en la versión 2,6. La arquitectura está formada por varias capas, esto es una gran ventaja para el desarrollador ya que por medio de librerías, no necesita programar a bajo nivel las funcionalidades de su aplicación.

Figura 2. **Arquitectura Android**



Fuente: Android Developer. <https://www.udacity.com/.../android-developer-nano>. Consulta: 2 de abril del 2013.

1.1.3.1. Núcleo Linux

Este es un Kernel de Linux versión 2,6 es una capa de atracción entre el hardware y el resto, por lo tanto es la única capa que depende del hardware. Proporciona el manejo de la memoria y el soporte de drivers para dispositivos,

de esta forma el desarrollador no debe acceder directamente, ya que puede utilizar librerías que están disponibles en otras capas.

1.2. Realidad Aumentada

Tiene como objetivo principal la visualización de objetos virtuales en el mundo real. Estos son generados por un software en tiempo real, de esta forma el usuario final ve el mundo real más la información que le muestran los objetos creados por el software.

La realidad aumentada tiene muchos usos en la actualidad como la medicina, arquitectura, turismo, entretenimiento, educación, entre otros y la mayoría pueden ser implementados en un dispositivo móvil. Para crear un software de realidad aumentada, en un dispositivo móvil, es necesario obtener la localización, datos del compás, la cámara y el acelerómetro.

Una de las muchas aplicaciones de la realidad aumentada es en el turismo, puede ser usada en una aplicación móvil para que los turistas puedan visualizar y encontrar lugares como restaurantes, hoteles, parques, y sitios turísticos.

1.2.1. Motor de Realidad Aumentada

Es un software que brinda al desarrollador una base para crear aplicaciones de realidad aumentada. Existe una gran variedad de motores de realidad aumentada de código abierto para múltiples plataformas como Microsoft Windows, Linux, Mac OS X, IOS y Android. Uno de estos motores de realidad aumentada es Mixare es de código abierto utiliza la licencia GPLv3,

funciona en IOS y Android. Este es el motor de realidad aumentada que se utiliza para el desarrollo del proyecto.

1.2.1.1. Motor de Realidad Aumentada Mixare

Es un motor de realidad aumentada de código abierto publicado bajo la licencia GPLv3. Está disponible para Android y del iPhone 3G en adelante.

1.2.1.1.1. Arquitectura de Mixare

La aplicación Mixare envía al servidor en una solicitud de geodata en un radio específico. La respuesta del servidor es en formato JSON, Mixare lo interpreta y lo despliega en el modo de realidad aumentada.

Figura 3. **Funcionamiento de Mixare**



Fuente: *Mixare*. <https://play.google.com/store/apps/details?id=org.mixare>. Consulta: 4 de abril del 2013.

1.3. Geolocalización

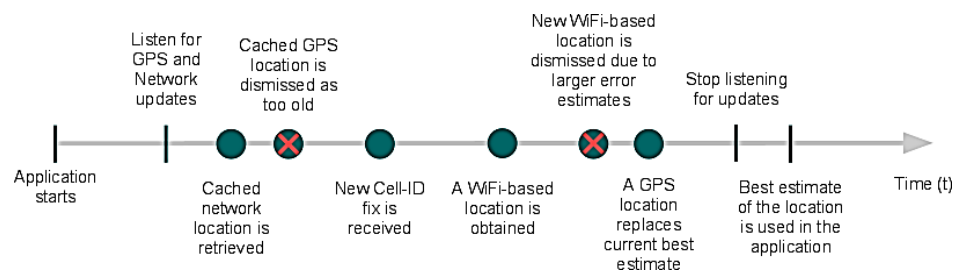
Es localizar un objeto en un sistema de coordenadas, por ejemplo las coordenadas latitud y longitud permiten localizar un punto en un mapa. Los dispositivos móviles facilitan el uso de la geolocalización por medio del

proveedor de internet o por GPS y han permitido una gran variedad de aplicaciones.

En los dispositivos Android el GPS solo funciona en el exterior y consume mucha batería, y la localización por internet es más rápida; funciona en el exterior como en interiores y no consume tanta batería, pero el GPS sigue siendo más preciso, Android permite escoger uno de los dos o utilizar ambos.

Algunos de los errores que se pueden tener al obtener la localización son generados por el movimiento del usuario, por lo que hay que actualizar la información cada cierto tiempo

Figura 4. Línea de tiempo en la que una aplicación obtiene la localización



Fuente: Google Developers. <https://www.udacity.com/.../android-developer-nano> Consulta: 27 de junio del 2013.

1.4. Teorías que respaldan el trabajo de graduación

La universidad de Brigham Young ha recopilado en su sitio web una serie de teorías utilizadas para sustentar los trabajos de investigación en el campo de tecnologías de la información. Los temas tratados son variados, incluyendo redes sociales, comportamiento humano, aceptación de la tecnología, el modelo

mental que existe en una organización, y otros. De todas esas teorías se han escogido algunas que forman una base teórica que soporta el presente documento y se presentan a continuación.

1.4.1. Objetos fronterizos

Boundary Object en inglés es un concepto usado en sociología introducido por Susan Star y James Griesemer en 1989. Esto dicta que la información es utilizada de diferentes maneras en diferentes entornos sociales. Los objetos son tan flexibles como para adaptarse a las necesidades locales, pero suficientemente rígidos para mantener una identidad a través de las fronteras. Sus propiedades pueden ser débiles cuando se usa en comunidad y fuertes cuando se interpreta individualmente. Pueden ser abstractos o concretos, lo importante es que el significado es circunstancial.

Una misma palabra puede significar cosas diferentes en países distintos, además de la clara diferencia de lenguajes. Un mismo objeto puede utilizarse de diferentes maneras en sociedades distintas, pero aun así tener una utilidad común reconocible en cualquier lugar. Estos objetos son de suma importancia para la unión de diferentes entornos sociales, proveen un concepto común a todos, fácilmente reconocible y comprensible, pero sin restarle identidad a las personas que los usan, en las diferentes maneras que los usan.

La relación entre la teoría y el presente trabajo es que ambos se valen de estos objetos para conectar diferentes sociedades. Cuando se habla de redes sociales, y en general con la tecnología, en un mundo conectado de tantas maneras se mencionan a personas con intereses en común y que vienen, en contextos tan variados como los intereses que manejan. Hay ciertas formas en las que muchas personas en el mundo coinciden y se relacionan, esto les

permite formar sociedades utilizando las redes sociales, sin importar sus orígenes, raza, idioma o religión. No hay necesidad de la presencia física para compartir grandes cantidades de información, conteniendo objetos fronterizos que conectan a las personas.

Un punto de interés podría ser un banco; para una persona podría ser un simple lugar de trabajo, para otra persona una oportunidad de negocios, una tercera persona puede ser un lugar donde puede obtener un servicio. Aquí se observa la plasticidad de los objetos fronterizos. Sin embargo, las tres personas tienen en común el concepto del banco, la función que cumple, su ubicación y la forma de acceder a los servicios que presta.

El concepto de banco es común en todo el mundo. Un banco puede tener distintas ubicaciones, diferentes formas de prestar servicios, pueden ser físicamente distintos, tener diferentes filosofías organizacionales, manejar diferentes monedas, y otras, de nuevo el concepto de banco es suficientemente flexible como para adaptarse a diferentes contextos sociales. Un banco es el lugar donde se encuentre o las personas que usen los servicios.

Otro ejemplo es una plaza, esta puede ser un buen lugar para tomar una fotografía, o un lugar de encuentro para los amigos, puede ser un monumento tributo a una figura importante en la historia, en otros países incluso un centro de comercio o un lugar sagrado con importancia puramente religiosa. El concepto de plaza es constante en todos estos casos: es un espacio abierto, de acceso público, donde muchas personas convergen por distintas razones. El objeto es tan flexible como para ser un lugar sagrado o de comercio, pero tan rígido como para significar un espacio abierto y público a través de las fronteras.

Con el uso de los dispositivos móviles inteligentes y su integración con las redes sociales, muchas personas en todo el mundo se pueden comunicar por medio de estos objetos, por medio de una fotografía, lugar, frase (sin importar el idioma) o gesto. Todos estos conceptos pueden carecer de significado alguno en un contexto distinto al original, pero las comunidades virtuales le dan el significado común a través del intercambio de información y la presencia virtual.

1.4.2. Riqueza de los medios

Media *richness*, en inglés, es una teoría que intenta categorizar los medios de comunicación según su capacidad para reproducir o recrear la información que contiene. Introducida por Richard Draft y Robert Hengel en 1984, evalúa la calidad de la información que se transmite en una organización en medios comunes. Según la teoría los medios se pueden categorizar de forma jerárquica según su riqueza. A continuación los medios de comunicación ordenados de menor a mayor nivel de riqueza:

- Documentos escritos sin recipiente específico: boletines, memos, *posters*, y otros.
- Documentos escritos a uno o varios recipientes específicos: una carta o un correo electrónico.
- Conversaciones grupales no presenciales: llamadas tripartitas, radios de corto alcance.
- Conversaciones no presenciales entre dos personas: líneas fijas, VoIP.
- Videoconferencias, a través de celulares, tablets o computadoras personales.
- Conversaciones cara a cara.

La riqueza de los medios se determina en función de ciertas características como las siguientes:

- Habilidad para manejar varios puntos de origen de la información
- Habilidad para facilitar la retroalimentación
- Habilidad para enfocarse en una o varias personas
- Habilidad para utilizar el lenguaje natural

De todas estas características se puede determinar la capacidad que tiene un medio de reproducir la información que el emisor desea que se transmita al momento de formular el mensaje.

La teoría surge por la necesidad de encontrar un balance entre el esfuerzo que se necesita para codificar un mensaje y la información que realmente se necesita para realizar una tarea. En general, el objetivo es reducir la incerteza y la equivocidad. La incerteza se refiere a no entender la tarea que fue encargada. Equivocidad, a no contar con la información necesaria para realizar una tarea, aun cuando el requerimiento fue comprendido.

Cuando una tarea tiene ambas incerteza y equivocidad, como convocar a los miembros de un equipo a una reunión, basta con un medio de comunicación de baja categoría para transmitir esa información. Una tarea más compleja podría resultar ambigua o incomprensible para el receptor y por tanto es necesario usar un medio más rico.

Encontrar el balance es clave para que la información fluya correctamente a través de una organización, y que el esfuerzo o costo de transmitirlo, no sean más grandes que el beneficio que conlleva la correcta transmisión del mensaje. Usar medios de bajo costo y esfuerzo como un memorandum puede resultar en

una tarea mal realizada y tener consecuencias negativas, las cuales se pudieron evitarse al usar un medio de mayor riqueza.

A medida que la tecnología avanza, los medios de comunicación se van centralizando en un solo canal o dispositivos, la internet por ejemplo. Una computadora conectada a internet puede ser usada para redactar correos, enviar mensajes de texto a celulares, tener conversaciones grupales con texto simple e incluso videoconferencias en tiempo real, todos con capacidad de retroalimentación instantánea. Esto mismo aplica a los dispositivos móviles.

Mientras las capacidades de cómputo de los dispositivos aumenten, incrementando la riqueza de la información que son capaces de transferir. Dos décadas atrás un teléfono celular era capaz de enviar mensajes de texto y realizar llamadas, hoy en día un teléfono inteligente puede compartir fotos, audio y video, algunos incluso pueden reconocer rostros y gestos humanos mediante la cámara de video. Un mismo aparato o dispositivo puede funcionar en distintas categorías de riqueza.

El uso de geolocalización en los teléfonos permite obtener grandes cantidades de información acerca de la ubicación, información que antes no estaba disponible o que era difícil de acceder. Un mapa podría categorizarse como un medio de baja riqueza.

Alguien con cierta experiencia puede usar un mapa para guiarse en un terreno desconocido, asumiendo que conoce su ubicación actual, de otra manera el mapa le resulta inútil. Un teléfono inteligente que cuenta con servicios de geolocalización puede guiar a una persona inexperta a través de una ciudad con comandos de voz que le indican la dirección a seguir, la ruta más óptima e incluso información de servicios próximos que podrían interesarle.

1.4.3. Capital social

La teoría de capital social se originó, probablemente, en una publicación de Mark Granovetter, seguida por otras varias entre 1985 y 2001 por parte de Pierre Bourdieu, James Coleman, Mark y Robert Putnam. El concepto ha existido en toda sociedad desde los comienzos de la humanidad, pero fue hasta esas fechas donde se comenzó a estudiar como una característica individual de un ser humano.

Capital social se refiere al conjunto de relaciones y normas que facilitan y propician acciones colectivas con beneficio mutuo. Una persona que vive aislada de las demás no cuenta con capital social ya que este se produce y acumula a medida que se relaciona de diferentes maneras con sus semejantes. Entre más relaciones crea una persona, más capital social acumula y más fácil es también generar. Los beneficios de acumular capital social pueden ser sociales, psicológicos, emocionales o económicos. Además, no necesariamente es personal y es posible identificar capital humano en un grupo de personas.

El capital humano está constituido, entre otras cosas, de los siguientes elementos (los cuales pueden funcionar en dos vías):

- Confianza: que inspira un individuo y que le tiene a sus compañeros y organizaciones que le rodean.
- Solidaridad: ser reconocido como una persona que está dispuesta a prestar ayuda y tener la seguridad de contar con la ayuda de sus iguales.
- Pertenencia: la noción de pertenecer a un grupo y sentirse identificado con las personas que lo integran.
- Normas generalizadas: el ser parte de un sistema de normas y acatarse a ellas sabiendo que existen para el bien común.

Con el uso de aparatos inteligentes conectados a diferentes redes sociales y medios de comunicación; una persona puede construir su capital social rápida y fácilmente. No es necesaria la presencia física para contactar a una persona y recibir retroalimentación instantánea. El conocer las reglas sobre lo que está permitido tratar en un foro, el contribuir respondiendo interrogantes de miembros nuevos, el ser conocido por los miembros antiguos, todo esto contribuye al capital social de una persona que es miembro de una red social virtual.

La facilidad de expresarse acelera la acumulación del capital pues una persona que tiene la capacidad de publicar a nivel global sus pensamientos, puede encontrar otras personas que comparten sus intereses o ideologías, además de darle reputación o desacreditarlo en función de cómo se exprese.

2. IDENTIFICACIÓN DEL PROBLEMA Y SOLUCIÓN PLANTEADA

2.1. Antecedentes

Un trabajo relacionado con el tema de geolocalización y realidad aumentada lleva por título *Utilización de servicios de geolocalización en dispositivos móviles sobre la ciudad universitaria usac, utilizando tecnologías Google*, de Cristian Chan y Rubén Quintana. Está enfocado a los nuevos estudiantes de la Universidad San Carlos de Guatemala, que no conocen las ubicaciones de edificios de las diferentes facultades, y otros lugares de interés como cafeterías y parqueos. En la aplicación también utilizaron como motor de realidad aumentada Mixare.

Analizando este trabajo se demuestran los beneficios de las tecnologías aplicadas a una población menor, pero con el mismo problema se aplica para una mayor población y de esta forma encontrar puntos de interés en el país.

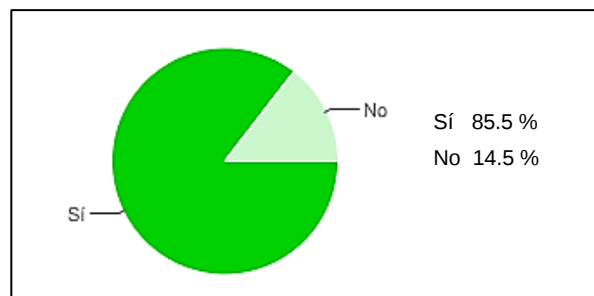
2.2. Descripción del problema

El uso del GPS para determinar la posición de una persona, objeto o lugar es de gran utilidad en varios campos de la ciencia, la industria y en la vida cotidiana. Hasta hace algunos años el conseguir un aparato que utiliza esta tecnología era costoso y complicado. Actualmente muchos teléfonos inteligentes traen integrada esta capacidad, abriendo oportunidad a explotar todo el potencial que esta tecnología ofrece.

Sucede algo similar con la realidad aumentada, cualquier persona debería beneficiarse de lo que la realidad aumentada ofrece. Las aplicaciones son variadas, por ejemplo: turismo, desarrollo comunitario, arte, seguridad, y otros. Es casi un estándar que los teléfonos inteligentes tengan una cámara integrada, de esta forma cualquier persona que posea uno puede implementar esa tecnología.

Según los resultados de la encuesta el 85,5 % de las personas que cuentan con un dispositivo móvil han utilizado aplicaciones para localizar un sitio al que deseaban ir.

Figura 5. **Porcentaje de personas que utilizan aplicaciones para localizar sitios de interés**

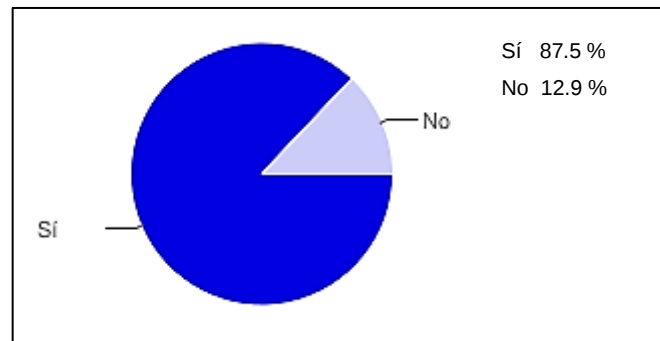


Fuente: elaboración propia.

Si bien existen algunos programas para dispositivos móviles que permiten compartir información acerca de la ubicación, la mayoría están destinados a un uso específico. Además, la mayoría del software que sirve para estos propósitos es cerrado y, por tanto, inaccesible a alguna persona que quisiera conocer cómo funciona o cómo ampliar la funcionalidad que ya tiene. Se necesita una aplicación de uso más general en donde una persona pueda definir varios puntos en su pueblo, ciudad, país o incluso en el mundo entero,

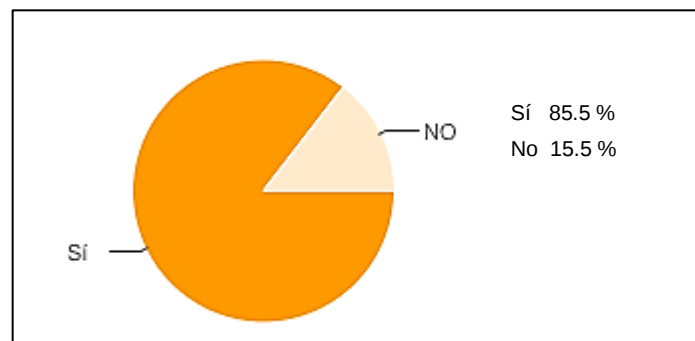
darles un interés o utilidad propia y elegir compartirlo con otras personas. La aplicación debe ser completamente accesible para cualquier estudiante o entusiasta de la tecnología que quiera conocer su funcionamiento y replicarlo o aumentarlo para sus propios propósitos.

Figura 6. **Porcentaje de personas interesadas en visualizar lugares turísticos con una aplicación de realidad aumentada**



Fuente: elaboración propia.

Figura 7. **Porcentaje de personas interesadas en visualizar y agregar lugares turísticos con una aplicación móvil**

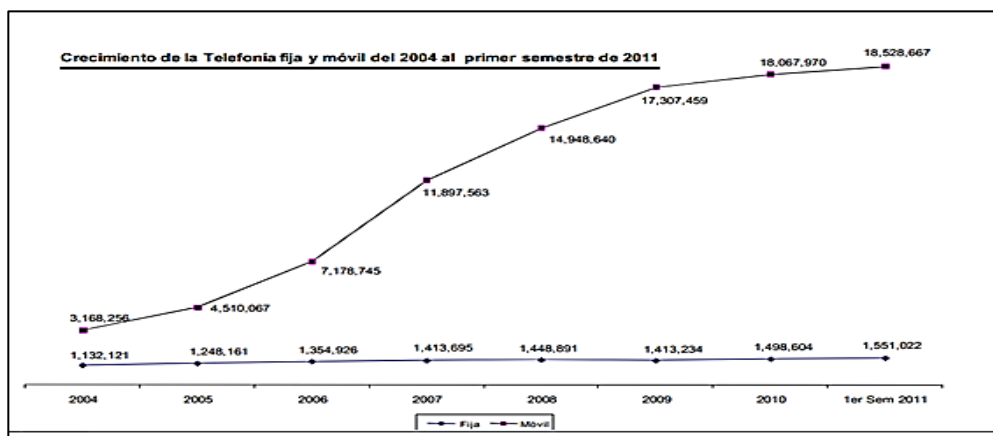


Fuente: elaboración propia.

2.3. Benchmark de la aplicación

En esta sección se listan las aplicaciones para móviles que tienen relación con los temas de geolocalización y realidad aumentada. Cada una de estas provee una funcionalidad similar, pero sin terminar de satisfacer la problemática planteada.

Figura 8. Crecimiento de la telefonía fija y móvil del 2004 al primer semestre 2011



Fuente: Superintendencia de Telecomunicaciones Guatemala. *Crecimiento de la telefonía fija y móvil*. https://es.wikipedia.org/.../Superintendencia_de_Telecomunicaciones. Consulta: 3 de marzo de 2015.

2.4. Waze

Es una aplicación que provee mapas e información acerca del tránsito en tiempo real. Se vale de la colaboración de sus usuarios para reportar eventos clave como un congestionamiento, un puesto de control de la policía, un

accidente, y otros. A continuación se muestran sus principales características y capacidades:

- Ruteo en vivo basado en la información de tránsito y vías generada en tiempo real por la comunidad de usuarios.
- Alertas de vía de la comunidad. Incluye accidentes, obstáculos, controles policiales, bloqueos de calles y más.
- Navegación por voz completa.
- Reruteo automático cuando cambian las condiciones en las calles.
- Aprende los destinos frecuentes, horas de conducción y rutas preferidas.
- Encuentra la gasolinera más barata en la ruta.
- Navegación a eventos de Facebook.
- Ver amigos de Facebook, conducir al mismo destino.
- Notifica a persona que llegará al destino enviándole una hora estimada de arribo y un *link* que muestra conduciendo sobre un mapa.
- Suma puntos y asciende en la tabla de posiciones del país contribuyendo con información de las calles.
- Mapas en vivo, constantemente editados y actualizados por la comunidad de editores de mapas de Waze.

Aunque un usuario es capaz de compartir información en tiempo real acerca de su ubicación Waze no se basa en lugares sino más bien en eventos asociados a una ubicación. Estos eventos son volátiles. Un accidente, por ejemplo, es un punto de interés solo mientras sea un problema para el tránsito, una vez está resuelto desaparece del mapa.

2.5. Junaio

Es una aplicación de realidad aumentada basada en geolocalización. Permite buscar eventos, lugares o incluso personas, a través de diferentes canales. Sus funciones son:

- Utilice el modo escáner y consigue acceso a información de productos, videos, páginas web o incluso echar un vistazo completo a una revista de Realidad Aumentada.
- Combina contenidos de algunas de las aplicaciones mejor valoradas, tales como 11811.es, Instagram, eBay clasificados en una sola.
- Usa los últimos avances tecnológicos sobre Realidad Aumentada, mostrando toda la información disponible en el lugar que le corresponde. Una experiencia pura y natural.
- El entretenimiento está a tan solo un “clic-y-desliza” se descubren juegos de Realidad Aumentada, concursos o contenidos emocionantes actualizados constantemente.

Si bien Junaio se basa en la ubicación del usuario para proveer su servicio es imposible para un usuario compartir su propia información o definir sus propios puntos de interés. Además, no es posible agregar más fuentes de información (canales), ni siquiera uno definido por el usuario mismo.

3. DOCUMENTACIÓN BASE PARA EL DESARROLLO DE LA APLICACIÓN

3.1. Documentación base para el desarrollo del sitio web

La aplicación de móviles guateAumentada se sirve de un sitio web que contiene servicios para almacenar, administrar y proveer la información de los puntos y usuarios. El objetivo es tener la información centralizada, segura y persistente, para no depender de los teléfonos celulares almacenando los datos. Además, se encuentra publicada en la red para que cualquier dispositivo y usuario pueda acceder a ella.

El sitio web complementa a la aplicación dándole oportunidad al usuario de consultar datos generales, administrar su perfil de usuario e ingresar puntos nuevos, todo desde un navegador. El sitio no sustituye la aplicación para móviles, dado que la función principal de la aplicación móvil es la realidad aumentada, característica que no está presente en el sitio.

3.1.1. Tecnologías utilizadas

A continuación se muestran algunas tecnologías utilizadas.

3.1.1.1. PHP

Hypertext Preprocessor es un lenguaje de programación de *scripting* que puede ser embebido dentro de páginas HTML para generar contenido web

dinámico. Este tiene una sintaxis basada en C, y en su versión 5 soporta el paradigma de objetos.

3.1.1.2. Zend Framework 2

Zend es un *framework* de código abierto para desarrollar aplicaciones web utilizando el lenguaje de programación PHP, en su versión 5,3 y más recientes. Está construido de forma modular de manera que sus componentes son independientes y se pueden utilizar solo las partes necesarias. Este *framework* incluye una colección de librerías que proveen las funciones básicas más usadas al momento de desarrollar aplicaciones web, como los son: autenticación, autorización, conexiones a bases de datos, maquetado de forma programática, y otros.

Zend está orientado al patrón de diseño MVC (modelo, vista, controlador), y aunque las librerías se pueden usar siguiendo otro patrón, facilita el diseño y la construcción de sitios usando este paradigma. Mediante el uso de plantillas y formas construidas de manera programática se facilita el mantenimiento de la interfaz gráfica independiente de la lógica de negocio y manejo de datos.

El desarrollo de Zend está patrocinado principalmente por la compañía Zend Technologies, pero ha recibido aportes de otras empresas de software grandes como Google, Microsoft y Strikelron. Además, por su condición de ser código abierto, recibe grandes aportes constantemente de la comunidad a través de listas de correo, canales IRC y similares.

3.1.1.3. Google Static Maps API

Es un servicio que provee Google para generar imágenes de mapas sin tener que utilizar JavaScript, PHP o ningún otro lenguaje de programación. Se hace una solicitud HTTP estándar con parámetros determinados y Google genera una imagen que se puede mostrar en cualquier sitio web. Se pueden dibujar marcadores, rutas, y áreas, Permitiendo personalizar los colores y posiciones del mapa y generar imágenes en distintos formatos.

3.1.1.4. Bootstrap

Es un conjunto de componentes visuales contruidos sobre CSS3 y JavaScript que facilita la construcción de sitios web mediante el uso de un sistema de filas y columnas adaptándose a diferentes resoluciones de pantallas. Con esto se evita el tener que usar varios diseños orientados a diferentes dispositivos. Además, tiene una colección de componentes generales para desplegar y recibir información, y contiene íconos, colores y tipos de letra que ayudan a la estética del sitio.

3.1.1.5. MySQL

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

3.1.2. Dependencias

Para que la aplicación web pueda funcionar es necesario contar con los siguientes componentes:

3.1.2.1. Servidor web

Es un programa que se dedica a atender peticiones entrantes que usan el protocolo HTTP y responden de acuerdo a la petición utilizando el mismo protocolo. Maneja archivos y directorios, y los accesos sobre estos. Puede además extender su funcionalidad para manejar lenguajes de *scripting* como PHP para generar contenido dinámico.

Para el trabajo de graduación se utilizó el software Apache HTTP Server. Este es un programa hecho en lenguaje C, de código abierto, mantenido por la Apache Software Foundation. Es la solución más popular dada su característica de ser código abierto, gratuito, altamente extensible a través de módulos, seguro y multiplataforma.

Existen otros programas que cumplen la función de servidor web y pueden sustituir a Apache HTTP Server, pero se recomienda el uso de este software dado que ha probado ser una solución robusta y confiable a través de los años. La versión utilizada al momento de generar este trabajo de graduación es la 2.2.22.

3.1.2.2. PHP

Se usó PHP para generar el contenido dinámico del sitio web y proveer los servicios de autenticación y persistencia de datos para la aplicación de móviles. El sitio web guateAumentada se creó usando la versión 5.4.3 de este lenguaje.

3.1.2.3. MySQL

Para almacenar la información de puntos se usó MySQL en su edición estándar, en la versión 5.5.24. No necesita ninguna configuración especial.

El presente documento no pretende cubrir a detalle el proceso de instalación de los prerequisites, pero hace evidente que si no se cubren estas dependencias la aplicación web no puede funcionar.

3.1.3. Arquitectura de la aplicación web

Para la aplicación web se implementó un patrón de diseño MVC (modelo-vista-controlador). Zend Framework está orientado a esta arquitectura y por tanto facilita la construcción de sitios de esa manera. A continuación se describe en qué consiste el patrón MVC.

MVC es un patrón de diseño que separa el manejo de datos, la lógica del negocio y la interfaz con el usuario, cada uno en su parte, para que puedan funcionar de manera independiente. Esta separación facilita el mantenimiento y la escalabilidad, y fomentando la reutilización de código. Para este propósito se propone el uso de tres componentes principales:

3.1.3.1. Modelo

Se encarga de manejar la lógica del negocio. Es independiente a la forma en que se almacenan y presentan los datos. Controla el flujo de la información desde el usuario al almacenamiento, la procesa, y devuelve una respuesta a la Vista para que sea desplegada.

3.1.3.2. Vista

Se encarga de desplegar y solicitar datos al usuario. Es indiferente al uso que se le dé a los datos que recibe y necesita del modelo para procesarlos y generar una respuesta.

3.1.3.3. Controlador

Es el intermediario entre el Modelo y la Vista. Cada petición de la Vista llega al Modelo a través del Controlador. Se encarga de traducir la respuesta del Modelo para entregarla a la vista.

3.1.4. Construcción de una aplicación utilizando Zend Framework

A continuación se muestra la construcción de una aplicación utilizando Zend Framework.

3.1.4.1. Aplicación esqueleto

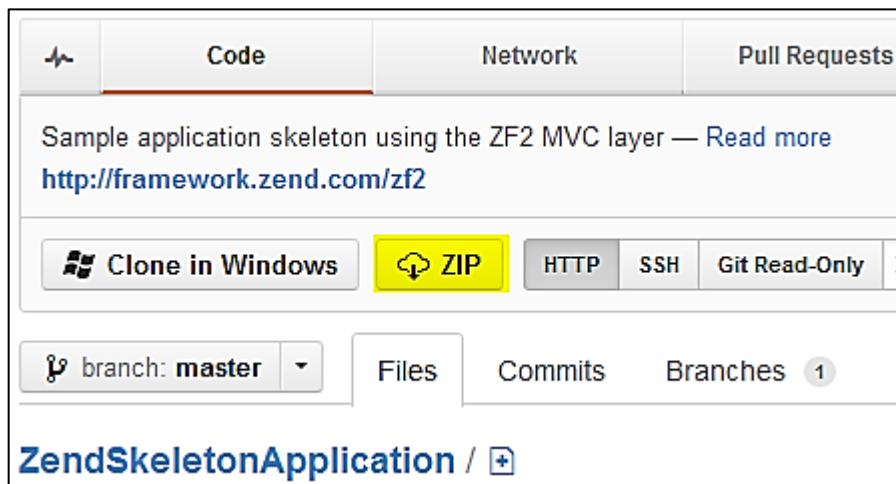
Zend provee una aplicación esqueleto con la estructura de clases y directorios necesarios para tener la funcionalidad básica de una aplicación web usando el patrón MVC. Esta aplicación nos servirá para construir sobre ella el proyecto entero.

La aplicación esqueleto se puede obtener en la siguiente dirección:

<https://github.com/zendframework/ZendSkeletonApplication>

Para descargar la aplicación en un solo archivo comprimido, se hace clic en el botón Zip, como se presenta a continuación:

Figura 9. **Github**



Fuente: elaboración propia, Snipping Tool.

Luego de esto se desempaqueta el archivo en alguna ubicación fácil de acceder, se renombra la carpeta y se configura el servidor Apache para que encuentre la aplicación. De ahora en adelante se asume que el directorio que contiene la aplicación web es C:\guateAumentada.

La aplicación esqueleto y este proyecto dependen de Zend Framework, por sí solos no pueden funcionar. Por eso es necesario obtener Zend Framework para usar la aplicación esqueleto.

3.1.4.2. Obteniendo el Zend Framework

La forma más fácil de descargar la última versión del Zend Framework y manejar futuras versiones es utilizando el programa composer. El mismo se puede obtener en la siguiente dirección (<http://getcomposer.org/>). Después de instalarlo se ingresa al directorio raíz de la aplicación esqueleto y se ejecuta el comando:

```
php composer.phar install
```

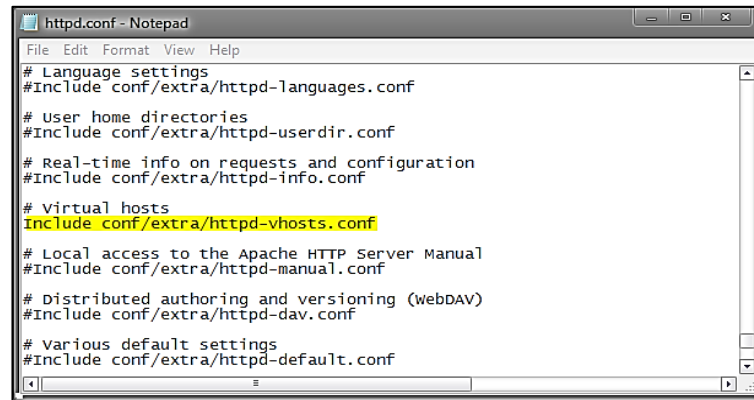
Con esto se descarga la colección de librerías que conforman el Zend Framework.

3.1.4.3. Configuración de Apache

Se necesita crear un *Virtual Host* que apunte a la carpeta de nuestra aplicación esqueleto para que se acceda a ella desde un navegador. Se edita el archivo de configuración principal del servidor Apache, que lleva por nombre httpd.conf. Este archivo se encuentra en el directorio ./conf.

Se busca la línea #Virtual hosts y se remueve el caracter '#' de la línea inferior, de la siguiente manera:

Figura 10. Archivo httpd.conf 1



```
httpd.conf - Notepad
File Edit Format View Help
# Language settings
#Include conf/extra/httpd-languages.conf

# User home directories
#Include conf/extra/httpd-userdir.conf

# Real-time info on requests and configuration
#Include conf/extra/httpd-info.conf

# Virtual hosts
Include conf/extra/httpd-vhosts.conf

# Local access to the Apache HTTP Server Manual
#Include conf/extra/httpd-manual.conf

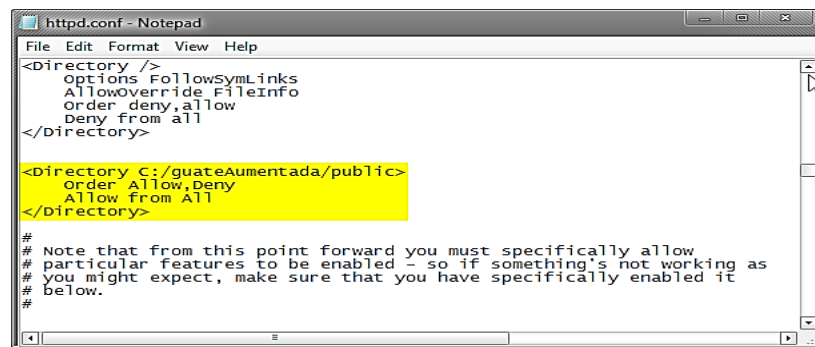
# Distributed authoring and versioning (WebDAV)
#Include conf/extra/httpd-dav.conf

# Various default settings
#Include conf/extra/httpd-default.conf
```

Fuente: elaboración propia, Snipping Tool.

Además, se debe hacer el directorio de la aplicación accesible. Esto se logra añadiendo las siguientes líneas:

Figura 11. Archivo httpd-conf 2



```
httpd.conf - Notepad
File Edit Format View Help
<Directory />
Options FollowSymLinks
AllowOverride FileInfo
Order deny,allow
Deny from all
</Directory>

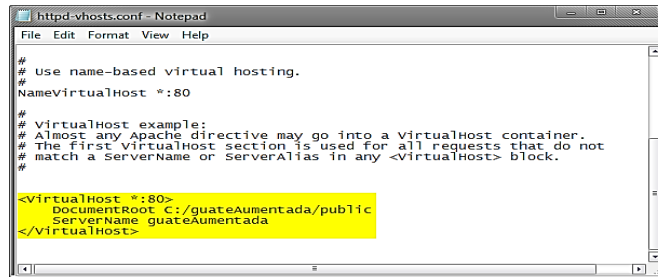
<Directory C:/guateAumentada/public>
Order Allow,Deny
Allow from All
</Directory>

# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
```

Fuente: elaboración propia, Snipping Tool.

Luego se crea el Virtual Host en el archivo ./conf/extra/httpd-vhosts.conf. Se agregan las siguientes líneas que se pueden apreciar en la figura 8.

Figura 12. Archivo httpd.conf 3



```
httpd-vhosts.conf - Notepad
File Edit Format View Help

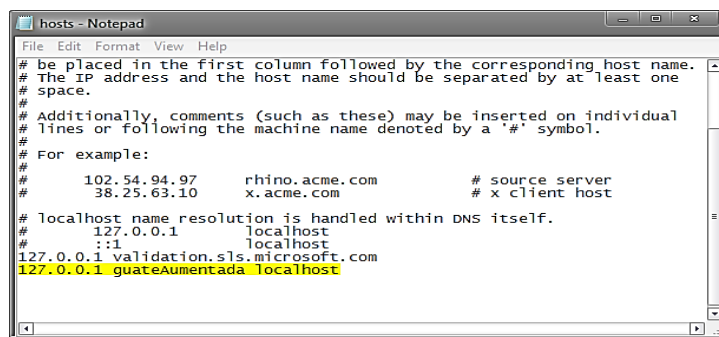
# Use name-based virtual hosting.
#
NameVirtualHost *:80

#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ServerName or ServerAlias in any <VirtualHost> block.
#
<VirtualHost *:80>
    DocumentRoot C:/guateAumentada/public
    ServerName guateAumentada
</VirtualHost>
```

Fuente: elaboración propia, Snipping Tool.

Ahora se procede a agregar la siguiente línea al archivo de *hosts* de Windows C:\Windows\System32\drivers\etc. Su equivalente en el sistema operativo Linux es /etc/hosts.

Figura 13. Archivo hosts



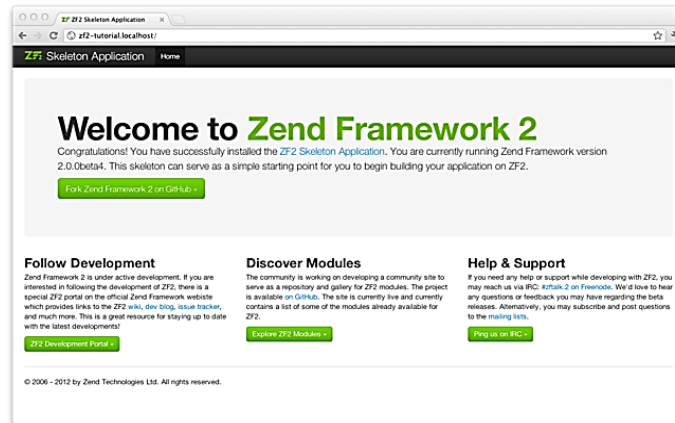
```
hosts - Notepad
File Edit Format View Help

# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97    rhino.acme.com        # source server
#       38.25.63.10    x.acme.com           # x client host
#
# localhost name resolution is handled within DNS itself.
#
#       127.0.0.1      localhost
#       ::1            localhost
127.0.0.1 guateAumentada localhost
```

Fuente: elaboración propia, Snipping Tool.

De esta manera cuando se ingrese a la dirección `http://guateaugmentada/` en el navegador, la máquina buscará el servidor que se está ejecutando localmente. Si todo fue configurado correctamente se debería ver el archivo `httpd.conf` figura 10.

Figura 14. **Zend Framework 2**

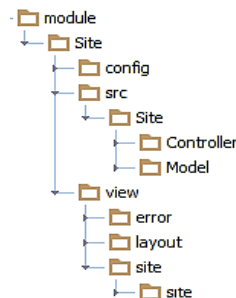


Fuente: elaboración propia, Snipping Tool.

3.1.4.4. **Estructura de directorios**

El siguiente paso es crear la estructura de directorios que usa Zend Framework. Se crearán los directorios necesarios para un árbol como en la figura 11.

Figura 15. **Estructura de directorios**



Fuente: elaboración propia, Snipping Tool.

A continuación se detalla el propósito de cada directorio:

- **Module:** es el lugar donde se almacena el código PHP de nuestra aplicación. Es el código que se crea para que este proyecto funcione.
- **Site:** Cada directorio que este dentro de Module representa un módulo de Zend. Un módulo es un componente independiente que puede tener o no una interfaz gráfica, y que se puede comunicar con otros módulos para extender su funcionalidad. En este proyecto, por facilidad y debido a que las funciones cumplidas no son muy variadas, se creó un único módulo genérico con nombre Site que atiende todas las necesidades de la aplicación.
- **config:** contiene el archivo `module.config.php` que describe cómo se comporta el módulo para atender las peticiones que recibe, qué peticiones debería manejar y con qué estructura las recibe (son las denominadas rutas que más adelante se explican a más detalle).
- **src:** contiene el código fuente del modelo y del controlador y deben cumplir con la estructura dada para que Zend pueda encontrarlos.
- **Controller:** contiene el código de uno o varios controladores.
- **Model:** abarca el código de uno o varios modelos.
- **view:** incluye las vistas, interfaces gráficas con que se puede dar una respuesta en HTML. Un módulo podría no tener vistas si no tiene interfaz con el usuario o si sus respuestas las da en otro formato (JSON, por ejemplo).
- **error:** vistas para las páginas que contiene respuestas con error.
- **layout:** contiene plantillas que se pueden reutilizar en uno o varios controladores. Aquí se pueden incluir encabezados y cabeceras, por ejemplo.
- **site:** contiene el HTML base con que se responde en cada acción.

3.1.4.4.1. Configuración del módulo

Se crea un archivo con nombre `module.config.php` que contendrá las rutas y los sobrenombres que se usarán para definir la forma en que la aplicación responde a las peticiones. Primero se explica el controlador para manejar nuestro modelo y vistas:

Figura 16. Archivo `module.config.php` I



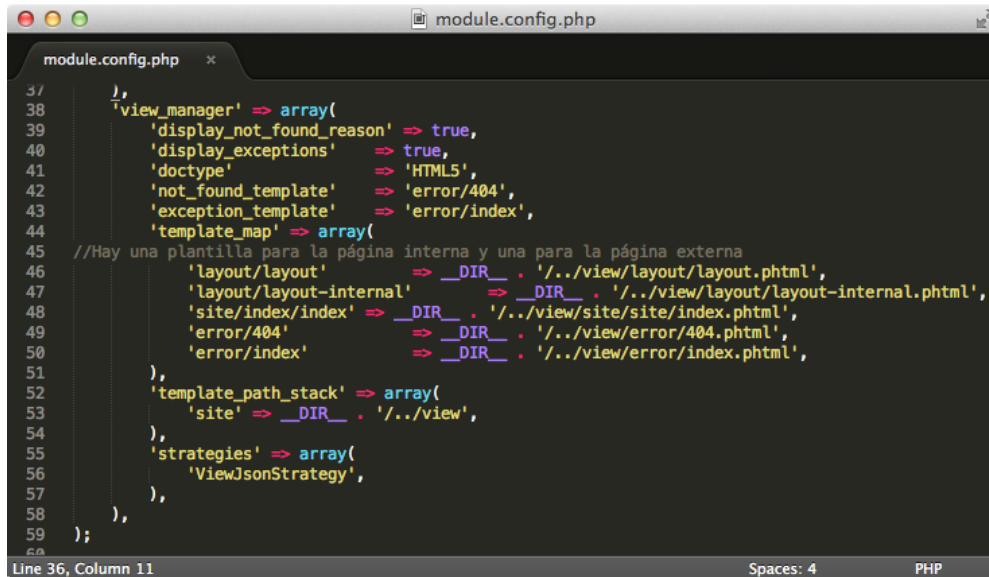
```
module.config.php
1 <?php
2 return array(
3     'controllers' => array(
4         'invokables' => array(
5             'Site\Controller\Site' => 'Site\Controller\SiteController',
6         ),
7     ),
8 );
9
10
```

Line 7, Column 6 Spaces: 4 PHP

Fuente: elaboración propia, Snipping Tool.

Luego se configura el manejador de vistas. Aquí entran en juego los directorios creados anteriormente. Se configuran los directorios en donde se encuentran las plantillas para las vistas regulares y las vistas de error. También se indica cuál es la vista predeterminada (`index`). Por último se agrega una configuración para que el controlador pueda responder con vistas en formato JSON en vez de HTML. Esto es clave para los servicios que se prestan a la aplicación móvil.

Figura 17. Archivo module.config.php II



```
37     },
38     'view_manager' => array(
39         'display_not_found_reason' => true,
40         'display_exceptions' => true,
41         'doctype' => 'HTML5',
42         'not_found_template' => 'error/404',
43         'exception_template' => 'error/index',
44         'template_map' => array(
45             //Hay una plantilla para la página interna y una para la página externa
46             'layout/layout' => __DIR__ . '/../view/layout/layout.phtml',
47             'layout/layout-internal' => __DIR__ . '/../view/layout/layout-internal.phtml',
48             'site/index/index' => __DIR__ . '/../view/site/site/index.phtml',
49             'error/404' => __DIR__ . '/../view/error/404.phtml',
50             'error/index' => __DIR__ . '/../view/error/index.phtml',
51         ),
52         'template_path_stack' => array(
53             'site' => __DIR__ . '/../view',
54         ),
55         'strategies' => array(
56             'ViewJsonStrategy',
57         ),
58     ),
59 );
```

Fuente: elaboración propia, Snipping Tool.

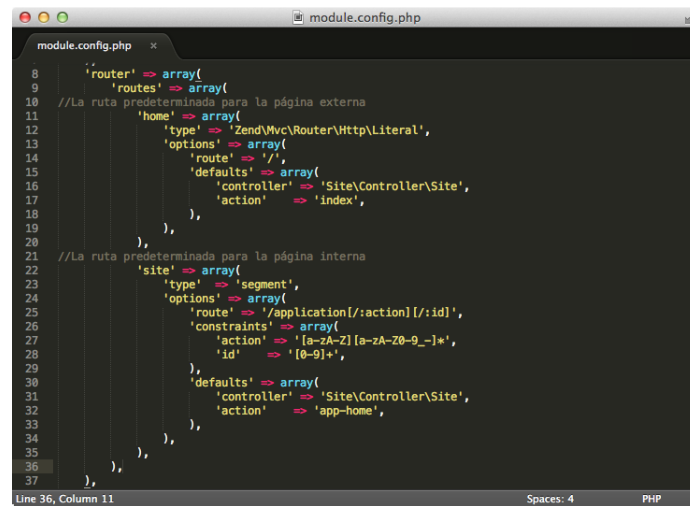
3.1.4.4.2. Rutas

Definen una estructura con que se forman las direcciones en el navegador y la forma en que la aplicación responde a cada dirección. Para este proyecto se crearon dos rutas diferentes, una para la aplicación interna y otra para la externa.

Existen diferentes tipos de rutas para generar patrones de URL distintos y con esto seleccionar la aplicación y las acciones que ejecuta. Para este proyecto se usó una ruta de tipo segment. Este tipo de ruta sirve para definir secciones que cumplen con una expresión regular, que pueden ir contiguas y donde se da un nombre y un significado a cada sección. Luego se obtiene el valor de cada sección y accionar acorde.

La ruta de la aplicación interna se observa en la figura 14.

Figura 18. **Rutas en archivo module.config.php**



```
8      'router' => array(
9          'routes' => array(
10             //La ruta predeterminada para la página externa
11             'home' => array(
12                 'type' => 'Zend\Mvc\Router\Http\Literal',
13                 'options' => array(
14                     'route' => '/',
15                     'defaults' => array(
16                         'controller' => 'Site\Controller\Site',
17                         'action' => 'index',
18                     ),
19                 ),
20             ),
21             //La ruta predeterminada para la página interna
22             'site' => array(
23                 'type' => 'segment',
24                 'options' => array(
25                     'route' => '/application[/:action[/:id]]',
26                     'constraints' => array(
27                         'action' => '[a-zA-Z][a-zA-Z0-9_-]*',
28                         'id' => '[0-9]+',
29                     ),
30                     'defaults' => array(
31                         'controller' => 'Site\Controller\Site',
32                         'action' => 'app-home',
33                     ),
34                 ),
35             ),
36         ),
37     ),
```

Fuente: elaboración propia, Snipping Tool.

La ruta definida consta de tres partes:

- /application: que es un valor constante y necesario. Con esto se separa la página externa de la interna.
- :action: define la acción que se va a tomar, siempre dentro de la página interna. Esta acción puede ser un valor cualquiera, pero debe coincidir con el nombre de alguna acción definida en el controlador, que a su vez podría tener una vista definida.
- :id: esto es un posible valor numérico que ayuda a identificar un elemento para alguna acción que así lo requiera.

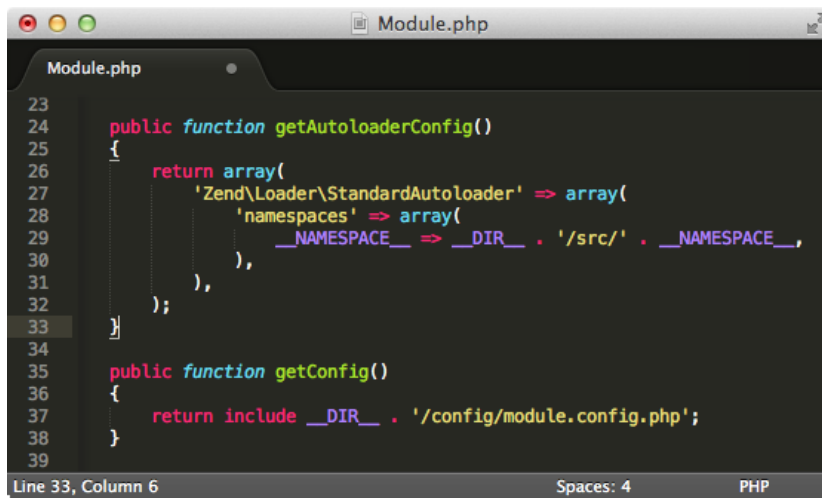
El uso de los corchetes implica que los valores para :action y :id son opcionales.

La sección constraints se refiere a los posibles valores que puede tener cada sección de la ruta. En la sección :action se definió una expresión regular para una cadena que comienza con una letra y luego puede contener cualquier caracter alfanumérico.

3.1.4.4.3. Carga e inicialización del módulo

Zend tiene un manejador de módulos que se encarga de buscar los módulos e inicializar los servicios que cada uno ha definido como necesarios. Eso lo hace usando el archivo module.php ubicado en el directorio Module. El contenido inicial es el siguiente:

Figura 19. Archivo Module.php

A screenshot of a code editor window titled 'Module.php'. The editor shows the following PHP code:

```
23
24 public function getAutoloaderConfig()
25 {
26     return array(
27         'Zend\Loader\StandardAutoloader' => array(
28             'namespaces' => array(
29                 __NAMESPACE__ => __DIR__ . '/src/' . __NAMESPACE__,
30             ),
31         ),
32     );
33 }
34
35 public function getConfig()
36 {
37     return include __DIR__ . '/config/module.config.php';
38 }
39
```

The status bar at the bottom indicates 'Line 33, Column 6', 'Spaces: 4', and 'PHP'.

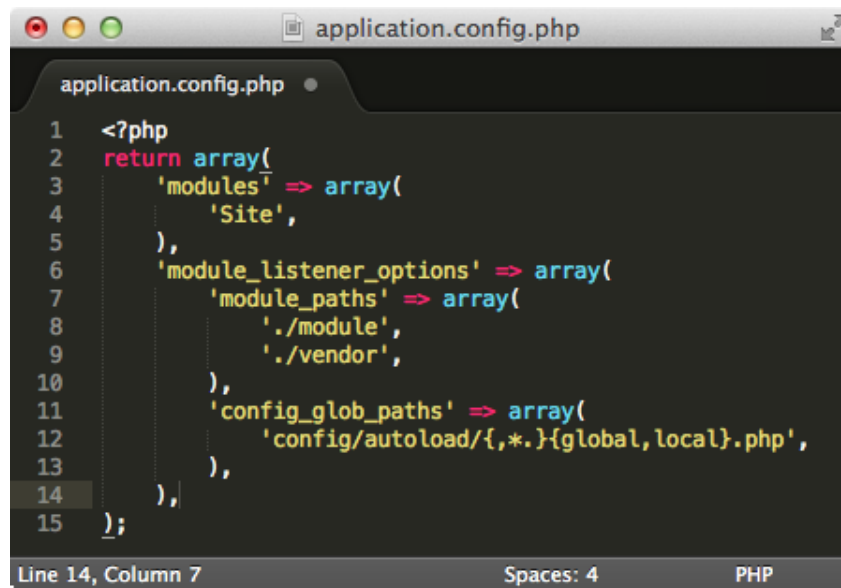
Fuente: elaboración propia, Snipping Tool.

Esta es configuración estándar y mínima para que el módulo funcione. Por el momento no necesita más.

3.1.4.4. Registrando el módulo

Para que Zend sepa de la existencia del módulo Site es necesario registrarlo en la configuración global de la aplicación. Esto se realiza usando el archivo `application.config.php`, que se encuentra en el directorio `config`, en la raíz de la aplicación. El contenido debe ser el siguiente:

Figura 20. Archivo `application.config.php`

A screenshot of a code editor window titled 'application.config.php'. The editor shows PHP code for registering a module. The code is as follows:

```
1 <?php
2 return array(
3     'modules' => array(
4         'Site',
5     ),
6     'module_listener_options' => array(
7         'module_paths' => array(
8             './module',
9             './vendor',
10        ),
11        'config_glob_paths' => array(
12            'config/autoload/{,*.}{global,local}.php',
13        ),
14    ),
15 );
```

The editor has a dark theme. The status bar at the bottom indicates 'Line 14, Column 7', 'Spaces: 4', and 'PHP'.

Fuente: elaboración propia, Snipping Tool.

Originalmente existía otra sección con nombre `Application`, pero esa ya no funciona porque era parte de la aplicación esqueleto y no tiene relación con el proyecto.

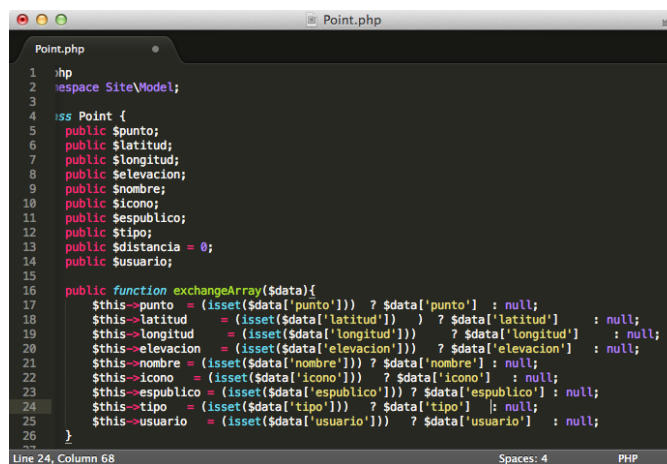
3.1.4.4.5. Modelo para los puntos

Para almacenar los puntos se creó la tabla punto, que tiene los campos:

- punto: la llave primaria de la tabla
- latitud: posición norte o sur respecto al Ecuador del planeta tierra
- longitud: posición este u oeste en relación al meridiano de Greenwich
- elevación: altura respecto al nivel del mar
- nombre: una palabra arbitraria para identificar el punto
- icono: una imagen que fue adjunta al guardar el punto
- tipo: una forma simple de catalogar varios puntos similares
- usuario: código del usuario que guardó el punto

Para representar el punto se crea un archivo Point.php en el directorio /module/Site/src/Site/Model con la clase Point. Este es el modelo que representa un punto, es independiente de la forma en que se almacenan y presentan los datos. El contenido del archivo es el siguiente:

Figura 21. Archivo Point.php



```
1  <?php
2  namespace Site\Model;
3
4  class Point {
5      public $punto;
6      public $latitud;
7      public $longitud;
8      public $elevacion;
9      public $nombre;
10     public $icono;
11     public $espublico;
12     public $tipo;
13     public $distancia = 0;
14     public $usuario;
15
16     public function exchangeArray($data){
17         $this->punto = (isset($data['punto'])) ? $data['punto'] : null;
18         $this->latitud = (isset($data['latitud'])) ? $data['latitud'] : null;
19         $this->longitud = (isset($data['longitud'])) ? $data['longitud'] : null;
20         $this->elevacion = (isset($data['elevacion'])) ? $data['elevacion'] : null;
21         $this->nombre = (isset($data['nombre'])) ? $data['nombre'] : null;
22         $this->icono = (isset($data['icono'])) ? $data['icono'] : null;
23         $this->espublico = (isset($data['espublico'])) ? $data['espublico'] : null;
24         $this->tipo = (isset($data['tipo'])) ? $data['tipo'] : null;
25         $this->usuario = (isset($data['usuario'])) ? $data['usuario'] : null;
26     }
27 }
```

Fuente: elaboración propia, Snipping Tool.

El modelo debe contener una variable pública por cada campo de la tabla, y un método `exchangeArray()` que le sirve a Zend para transformar la información que viene en forma de arreglo a un objeto de tipo `Point`. El método `toArray()` hace el proceso inverso de `exchangeArray` y `toMixareArray`, convierte el objeto al formato que usa Mixare para intercambiar información de los puntos que se despliegan en la aplicación móvil.

Figura 22. Funciones en el archivo **Point.php**

```
28 public function toArray(){
29     return array(
30         'punto' => $this->punto,
31         'latitud' => $this->latitud,
32         'longitud' => $this->longitud,
33         'elevacion' => $this->elevacion,
34         'nombre' => $this->nombre,
35         'icono' => $this->icono,
36         'espublico' => $this->espublico,
37         'tipo' => $this->tipo,
38         'usuario' => $this->usuario,
39     );
40 }
41
42 public function toMixareArray(){
43     return array(
44         'id' => $this->punto,
45         'lat' => $this->latitud,
46         'lng' => $this->longitud,
47         'elevation' => $this->elevacion,
48         'title' => $this->nombre,
49         'distance' => $this->distancia,
50         'has_detail_page' => 0,
51         'webpage' => '',
52         'espublico' => $this->espublico,
53         'tipo' => $this->tipo,
54         'imagen' => $this->icono,
55     );
56 }
57 }
```

Fuente: elaboración propia, Snipping Tool.

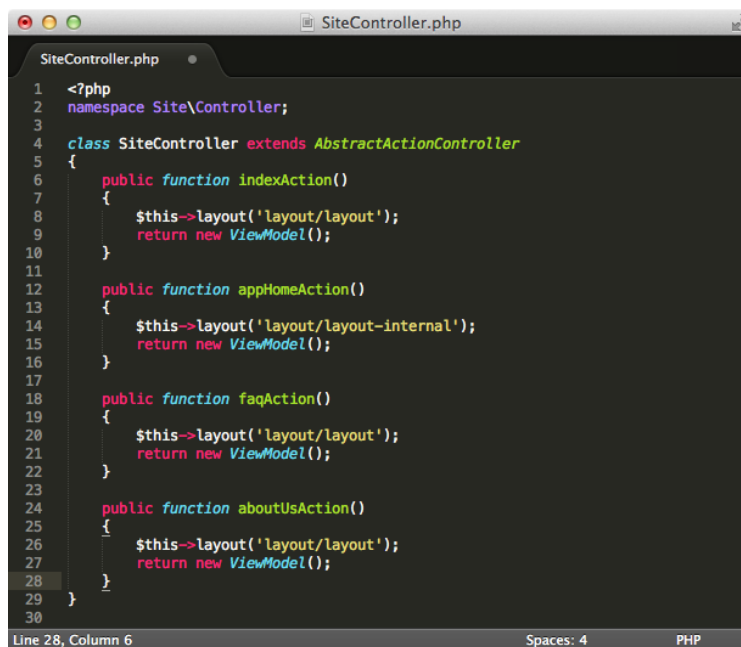
Además de `Point` se necesita una clase que sirva de interfaz entre el medio de almacenamiento y el controlador, y que provea los métodos necesarios para agregar, consultar, modificar y eliminar puntos. La clase se llama `PointTable` (el archivo es `PointTable.php`) y tiene los siguientes métodos:

3.1.4.4.6. Creando el controlador

Para atender los llamados que se hagan a la aplicación se debe contar con un controlador que ejecuta las acciones y devuelve una respuesta, ya sea en HTML o en JSON.

El archivo para el controlador lleva por nombre SiteController y se encuentra en el directorio /Module/Site/src/Site/Controller. El contenido inicial del archivo es el siguiente:

Figura 23. Archivo SiteController.php

The image shows a code editor window titled 'SiteController.php'. The code is written in PHP and defines a class 'SiteController' that extends 'AbstractActionController'. The class contains four public methods: 'indexAction()', 'appHomeAction()', 'faqAction()', and 'aboutUsAction()'. Each method calls '\$this->layout()' with a specific layout name and then returns a new 'ViewModel()' object. The code is formatted with syntax highlighting, and the editor shows line numbers from 1 to 30. The status bar at the bottom indicates 'Line 28, Column 6', 'Spaces: 4', and 'PHP'.

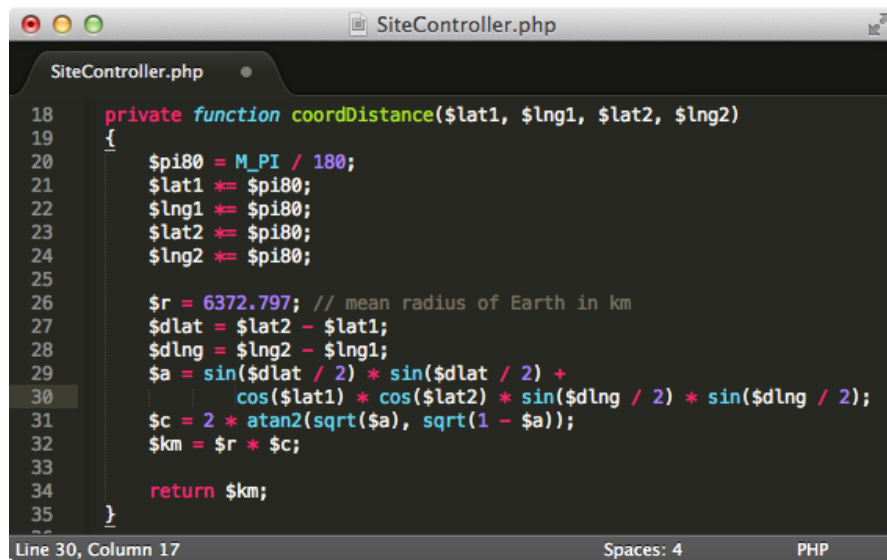
Fuente: elaboración propia, Snipping Tool.

Estas acciones proveen a la aplicación móvil la posibilidad de guardar puntos y obtenerlos después. Los puntos pueden ser públicos o pueden estar asociados a un usuario.

3.1.4.4.7. Función coordDistance

Sirve para calcular la distancia entre dos puntos conociendo sus coordenadas. Recibe cuatro parámetros: latitud y longitud del primer punto, y latitud y longitud del segundo punto.

Figura 24. Función coordDistance en archivo SiteController.php



```
SiteController.php
18 private function coordDistance($lat1, $lng1, $lat2, $lng2)
19 {
20     $pi80 = M_PI / 180;
21     $lat1 *= $pi80;
22     $lng1 *= $pi80;
23     $lat2 *= $pi80;
24     $lng2 *= $pi80;
25
26     $r = 6372.797; // mean radius of Earth in km
27     $dlat = $lat2 - $lat1;
28     $dlng = $lng2 - $lng1;
29     $a = sin($dlat / 2) * sin($dlat / 2) +
30         cos($lat1) * cos($lat2) * sin($dlng / 2) * sin($dlng / 2);
31     $c = 2 * atan2(sqrt($a), sqrt(1 - $a));
32     $km = $r * $c;
33
34     return $km;
35 }
```

Line 30, Column 17 Spaces: 4 PHP

Fuente: elaboración propia, Snipping Tool.

3.1.4.4.8. Función addPointAction

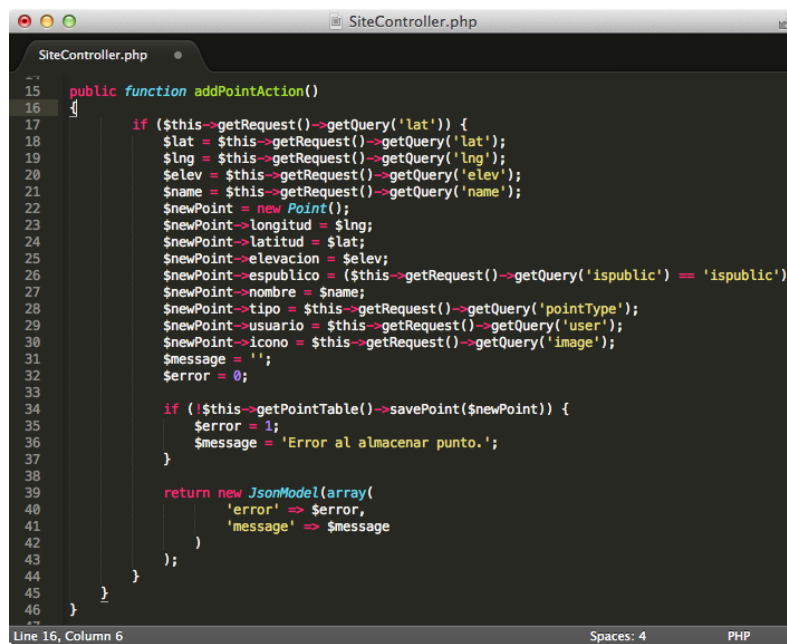
Sirve para guardar puntos nuevos. Esta función es llamada desde la aplicación móvil de la siguiente forma:

add-point?lat=14.600801&lng=-90.521041&elev=1500&name=prueba
&pointType=1&ispublic=ispublic

La función espera los siguientes parámetros:

- lat: latitud del punto a guardar
- lng: longitud del punto a guardar
- elev: elevación del punto a guardar
- name: un nombre para identificar al punto
- pointType: un tipo de punto, de un catálogo de tipos de punto
- ispublic: indica si el punto es público o privado

Figura 25. Función `addPointAction` en archivo `SiteController.php`



```
SiteController.php
15 public function addPointAction()
16 {
17     if ($this->getRequest()->getQuery('lat')) {
18         $lat = $this->getRequest()->getQuery('lat');
19         $lng = $this->getRequest()->getQuery('lng');
20         $elev = $this->getRequest()->getQuery('elev');
21         $name = $this->getRequest()->getQuery('name');
22         $newPoint = new Point();
23         $newPoint->longitud = $lng;
24         $newPoint->latitud = $lat;
25         $newPoint->elevacion = $elev;
26         $newPoint->espublico = ($this->getRequest()->getQuery('ispublic') == 'ispublic');
27         $newPoint->nombre = $name;
28         $newPoint->tipo = $this->getRequest()->getQuery('pointType');
29         $newPoint->usuario = $this->getRequest()->getQuery('user');
30         $newPoint->icono = $this->getRequest()->getQuery('image');
31         $message = '';
32         $error = 0;
33
34         if (!$this->getPointTable()->savePoint($newPoint)) {
35             $error = 1;
36             $message = 'Error al almacenar punto.';
37         }
38
39         return new JsonModel(array(
40             'error' => $error,
41             'message' => $message
42         ));
43     }
44 }
45
46
Line 16, Column 6 Spaces: 4 PHP
```

Fuente: elaboración propia, Snipping Tool.

3.2. Documentación base para el desarrollo de la aplicación móvil

En este capítulo se detallan los procesos de instalación, configuración de las herramientas para la creación de la aplicación móvil, así como el desarrollo y pruebas de la aplicación móvil. El lenguaje utilizado para desarrollar

aplicaciones en Android es Java, un lenguaje orientado a objetos, y tener conocimientos de la arquitectura de Android. Es necesario también tener conocimientos de XML para la parte visual, y para los textos que son desplegados en los botones, etiquetas y mensajes.

3.2.1. Requerimientos para el desarrollo

En esta sección se especifican los requerimientos para el desarrollo de la solución en el sistema operativo Android.

3.2.1.1. Android SDK

Android Software Development Kit provee un API con las herramientas para compilar, probar aplicaciones para Android.

Una de las grandes ventajas que el Android SDK provee al desarrollador son los dispositivos virtuales Android (AVD), ya que los permite realizar obteniendo los mismos resultados que se tendrían en un dispositivo real. El Android AVD permite emular un dispositivo real definiendo las especificaciones de hardware y software del dispositivo virtual.

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

3.2.1.2. Android Development Tools (ADT)

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

3.2.1.3. Instalación y configuración

En esta sección se detallan los pasos para la instalación y configuración del ambiente de trabajo. Para el desarrollo de la aplicación en dispositivos móviles son el sistema operativo Android.

3.2.1.3.1. Descarga e instalación del Android SDK

Desde la página <http://developer.android.com/sdk/index.html> se puede descargar Eclipse y el ADT plugin con el Android SDK tools y todas las herramientas necesarias para el desarrollo.

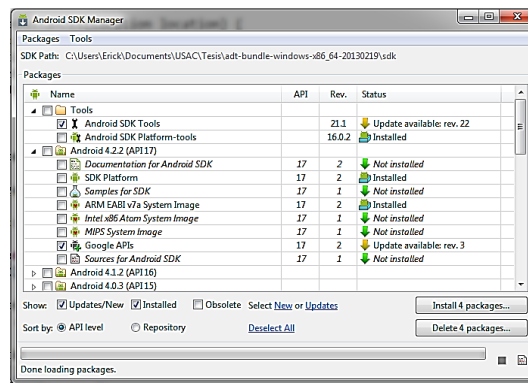
Al finalizar la descarga hay que descomprimir el archivo descargado, en la ruta donde sea desee instalar el SDK.

En la ruta de instalación existen dos directorios uno contiene el SDK de Android y en el otro se encuentra Eclipse.

3.2.1.3.2. Android SDK Manager

Es una herramienta aparte que permite al usuario obtener componentes para el desarrollo. Para entrar al Android SDK manager desde Eclipse se ingresa al menú Window>Android SDK Manager. También se encuentra el ejecutable en el directorio raíz del Android SDK.

Figura 26. Android SDK Manager

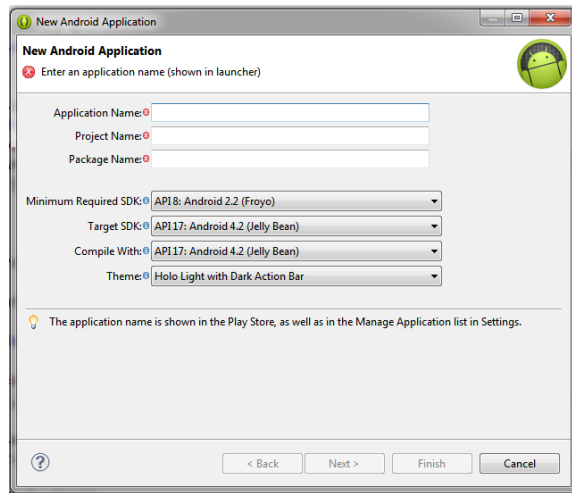


Fuente: elaboración propia.

3.2.1.3.3. Crear un nuevo proyecto en Eclipse

Se ingresa al menú File > New > Android Project, luego el nombre de la aplicación y Eclipse autocompleta el nombre del proyecto y el nombre del paquete. Como recomendación es preferible que el nombre de la aplicación inicie con una letra mayúscula.

Figura 27. **New Android Application**



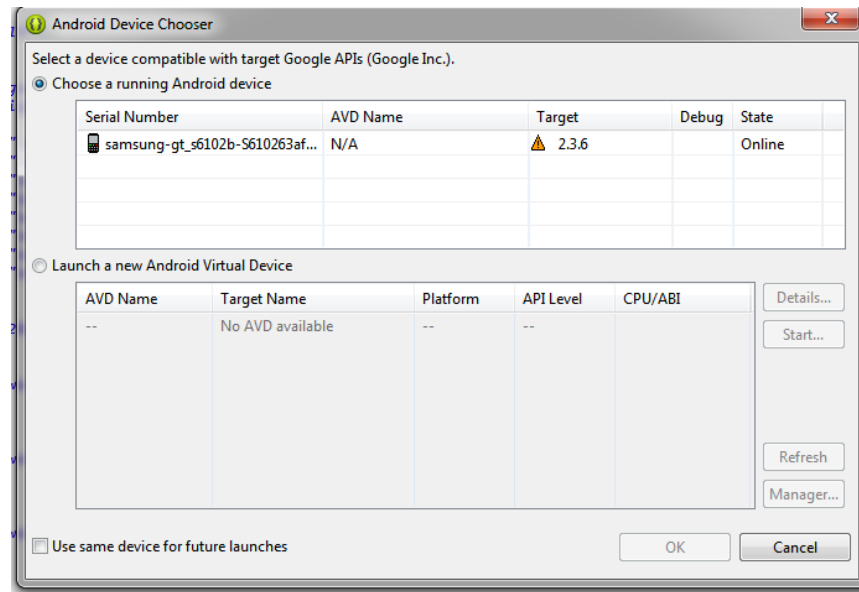
Fuente: elaboración propia, Snipping Tool.

3.2.1.4. **Ambiente de real pruebas**

Para realizar pruebas de una aplicación para Android en un dispositivo real es necesario tener instalado el driver del dispositivo en la computadora. También es necesario que el dispositivo permite la instalación de aplicación que sean del Market.

Al ejecutar la aplicación se debe tener conectado el dispositivo vía USB, la siguiente imagen muestra la ventana que despliega Eclipse donde se selecciona si se ejecuta la aplicación en el dispositivo o en un AVD.

Figura 28. **Android Device Chooser**



Fuente: elaboración propia, Snipping Tool.

3.2.2. **Desarrollo de aplicaciones para Android**

A continuación se muestra el desarrollo de aplicaciones para Android.

3.2.2.1. **Conceptos generales**

Existe una serie de conceptos clave que se deben establecer para que se puedan desarrollar aplicaciones de acuerdo a la arquitectura de Android.

3.2.2.1.1. Actividad

Es la representación gráfica de la aplicación. Es el bloque principal de construcción de todos los componentes visuales de la aplicación. Toda aplicación en Android está formada por múltiples actividades y tiene una actividad principal que es con la que interactúa el usuario al ejecutar la aplicación.

Una vista provee la interfaz gráfica de una actividad, cada vista controla un espacio rectangular donde la actividad puede interactuar con el usuario.

3.2.2.1.2. Servicios

Son procesos que se ejecutan en segundo plano. Es activado cuando un componente de la aplicación lo requiere sin que el usuario lo note, y solo se mantiene activo el tiempo que es requerido.

3.2.2.1.3. Intención (*Intent*)

Es uno de los componentes principales permitiendo la comunicación entre las actividades, por ejemplo se pueden enviar datos al ejecutar una actividad, así como recibir datos al finalizar una actividad

3.2.2.1.4. Receptor de transmisión (*Broadcast Receiver*)

Permite registrar los eventos del sistema, cada vez que ocurre un evento se ejecuta un receptor y todos estos serán notificados por el Android runtime cuando el evento sucede.

3.2.2.1.5. Proveedor de contenidos (Content Provider)

Es esencialmente una base de datos, que provee una interfaz estructurada a los datos para compartirlos con otras aplicaciones. La mayoría de aplicaciones utiliza SQLite en combinación con un Content Provider.

3.2.2.1.6. Android Manifest

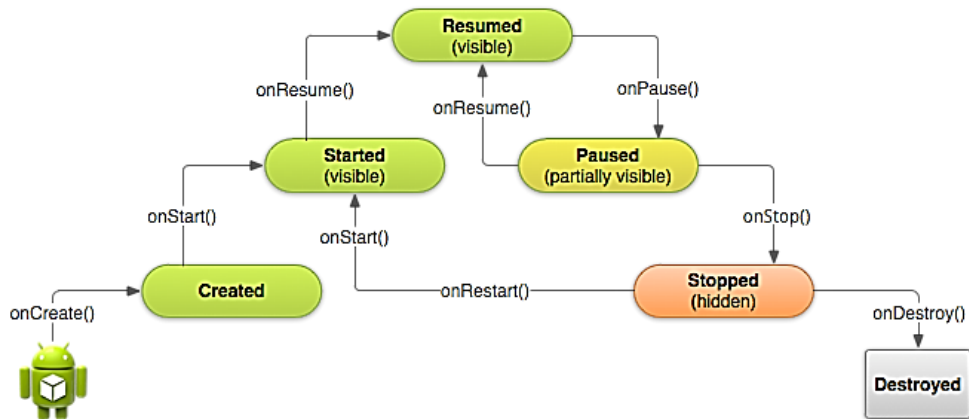
El Android Manifest es un archivo en formato XML se encuentra en el directorio raíz de la aplicación con el nombre de Androidmanifest.xml. Se encarga de informar al sistema los componentes de la aplicación, el nivel mínimo del API requerido y los requerimientos de hardware.

3.2.2.1.7. Ciclo de vida de los componentes

Esto hace posible ejecutar multiples aplicaciones de forma simultánea, para esto Android implementa una serie de llamadas a cada componente para manipular su ciclo de vida.

El ciclo de vida de las actividades inicia con la llamada onCreate() y termina cuando se hace la llamada onDestroy().

Figura 29. **Ciclo de vida de una Actividad**



Fuente: *Android Developer*. <https://developer.android.com/.../deep-linking.html>. Consulta: 22 de mayo del 2013.

3.2.3. **Desarrollo de la aplicación como solución al trabajo de investigación**

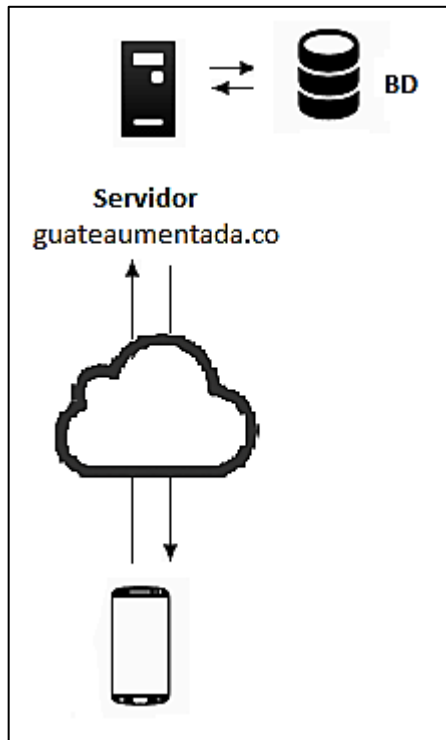
En esta sección se explica la arquitectura y el desarrollo la aplicación para los dispositivos móviles.

3.2.3.1. **Arquitectura**

La arquitectura de la aplicación se divide en dos áreas:

- Aplicación Web
- Aplicación para dispositivos Android

Figura 30. **Arquitectura del sistema**



Fuente: elaboración propia, con Microsoft Project.

Como lo describe la figura 15, el usuario al ejecutar la aplicación en su dispositivo móvil, envía solicitudes al servidor web, este le responde en formato JSON para analizar los datos recibidos en la aplicación móvil se utilizó un parser para JSON y se crean los puntos que contiene la información del lugar, la imagen, latitud, longitud, elevación, identificador y URL. Con estos datos se puede representar el lugar tanto en el modo de realidad aumentada como en el mapa.

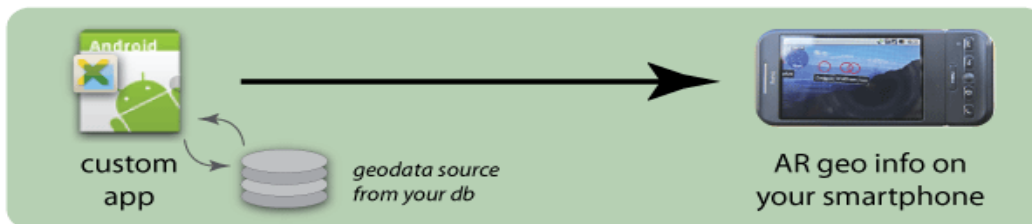
3.2.3.2. Componentes de la aplicación

En esta sección se van a describir los componentes que la aplicación para el sistema operativo Android.

3.2.3.2.1. Integración con Mixare

Mixare se puede integrar a otra aplicación, y se pueden editar sus fuentes de geodata, como se realizó con la aplicación de GuateAumentada. Se creó una aplicación nueva a la cual se integró Mixare para visualizar la geoinformación en realidad aumentada, la geodata es obtenida desde un servidor propio en formato JSON.

Figura 31. **Funcionalidad Mixare**



Fuente: Mixare. <https://code.google.com/.../mixare/.../DisplayYourO>. Consulta: 23 de mayo del 2013.

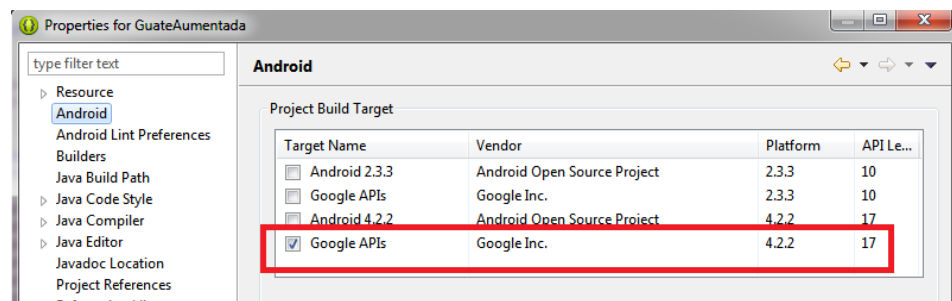
3.2.3.2.2. Integrando Mixare en Eclipse

El primer paso es descargar el código fuente de Mixare desde el sitio web <https://github.com/mixare/mixare> y exportarlo a la aplicación en Eclipse. Una vez realizado, Eclipse marcara múltiples errores en el código de Mixare.

Los errores que marca Eclipse en el código de Mixare son provocados porque la aplicación debe ser construida usando Google API y Android, ya que no cuenta con todas las librerías.

Para utilizar la versión correcta del SDK se descargar desde el Android SDK Manager. Una vez finalizada hay que asociar nuestro proyecto al nuevo SDK.

Figura 32. **Propiedades del proyecto 1**



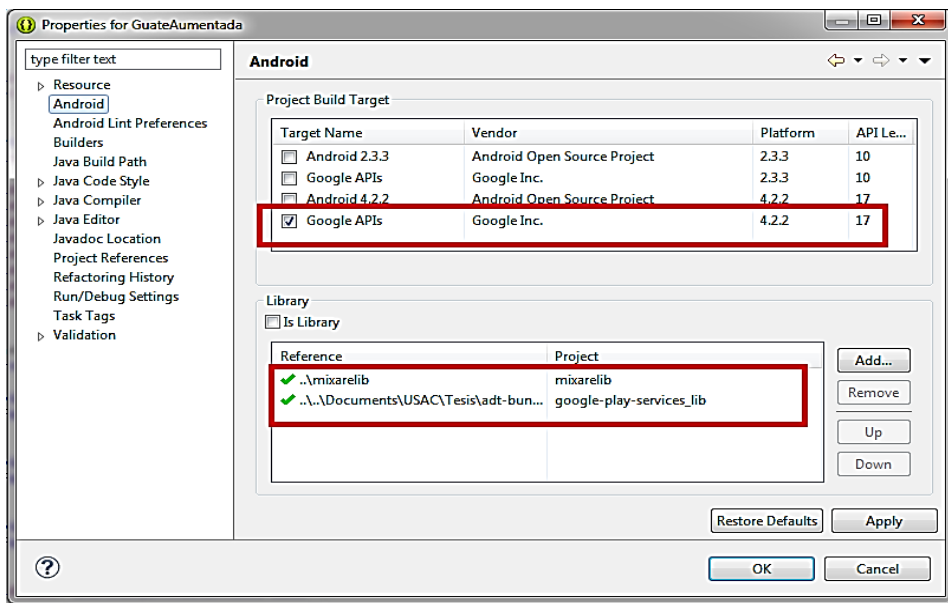
Fuente: elaboración propia.

Es necesario exportar como un nuevo proyecto la librería de Mixare, y agregarla, al hacer esto cada vez que se construye el proyecto también se construye la librería de esta forma es sencillo realizar cambios en esta.

Para exportar la librería en la ventana, para importar proyectos, se selecciona la ruta de la librería de Mixare, que está ubicada en la carpeta plugins del proyecto principal. Cuando se exporta la librería, como un nuevo proyecto, Eclipse despliega un error ya que no contiene la carpeta red. Para solucionar esto solo hay que crear la carpeta, al exportar la librería y solucionar el error, en el explorador de paquetes de Mixare debe aparecer la librería.

Hay que agregar la referencia de la librería de Mixare al proyecto para ingrese en las propiedades del proyecto, en la opción Android y se hace referencia a la librería, después de realizar las propiedades del proyecto debe verse como la figura 32.

Figura 33. **Propiedades del proyecto 2**



Fuente: elaboración propia.

El siguiente paso es agregar a la opción Java Build Path, el .jar de la librería de Mixare, y cuando se genere el apk de la aplicación, Eclipse de forma automática genera el .jar de la librería de Mixare y la agrega al proyecto, de la misma forma hay que agregar el jar en la pestaña de librerías. Una vez realizado esto se ha importado de forma correcta el motor de realidad aumentada Mixare dentro de la aplicación.

3.2.3.2.3. Actividades

Son la representación de la interfaz al usuario, todas las interacciones del usuario con la aplicación son por medio de las actividades. En el archivo AndroidManifest.xml se define la información básica de las actividades como el nombre de la clase. Un label definido por el desarrollador se puede encontrar en el archivo String.xml.

A continuación se listan las actividades que conforman la aplicación GuateAumentada:

- MainActivity
- MixView
- Login
- Mapa
- AgregarLugar
- Registro

3.2.3.2.4. Permisos en el Android Manifest

Los permisos en Android se utilizan para proteger o limitar el acceso a ciertos componentes. Estos se declaran en el archivo Android Manifest.

Al momento de la instalación de la aplicación se listan los permisos que utiliza, de esta forma el usuario puede conocer si la aplicación que necesita acceso a internet o el GPS. Para instalar la aplicación en el dispositivo móvil el usuario debe aceptar los permisos que solicita la aplicación.

En la figura 33 se listan los permisos que requiere la apelación GuateAumentada.

Figura 34. **Permisos en el archivo Android Manifest**

```
<uses-permission android:name="com.example.guateaumentada.permission.MAPS_RECEIVE" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.VIBRATE" />
```

Fuente: elaboración propia, Snipping Tool.

3.2.3.2.5. Linear layout

Es un contenedor de vistas y las ordena de forma vertical. Los elementos que estén contenidos dentro del *linear layout* van a ajustarse para ocupar el largo y ancho de ellos. Se puede utilizar para crear una lista de vistas como, por ejemplo, un grupo de text view orientados de forma vertical.

3.2.3.2.6. Text View

Es un tipo de vista, que permite al usuario desplegar texto dentro de un área. Al definir un *text view*, en el *layout*, se pueden definir sus propiedades como tamaño, alineación, color de texto, tipo de letra, tamaño del texto entre otras.

3.2.3.2.7. Edit Text

Es una subclase del *text view* que ha sido configurado para que el usuario pueda ingresar texto, para esto el usuario debe presionar el *text view* y se

despliega el teclado. Este puede variar dependiendo del valor en la propiedad `inputType` del `textview`, algunos tipos son:

- `normal abc`: muestra el teclado normal.
- `textPassword`: muestra el teclado normal con la diferencia que al ingresar el texto el `edit text` solo muestra puntos.
- `textEmailAddress`: muestra un teclado normal, pero agrega un botón la arroba y un botón que agrega el texto “.com” lo que facilita al usuario el ingreso de direcciones de correo electrónico.
- `number`: muestra al usuario un teclado numérico con algunos símbolos.

3.2.3.2.8. *Button*

Se extiende a la clase *text view* que generalmente se le agrega un *listener* que está a la espera que el usuario presione el botón y de esta forma generar un evento. También se puede crear un listener para varios botones.

3.2.3.2.9. Alert Dialog

Son ventanas utilizadas para notificar al usuario de un evento y en algunas ocasiones para que tome una decisión sobre un evento de la aplicación. Los mensajes son mostrados sobre la aplicación y la bloquean hasta que el usuario cierra la ventana.

Android define un layout por defecto a los Alert Dialog y el usuario lo puede personalizar por medio del código, pero también se puede crear un layout personalizado y definirlo al dialogo que se va a desplegar. En la aplicación para todos los Alert Dialog se utilizó el siguiente *layout*:

Figura 35. **Layout**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <ImageView
        android:id="@+id/image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="5dp" />

    <TextView
        android:id="@+id/text"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#FFF"
        android:layout_toRightOf="@+id/image"/>

    <Button
        android:id="@+id/dialogButtonOK"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text=" Ok "
        android:layout_marginTop="5dp"
        android:layout_marginRight="5dp"
        android:layout_below="@+id/image"/>
</RelativeLayout>
```

Fuente: elaboración propia, con programa Sublime Text.

3.2.3.2.10. Progress Dialog

Es una pequeña ventana que le indica al usuario que se está realizando alguna actividad en segundo plano. En general se despliegan cuando se esperan datos de internet o cuando se suben datos. En la aplicación son utilizados para indicar al usuario que se está guardando un nuevo lugar, mientras en segundo plano la imagen del nuevo lugar se está transfiriendo al servidor así como la información del nuevo lugar.

3.2.3.2.11. JSON Parser

Se necesita que la aplicación en Android pueda interpretar los documentos JSON que se obtienen de los servicios web, para esto es necesario mapear el documento JSON a un objeto que java para poder interpretarlo. Se utiliza la librería org.json de código abierto, en ella se tienen diferentes clases que abstraen un documento JSON. Existe la clase JSONObject que tiene una colección de nombres y valores, se obtienen valores *string*, *integer*, otros JSONObject o un JSONArray que es un arreglo de JSONObject.

3.2.3.2.12. Strings

Esto permite a la aplicación adaptarse al lenguaje del dispositivo móvil. Todo el texto que se muestra al usuario está en los archivos *Strings*, un archivo con formato XML, y existe uno por cada lenguaje en el que se puede utilizar la aplicación.

Ejemplo para la ventana de iniciar sesión en español:

```
<string name="title_activity_login">Login</string>
<string name="user_activity_login">Usuario</string>
<string name="passwd_activity_login">Contraseña</string>
<string name="boton_activity_login">Iniciar sesión</string>
<string name="register_activity_login">Nuevo? Regístrate aquí</string>
```

Ejemplo para la ventana de iniciar sesión en inglés:

```
<string name="title_activity_login">Login</string>
<string name="user_activity_login">User</string>
```

```
<string name="passwd_activity_login">Password</string>
<string name="boton_activity_login">Login</string>
<string name="register_activity_login">New? Register here</string>
```

3.2.3.3. Integración con el API de elevación de Google

Este servicio de google se utilizó al obtener las coordenadas, en las que se encuentra el usuario, de la aplicación con el servicio de localización por internet. La altitud siempre es cero y se necesita conocer la altitud a la que se encuentra el usuario para que en la vista de realidad aumentada los lugares se puedan visualizar en la posición correcta.

Para utilizar este servicio de Google hay que hacer una solicitud por medio de un `HttpGet`, un ejemplo seria `http://maps.googleapis.com/maps/api/elevation/json?locations=39.7391536,-104.9847034&sensor=true`, los parámetros que se envían son:

- `Locations= <latitud> , <longitud>`.
- `Sensor= true o false`, es true si la latitud y longitud se obtuvieron del sensor GPS.

La respuesta de este servicio puede ser formato JSON. un ejemplo de la elevación en este formato es:

Figura 36. **Resultados en formato JSON el API**

```
{
  "results": [
    {
      "elevation": 1608.637939453125,
      "location": {
        "lat": 39.73915360,
        "lng": -104.98470340
      },
      "resolution": 4.771975994110107
    }
  ],
  "status": "OK"
}
```

Fuente: *Google Developers*. <https://developers.google.com/speed/libraries/>. Consulta: 27 de junio del 2013.

3.2.3.4. Integración con Google Maps v2

El primer paso para la integración del API de Google Maps v2 en un proyecto en Eclipse es instalar el paquete Google Play Services. Esto provee acceso a los servicios de Google.

El segundo paso es generar un API key para que los usuarios puedan utilizar el servicio en los dispositivos móviles. Para obtener un API key se debe obtener la finger print (SHA1), del certificado de la aplicación, también se pueden generar varios API key, por ejemplo si aún no se cuenta con un certificado se puede generar una API key para el certificado que provee Eclipse. También se puede generar una API key para un certificado propio.

Se creó una actividad mapa en aplicación GuateAumentada, utilizando los marcadores que el usuario puede visualizar en el modo de realidad aumentada. Con ello localiza mapas.

Para agregarle mayor usabilidad a los marcadores, se creó una vista para reemplazar a la ventana de información que usa por defecto el API de Google Maps.

Figura 37. **Archivo XML de configuración de la interfaz web**

Archivo XML del infowindow personalizado

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="@color/bg_color" >
    <ImageView
        android:id="@+id/lugar_image"
        android:layout_width="96dp"
        android:layout_height="96dp"
        android:layout_alignParentTop="true"
    />
    <TextView
        android:id="@+id/nombre_lugar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/tv_lng"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

Fuente: elaboración propia, con programa SublimeText.

Los marcadores que se muestran en el mapa cuentan con un icono personalizado dependiendo del tipo de lugar. Al presionar una vez el marcador se despliega la ventana de información personalizada, si se presiona la ventana de información, se calcula la ruta hacia el marcador.

3.2.3.5. Integración con el API de rutas de Google

En la aplicación para dispositivos móviles se integró la funcionalidad de mostrar al usuario una ruta hacia el punto de interés. Para lograr definir una ruta se utiliza un servicio que provee Google, el API de rutas de Google retorna un JSON con la ruta recomendada, se utiliza haciendo una solicitud http a `http://maps.googleapis.com/maps/api/directions/output?parameters` donde *parameters* se reemplaza con los parámetros.

Los parámetros obligatorios son *origin*, *destination* y *sensor*, en *origin* se define la latitud y longitud del punto en el que se encuentra el usuario, en *destination* se define la latitud y longitud del punto al que se quiere llegar, el parámetro *sensor* se utiliza para indicarle al API si las coordenadas proceden de un dispositivo con sensor. En el JSON que retorna el API de Google tiene un arreglo de pasos en cada uno se muestra la distancia, la duración, las instrucciones en formato HTML y una polilínea codificada. Para dibujar la ruta en el mapa en Java hay que decodificar las polilíneas de cada uno de los pasos.

4. MANUAL DE USO DE LA APLICACIÓN

4.1. Aplicación Web

Esta se divide en dos partes. La primera es visible al usuario, sirve para consultar y agregar puntos de interés de manera rápida y sencilla utilizando una computadora. Esta es una práctica común para las personas que no están acostumbradas a trabajar con los dispositivos móviles o encuentra más práctico trabajar con una computadora. La segunda parte de la aplicación oculta al usuario y se encarga de proveer, a la aplicación, los servicios para almacenar, consultar y modificar puntos, creando y autenticando usuarios.

A continuación se muestra cómo utilizar las funciones de la aplicación web que son visibles al usuario.

4.1.1. Creando usuarios

Para hacer uso de las aplicaciones, tanto la móvil como la web, es necesario con un usuario registrado. A este usuario se le asocian los puntos de interés. Para crear un usuario desde la página web se requieren los siguientes datos:

Figura 38. Creación de usuarios

¡Bienvenido!

Necesitamos algunos datos:

Usuario	Elige un usuario
Pase	Pase
Confirmar pase	Confirma tu pase
Nombres	Nombre
Apellidos	Apellido
Correo	e-mail
Sitio web	Sitio web

Fuente: elaboración propia, con programa Snipping Tool.

A excepción del sitio web todos los demás datos son obligatorios. Una vez ingresados los datos se presiona el botón “Enviar”. Si existiera algún error la aplicación se lo indica.

Figura 39. Error al crear usuarios

Necesitamos algunos datos:

Usuario	Elige un usuario
Pase	Pase
Confirmar pase	Confirma tu pase
Nombres	Nombre
Apellidos	Apellido
Correo	e-mail
Sitio web	Sitio web

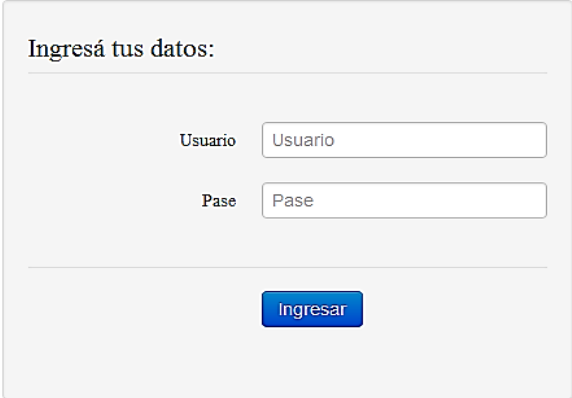
Usuario ya existe.

Fuente: elaboración propia, con programa Snipping Tool.

4.1.2. Autenticando los usuarios

El primer paso para usar la aplicación web es la autenticación de los usuarios. Todo en la aplicación está relacionado con el usuario autenticado, excepto los puntos públicos. Con base en ese usuario se sabe qué puntos desplegar y a qué usuario agregar los puntos que son creados. Todo esto se almacena en la sesión.

Figura 40. Autenticación de usuarios



Formulario de autenticación de usuarios. El formulario tiene un fondo gris claro y una sombra. En la parte superior, el texto "Ingresa tus datos:" está subrayado. Debajo, hay dos campos de entrada: "Usuario" y "Pase", cada uno con un botón de "X" para borrar. El campo "Pase" tiene un ícono de ojo para alternar la visibilidad. En la parte inferior, hay un botón azul con el texto "Ingresar".

Fuente: elaboración propia, con programa Snipping Tool.

Se digita un usuario y una contraseña para ingresar y si los datos son correctos, la aplicación redirecciona a la página interna. De haber un error, también se hace evidente.

Figura 41. **Error al iniciar sesión**



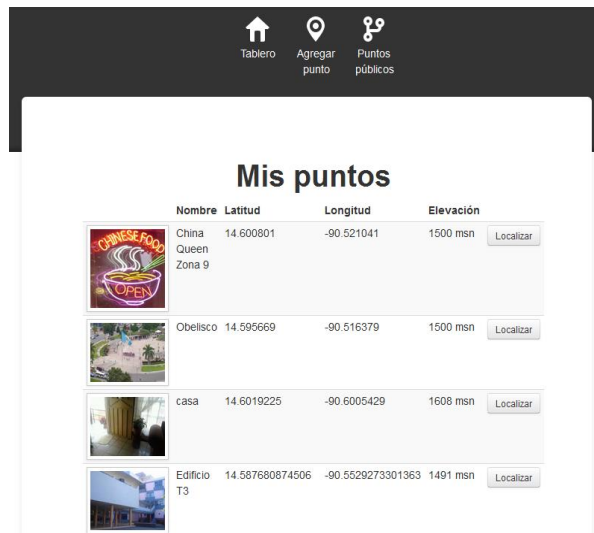
The image shows a login form with a light gray background. At the top, the text "Ingresá tus datos:" is followed by a horizontal line. Below this, there are two input fields: the first is labeled "Usuario" and contains the text "Usuario"; the second is labeled "Pase" and contains the text "Pase". Below the input fields, there is a red rectangular box with the text "Usuario o pase incorrecto." in a dark red font. At the bottom of the form, there is a blue button with the text "Ingresar" in white.

Fuente: elaboración propia, con programa Snipping Tool.

4.1.3. Puntos del usuario

Al ingresar al sitio lo primero que se despliega son los puntos del usuario, que podrían ser públicos o no.

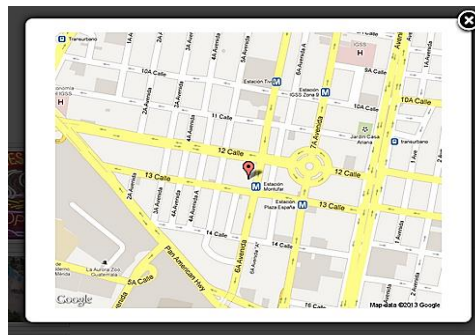
Figura 42. **Mis puntos**



Fuente: elaboración propia, con programa Snipping Tool.

Para cada punto se despliega la imagen almacenada por el usuario, el nombre, ubicación y un botón “Localizar” que sirve para desplegar la ubicación del punto en un mapa.

Figura 43. **Punto sobre el mapa**



Fuente: elaboración propia, con programa Snipping Tool.

Para listar los puntos de un usuario se puede hacer clic en el ícono del “Tablero” en cualquier momento.

4.1.4. Agregando puntos

Se puede hacer clic en cualquier parte del mapa. Con esto se agrega un marcador y automáticamente se obtiene la longitud, latitud y elevación de ese punto. Se debe adjuntar una imagen, asignarle un tipo y un nombre al punto. Por último se debe especificar si es un punto público o privado.

Figura 44. **Agregar punto**

Click para agregar un punto



Map data ©2013 Google - Terms of Use

Latitud: 14.61643 Longitud: -90.53351 Elevación: 1530.242

Imagen: Browse...

Tipo: Residencia Nombre: Mi casa ☐ Público

Fuente: elaboración propia, con programa Snipping Tool.

Cuando un punto es agregado correctamente se desplegará un mensaje confirmando el evento.

4.1.5. Consultando puntos públicos

El listado de puntos públicos es muy similar al de los puntos del usuario. La diferencia es que aparecerán todos los puntos que definieron como públicos los usuarios registrados. Se presenta la misma información incluyendo el botón de “Localizar”.

Figura 45. Puntos públicos

Puntos públicos				
	Nombre	Latitud	Longitud	Elevación
	China Queen Zona 9	14.600801	-90.521041	1500 msn
	Obelisco	14.595669	-90.516379	1500 msn
	casa	14.6019225	-90.6005429	1608 msn
	Edificio T3	14.587680874506	-90.5529273301363	1491 msn

Fuente: elaboración propia, con programa Snipping Tool.

4.2. Aplicación móvil

Esta inicia con la ventana de inicio de sesión, ya que el usuario debe estar registrado en el sistema para utilizar la aplicación. Una vez registrado el usuario puede empezar a utilizar la aplicación y generar y consultar puntos de interés.

4.2.1. Autenticando usuarios

Al iniciar la aplicación, en el dispositivo Android, este solicita al usuario iniciar sesión. El usuario debe ingresar un nombre de usuario y contraseña válidos para acceder a la aplicación, ya que para realizar alguna acción se necesita conocer que usuario es.

Figura 46. Autenticación de usuario

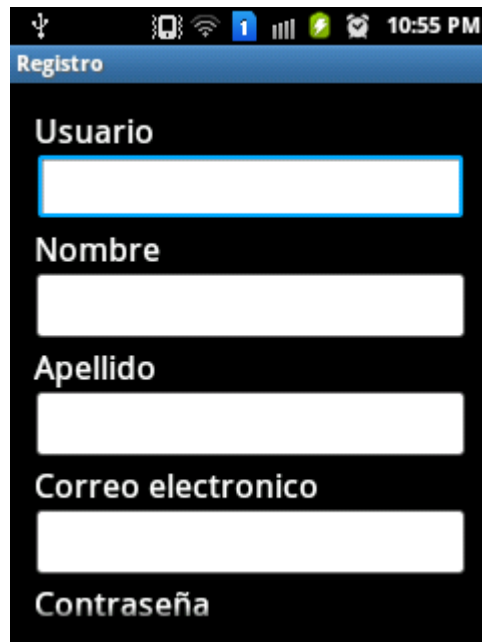


Fuente: elaboración propia, con programa Snipping Tool.

4.2.2. Creación de usuarios

El usuario se puede registrar desde la aplicación móvil, de la misma forma que en la aplicación web y todos los campos son obligatorios, a excepción del sitio web. Una vez ingresados los datos se presiona el botón “Registrar”. Si existiera algún error la aplicación se lo indica.

Figura 47. **Registro de usuarios**



The image shows a mobile application interface for user registration. At the top, there is a status bar with various icons and the time 10:55 PM. Below the status bar is a blue header with the word 'Registro'. The main form area has a black background with white text labels and white input fields. The labels are 'Usuario', 'Nombre', 'Apellido', 'Correo electronico', and 'Contraseña'. Each label is followed by a corresponding white input field. The 'Usuario' field is currently active, indicated by a blue border.

Fuente: elaboración propia, con programa Snipping Tool.

4.2.3. Agregar puntos

El usuario puede agregar puntos de interés desde la aplicación móvil. Para agregar un nuevo lugar tiene que desplegar el menú y presionar el botón "agregar lugar", la aplicación despliega una pantalla donde el usuario debe ingresar el nombre del lugar, seleccionar en una lista de valores el tipo de lugar, si es público o privado, y debe capturar una fotografía. La aplicación se encarga de obtener las coordenadas y se le indica al usuario si pudo agregar el sitio con éxito.

Figura 48. **Agregar puntos**



Fuente: elaboración propia, con programa Snipping Tool.

4.2.4. Modo Realidad Aumentada

Este es el modo principal de la aplicación. Es la pantalla que despliega la aplicación después de iniciar sesión, en ella se visualizan imágenes de los lugares y un radar que indica dónde están los diferentes puntos. El radar indica hacia qué dirección hay que mover el teléfono para visualizar los puntos.

Figura 49. **Modo Realidad Aumentada**



Fuente: elaboración propia, con programa Snipping Tool.

4.2.5. Menú

Al presionar el botón menú, en el dispositivo móvil, la aplicación muestra las opciones fuente de datos, agregar lugar, mapa, zoom, búsqueda e información. A continuación se especifica la funcionalidad de las opciones del menú:

- La opción fuente de datos se seleccionan los puntos públicos o privados.
- La opción agregar lugar abre la pantalla permitiendo al usuario crear un nuevo lugar.
- La opción mapa abre una pantalla donde se agregan los puntos de interés que está viendo el usuario en un mapa.
- La opción zoom se utiliza para que el usuario defina un radio en kilómetros visualizando puntos de interés dentro de ese radio.
- La opción búsqueda es para mostrar solo lugares cuyo nombre coincide con la cadena que el usuario está buscando.

Figura 50. Menú de usuario

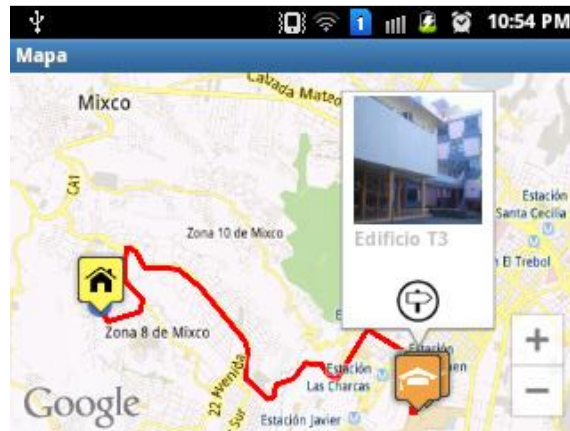


Fuente: elaboración propia, con programa Snipping Tool.

4.2.6. Mapa

En la pantalla de mapa se muestran todos los puntos dentro del radio definido por el usuario. Si el usuario, presiona con el dedo, sobre el punto se despliega una ventana de información en el lugar. Se muestra una foto del lugar y el nombre, para trazar una ruta hacia el lugar, se presiona en la pantalla de información.

Figura 51. Mapa



Fuente: elaboración propia, con programa Snipping Tool.

CONCLUSIONES

1. La evolución en las tecnologías móviles y su creciente demanda permiten, cada día a más usuarios, facilidades como localizar lugares desde su dispositivo móvil.
2. El uso de la localización, en los dispositivos móviles combinado con otras tecnologías como la realidad aumentada, es una solución a la necesidad de la población que busca sitios de interés en el país que por otros medios es difícil encontrar.
3. La tecnología crece y se adapta a las cambiantes necesidades de la sociedad, estableciéndose como un insumo más para la vida diaria.
4. En el mundo existen muchas culturas y agrupaciones de personas, pero hay ciertos conceptos que son constantes a todos. Esto se materializan de distintas formas, en relación a la ubicación geográfica.
5. La cantidad de información, que consume y transmite una persona a sus semejantes, crece constantemente y por esa razón surgen nuevas tecnologías que le agregan significado a las relaciones que forman las personas a través de la internet y las redes sociales.

RECOMENDACIONES

1. Utilizar este trabajo de investigación como referencia para el desarrollo de aplicaciones que usen las mismas tecnologías que se emplearon en el desarrollo del proyecto.
2. Continuar con el desarrollo de la aplicación implementado el uso de las redes sociales como Facebook y Twitter, para ir promoviendo tanto el uso de la aplicación como lugares de importancia en el país.
3. Escalar la aplicación para dispositivos móviles y tabletas, de esta forma la aplicación pueda ser utilizada en las diferentes plataformas que existen.
4. Continuar con el uso de tecnologías, como la realidad aumentada, que tiene múltiples aplicaciones en diferentes campos.

BIBLIOGRAFÍA

1. Android. *API Guides*. [en línea] <<http://developer.android.com/guide/index.html>>. [Consulta: 3 de abril de 2013].
2. Brigham Young University. *Theories Used in IS Research*. [en línea] <http://istheory.byu.edu/wiki/Main_Page>. [Consulta: 18 de marzo de 2013].
3. Google. *Guía para desarrolladores de la versión 2 del API de Google Static Maps*. [en línea] <<https://developers.google.com/maps/documentation/staticmaps/>> [Consulta: 16 de octubre de 2013].
4. Guatemala Superintendencia de Telecomunicaciones. *Crecimiento de telefonía móvil y fija*. [en línea] <http://www.sit.gob.gt/files/8913/9636/9391/crecimiento_de_la_telefonia_fija_y_movil_2do_2013.pdf> [Consulta: 30 de marzo de 2013].
5. GuiaLocal. *Invasión Mobile Latinoamérica 2014*. [en línea] <<http://guialocal.com/blog/es/tag/invasion-mobile-latinoamerica-2014/>>. [Consulta enero 2015].

6. GuiaLocal. *Android, el favorito de Latinoamérica*. [en línea] <<http://guialocal.com/blog/es/2013/12/17/sistemas-operativos-mobile-quien-es-quien-en-latinoamerica/>>. [Consulta enero 2014].
7. OSTANDER, Jason. *Android UI Fundamentals Develop and Design*. California, United States of America: Peachpit Press, 2005. 67 'p.
8. SOOD, Raghav *Pro Android Augmented Reality*. New York, Estados Unidos: Apress, 2002. 90 p.
9. YAGHMOUR, Karim. *Embedded Android*. California, Estados Unidos de America: O'REILLY, 2007. 167 p.
10. Zend. Getting Started with Zend Framework 2. [en línea]. <<http://framework.zend.com/manual/current/en/user-guide/overview.html>>. [Consulta: 05 de octubre de 2013].

APÉNDICES

1. ENCUESTA DE ACEPTACIÓN DE LA TECNOLOGÍA DE REALIDAD AUMENTADA

1. ¿Cuenta con un teléfono inteligente (smartphone)?
2. ¿Qué sistema operativo tiene su teléfono inteligente (smartphone)?
3. ¿Sabe qué es la Realidad Aumentada en teléfonos inteligentes?
4. ¿Ha utilizado alguna aplicación con Realidad Aumentada en su teléfono?
5. ¿Ha utilizado alguna vez una aplicación en su teléfono para localizar algún lugar de interés (restaurante, banco, centro comercial, y otros.)?
6. ¿Le gustaría utilizar una aplicación para encontrar lugares turísticos del país?
7. ¿Le gustaría visualizar por medio de Realidad Aumentada lugares de interés en el país?
8. ¿Le gustaría poder agregar sus propios lugares de interés en el país desde una aplicación móvil y web?

2. RESULTADOS TABULADOS DE LA ENCUESTA 1

¿Cuenta con un teléfono inteligente (smartphone)?	¿Qué Sistema Operativo tiene su teléfono inteligente (smartphone)?	¿Sabe qué es la Realidad Aumentada en teléfonos inteligentes?	¿Ha utilizado alguna aplicación con Realidad Aumentada en su teléfono?	¿Ha utilizado alguna vez una aplicación en su teléfono para localizar algún lugar de interés (restaurante, banco, centro comercial, etc.)?	¿Le gustaría utilizar una aplicación para encontrar lugares turísticos del país?	¿Le gustaría visualizar por medio de Realidad Aumentada lugares de interés en el país?	¿Le gustaría poder agregar sus propios lugares de interés en el país desde una aplicación móvil y web?
Sí	Android (Cualquier versión)	Sí	Sí	Sí	Sí	Sí	Sí
Sí	iOS (iPhone, cualquier modelo)	Sí	Sí	Sí	Sí	Sí	NO
Sí	Android (Cualquier versión)	No	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	No	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	No	No	No	Sí	Sí	Sí
Sí	Windows	Sí	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	Sí	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	No	No	Sí	Sí	No	Sí
Sí	Android (Cualquier versión)	No	Sí	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	Sí	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	No	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	Sí	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	Sí	Sí	Sí	Sí	NO
Sí	Android (Cualquier versión)	Sí	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	No	Sí	Sí	Sí	Sí
Sí	iOS (iPhone, cualquier modelo)	Sí	Sí	Sí	Sí	Sí	Sí
No		No	No	No	No	No	NO
Sí	Windows	Sí	Sí	Sí	Sí	Sí	Sí
Sí	Windows	Sí		Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	No	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	No	No	Sí	Sí	Sí	Sí
Sí	iOS (iPhone, cualquier modelo)	No	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	No	No	Sí	Sí	Sí	Sí
No	Otro	No	No	No	Sí	No	Sí
Sí	Android (Cualquier versión)	No	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	No	No	Sí	Sí	Sí
Sí	iOS (iPhone, cualquier modelo)	Sí	Sí	Sí	Sí	No	Sí
Sí	Android (Cualquier versión)	Sí	Sí	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	No	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	Sí	No	Sí	No	Sí
Sí	Android (Cualquier versión)	No	No	Sí	Sí	Sí	Sí
Sí	iOS (iPhone, cualquier modelo)	Sí	No	Sí	Sí	Sí	Sí
Sí	Android (Cualquier versión)	Sí	No	No	No	No	NO
Sí	iOS (iPhone, cualquier modelo)	Sí	Sí	Sí	Sí	No	NO
Sí	Android (Cualquier versión)	Sí	Sí	Sí	No	Sí	NO