

**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**TESIS DE GRADO**

**“GEOREFERENCIACIÓN DE  
CAJEROS AUTOMÁTICOS Y AGENCIAS BANCARIAS  
MEDIANTE APLICACIÓN ANDROID”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

**POSTULANTE : JUAN CARLOS CHUQUIMIA CONDORI**  
**TUTOR METODOLÓGICO : Lic. FREDDY MIGUEL TOLEDO PAZ**  
**ASESOR : M. Sc. CARLOS MULLISACA CHOQUE**

**LA PAZ – BOLIVIA**

**2015**



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON EstrictAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

## DEDICATORIA

*Quiero dedicar este trabajo en primer lugar a Dios, por ser quien guía mi vida, por darme la fortaleza y así lograr alcanzar esta meta.*

*A mis padres Félix Chuquimia Fabián y Eusebia Condori Mamani, a quienes debo mi vida, por el cariño, paciencia y comprensión que me han tenido, los admiro y los amo con todo mi corazón, gracias por su apoyo, por su amor, por su esfuerzo y por sus ánimos.*

*A mis hermanos, tíos, primos, cuñados y sobrinos, por su constante motivación y los buenos ratos en su compañía, que ciertamente me han permitido superar todos los obstáculos.*

*A mis amigos y compañeros por compartir los buenos y malos momentos, y sobre todo los conocimientos compartidos.*

## AGRADECIMIENTOS

*Un agradecimiento muy especial a mi Tutor Metodológico Lic. Freddy Miguel Toledo Paz, por toda su sabiduría y experiencia profesional, respaldándome durante la elaboración y la culminación de la misma.*

*De la misma manera un agradecimiento sincero a mi Asesor M.Sc. Carlos Mullisaca Choque, por guiarme en las diferentes etapas de mi tesis, por su asesoramiento, colaboración y consejos, pero sobre todo por su paciencia y tiempo.*

*También agradezco a todos los Docentes de la Carrera de Informática por los conocimientos transmitidos durante todo este tiempo de formación en la Universidad.*

## **RESUMEN**

El trabajo de grado “GEOREFERENCIACIÓN DE CAJEROS AUTOMÁTICOS Y AGENCIAS BANCARIAS MEDIANTE APLICACIÓN ANDROID”, se fundamenta en desarrollar un prototipo el cual será usado en su mayoría por personas que dependen de cuentas bancarias, los mismos que realizan transacciones asistiendo a una sucursal o mediante el uso de cajeros automáticos (ATM), permitiéndoles geolocalizar estos centros en cualquier región y localidad de Bolivia.

El prototipo será desarrollado para dispositivos móviles que cuentan con una plataforma Android, debido a que en su gran mayoría utiliza este sistema, el mismo permitirá un acercamiento entre las personas y estos centros de transacción, aprovechando la portabilidad y movilidad de las personas para usar esta aplicación en cualquier momento y lugar, dando la oportunidad de poder consultar sobre los Cajeros Automáticos y Agencias de Banco más próximos que se encuentran alrededor de los usuarios, cuando ya se encuentren en una zona específica.

La aplicación se desarrollara de tal manera que sea utilizable tanto en Smarthphones como en Tablets, manteniendo la usabilidad y la estética para las diferentes posibilidades de resoluciones y tamaños de pantallas, para lograr abarcar un mayor número de posibles usuarios.

## **ABSTRACT**

The degree work "GEOREFERENCING ATM AND AGENTS BANK BY APPLICATION ANDROID" is based on developing a prototype which will be used mostly by people who depend on bank accounts, the same transacting attending a branch or by use of automated teller machines (ATM), allowing these centers to geotag any region and town of Bolivia.

The prototype will be developed for mobile devices with an Android platform, because they mostly use the system, it will allow a rapprochement between people and these trading centers, taking advantage of the portability and mobility of people to use application at any time and place, giving the opportunity to consult on ATMs and Agencies closest Bank found around users, when they are in a specific area.

The application is developed in a way that is usable both Smarthphones as Tablets, keeping the usability and aesthetic possibilities for different resolutions and screen sizes, in order to cover a wider range of potential users.

# ÍNDICE

## CAPÍTULO 1

<b>MARCO INTRODUCTORIO</b> .....	1
1.1. INTRODUCCIÓN.....	2
1.2. ANTECEDENTES.....	3
1.3. EL PROBLEMA.....	5
1.3.1. Planteamiento del Problema.....	5
1.3.2. Problemas Específicos.....	5
1.3.3. Formulación del Problema.....	6
1.4. OBJETIVOS.....	6
1.4.1. Objetivo General.....	6
1.4.2. Objetivos Específicos.....	6
1.5. HIPÓTESIS.....	7
1.6. JUSTIFICACIÓN.....	7
1.6.1. Justificación Social.....	7
1.6.2. Justificación Técnica.....	7
1.6.3. Justificación Económica.....	8
1.7. IDENTIFICACION DE VARIABLES.....	8
1.7.1. Variable Independiente.....	8
1.7.2. Variable Dependiente.....	8
1.8. LIMITES Y ALCANCES.....	8
1.8.1. Límites.....	8
1.8.2. Alcances.....	8
1.9. APORTES.....	9
1.10. METODOLOGIA DE INVESTIGACION.....	9
1.10.1. Recolección de Información.....	9
1.10.2. Tipo de Investigación.....	9
1.10.2.1. Investigación Aplicada.....	10
1.10.2.2. Demostración (Factibilidad del proyecto).....	10

## CAPÍTULO 2

<b>MARCO TEÓRICO</b> .....	11
<b>2.1. GEORREFERENCIACIÓN</b> .....	12
2.1.1. Definición.....	12
2.1.2. Aplicaciones tecnológicas.....	12
2.1.3. Sistemas de localización de dispositivos móviles.....	12
2.1.4. Sistemas de localización basados en la identidad celular.....	14
2.1.4.1. Identidad Celular Global (CGI).....	14
2.1.4.2. Identidad Celular Perfeccionada (CGI-TA).....	15
2.1.5. Sistemas de localización basados en la red.....	15
2.1.5.1. Ángulo de llegada (AOA, DOA).....	15
2.1.5.2. Tiempo de llegada (TOA).....	16
2.1.5.3. Diferencia en el tiempo de llegada (TDOA).....	16
2.1.5.4. Huella multitrayecto.....	17
2.1.6. Técnicas basadas en la modificación del terminal móvil.....	17
2.1.6.1. Diferencia en el tiempo de llegada con terminal modificado....	17
2.1.6.2. Diferencia en el tiempo de llegada perfeccionada (E-ODT).....	17
2.1.6.3. Triangulación avanzada de enlace hacia delante (A-FTL).....	18
2.1.6.4. Global Positioning System (GPS).....	18
2.1.6.5. Integración con Telefonía Móvil.....	20
2.1.6.6. GPS actuales.....	20
2.1.6.7. Arquitectura GPS.....	22
2.1.6.8. A-GPS.....	24
2.1.7. API's de Geolocalización.....	25
2.1.7.1. Location API for JAVA (OpenLAPI).....	25
2.1.7.2. Google Gear API.....	26
2.1.7.3. Android Location Services.....	26
2.1.8. Trust Managment Service (TMS).....	27
2.1.9. Aplicaciones actuales con esta tecnología.....	27



2.1.9.1. Sky Map.....	27
2.1.9.2. Places Directory.....	27
2.1.9.3. Search with my Location.....	28
2.1.10. Google Maps.....	28
2.1.10.1. Ventajas de Google Maps.....	29
2.1.10.2. Las API's de Google Maps Versión 3.....	30
2.2. COORDENADAS GEOGRÁFICAS.....	31
2.2.1. Latitud.....	31
2.2.2. Longitud.....	31
2.2.3. Sistemas de Coordenadas.....	32
2.2.4. Geocodificación.....	32
2.3. CONCEPTOS MATEMATICOS.....	32
2.3.1. Sistemas de representación.....	32
2.3.2. Representación esférica.....	33
2.3.3. Representación elipsoidal.....	34
2.3.4. Distancia Euclidiana.....	34
2.3.5. Fórmula de Haversine.....	34
2.3.6. Ley de Haversine.....	35
2.4. DISPOSITIVOS MOVILES.....	36
2.4.1. Arquitectura de sistemas móviles.....	37
2.4.2. Arquitectura del Hardware.....	38
2.5. PLATAFORMA ANDROID.....	39
2.5.1. Características y ventajas de Android.....	39
2.5.2. Arquitectura Android.....	41
2.5.3. Como trabaja Android.....	43
2.5.4. Life Cycle (Ciclo de Vida de una Actividad).....	44
2.6. HERRAMIENTAS Y TECNOLOGÍAS.....	46
2.6.1. Eclipse IDE+ADT plugin+SDK android.....	46
2.6.2. Adobe Photoshop CS5.....	47

2.6.3.	Json.....	47
2.6.4.	Web Service.....	48
2.6.4.1.	Estándares empleados.....	48
2.6.4.2.	Ventajas de los Servicios Web.....	48
2.6.4.3.	Razones para crear Servicios Web.....	49
2.7.	LENGUAJES DE PROGRAMACION.....	49
2.7.1.	JAVA.....	49
2.7.2.	XML.....	50
2.7.3.	PHP.....	50
2.7.4.	HTML5.....	51
2.7.5.	Javascript.....	51
2.8.	GESTORES DE BASES DE DATOS.....	52
2.8.1.	Definición de Base de Datos.....	52
2.8.2.	Sistemas de Gestión de Base de Datos (SGBD).....	53
2.8.2.1.	MySQL.....	53
2.8.2.2.	SQLite.....	54
2.9.	METODOLOGIA Y TECNICAS DE DESARROLLO.....	56
2.9.1.	Metodología para el Desarrollo de Aplicaciones Móviles (MDAM).....	56
2.9.2.	Fases de la metodología (MDAM).....	58
1)	FASE DE ANÁLISIS.....	58
2)	FASE DE DISEÑO.....	60
3)	FASE DE DESARROLLO.....	62
4)	FASE DE PRUEBA DE FUNCIONAMIENTO.....	62
5)	FASE DE ENTREGA.....	63
2.9.3.	Lenguaje Unificado de Modelado (UML).....	64
2.9.3.1.	Beneficios de UML.....	65
2.9.3.2.	UML ¿Método o Lenguaje de Modelado?.....	65
2.10.	MÉTRICAS DE CALIDAD DEL SOFTWARE.....	65
2.10.1.	Funcionalidad.....	66
2.10.2.	Usabilidad.....	67

2.10.3. Portabilidad.....	68
2.10.4. Mantenibilidad.....	70

### **CAPÍTULO 3**

<b>MARCO APLICATIVO.....</b>	<b>71</b>
3.1. INTRODUCCION.....	72
3.2. ANÁLISIS METODÓLOGICO.....	72
3.3. METODOLOGIA DE DESARROLLO.....	72
3.4. FASE DE ANALISIS.....	73
3.4.1. Definición de Actores.....	73
3.4.2. Requerimientos.....	73
3.4.2.1. Requerimientos Funcionales.....	74
3.4.2.2. Requerimientos No Funcionales.....	75
3.5. FASE DE DISEÑO.....	75
3.5.1. Estructura del Sistema.....	75
3.5.2. Diseño de Navegación.....	77
3.5.3. Diagramas de Casos de Uso y Especificaciones.....	77
3.5.4. Diagramas de Secuencia.....	86
3.5.5. Diagrama de clases.....	91
3.5.6. Estructura de la Base de Datos.....	93
3.5.7. Diseño de Interfaces.....	95
3.5.8. Interfaz del usuario.....	96
3.6. FASE DE DESARROLLO.....	98
3.6.1. Implementación Web.....	98
3.6.2. Implementación Aplicación Android.....	104
3.7. FASE DE PRUEBAS DE FUNCIONAMIENTO.....	108
3.7.1. Visualización de interfaces.....	108
3.8. FASE DE ENTREGA.....	113
3.9. CALIDAD DEL SOFTWARE.....	113

## **CAPÍTULO 4**

<b>PRUEBA DE HIPOTESIS</b> .....	104
4.1. PLANTEAMIENTO DE LA HIPOTESIS NULA Y ALTERNATIVA.....	115
4.1.1. Hipótesis Nula ( $H_0$ ).....	115
4.1.2. Hipótesis Alternativa ( $H_1$ ).....	115
4.2. ESTADISTICO DE PRUEBA.....	115
4.2.1. Frecuencias Observadas.....	116
4.2.2. Tabla de contingencia para las probabilidades.....	117
4.2.3. Tabla de Frecuencias Esperadas.....	117
4.2.4. Nivel de Significación.....	118
4.2.5. Nivel de Confianza.....	118
4.2.6. Estadístico Chi cuadrado.....	118
4.2.7. Valor Crítico.....	118
4.2.8. Valor Empírico.....	119
4.3. DECISION.....	120
4.4. CONCLUSION.....	120

## **CAPÍTULO 5**

<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	121
5.1. CONCLUSIONES.....	122
5.2. RECOMENDACIONES.....	122
<b>BIBLIOGRAFÍA</b> .....	124
- Referencias Bibliográficas.....	125
- Referencias Electrónicas.....	125
<b>ANEXOS</b> .....	127
- Anexo A: Código Fuente Servicio Web.....	128
- Anexo B: Código Fuente de la Aplicación.....	133
<b>DOCUMENTOS</b> .....	135

# ÍNDICE DE FIGURAS

Figura 2.1: Georreferenciación, geoposicionamiento y geolocalización.....	12
Figura 2.2: Estructura de red en celdas.....	14
Figura 2.3: Sistemas de localización por identidad celular.....	15
Figura 2.4: Localización por ángulo de llegada.....	16
Figura 2.5: Localización GPS.....	19
Figura 2.6: GPS en Telefonía Móvil.....	21
Figura 2.7: Arquitectura GPS.....	22
Figura 2.8: Elipsoide terrestre.....	33
Figura 2.9: Triangulo esférico de haversine.....	36
Figura 2.10: Arquitectura de hardware de un dispositivo móvil moderno.....	39
Figura 2.11: Arquitectura de Android.....	41
Figura 2.12: Ciclo de vida de una Actividad .....	44
Figura 2.13: Etapas de la Metodología para el Desarrollo de Aplicaciones Móviles...	58
Figura 2.14: Posibles diagramas para el desarrollo de aplicaciones móviles.....	61
Figura 3.1: Estructura del Sistema.....	76
Figura 3.2: Mapa de Navegación.....	77
Figura 3.3: Caso de uso “ingreso a la aplicación”.....	78
Figura 3.4: Caso de uso “búsqueda de sucursales”.....	79
Figura 3.5: Caso de uso “ver más”.....	80
Figura 3.6: Caso de uso “ingreso a la plataforma de administración”.....	81
Figura 3.7: Caso de uso “agregar sucursal”.....	82

Figura 3.8: Caso de uso “actualizar información” ..	83
Figura 3.9: Caso de uso “eliminar sucursal” ..	85
Figura 3.10: Secuencia “ingreso a la aplicación” ..	86
Figura 3.11: Secuencia “búsqueda de sucursales” ..	87
Figura 3.12: Secuencia “ver más” ..	88
Figura 3.13: Secuencia “ingreso administrador” ..	89
Figura 3.14: Secuencia “agregar sucursal” ..	90
Figura 3.15: Diagrama de clases de la interfaz de la aplicación ..	91
Figura 3.16: Esquema de la Base de datos ..	94
Figura 3.17: Pantalla de inicio de la aplicación ..	96
Figura 3.18: Guía visual de eventos de configuración ..	97
Figura 3.19: Guía visual de eventos de búsqueda y créditos ..	97
Figura 3.20: Diagrama de clases servicio web ..	100
Figura 3.21: Pantalla de inicio de sesión del administrador ..	109
Figura 3.22: Pantalla de error de acceso (usuario y/o incorrectos) ..	109
Figura 3.23: Pantalla con vista satelital ..	110
Figura 3.24: Pantalla con vista de mapa ..	110
Figura 3.25: Pantalla de agregar nuevas sucursales ..	111
Figura 3.26: Pantalla de edición de sucursales ..	111
Figura 3.27: Pantalla de trazado de ruta ..	112
Figura 3.28: Pantalla de navegación de la ruta ..	112
Figura 4.1: Tabla de Chi cuadrada ..	119
Figura 4.2: Gráfica de decisión en Winstats ..	120

# ÍNDICE DE TABLAS

Tabla 2.1: Ventajas de Google Maps.....	30
Tabla 2.2: Arquitectura de Sistema Móviles.....	37
Tabla 2.3: Métricas de Adecuidad.....	67
Tabla 2.4: Métricas de Entendibilidad.....	68
Tabla 2.5: Métricas de Portabilidad.....	69
Tabla 3.1: Definición de Actores.....	73
Tabla 3.2: Requerimientos Funcionales.....	74
Tabla 3.3: Requerimientos No Funcionales.....	75
Tabla 3.4: Especificación caso de uso “ingreso a la aplicación”.....	78
Tabla 3.5: Especificación caso de uso “búsqueda de sucursales”.....	79
Tabla 3.6: Especificación de uso “Ver más”.....	80
Tabla 3.7: Especificación caso de uso “ingreso administrador”.....	81
Tabla 3.8: Especificación del caso de uso “agregar sucursal”.....	82
Tabla 3.9: Especificación caso de uso “actualizar información”.....	84
Tabla 3.10: Especificación caso de uso “eliminar marcador”.....	85
Tabla 4.1: Tabla de contingencia de frecuencias observadas.....	116
Tabla 4.2: Tabla de contingencia para las probabilidades.....	117
Tabla 4.3: Tabla de Frecuencias esperadas.....	117
Tabla 4.4: Tabla de valor empírico.....	119

# **CAPÍTULO 1**

## **MARCO INTRODUCTORIO**

---



## 1.1. INTRODUCCIÓN

En los últimos años, las nuevas tecnologías en dispositivos móviles han avanzado hasta un nivel en el que se puede disponer en todo momento de aplicaciones capaces de gestionar grandes cantidades de información y de realizar costosas tareas y operaciones.

La disponibilidad de esta tecnología en dispositivos como teléfonos móviles o tablets, nos abre un mar de oportunidades a aplicaciones que podemos usar o necesitar en cualquier momento o situación en el día a día. Si a ello le añadimos que hoy en día es posible conocer nuestra propia posición en tiempo real, se da la posibilidad de desarrollar una herramienta capaz de conocer, analizar y almacenar información sobre lo que nos rodea.

El objetivo básico y principal del presente trabajo, es desarrollar una aplicación que conozca nuestras preferencias, nuestra posición, y que nos ayude a encontrar con facilidad en cualquier región y localidad de Bolivia sucursales tales como cajeros automáticos y agencias bancarias donde se puedan realizar transacciones financieras, los mismos brinden información básica y necesaria para acudir o geolocalizar otro centro asequible en el momento que lo necesitemos.

Para ello se decidió utilizar dispositivos móviles, dada su gran disponibilidad por parte de la mayoría de personas, y el avance tecnológico cada vez más grande de estos. En cuanto a los sistemas que disponen los dispositivos móviles se decidió usar Android por la facilidad de acceso al SDK de desarrollo, aparte de su apoyo por el software libre. El hecho de que Android esté desarrollado sobre código abierto, lo ha impulsado fuertemente consiguiendo ocupar una de las posiciones más vendidas, y también ha impulsado a muchos otros desarrolladores de software libre a ofrecer su código de aplicaciones Android de manera abierta.

La tecnología del reconocimiento de marcadores, parecidos a los códigos de barras pero con la característica de poder contener mucha más información, y con los que se pueden identificar a los cajeros automáticos y agencias bancarias que se encuentran a nuestro alrededor y disponen de estos marcadores.

El uso de memoria interna y posicionamiento nos permite también utilizar mapas en modo offline en los que se muestra la posición del dispositivo, y la posición de aquellos centros de transacciones que son de nuestro interés.

## 1.2. ANTECEDENTES

En el país, el crecimiento poblacional hace que las regiones y localidades se expandan con nuevas urbanizaciones, esto requiere la implementación de nuevos servicios financieros de modo que beneficien a la mayor cantidad de las personas en sus respectivas regiones y localidades.

Las entidades financieras, cuyo principal compromiso con la sociedad está relacionado con la excelencia en la prestación de sus servicios, evalúan y revisan, constantemente los efectos causados por los avances tecnológicos y el impacto que producen en la operación bancaria, por ello toda entidad financiera adoptan nuevos mecanismos para la realización de transacción para con sus clientes, como son la puesta de sucursales bancarias y los cajeros automáticos en lugares estratégicos, así también algunos de ellos ya cuentan con aplicaciones de banca virtual.

Google Maps pone a disposición de los desarrolladores sus API's los cuales brindan soporte de localización geográfica y funciones propias dentro de un entorno de trabajo personalizable, sobre el cual ubicar nuestra aplicación, sea cual sea su finalidad y utilidad.

Entre algunas aplicaciones que se destacan en Play Store son las siguientes:

- a) ***TripAdvisor***: Aplicación siempre muy nombrada especialmente en viajes, playa, hoteles, entre otras, y es que por algo se caracteriza *TripAdvisor* es por ser de las más completas. Nos sacará de un apuro tanto si estamos de viaje como si queremos conocer los comentarios de los sitios, pero también destaca por su función para descubrir lo que tenemos cerca nuestro, queremos descubrir que hay más allá.
- b) ***Foursquare***: Un clásico, *la red social geolocalizada por excelencia*. Su aplicación para Android está muy cuidada, más últimamente que está efectuando varios movimientos para renovar su imagen. Empezando por el logo y acabando por Swarm, la aplicación centrada en checkins. Sus mejores armas son la gran base de usuarios, los

sitios, comentarios, alianza con otras marcas como Telepizza y su sistema de badges que recompensan y animan a participar. Sin duda el rival a batir y uno de nuestros preferidos.

- c) **Tinder:** No todo tiene que ir de descubrir lugares, también pueden ser personas. Y aquí Tinder, una de las últimas aplicaciones de moda, es la ganadora. Depende de donde estemos podemos darle a buscar para *encontrar personas afines cercanas*, una utilidad de la geolocalización que Tinder aprovecha para darle el mejor enfoque comercialmente hablando.
- d) **Meetup:** Otra aplicación genial para descubrir lo que te rodea. Esta vez es Meetup y está *centrada en eventos, reuniones y grupos*. Gracias a la aplicación de Android podremos explorar nuestro alrededor y aprender idiomas, unirse a un club de lectura o hacer ejercicio con otra gente. Hay meetups por todas partes.
- e) **Groupon:** Si lo que quieres son ofertas, entonces el campeón en esta categoría es Groupon. Dale a buscar alrededor y te mostrará en un mapa *ofertas activas*. Ahora viajando de un lado a otro, aunque eso sí, hay que tener en cuenta que las ofertas que ofrecen suelen ser por tiempo limitado. Una aplicación perfecta para dejar nuestra ubicación activa.
- f) **Minube:** Otro servicio parecido a 11870.com. empezó siendo un buscador de vuelos para después expandir sus servicios en la búsqueda de sitios que visitar. Como la mayoría de este tipo de herramientas, también se aprovecha de la ubicación para ofrecer resultados geolocalizados.
- g) **Banjo:** Es una aplicación muy lograda centrada en noticias y actualizaciones. Muchas veces vemos las noticias clasificadas por categorías, pero no por su cercanía a nosotros. Esta aplicación quiere resolver este problema, aunque de momento está muy centrada en los Estados Unidos. Se buscó algún servicio similar en español, pero no tenemos ninguno con número de usuarios suficiente.
- h) **Android Device Manager:** Si hablamos de geolocalización no podemos olvidar el Android Device Manager. *¿Qué es lo busca? Pues nuestro móvil.* Tan simple y potente

como eso. La aplicación funciona a la perfección y nos permitirá además de saber la posición, bloquearlo, hacer que suene o borrarlo en caso necesario.

La geolocalización y la ubicación juegan un papel clave en muchas Aplicaciones para Android pero confiamos en que el auge de sistemas como Android Wear (movilidad), Android Auto (carretera) y Android TV (hogar) la localización siga teniendo ese rol tan importante en Android.

### **1.3. EL PROBLEMA**

El mundo ha evolucionado, de la misma forma ha ocurrido con las personas, las empresas y con ello las agencias financieras con sus diversas sucursales de cajeros automáticos y agencias bancarias, estos últimos años han ido cambiando la manera de hacer las cosas, buscando siempre la eficiencia y la eficacia de sus procesos, esto implica acortar los tiempos de las actividades, para ello es preciso tener la información siempre a la mano.

#### **1.3.1. Planteamiento del Problema**

Personas que dependen de una o varias cuentas bancarias en su mayoría cuentan con sus respectivas tarjetas de débito para realizar transacciones financieras en cajeros automáticos, en cualquier momento y lugar, pero muchas veces los usuarios no siempre conocen las localizaciones de las sucursales de su preferencia, muchas veces siempre acude a ciertos cajeros automáticos y agencias bancarias en lugares céntricos de las ciudades, desconociendo otras sucursales más cercanas y con poca afluencia de usuarios, siendo esto un gasto de transporte y pérdida de tiempo de las personas que quieran realizar transacciones financieras.

#### **1.3.2. Problemas Específicos**

- Personas que realizan, viajes a distintas regiones del país no conocen siempre los centros para la realización de una transacción para ello solo recurren a preguntar a transeúntes sobre la ubicación de cajeros y agencias de banco.

- En cuestiones de emergencia puede recorrer una distancia considerable para ubicar un cajero automático, no sabiendo que a una cuadra se encuentra un cajero de su preferencia.
- Al querer realizar una transacción en un cajero que no pertenece a su banco por alguna necesidad de urgencia, el usuario desconoce la información sobre las comisiones que se debitará al realizar la transacción.
- Existen algunos cajeros que solo funcionan en horarios específicos, y muchos usuarios suelen recurrir a distintos cajeros para realizar una transacción, y suelen encontrarse que el cajero tiene horarios de funcionamiento.

### **1.3.3. Formulación del problema**

¿Por qué recorrer largas distancias en busca de una sucursal bancaria o cajero automático, que pueden estar a la vuelta de la esquina a las cuales no accedemos solo por desconocimiento?

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo General**

Desarrollar una aplicación para dispositivos móviles con sistema Android que utilicen las tecnologías y servicios de localización de Google Maps para ubicar y encontrar cajeros automáticos y agencias bancarias más próximos de su posición, provistos de información necesaria e importante, mediante la georeferenciación.

### **1.4.2. Objetivos Específicos**

- Realizar un análisis de las aplicaciones actuales que se puedan considerar competencia, afines a la problemática mencionada.
- Analizar y seleccionar la plataforma en la cual se desarrollará la aplicación móvil y el servicio web.



- Diseñar la base de datos en la que estará soportada toda la información brindada por las entidades financieras.
- Integrar un servicio de mapas a la plataforma de desarrollo.
- Desarrollar módulos del prototipo en la plataforma seleccionada.
- Desarrollar la interfaz de la aplicación móvil.
- Validar la usabilidad y rendimiento del prototipo de la aplicación.

## **1.5. HIPÓTESIS**

La implementación del sistema Georreferenciación de cajeros automáticos y agencias bancarias mediante aplicación android, responde a las necesidades de obtener la ubicación geográfica de manera ágil y precisa las sucursales bancarias y cajeros automáticos.

## **1.6. JUSTIFICACIÓN**

### **1.6.1. Justificación Social**

Mucha gente con esta aplicación podrá tener la certeza encontrar una sucursal de cajero automático y agencia bancaria en cualquier lugar del territorio nacional haciendo consultas sobre las mismas cualquier momento, ahorrándose tiempo y recursos en tratar de buscar sucursales de su preferencia.

### **1.6.2. Justificación Técnica**

Se justifica tecnológicamente con la elección de la plataforma Android, por expandirse exponencialmente en los últimos años, siendo Android una de las plataformas más usadas por la gente, basándose en un lenguaje de programación Java, introduciendo el concepto de geolocalización que es muy importante en la actualidad, teniendo a nuestra disposición los mapas de google.

### **1.6.3. Justificación Económica**

Las herramientas a utilizar son de costo cero de Licenciamiento por ser Android de código abierto, dado que los usuarios pueden adquirirlo de manera gratuita, en tanto quienes generan utilidades son las empresas proveedoras de internet y telecomunicaciones por los servicios de conexión al consumo de datos.

## **1.7. IDENTIFICACIÓN DE VARIABLES**

### **1.7.1. Variable Independiente**

*V.I. (x)*: El Diseño y la implementación de un sistema georreferenciación de cajeros automáticos y agencias bancarias mediante aplicación android.

### **1.7.2. Variable Dependiente**

*V.D. (y)*: Responde a las necesidades de obtener la ubicación geográfica de manera ágil y precisa las sucursales bancarias y cajeros automáticos.

## **1.8. LÍMITES Y ALCANCES**

### **1.8.1. Limites**

- La aplicación propuesta está diseñada para el sector financiero regularizados por la Autoridad de Supervisión del Sistema Financiero (ASFI).
- El funcionamiento está garantizado en los dispositivos que cuenten con los servicios de GPS y conexiones por redes móviles o redes inalámbricas WiFi.

### **1.8.2. Alcances**

- El presente proyecto pretende generar un prototipo funcional de la aplicación de georreferenciación de cajeros automáticos y agencias bancarias para la plataforma Android.
- Con la interacción de los mapas de Google Maps y sus constantes actualizaciones se muestran algunas sucursales representativas.

- La aplicación es compatible con las versiones de Android 2.3.6 o superiores a esta.

## **1.9. APORTES**

- La presente investigación servirá a las próximas generaciones como una base de conocimiento hacia las nuevas tendencias de la tecnología móvil de geolocalización, asimismo ilustrará como los móviles pueden ayudar a las personas a encontrar lugares de su interés.
- Esta tesis se basará en la Metodología para el Desarrollo de Aplicaciones Móviles (MDAM) para su desarrollo, método que se basa en la conceptualización de las tecnologías y las metodologías ágiles para el desarrollo de aplicaciones móviles, y su objetivo principal es facilitar la creación de nuevas aplicaciones y servicios exitosos.

## **1.10. METODOLOGÍA DE INVESTIGACIÓN**

Para el desarrollo del proyecto, la metodología de a seguir, incluirá los siguientes puntos:

### **1.10.1. Recolección de Información**

Para conocer las principales características acerca de los mapas de Google Maps, se ha buscado la información apropiada desde el sitio web de Google y así estar al tanto de los beneficios que ofrece a las diferentes aplicaciones para dispositivos móviles que utilizan los servidores Google Maps, además de conocer los procesos de utilización de los API's de Google Maps (sintaxis y utilización).

### **1.10.2. Tipo de investigación.**

Para llevar a cabo esta investigación, se ha tomado como referencia la investigación científica, en la cual se tendrá presente cuatro elementos principales en el proceso de desarrollo de la misma, los cuales son mencionados a continuación:

- **Sujeto:** El investigador del presente proyecto.
- **Objeto:** Desarrollo de una aplicación “Georreferenciación de Cajeros Automáticos y Agencias Bancarias Mediante Aplicación Android”.



- **Medio:** Servicios de Google Maps, Sistema de desarrollo para aplicaciones Android y API's; los mismo que permitirán visualizar las sucursales de cajeros automáticos y agencias bancarias en los mapas de Google Maps a partir de información de una Base de datos en producción, los mismos tendrán un mejor análisis ya que tendremos imágenes satelitales.
- **Fin:** Demostrar la versatilidad y funcionalidad que brindan las aplicaciones móviles de geolocalización, a los diferentes usuarios y las imágenes de los mapas de Google Maps al mostrar cualquier tipo de punto geográfico.

Dentro de este tipo de investigación encontramos la investigación aplicada que es la que se ha utilizado en el proceso de desarrollo de presente trabajo.

#### **1.10.2.1. Investigación Aplicada**

Se ha tomado este tipo de investigación, ya que se desarrolla una propuesta viable aprovechando la aportación de la teoría encontrada con respecto al tema investigado llevándolo a la práctica, en la cual se obtiene como resultado una aplicación para georreferenciar cajeros automáticos y agencias bancarias, a los diferentes usuarios los mismos que pueden tener acceso a la información de manera rápida y eficaz.

#### **1.10.2.2. Demostración (Factibilidad del Proyecto)**

El proyecto se basa en la georreferenciación de cajeros automáticos y agencias bancarias, el mismo que se declara como un proyecto factible, es decir que se puede aplicar a muchas áreas según lo que se pretende demostrar.

En el proceso del desarrollo de la aplicación para demostrar la funcionalidad que poseen los mapas de Google, se cuenta con la información recopilada de las fuentes de información que en este caso han sido el sitio web de Google, además se han utilizado las herramientas apropiadas para el desarrollo de la aplicación.



## **CAPÍTULO 2**

# **MARCO TEÓRICO**

---

# MARCO TEÓRICO

## 2.1. GEORREFERENCIACIÓN

### 2.1.1. Definición

Se entiende por Georreferenciación o Geolocalización, a la identificación de la posición geográfica real de un objeto o persona, ya sea un dispositivo conectado a Internet, un teléfono móvil o cualquier otro aparato que sea posible rastrear. Dicha localización puede ser en un plano de dos dimensiones (por ejemplo, Google Maps), como en un plano de tres dimensiones (GPS). En los últimos años, diferentes tipos de tecnologías han apostado por la Georreferenciación, siendo extraordinario el auge de esta en las tecnologías móviles de última generación.



Figura 2.1: Georreferenciación, geoposicionamiento y geolocalización

Fuente: <http://nuevastecsomamfyc.files.wordpress.com>

Actualmente es posible el uso de la Geolocalización en la mayoría de plataformas:

- En el caso de la Geolocalización de un ordenador, esta se hace a través de una serie de bases de datos que “aproximan” la zona en la que el usuario se encuentra.
- En el caso de un dispositivo Móvil, existen diferentes tecnologías actualmente, como Sistema de Posicionamiento Global (GPS) o la localización por celdas.

### 2.1.2. Aplicaciones tecnológicas

El conocimiento de la posición de un aparato electrónico puede definir la posición de una persona física en el mundo, lo cual tiene una infinidad de posibilidades.

Con la aplicación de geolocalización o en un PC de escritorio, podemos definir diferentes parámetros sobre el usuario que accede a un sistema a través del ordenador, como por ejemplo el idioma del usuario, publicidad de empresas cercanas, zona horaria, etc. También se han encontrado sistemas de Hacking de Ingeniería Social que hacen uso de la Geolocalización para hacer más creíble el ataque.

Sobre la Geolocalización en dispositivos Móviles, los usos se disparan. El conocer la posición actual en un momento determinado nos ofrece el poder disponer de un sinnúmero de información de nuestro alrededor. La primera y más conocida tecnología es el sistema GPS, que informa de nuestra posición actual, lo cual ha derivado a los conocidos navegadores GPS, que a través de mapas y la posición actual, indican diferentes rutas a fin de llegar al destino. Actualmente, esta tecnología ofrece información como bares, restaurantes, puntos de encuentro, localización de móviles para niños, etc.

### 2.1.3. Sistemas de localización de dispositivos móviles

Existen diferentes maneras de localizar un dispositivo móvil, pero la efectividad del método dependerá de algunas variables como el medio o la disponibilidad de esta medición en el terminal.

Es posible clasificar los diferentes sistemas en tres grupos:

- **Basados en la red:** Estos sistemas utilizan los sistemas del proveedor de servicios para determinar la posición del terminal, por lo que no necesitamos ninguna aplicación específica funcionando en el móvil. El problema principal de este sistema es que es preciso estar cerca del proveedor para que funcione.
- **Basados en el terminal:** Los dispositivos que utilizan estos sistemas disponen de un receptor de señales y un software cliente para determinar la posición del terminal a través de las señales externas. Cabe destacar que es preciso instalar una aplicación en

el móvil, haciendo que el funcionamiento de esta dependa de la adaptación de los diferentes sistemas operativos.

- **Híbridos:** Los sistemas híbridos son una combinación de sistemas basados en el terminal y sistemas basados en la red. Aunque contenga los métodos más fiables, también adquiere los problemas de los grupos anteriores.

#### 2.1.4. Sistemas de localización basados en la identidad celular

##### 2.1.4.1. Identidad Celular Global (CGI)

Esta técnica fue inventada primeramente para la localización en redes GSM, aunque actualmente funciona en redes GSM, GPRS, UMTS y CDMA. Debido a su sistema de funcionamiento la red se conoce como red celular, ya que esta se compone de células que emiten a una cierta distancia. Cada célula tiene otras 6 contiguas a su alrededor, para diferenciar las señales de las diferentes celdas, cada celda no puede usar la misma frecuencia que sus otras celdas contiguas, por lo cual obliga a usar un mínimo de siete frecuencias diferentes.

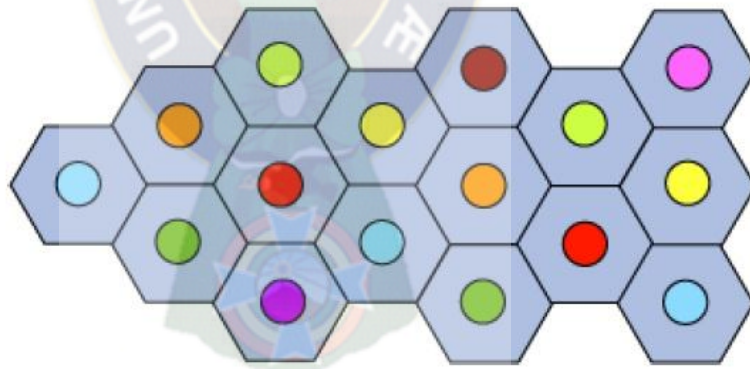


Figura 2.2: Estructura de red en celdas

Fuente: Wikipedia, 2015

Con ello, si conocemos la celda en la que nos encontramos conectados, podemos concretar que nos encontramos en algún punto del radio de esa célula. Esta técnica no necesita ningún tipo de modificación en el dispositivo móvil ya que el cálculo lo hace la propia red de comunicaciones.

El margen de error del sistema de identidad celular depende del radio de la señal. Así pues, en zonas de mucha afluencia como ciudades, las celdas tienen un radio más pequeño, por lo que podemos conseguir un posicionamiento con un error de tan solo varios metros, mientras que en zonas de baja afluencia de gente como pueblos o zonas rurales, podemos variar desde un error de 500 metros hasta kilómetros. Esto lo convierte en un método poco preciso fuera de las ciudades capitales.

#### **2.1.4.2. Identidad Celular Perfeccionada (CGI-TA)**

Timing Advance (TA) es una técnica para conocer la distancia de la celda al dispositivo a través del tiempo de retraso de la señal. La integración de este método con CGI produce el método CGI-TA, que sitúa nuestra posición de una circunferencia a un círculo. Para hacer este cálculo, se precisa un Location Measurement Unit (LMU), que ayudará a la célula en los cálculos de distancia de la señal.



Figura 2.3: Sistemas de localización por identidad celular

Fuente: Wikipedia, 2015

### **2.1.5. Sistemas de localización basados en la red**

#### **2.1.5.1. Ángulo de llegada (AOA, DOA)**

Este método utiliza antenas multiarray en las estaciones telefónicas para calcular el ángulo con el que llega la señal del teléfono. Para conocer la posición de este móvil es preciso tener al menos dos mediciones de dos antenas diferentes. Normalmente en entornos rurales es suficiente para hacer un cálculo de la posición



se necesitan al menos tres antenas. Por otra parte esta tecnología es relativamente cara, y la posición de la antena multiarray es muy importante, cualquier desajuste de esta implicaría errores importantes en la estimación.

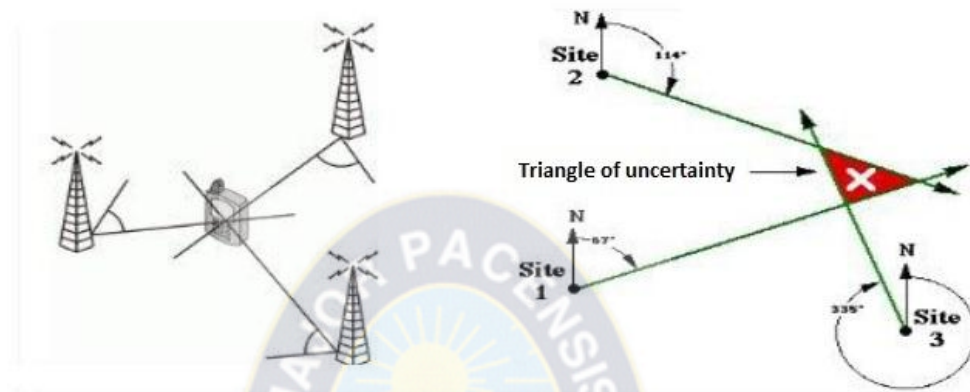


Figura 2.4: Localización por ángulo de llegada

Fuente: Wikipedia, 2015

#### 2.1.5.2. Tiempo de llegada (TOA)

Esta técnica se basa en el cálculo del tiempo de llegada de una señal a las antenas cercanas. Considerando que la señal viaja a la velocidad de la luz, si conocemos el tiempo que tarda la señal en llegar de una antena al dispositivo móvil y volver, podemos conocer la distancia a la que este se encuentre. Si trazamos un círculo con al menos tres de estas antenas, estos círculos se cruzaran en un punto donde se encontrara situado el móvil. Hay que tener en cuenta de que el hecho que el móvil deba responder a una señal crea un error de respuesta, ya sea por el procesado del terminal o por la carga de este.

#### 2.1.5.3. Diferencia en el tiempo de llegada (TDOA)

El método es parecido a TOA, pero en vez de medir el tiempo de llegada de la señal del móvil a una antena, se mide la diferencia en tiempo de la llegada de la señal del móvil a un par de antenas. Una vez tenemos la diferencia del tiempo, si marcamos todos los puntos en que la distancia de cada punto a las antenas este definida por una constante (la constante marca la diferencia de tiempo de la

llegada de la señal), esto genera una Hipérbola. Si repetimos este proceso para todos los pares posibles de antenas de las que disponemos, conseguiremos que las Hipérbolas se crucen en un punto, donde estará situado el dispositivo móvil. Este sistema, al estar basado en mediciones de tiempo en relación a dos antenas y no solo a una, reduce los errores a causa de la reflexión. Para que este sistema funcione, es importante que todos los relojes estén sincronizados.

#### **2.1.5.4. Huella multitrayecto**

la técnica de la huella multitrayecto aprovecha el fenómeno de refracción de las ondas del móvil al llegar a la antena. La señal del móvil rebota en edificios, casas, etc, hasta llegar al receptor. Esto hace que el dispositivo reciba la misma señal varias veces con diferentes tiempos de llegada. Con esto, el operador ha de enviar diferentes señales de prueba para que se guarden las diferentes refracciones en una base de datos que luego permita con una serie de cálculos saber de dónde procede una señal. Este sistema permite usar tan solo una sola antena para calcular la posición de un dispositivo.

### **2.1.6. Técnicas basadas en la modificación del terminal móvil**

#### **2.1.6.1. Diferencia en el tiempo de llegada con terminal modificado**

Esta técnica tiene el mismo funcionamiento que TOA, con la diferencia de que el dispositivo móvil es capaz de saber el momento exacto en que la antena mandó la señal, por lo que solo tiene que hacer la diferencia entre el momento en que recibió la señal y el momento que esta se envió para saber la distancia hasta la antena. El problema de este sistema es que requiere que los relojes del dispositivo móvil y la antena estén sincronizados.

#### **2.1.6.2. Diferencia en el tiempo de llegada perfeccionada (E-ODT)**

E-ODT es una tecnología para GSM basada en el concepto de TDOA, con diferencia de tiempos de llegada. A diferencia de TDOA, es el dispositivo móvil el que calcula la posición, y no el operador. Para que este sistema funcione, primero



se necesita, igual que en TDOA, que todos los relojes de los dispositivos estén bien sincronizados. También se necesitan diversas unidades MTU que medirán el tiempo de llegada de la señal de la antena al receptor. Con la diferencia del tiempo de la antena al móvil y al dispositivo MTU, el móvil puede calcular una Hipérbola, igual que en el caso de TDOA. Si repetimos este proceso para diversas antenas y diversos MTU, el móvil será capaz de calcular la posición. Para regular posibles desajustes en los relojes de los dispositivos, normalmente se utiliza conjuntamente RTD, que calcula las diferencias de tiempo de una antena con las otras antenas.

#### **2.1.6.3. Triangulación avanzada de enlace hacia delante (A-FTL)**

A-FTL utiliza el mismo procedimiento que TDOA, pero únicamente funciona con CDMA, ya que este tipo de red es síncrona en operación, por lo cual no es necesario ningún sistema de sincronización de reloj añadido.

#### **2.1.6.4. Sistema de Posicionamiento Global (GPS)**

Global Positioning System, o conocido más comúnmente por sus siglas (GPS), es un “sistema de posicionamiento global” basado en terminal que permite conocer la situación de un objeto o persona en cualquier lugar del mundo. Se trata de una red de 27 satélites que emiten una señal con el tiempo de emisión y su posición. Esta señal llega al GPS con un cierto retraso, lo cual nos permite calcular de una manera aproximada la distancia del satélite, ya que sabemos que esa señal viaja a la velocidad de la luz.

En cuanto conocemos la distancia de un satélite y su posición, podemos definir una esfera con el satélite como centro. Conociendo también la posición y distancia de un segundo satélite, podemos conocer también su esfera, que cortara con la del primer satélite en un círculo, definiendo nuestra posición dentro de ese círculo. Con un tercer satélite, conseguiremos de nuevo una esfera que esta vez nos cortara el círculo en dos puntos. Uno de ellos se podrá rechazar por no ser real dentro de la superficie de la tierra.



Figura 2.5: Localización GPS

Fuente: Wikipedia, 2015

Desgraciadamente, los relojes atómicos de los satélites GPS no están sincronizados con los relojes del receptor GPS, por lo que nuestra posición no resulta del todo precisa. La ayuda de un cuarto satélite dará una mejor precisión sobre la posición donde nos encontramos, ya que al no tener los relojes correctamente sincronizados, la intersección de tres de las cuatro esferas dará un pequeño volumen. La corrección de la posición consiste en buscar el ajuste en el reloj para que el corte de las cuatro esferas se dé en un punto, o dicho de otra forma, que el volumen se convierta en un punto. En este caso, el problema es que las señales nunca viajan a la velocidad de la luz exactamente, varían según el medio en el que viajan, por eso la posición no es 100% exacta. Por ello, los sistemas de GPS suelen usar la señal de otros satélites, ya que la cobertura suele oscilar entre 6 y 10 satélites.

Actualmente, los sistemas de GPS, con una buena cobertura (7 o más satélites), son capaces de ofrecer una posición con una precisión inferior a 2'5 metros. Los principales errores producidos al usar GPS son causados por:

- Errores en los parámetros orbitales: El satélite tiene un margen de error a la hora de calcular su posición exacta, por lo que este error se transfiere al cálculo de nuestra posición.
- Refracción del medio de transporte: El medio de transporte de la señal hace que no llegue a viajar a la velocidad de la luz exactamente.
- Disponibilidad selectiva: Manipulación que el departamento de defensa de los Estados Unidos hace sobre los parámetros orbitales y el reloj.
- Efecto Multipath: El efecto del rebote de la señal en una superficie reflectante, hace que aumente el tiempo que tarda en llegar esta señal.

Todo y tener diferentes errores de precisión, todos tienen sistemas para ajustar este error, por lo que se puede afirmar que la precisión de este sistema es bastante fiable.

#### **2.1.6.5. Integración con Telefonía Móvil**

Actualmente dentro del mercado de la telefonía móvil la tendencia es la de integrar, por parte de los fabricantes, la tecnología GPS dentro de sus dispositivos. El uso y masificación del GPS está exclusivamente extendido en los teléfonos móviles Smartphone, lo que ha hecho surgir todo un ecosistema de software para este tipo de dispositivos, así como nuevos modelos de negocios que van desde el uso del terminal móvil para la navegación tradicional punto-a-punto hasta la prestación de los llamados Servicios Basados en la Localización. Ver Figura 2.6.

#### **2.1.6.6. GPS actuales**

Existen varios modelos, algunos ya antiguos, y otros todavía operativos. Pero enfocados a dispositivos móviles, se puede mencionar tres:

1. GPS: La base en la que se sustentan casi todos los dispositivos móviles. Como principal ventaja está el hecho de ser el estándar en geolocalización y que no depende de la cobertura 3g. Como desventaja, la poca fiabilidad, la

dependencia de buena cobertura exterior y los tiempos de espera entre envío y recepción de información.

2. A-GPS: El A-GPS fue creado precisamente por el tiempo de espera inicial del GPS, A-GPS hace uso de Internet para dar una posición más o menos acertada y ayudar al GPS a triangular más rápido. Depende de datos móviles, y no mejora al GPS simplemente le da una pequeña ayuda inicial. Su funcionamiento es sencillo y rápido. Se conecta a internet para descargar un paquete de datos con la información de la antena a la que está conectado.
3. GLONASS: Los satélites GPS son dominio de EEUU, los de GLONASS son rusos. Este sistema lleva más de 30 años, pero por ser soviéticos, no es muy conocido. Por este motivo Rusia amenazó con aumentar los impuestos de dispositivos móviles que no tuvieran soporte a GLONASS para que los fabricantes comenzarán a tomarlo en serio. GLONASS entra en funcionamiento cuando el GPS falla. No hace falta estar perdido en el desierto para que no haya cobertura GPS, sino que simplemente una ciudad con edificios altos tiene a su vez puntos ciegos, y la ayuda de GLONASS es mejor. Tiene más margen de error que el GPS, pero la unión de estos dos hace más efectivo el posicionamiento.

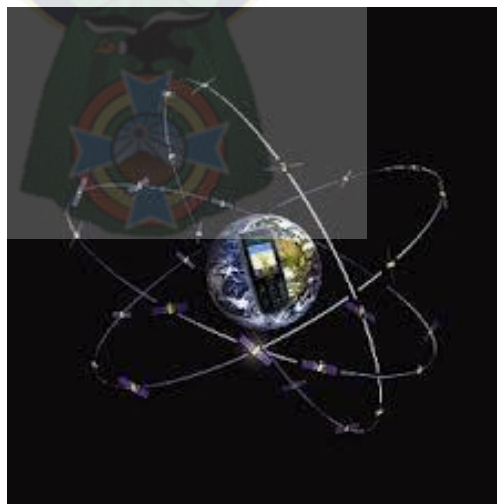


Figura 2.6: GPS en Telefonía Móvil

Fuente: Wikipedia, 2015

### 2.1.6.7. Arquitectura GPS en dispositivos móviles

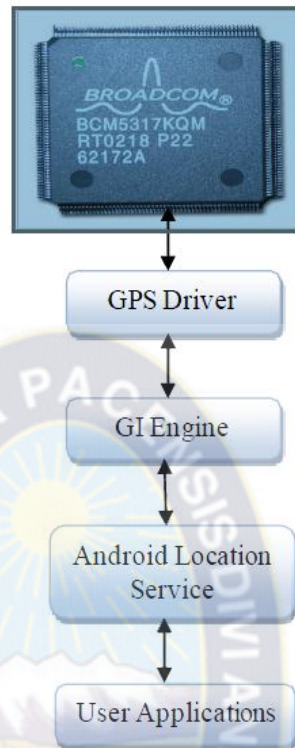


Figura 2.7: Arquitectura GPS

Fuente: Wikipedia, 2015

#### - El Chip

Se trata de un receptor de radiofrecuencia que se comunica directamente con los satélites GPS.

#### - El Driver

Sirve de controlador para el chip, el GPS Driver usa API's de nivel Bajo para comunicarse con el Chip GPS. Consiste sencillamente en una serie de archivos que se encuentran en /System/lib/hw o /Vendor/Lib/hw y usualmente del tipo gps.\*.os dependiendo de la versión de Android (ejemplos: gps.default.so o gps.aries.so). Cabe mencionar que son los propios fabricantes quienes liberan este producto.



## - **GL Engine**

Este es el encargado de suministrar la localización de los nodos de radiofrecuencia cercanas, guardando los datos obtenidos en el GPS Lock.

Consiste de un path (/system/bin) acompañado de nombres como glgps o gpsd. Funciona utilizando unos parámetros de configuración que consisten en archivos .xml o .conf (glconfig.xml, gps.xml, gpsconfig.xml, gps.conf). Estos nombres dependen de la versión de Android y se encuentran generalmente en (/system/etc, /system/etc/gps o /vendor/etc).

Dependiendo de la configuración y de la plataforma, GL Engine toma la información de Localización de las torres móviles y lee el NVRAM (es el sitio donde se almacena data del GPS lock), esta información se encuentra generalmente en /data/gps en archivos acabados en .sto y .dat (ejemplo: gldata.sto, lto.dat, xtra.bin, epo.dat).

Al hacer uso de toda esta información, el GL Engine asiste al GPS Driver: por lo tanto es capaz de detectar múltiples satélites GPS para los que está programado el GPS Driver, pero para fijar la posición necesita información adicional que puede descargar de los satélites GPS (de manera muy lenta) o bien puede acceder a Internet para usar los servidores SUPL/NTP (mayor velocidad), inmediatamente guarda todos estos datos en el NVRAM para próximos usos.

## - **Android Framework o Servicio de Localización**

Consiste en una serie de clases como puede ser el Location Manager, que proporcionan servicios para que una aplicación use el GL Engine. Es una especie de puente entre la aplicación y el GL Engine.

Pertenecen al Framework de Android, y por lo tanto de java. Se trata de un grupo de funciones contenidas en el paquete android.location del

framework que controlan diferentes aspectos del GPS con el fin de comunicarse entre Apps y GL Engine.

Algunos ejemplos son:

- a) addGpsStatusListener (GpsStatus.Listener listener): Cambia el estado del GPS a escuchando.
- b) getProviders (Criteria criteria, boolean enabledOnly): Retorna una lista de torres cercanas que satisfacen X criterios pasados a la función.

#### - **Aplicaciones de Usuario**

La última capa son las aplicaciones que se tiene instalada en el terminal, se trata de cualquier aplicación que use el GPS. Google Maps, Navigation, etc.

#### **2.1.6.8. A-GPS**

El A-GPS, es un sistema híbrido basado en GPS, creado para solucionar los problemas de localización con GPS en ciudades grandes o interiores con poca cobertura de señal de los satélites, y aumentar su precisión. Se trata de recibir la señal de GPS con algún otro tipo de señal, ya sea Wireless, telefonía, etc.

Tiene dos sistemas de funcionamiento, el modo online, donde el A-GPS necesita una conexión activa y constante a una red de teléfono (por ejemplo GSM), de la cual puede recibir su posición gracias a la celda en que se encuentra, recibir información sobre el entorno como posición de los satélites o condiciones ionosféricas para aumentar la precisión del GPS, o incluso enviar el computo con la información del GPS al servidor de asistencia. En el modo offline, la conexión (ya sea a través de GPRS, Ethernet, Wireless), no es constante, por lo que el A-GPS descarga un fichero con información de su celda o posición de satélites, entorno, etc., mientras disponga de esta conexión, para más tarde usar dicha información incluso durante varios días.

## **2.1.7. API's de Geolocalización**

### **2.1.7.1. Location API for JAVA (OpenLAPI)**

“Location API” es una interfaz de localización de dispositivos móviles para J2ME, la versión reducida de Java para móviles. Fue creada bajo “Java Community Process” en septiembre de 2003. Nokia es el principal autor y encargado del mantenimiento de esta plataforma. Es posible utilizarla mediante el paquete `Javax.microedition.location`. Esta API funciona para dispositivos móviles, funcionando con la plataforma CLDC v1.1 como mínimo, ya que las antiguas versiones no dejan trabajar con números en coma flotante. Se encuentra bajo la licencia de GNU (LGPL) de código abierto, por lo que es posible usarla como librería en alguna aplicación, ya sea esta de código abierto o cerrado. La API funciona con técnicas diferentes de posicionamiento, a través de GPS, AGPS, AOA.

Esta API soporta información como posición (latitud, longitud, altura, dirección y velocidad de movimiento), e incluye soporte para la creación y uso de lugares de referencia “landmarks”, datos sobre los puntos de referencia (dirección, tipo de lugar, código postal, etc.), y soporte para el lanzamiento de aplicaciones por posición. Se pueden definir diferentes métodos para una única localización, pudiendo establecer unas reglas como rango de error, tiempo de respuesta, necesidad de altura o velocidad.

Dado que este sistema está implementado para dispositivos con especificaciones limitadas, se recomienda no cargar la aplicación más de 2 kB para RAM, y que no ocupe más de 20kB de memoria.

Hay que tener en cuenta que es parte de la implementación de la aplicación el poder tener acceso a ciertos recursos de posicionamiento del terminal, teniendo en la mayoría de ellos una capa de seguridad (como por ejemplo MIPD 2.0), con la que la API no trabaja directamente sino que la aplicación ha de pedir permiso para usar estos recursos.



### **2.1.7.2. Google Gear API**

Google Gears es una API desarrollada por Google que se instala como extensión del navegador para añadir funcionalidades adicionales a la creación de Webs. Dentro de sus diferentes posibilidades, Gears permite a la web la localización por GPS o por red (por IP + células que detecta o por redes Wireless). Esta última no es eficiente en tiempo real, por lo que no se aconseja para aplicaciones de seguimiento como navegadores para mapa. Funciona con la mayoría de SO y Navegadores más comerciales, pudiendo ser utilizada también en dispositivos móviles sobre IExplorer en Windows Mobile 5 & 6, Opera Mobile en Windows Mobile 6 y todos los dispositivos Android. Gears permite el cálculo de la posición actual, la monitorización del cambio de posición a través del tiempo, y conocer la última posición conocida del dispositivo.

Hay que destacar que en este caso, el cálculo de la posición la hace el servidor web a través de una consulta a la base de datos de Google, por lo tanto, el dispositivo solo envía información de su red al servidor. Por ello, es posible que una aplicación conozca la posición sin tener que instalar ningún servicio al móvil más que Google Gears.

Cuando Google Gears intenta acceder a este servicio, por temas de privacidad, aparece una ventana de aceptación por parte del usuario.

### **2.1.7.3. Android Location Services**

Google ofrece en su área de desarrollo Android para móviles, una API que se encuentra en el paquete android.location. La API de Android permite conocer las últimas posiciones del móvil, así como monitorizar en tiempo real la posición del dispositivo, o llamar a una aplicación cuando el dispositivo se acerca a una zona previamente marcada. Permite trabajar con diferentes tecnologías como GPS o otras basadas en red. Uno de los beneficios de usar esta plataforma es que está diseñada para trabajar con facilidad con el sistema de Google Maps.

### **2.1.8. Trust Managment Service (TMS)**

Trust Management Service funciona como una entidad certificadora, algo parecido al HTTPS para páginas web, que certifica como confiable a una aplicación que intenta acceder a la información sobre nuestra posición. Este servicio nació por diferentes causas, una de ellas es la poca usabilidad de tener que aceptar manualmente los términos cada vez que iniciamos una aplicación de estas características. Además, normalmente los usuarios disponen de poca información sobre la empresa / aplicación a la que le están dando su consentimiento. Con TMS, se delega a esta entidad el decidir sobre la aceptación o no de estos términos, según si es una aplicación en la que se puede confiar o no.

Primero, el cliente llama a una capa de seguridad local cuando una aplicación intenta acceder a funcionalidades restringidas del dispositivo móvil. A continuación, el cliente hace una llamada remota a TMS, enviándole la información sobre la aplicación que intenta acceder a estas funcionalidades. Por último, TMS decide si debe o no debe ceder estas funcionalidades según su base de datos y las preferencias predefinidas del usuario.

### **2.1.9. Aplicaciones actuales con esta tecnología**

Actualmente ya hay diferentes aplicaciones comerciales que hacen uso de esta tecnología. A continuación se listan algunos ejemplos:

#### **2.1.9.1. Sky Map**

Funciona con el sistema operativo Android. Conociendo la posición del usuario, gracias al navegador GPS y a una brújula digital que indica el lugar hacia el que la persona desea observar, muestra en la pantalla los nombres de las estrellas que se ven en tiempo real.

#### **2.1.9.2. Places Directory**

Este programa, también diseñado para Android, muestra una guía de locales comerciales, como restaurantes, bares o gasolineras, que se encuentren cerca de la posición del usuario. La aplicación indica la dirección geográfica y la distancia a la

que se encuentran. La guía se complementa con comentarios y puntuaciones de otros usuarios.

### **2.1.9.3. Search with my Location**

Se trata de una función que el buscador Google ha incorporado a su versión para móviles que permite determinar la posición del usuario y le ofrece resultados de búsqueda relacionados con esta.

### **2.1.10. Google Maps**

Uno más de los productos llevados al mercado por el gigante Google Inc., mismo que anunciaba en Google Blog el 8 de febrero del 2005 el inicio y puesta en marcha de sus mapas digitales, soportado en una primera fase solo por los usuarios de Internet Explorer y Mozilla Firefox, agregándose el soporte para Opera y Safari el 25 de febrero del mismo año.

Se trata de un servidor de aplicaciones de mapas en la Web (Internet), con la capacidad de hacer acercamientos o alejamientos (Zoom) al mapa, controlando con el mouse o las teclas de dirección los movimientos para encontrar la ubicación que se desee; además los usuarios pueden ingresar una dirección, una intersección o un área en general para buscar en el mapa y encontrar los resultados.

Google Maps, permite la creación de pasos para llegar a alguna dirección creando una lista paso a paso para saber el cómo llegar a su destino, calculando el tiempo necesario y la distancia recorrida entre las ubicaciones.

Google Maps puso a disposición de los desarrolladores sus códigos fuentes llamados APIS, los mismos que permiten introducir los mapas de Google Maps en cualquier aplicación con el uso de su codificación y con ello se pueden aplicar nuevas formas de ver el mundo.

Con la innovación de las herramientas de búsqueda y el movimiento en el mapa de Google Maps, se ha incrementado el interés en el uso de las imágenes satelitales, se han creado sitios que buscan crear la base de datos de lugares conocidos y vistos desde el

espacio, como por ejemplo: estadios, construcciones antiguas, carreteras, edificios, ciclo vías, etc.

#### 2.1.10.1. Ventajas de Google Maps

VENTAJAS	DESCRIPCION
<b>La arquitectura hardware (servidores).</b>	Se puede decir que la gran ventaja de Google Maps es su arquitectura (servidores); así como también el ancho de banda utilizado. Hoy en día ningún sistema propio o propietario puede ofrecer su potencial de cálculo.
<b>La cartografía</b>	Se encuentra más actualizada que cualquiera de sus competidores, y a disposición del público a coste cero.
<b>El sistema es abierto y flexible</b>	Es muy fácil la introducción de capas de datos, datos geolocalizados, modelos 3D, incluso nuestra propia cartografía, desde diferentes fuentes de datos.
<b>Todos los datos están en la nube</b>	Lo que permite a los usuarios un acceso rápido y eficiente desde cualquier lugar, lo que Google ha denominado FusionTables.
<b>No requiere la instalación / configuración</b>	Tan solo con un navegador web que permita JavaScript, lo que hoy en día son prácticamente todos.

<b>Los datos están más Seguros</b>	Los mismos que son cifrados, se hacen copias de seguridad, balanceos de carga, etcétera.
<b>Acceso a los datos</b>	Con acceso desde cualquier lugar del mundo, a cualquier hora, y desde cualquier dispositivo (ordenadores, teléfonos móviles, tablets, etc.).
<b>En su versión 3</b>	Google Maps se ha optimizado para el uso en dispositivos móviles.

Tabla 2.1: Ventajas de Google Maps

#### 2.1.10.2. Las API's de Google Maps Versión

Estos códigos fuentes proporcionan a los desarrolladores diversas formas de insertar Google Maps en páginas web permitiendo un uso sencillo y una amplia personalización.

La versión 3 de las APIS de Google Maps, presenta novedades que aportan a los habituales desarrollos en el escritorio son pocas (añadidas funciones como geolocalización del cliente de manera nativa y soporte para direcciones, por ejemplo), sin embargo ha sido modificada por completo desde sus entrañas, al punto de que casi se la ha reescrito prácticamente usando un framework MVC consiguiendo lo siguiente:

- No se necesita solicitar la tediosa API Key.
- Aumentar la velocidad de descarga, pues ocupa menos espacio y por tanto debe descargarse en menos tiempo.
- Funciona razonablemente bien en dispositivos móviles como iPhone y Android, para este último incluso se han añadido controles de navegación con un aspecto similar a la aplicación de mapas nativa.



Hablar de Google Maps es hablar de desarrollo e innovación ya que el aporte a la sociedad es muy tangible, dado que el servicio se encuentra disponible en el internet con tan solo digitar Google Maps en un navegador web, además los desarrolladores de todo el mundo logran aplicar Google Maps en sus aplicaciones sin ninguna restricción.

Con conocer y aplicar las estructuras y sintaxis de los códigos fuentes, los mismos que se encuentran totalmente disponibles; es posible obtener el entorno de un mapa digital totalmente actualizado, es por todo esto que, Google Maps es una herramienta de vital importancia que se aplicará en el desarrollo de la GEOREFERENCIACIÓN DE CAJEROS AUTOMÁTICOS Y AGENCIAS BANCARIAS MEDIANTE APLICACIÓN ANDROID.

## **2.2. COORDENADAS GEOGRÁFICAS**

Las coordenadas geográficas son un sistema de referencia que utiliza las dos coordenadas angulares, latitud (Norte y Sur) y la longitud (Este y Oeste) y sirve para determinar los ángulos laterales de la superficie terrestre.

### **2.2.1. Latitud**

La latitud mide el ángulo entre cualquier punto y el ecuador. Las líneas de latitud son círculos que cortan al ecuador en la superficie de la tierra.

### **2.2.2. Longitud**

La longitud mide el ángulo a lo largo del ecuador desde cualquier punto de la tierra. Se acepta que Greenwich en Londres es la longitud 0 en la mayoría de las sociedades modernas. Las líneas de longitud son círculos máximos que pasan por los polos y se llaman meridianos.

La combinación de estas 2 coordenadas permite ubicar cualquier punto en la Tierra, y fueron utilizadas en este proyecto para conocer la ubicación del usuario y las sucursales cercanas. Las coordenadas del usuario se obtienen mediante el uso del GPS del dispositivo y las coordenadas de las sucursales se obtienen de una base de datos.

### **2.2.3. Sistema de Coordenadas**

En geometría, un sistema de coordenadas es un sistema que utiliza uno o más números (coordenadas) para determinar unívocamente la posición de un punto o de otro objeto geométrico. El orden en que se escriben las coordenadas es significativo y a veces las identifica por su posición en una tupla ordenada.

En esta aplicación se utilizará siempre como origen del sistema de coordenadas la ubicación del usuario y se mostrarán las sucursales basándose en su posición relativa a él, es decir, cada vez que el usuario se mueve o cambia de posición, se vuelve a generar un sistema de coordenadas con la nueva ubicación de todos los puntos, manteniendo siempre al usuario en el origen.

### **2.2.4. Geocodificación**

La geocodificación es el proceso de asignar coordenadas geográficas a puntos del mapa (que pueden representar diferentes puntos de interés). Con esta información, el SIG es capaz de especular generando información que no dispone a través de la que conoce. Por ejemplo, si se conoce algunos números de casas de una calle, se puede interpolar, suponiendo que entre casa y casa hay la misma distancia, la posición de los otros números.

Otro sistema parecido es la geocodificación inversa. En este caso, lo que se hace es asignar un punto de interés a unas coordenadas concretas. Un ejemplo, podría ser, con una aplicación móvil, estando en frente de un museo, guardar en una base de datos su posición.

## **2.3. CONCEPTOS MATEMÁTICOS**

### **2.3.1. Sistemas de Representación**

La representación geométrica exacta de la Tierra no es posible, dado la complejidad del cuerpo. Es por esto que en la mayoría de los casos se usan modelos simplificados para este fin. Para este trabajo utilizaremos en primera medida el modelo de Esfera Terrestre y para una aproximación más cercana el modelo de Elipsoide Terrestre. Un modelo más

complejo aun es el de Geoide, éste solido toma como nivel medio al Elipsoide Terrestre y aproxima sobre este la superficie topográfica. El geoide es muy difícil de representar geométricamente y por ello que no fue usado en este trabajo.

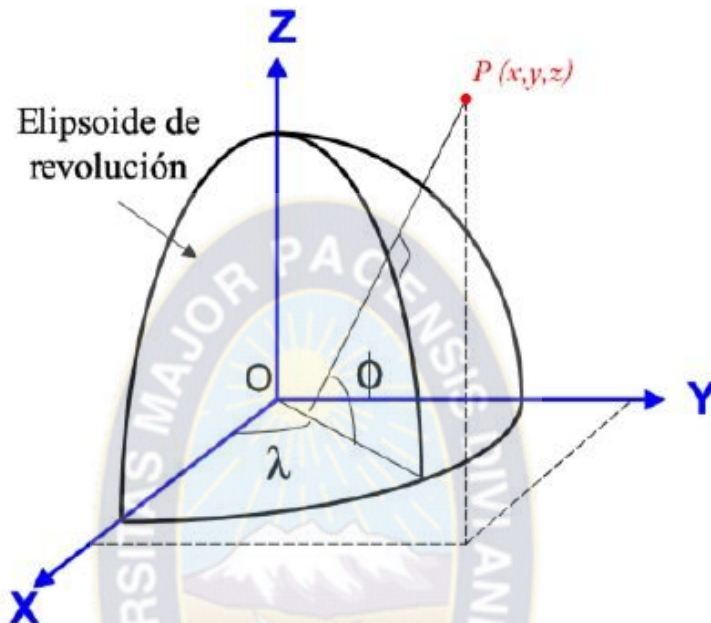


Figura 2.8: Elipsoide terrestre

Fuente: Wikipedia, 2015

Para la realización de algunos cálculos es necesario transformar las coordenadas geográficas en coordenadas cartesianas ortonormales en el espacio (x,y,z) con origen en el centro de la Tierra.

### 2.3.2. Representación esférica

Para esta representación se tiene como dato que 6,371 Km, es el radio propuesto para la tierra.

$$\begin{cases} x = 6371 * \cos(\phi) * \cos(\lambda) \\ y = 6371 * \cos(\phi) * \sin(\lambda) \\ z = 6371 * \sin(\phi) \end{cases}$$

$(\phi)$  = Valor que resulta de la Latitud  $(\lambda)$  = Valor que resulta de la Longitud



### 2.3.3. Representación elipsoidal

La representación en este caso está dada por:

$$\begin{cases} x = 6378,14 * \cos(\phi) * \cos(\lambda) \\ y = 6378,14 * \cos(\phi) * \sin(\lambda) \\ z = 6356,755288 * \sin(\phi) \end{cases}$$

Donde 6378,14 Km es el radio mayor medido sobre el Ecuador y 6356,755288 Km es el radio menor dado por el achatamiento 1/298,257. Estos valores de constantes astronómicas fueron adoptados en la Asamblea General de la Unión Astronómica Internacional que se celebró en Grenoble en 1976, designado por IAU (1976), que entró en vigor el 1 de enero de 1984. Con estos valores obtenemos al elipsoide denominado WGS 84.

### 2.3.4. Distancia Euclidiana

En el caso de la geometría de la Tierra, la ecuación Euclidiana de la distancia no resulta adecuada, debido que la medida no se toma sobre la superficie. Esta resulta una buena aproximación para distancias en las cuales la curvatura de la Tierra puede ser despreciable.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

### 2.3.5. Fórmula de Haversine

La fórmula de haversine es un caso especial de la ley de haversines, una formula general de la trigonometría esférica muy utilizada en la navegación. La función se expresa en términos de funciones de haversine o múltiplos de esta. Se define como:

$$\text{haversin}\left(\frac{D}{R}\right) = \text{haversin}(\phi_1 - \phi_2) + \cos(\phi_1) * \cos(\phi_2) * \text{haversin}(\Delta\lambda)$$

Dónde:

$$\text{haversin}(\theta) = \sin^2\left(\frac{\theta}{2}\right)$$

$D = \text{distancia entre los 2 puntos}$

$R = \text{radio de la esfera}$

$\phi = \text{latitud}$

$\lambda = \text{longitud}$

$\Delta\phi = \text{diferencia de latitud}$

$\Delta\lambda = \text{diferencia de longitud}$

Al despejar la distancia de la fórmula de haversine se obtiene una nueva ecuación, que presenta algunos problemas para ciertos hemisferios, para solucionar estos se propone la formula directa de la distancia de haversine:

$$\Delta\phi = \phi_1 - \phi_2$$

$$\Delta\lambda = \lambda_2 - \lambda_1$$

$$A = \sin^2\left(\frac{\Delta\phi}{2}\right) * \cos(\phi_1) * \cos(\phi_2) * \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$C = 2 * \text{atan2}(\sqrt{A}, \sqrt{1-A})$$

$$D = R * C$$

### 2.3.6. Ley de Haversine

Dada una esfera unitaria, un "triángulo" en la superficie de la esfera se define por gran círculo s de conectar tres puntos o , v y w en la esfera. Si las longitudes de estas tres partes están al (de uv por v ), b (desde o a w ), yc (a partir de W a ), y el ángulo de la esquina opuesta c es C , entonces la ley del haversine dice:

$$\text{haversin}(c) = \text{haversin}(a - b) + \sin(a) \sin(b) \text{haversin}(c)$$

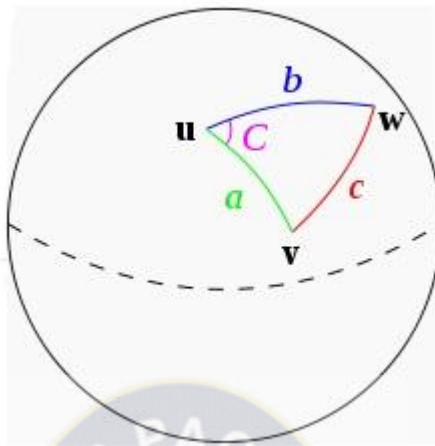


Figura 2.9: Triángulo esférico de haversine

Fuente: Wikipedia, 2015

## 2.4. DISPOSITIVOS MOVILES

Hoy, en que el mundo es cada vez más móvil, los usuarios desean acceder a la información desde cualquier lugar en que se encuentren; esto es, a través de dispositivos PDA, teléfonos móviles o incluso en automóviles.

Un Smartphone (cuya traducción en español sería “teléfono inteligente”) es una evolución del teléfono móvil tradicional que cuenta con ciertas características y prestaciones que lo acercan más a un ordenador personal que a un teléfono tradicional. Entre dichas características, se puede encontrar una mejora en la capacidad de proceso y almacenamiento de datos, conexión a Internet, dispositivos de seguimiento, métodos de entrada avanzados y diversas aplicaciones de usuario como navegadores, aplicaciones ofimáticas, reproductores multimedia, etc.

A pesar de estas importantes mejoras con respecto a sus predecesores móviles, el reducido tamaño de los Smartphones conlleva limitaciones de hardware que los mantienen claramente diferenciados de los ordenadores convencionales. Estas limitaciones se reflejan principalmente en pantallas más pequeñas, menor capacidad de procesamiento, restricciones de memoria RAM y persistente, y necesidad de adaptar el consumo de energía a la capacidad de una pequeña batería.

Aun así, los teléfonos inteligentes son lo más codiciado actualmente en tecnología, y ello no ha pasado desapercibido para las grandes compañías, las empresas ha llevado algunas de sus mejores aplicaciones al mercado de los teléfonos móviles. Además de incluir nuevas.

Muchas veces se asume que la arquitectura de Internet es exactamente igual a la arquitectura móvil es a la de Internet, aunque está lejos de ser verdad. La arquitectura móvil es una arquitectura única y completa. Al igual que sucede con Internet, se compone de piezas diferentes donde todas deben trabajar de forma integrada. Sin embargo, en la tecnología móvil, las partes que la componen son diferentes, y porque los dispositivos móviles tener acceso a Internet, significa que esta última puede ser una parte de la arquitectura de redes móviles.

#### **2.4.1. Arquitectura de sistemas móviles**

Para decirlo de otra manera, pensemos en Internet como una gran nube. Cuando queremos para sacar algo de ella, usamos una herramienta, como una pieza de software o dispositivo. Estas herramientas pueden ser los dispositivos móviles.

Entonces vemos el sistema móvil como un sistema de capas, donde cada capa depende de las otras para funcionar:

<b>Servicios</b>	
<b>Aplicaciones</b>	
<b>Framework de Aplicación</b>	
<b>Sistema Operativo</b>	
<b>Plataforma</b>	
<b>Dispositivo</b>	
<b>Red Móvil</b>	
<b>Operadora</b>	

Tabla 2.2: Arquitectura de Sistema Móviles

Fuente: <http://www.android.com>, 2015

La capa de base en la arquitectura móvil es el operador. Son los que fundamentalmente hacen el trabajo de toda arquitectura móvil: instalan torres de comunicación, gestionan la red celular, hacen que los servicios (como Internet) estén disponible para los suscriptores móviles. El papel del operador en la arquitectura es crear y mantener de manera segura un conjunto específico de servicios de tecnología inalámbrica a través de una red celular.

Las operadoras operan las redes inalámbricas. La tecnología celular, de manera simplista, es una radio que recibe una señal de una antena. El tipo de la radio y la antena determinan las capacidades de la red y los servicios que puede habilitar en ella. La gran mayoría de las redes de todo el mundo utilizan el estándar GSM (Global System for Mobile), pero también existen CDMA (Code Division Multiple Access) y su híbrido CDMA2000 2.5G, 3G, 4G que aunque ofrece una mayor cobertura, limitan la cantidad y capacidad de servicios que pueden transferirse sobre ella.

Los dispositivos son los teléfonos celulares, aunque existen tipos de dispositivos inalámbricos que se basan en redes de los operadores, pero no hacen llamadas telefónicas. De ahora en más al referirnos a dispositivos móviles nos referiremos a los teléfonos móviles. La mayoría de estos dispositivos son los teléfonos multifunciones, que constituyen la mayoría del mercado. Los teléfonos Smartphones constituyen una pequeña porción del mercado en todo el mundo. Como los dispositivos de próxima generación son una realidad, la distinción entre términos teléfonos y teléfonos inteligentes tiende a desaparecer.

#### **2.4.2. Arquitectura del Hardware**

Hoy día, un dispositivo móvil ofrece un gran número de posibilidades, como si de una computadora se tratará. Todo esto combinado con una gran potencia de procesamiento y volumen de almacenamiento que permiten la ejecución no sólo de sencillas aplicaciones sino de aplicaciones realmente funcionales.

Todo esto sin olvidar otras funcionalidades incorporadas a los móviles, como son el GPS, pantallas táctiles, capturadoras de video, sensores, capacidades multimedia, etc.



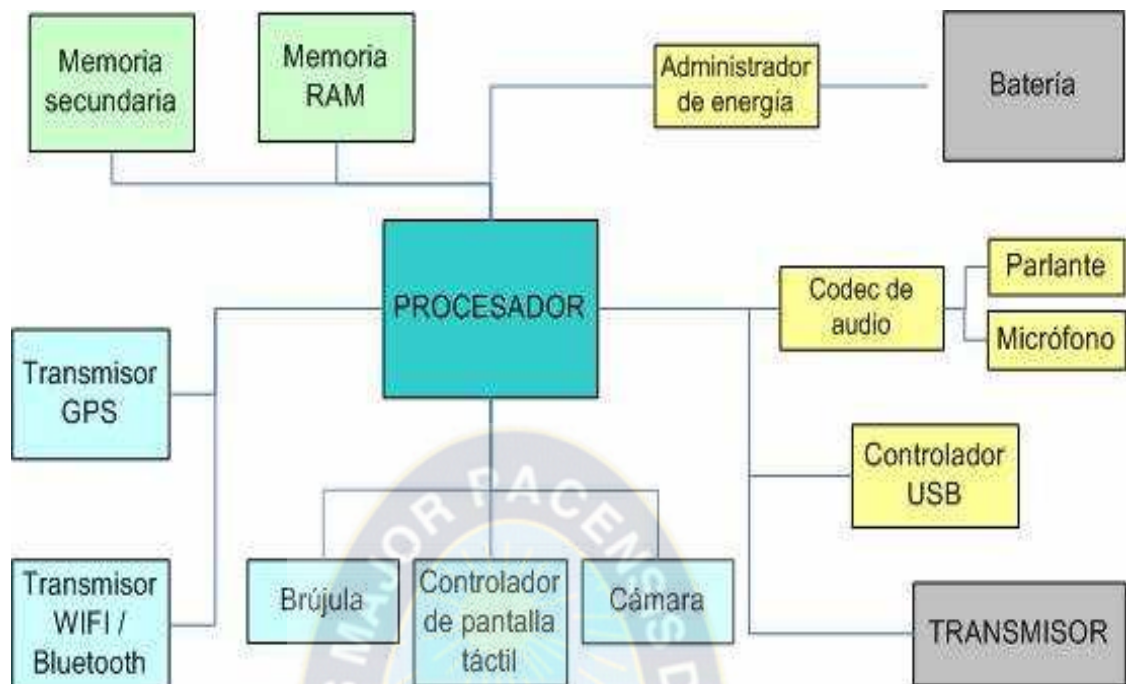


Figura 2.10: Arquitectura de hardware de un dispositivo móvil moderno

Fuente: <http://www.android.com>, 2015

## 2.5. PLATAFORMA ANDROID

### ¿Qué es Android?

Android es un conjunto de herramientas software de código abierto para dispositivos móviles creado por Google y en colaboración con “Open Handset Alliance”. Esta plataforma de software está basada en un kernel Linux y permite su desarrollo mediante la utilización de librerías Java creadas por Google. Está dentro de millones de dispositivos como teléfonos móviles, tablets y un largo etc, haciendo de Android una de las mayores plataformas para programadores.

#### 2.5.1. Características y ventajas de Android

Aunque podemos encontrar en el mercado una gran variedad de dispositivos con diferentes S.O, ninguno de ellos tiene una combinación de las siguientes características.

- *Es una plataforma totalmente libre basada en Linux y de código abierto:* le gusta a los desarrolladores porque pueden usarla y customizarla a su gusto sin tener que pagar unos derechos de autor.
- *Una arquitectura basada en componentes inspirados en “mashups” de internet:* partes de una aplicación puede ser usada para otro objetivo diferente al cual fue diseñado originalmente por el desarrollador. Es posible incluso sustituir o ampliar código en tu propio beneficio para la creación de tus versiones del programa. Esto proporciona una creatividad ilimitada.
- *Toneladas de servicios integrados:* servicios basados en localización utilizando GPS o triangulación mediante torres de telefonía. Una potente y completa base de datos SQL. Mapas que puedes incrustar en tus aplicaciones. Todos estos servicios integrados ayudan a incrementar la funcionalidad pero lo más importante, a reducir el tiempo de implementación.
- *Gestión automática de la vida de las aplicaciones:* los programas están aislados unos de otros por varias capas de seguridad, lo que proveerá de un nivel de estabilidad nunca visto antes en un smart phone. El usuario final no tiene que preocuparse de que programas están abiertos o cerrados si quiere ejecutar cualquier aplicación. Android lo gestiona en función de la memoria sobrante.
- *Gráficos y sonido de alta calidad:* gráficos vectoriales 2D y animaciones en Flash se mezclan con gráficos Open GL en 3D preparados para nuevos juegos y aplicaciones. También se incluyen los códec de audio y video estándar como H.264 (AVC), MP3 y ACC.
- *Funcionamiento en una gran variedad de hardware actual y futuro:* todos los programas están escritos en Java y ejecutados por la máquina virtual Dalvik es decir, el código desarrollado se puede trasladar a arquitecturas ARM, x86, etc.... Además soporta una amplia variedad de periféricos de entrada como teclados, paneles táctiles y trackball.

## 2.5.2. Arquitectura Android

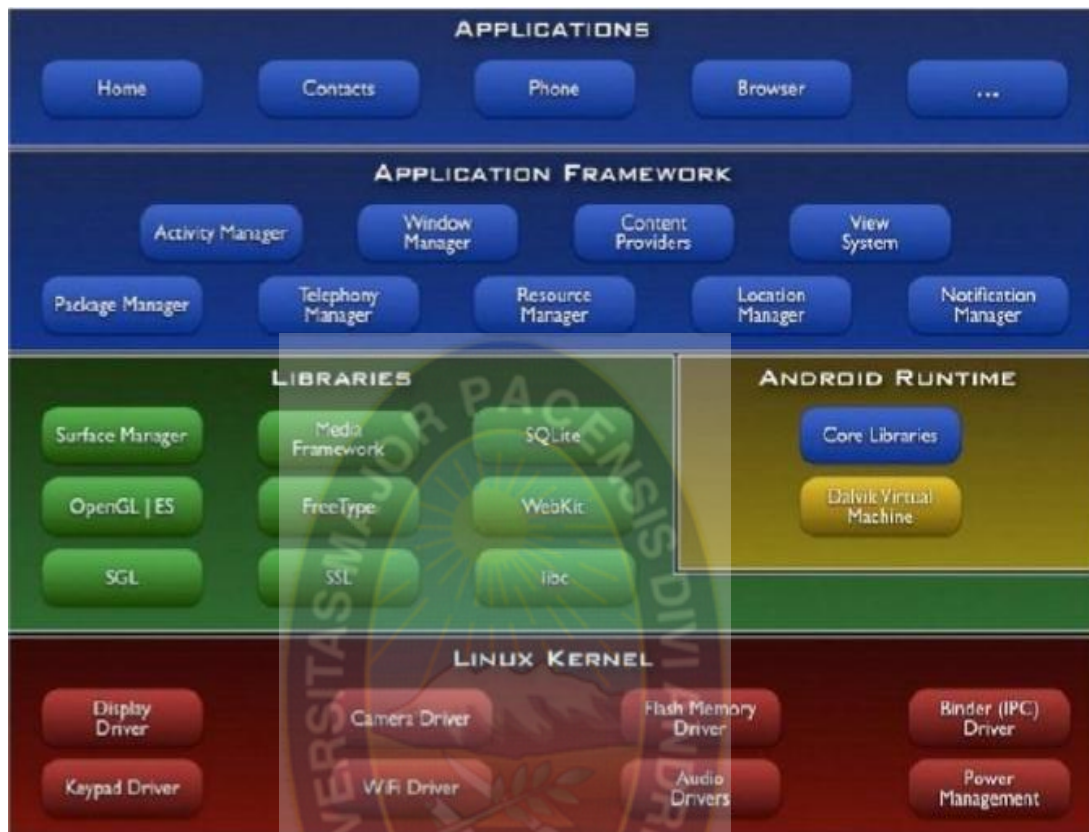


Figura 2.11: Arquitectura de Android

Fuente: <http://www.android.com>, 2015

A continuación detallaremos las principales capas y componentes que forman parte en esta pila de software en código abierto. Cada una de las capas usa servicios que provienen de la capa inferior.

- **Linux Kernel:** Android está construido sobre una firme base que provee la capa de abstracción hardware permitiendo ser transferido a un amplio rango de plataformas en el futuro. Android usa Linux para su gestión de memoria, gestión de procesos, red y otros servicios del sistema operativo.
- **Librerías:** Esta capa contiene librerías escritas en C o C++. Algunas de las principales librerías son:



- *Surface Manager*: se encarga de la gestión de efectos de ventana similar a Vista o Compiz.
- *Graficos 2D y 3D*: los graficos 2D y 3D pueden ser combinados en una única interface de usuario.
- *Media Codecs*: Android puede reproducir video y audio así como grabarlo en una gran variedad de formatos incluyendo AAC, AVC (H.264), H263, MP3 y MPEG-4.
- *Base de datos SQL*: Android incluye una ligera base de datos, la misma que se utiliza en Firefox y Iphone.
- *Browser Engine*: Android utiliza el WebKit library para la rápida reproducción de contenido HTML.
- **Android Runtime**: esta capa está situada también encima de la capa de hardware y contiene la máquina virtual Dalvik y las librerías Java.

Dalvik es una implementación de Google para Java y optimizada para dispositivos móviles. Todo el código que se escribe para Android estará escrito en Java y se ejecutara dentro de Dalvik. Dalvik difiere respecto a Java tradicional en dos aspectos:

- Dalvik ejecuta archivos .dex, los cuales son convertidos en tiempo de compilación desde los tradicionales .class y .jar. Los archivos .dex son más compactos y eficientes que los archivos .class lo cual es bastante importante teniendo en cuenta la limitada memoria y batería de los dispositivos móviles.
- La librería Java que viene con Android difiere de la estándar y de la versión para móviles. Hay una substancial cantidad de solapamiento.
- **Application Framework**: en esta capa encontraremos las herramientas que usaremos para crear las aplicaciones. Las partes más importantes del Framework son:
  - *Activity Manager*: controla el ciclo de vida de las aplicaciones.

- *Content Providers*: estos objetos encapsulan datos para poder ser compartidos entre aplicaciones.
- *Resource Manager*: es todo lo que acompaña a nuestro programa que no es código.
- *Location Manager*: todo lo relacionado con el sistema de localización GPS.
- *Notification Manager*: toda clase de eventos como alertas de mensajes, etc.
- **Applications and Widgets**: Son los programas que ve el usuario final. Normalmente vienen pre-instaladas algunas aplicaciones como un cliente e-mail, programa de llamadas, un buscador, Android Market, etc.

### 2.5.3. Como trabaja Android

En nuestro Linux o Windows estándar, podemos tener tantas aplicaciones abiertas a la vez como queramos y todas ejecutándose a la vez en diferentes ventanas. Podemos intercambiar actividades como nosotros queramos. Es responsabilidad del usuario abrir y cerrar tantas como se quiera.

Android trabaja de un modo diferente. En Android solo se trabaja con una aplicación en primer plano la cual normalmente se hace cargo de toda la pantalla. Por ejemplo, cuando un usuario enciende el teléfono, la primera aplicación que ve es la Home application o Aplicación Principal.

Cuando se ejecuta una aplicación, Android la lleva al primer plano y desde esa se podría invocar otra dejando esta almacenada en una pila de aplicaciones. En el momento el usuario desee retornar a la anterior presionaría el botón de retroceso para acceder a la posición anterior de la pila de actividades.

Por otro lado cabe diferenciar entre Actividad y Proceso. Cada actividad tiene su propio ciclo de vida pero una aplicación puede estar en ejecución incluso cuando el proceso ha sido interrumpido (killed). Es decir, el ciclo de vida de una actividad no está atado con el ciclo de vida de los procesos. Los procesos son contenedores de actividades.

#### 2.5.4. Life Cycle (ciclo de vida de una Actividad)

Durante el periodo de vida de una Actividad en un programa Android puede encontrarse en uno de los estados representados en la Figura 2.13.

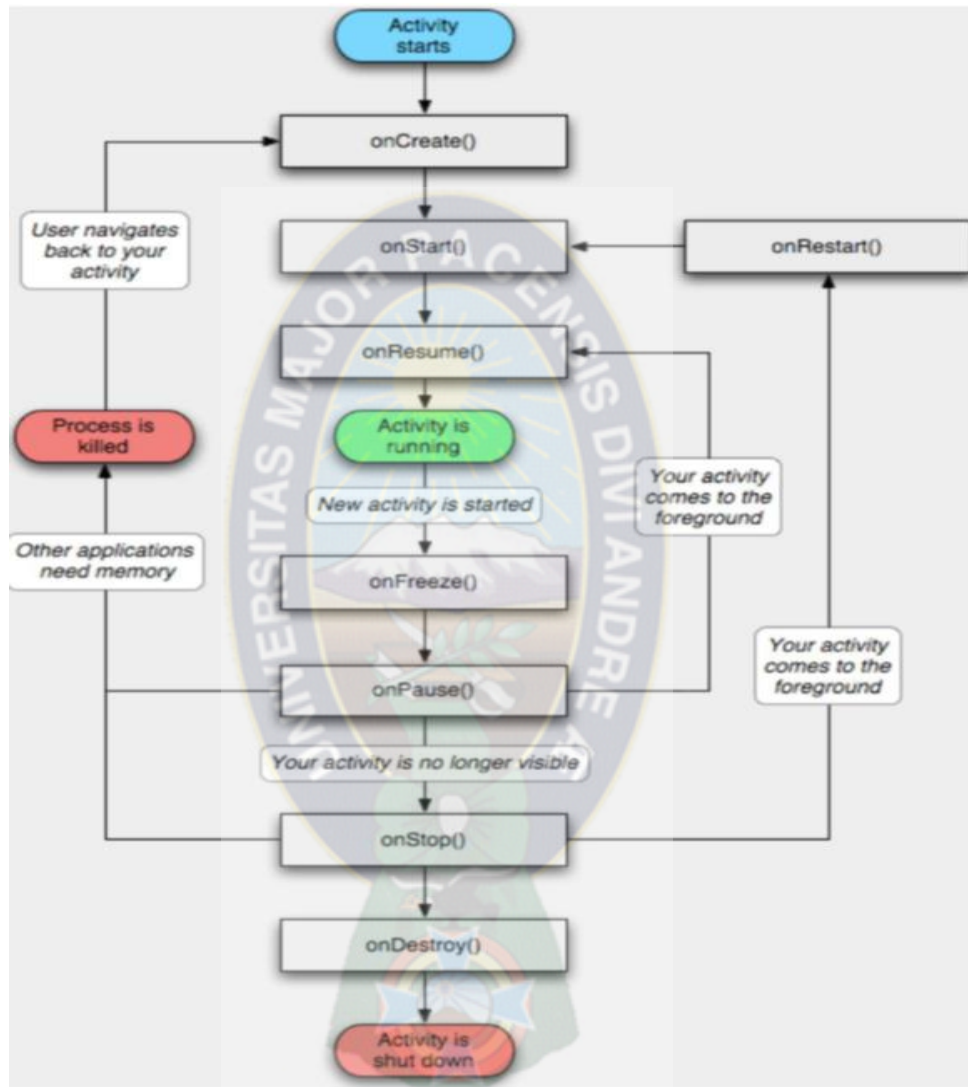


Figura 2.12: Ciclo de vida de una Actividad

Fuente: <http://www.android.com>, 2015

El usuario no tendrá el control sobre el estado en el que se encuentra el programa. Todo esto vendrá gestionado por el sistema. Sin embargo se recibirán notificaciones de cambio de estado.

Se deberán de implementar estos métodos que citaremos a continuación en la actividad principal y el sistema se encargará de llamarlos a su debido tiempo.

- `onCreate()`: este método es llamado cuando se inicia la actividad. El método tiene un valor inicial de null o alguna información previamente salvada por el método `onSaveInstanceState()`.
- `onStart()`: indica que la actividad va a ser mostrada al usuario.
- `onResume()`: se llama cuando la actividad puede empezar a interactuar con el usuario. Este es buen momento para ejecutar animaciones y música.
- `onPause()`: se llama a este método cuando la actividad va a ser desplazada a un segundo plano. Normalmente otra actividad va a ser lanzada a primer plano.
- `onStop()`: se llama cuando la actividad no es visible para el usuario y no se necesitará de momento. Si hay falta de memoria el método `onStop()` nunca será llamado y el sistema simplemente terminara con tu proceso.
- `onRestart()`: si este método es llamado significa que la actividad va a volver a ser mostrada al usuario desde un estado `onStop()`.
- `onDestroy()`: este método es llamado justo antes de que la actividad sea destruida. Si la memoria es escasa este proceso nunca será llamado. El sistema simplemente terminara con el proceso.
- `onSaveInstanceState()`: Android llamara a este método para permitir que la actividad salve un estado como la posición de un cursor en un campo de texto. Normalmente no se necesitara incluir este método debido a que el sistema guarda el estado de todo el entorno de usuario y todos los controles automáticamente.
- `onRestoreInstanceState()`: este método es llamado cuando una actividad está siendo recuperada desde un estado anterior salvada por `onSaveInstanceState()`. La implementación por defecto recupera el estado del entorno de usuario.

## 2.6. HERRAMIENTAS Y TECNOLOGIAS

Para la realización de este proyecto, ha sido necesario: el entorno de desarrollo Eclipse con el plugin ADT (Android Developer Tools) para la programación de la aplicación y Adobe Photoshop CS5 para el diseño de las interfaces.

### 2.6.1. Eclipse IDE+ADT plugin+SDK android

Para la implementación de la aplicación se va a emplear el Framework Eclipse, al que se le configurará el plugin ADT para poder trabajar con aplicaciones de Android, con el cual, Eclipse se convierte en una herramienta muy eficaz para el desarrollo de aplicaciones para Android.

Android Development Tools (ADT) es un plugin para Eclipse IDE que se ha diseñado para proporcionar un entorno poderoso e integrado para el desarrollo de aplicaciones Android.

ADT extiende las funcionalidades de Eclipse para ayudar en el desarrollo de aplicaciones Android, permitiendo crear interfaces de usuario de manera sencilla, añadir componentes basados en el API de Android, depurar las aplicaciones utilizando el SDK de Android o exportar ficheros .apk, firmados o sin firmar, para distribuir tu aplicación.

Para el desarrollo de una aplicación móvil en Android se emplea una arquitectura multicapa, cada una de las capas una funcionalidad concreta y bien definida.

- **Capa interfaz:** La capa de interfaz o de presentación de la aplicación está compuesta por las vistas definidas en ficheros XML. Es muy importante en la aplicación, ya que es la que se encarga de la presentación de la aplicación al usuario.
- **Capa de modelo de negocio:** Esta capa es la encargada de manejar los datos enviados por la capa de interfaz, es decir, en función de los eventos recibidos por la capa de presentación deberá tomar decisiones de modificar el modelo y mostrar los resultados. Aquí se implementa toda la lógica de negocio. En la aplicación, esta capa se implementa en Java.



- **Capa de datos:** Es la encargada de acceder a los datos según reciba las solicitudes de la capa anterior. En el caso concreto de esta aplicación se cargarán los datos desde el inicio en unas clases globales que serán accesibles en cualquier contexto.

### 2.6.2. Adobe Photoshop CS5

Para el diseño de las interfaces de usuario de la aplicación será de gran ayuda esta herramienta ya que es muy potente y fácil de utilizar. Se empleará para el diseño de iconos, cabeceras, botones y demás elementos estéticos de la aplicación.

### 2.6.3. Json

JSON es una notación Javascript para escribir objetos que se ha hecho bastante popular en el mundo del desarrollo de webs y que se utiliza en diversos lenguajes de programación, componentes (habitualmente Ajax), etc. Su éxito se debe a que es una excelente forma para almacenar información que deseamos compartir entre distintos componentes o lenguajes de las aplicaciones web. Si trabajamos con Ajax y alguno de los frameworks Javascript existentes, ya habremos notado esta posibilidad y posiblemente ya estemos utilizando JSON o una notación similar.

JSON, cuyas siglas significan JavaScript Object Notation (Notación de Objetos de JavaScript), es un formato ligero, fácil de escribir o codificar, así como también es fácil de leer por los seres humanos. Desde Javascript podemos procesar directamente cualquier objeto

JSON y existen librerías para la mayoría de los lenguajes de programación que tienen funciones para interpretar este formato. Por ello se ha adoptado universalmente. Para más información podemos visitar el sitio web de JSON en <http://www.json.org/>.

Esto quiere decir que con JSON podemos comunicar datos fácilmente entre scripts Javascript y scripts PHP. Por ejemplo, pensemos en una validación de formulario que se desea hacer con Ajax. Los datos del formulario se pueden enviar a PHP por medio de POST y luego podríamos desde PHP enviar a Javascript el resultado de validar esos datos

en el servidor. Como la validación puede ser positiva o negativa, así como puede tener más o menos códigos de error y acciones a realizar dependiendo de la información procesada, el script PHP tiene que mandar una respuesta más o menos elaborada al script Javascript y una posibilidad es enviar esos datos desde PHP utilizando la notación JSON.

#### **2.6.4. Web Service**

Es una pieza de software que usa para su funcionamiento un conjunto de estándares y protocolos que sirven para intercambiar datos entre aplicaciones. Diversas aplicaciones de software desarrolladas en lenguajes de programación distintos, y ejecutadas sobre cualquier plataforma, pueden usar los servicios web para intercambiar datos en redes como Internet. Mediante la adopción de estándares abiertos se consigue la ansiada interoperabilidad.

##### **2.6.4.1. Estándares empleados**

- *SOAP (Simple Object Access Protocol)*: Protocolos sobre los cuales se establece el intercambio.
- *Web Services Protocol Stack*: Denominado al conjunto de servicios y protocolos de los servicios Web.
- *Otros Protocolos*: los datos en XML pueden también enviarse de una aplicación a otra usando protocolos normales como HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), o SMTP (Simple Mail Transfer Protocol).

##### **2.6.4.2. Ventajas de los Servicios Web**

Fomentan los estándares y protocolos basados en texto, los cuales hacen más fácil acceder a su contenido y entender sus principios y funcionamiento.

Permiten interoperabilidad entre aplicaciones de software de forma independiente de sus propiedades o de las plataformas en las que éstas se encuentren instaladas.

De igual manera, aportan para que los servicios y software de distintas compañías ubicadas en diferentes sitios geográficos puedan ser combinados sin problema para proveer servicios integrados.

#### **2.6.4.3. Razones para crear Servicios Web**

Principalmente la razón para usar Web services es que pueden usarse con el protocolo HTTP sobre TCP en el puerto 80. Debido a que las organizaciones protegen sus redes mediante firewalls, que bloquean y filtran gran parte del tráfico hacia y desde Internet, mantienen cerrados casi todos los puertos TCP excepto el 80, que es, justamente, el que usan los navegadores. Hay que señalar que los servicios web se pueden usar sobre cualquier tipo de protocolo, de cualquier manera, TCP es el más comúnmente utilizado.

### **2.7. LENGUAJES DE PROGRAMACION**

Los lenguajes de programación que se van a utilizar para el desarrollo de este proyecto son: Java y XML. A continuación se introducirá brevemente cada uno:

#### **2.7.1. JAVA**

Java es un lenguaje de programación orientado a objetos que permite realizar todo tipo de aplicaciones, el cual es de uso libre y gratuito, y está enfocado a cubrir las necesidades tecnológicas más punteras. Actualmente es propiedad de la empresa Oracle, la cual adquirió a Sun Microsystems.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma, dado que es un lenguaje interpretado, diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes de compilado, lo que posibilita que un código interpretado pueda ser ejecutado en cualquier otro tipo de sistema operativo, si se cuenta con un intérprete codificado para ese sistema operativo.



Las aplicaciones para Android se escriben en este lenguaje, con la única diferencia que al SDK de Java se añaden las funciones propias del Android SDK que le permiten adaptarse y controlar dispositivos con sistema operativo Android.

### **2.7.2. XML**

El lenguaje XML estará muy presente en el desarrollo de la aplicación, ya que se utilizará tanto para el diseño de las interfaces de usuario de Android, como para el almacenamiento de los datos estáticos de la aplicación.

No será necesaria ninguna herramienta adicional para el trabajo con los XML ya que Eclipse también da soporte en el trabajo con este formato de ficheros.

### **2.7.3. PHP**

El lenguaje PHP (cuyo nombre es acrónimo de PHP: Hipertext Preprocessor) es un lenguaje interpretado con una sintaxis similar a la de C++ o JAVA. Aunque el lenguaje se puede usar para realizar cualquier tipo de programa, es en la generación dinámica de páginas web donde ha alcanzado su máxima popularidad. En concreto, suele incluirse incrustado en páginas HTML (o XHTML), siendo el servidor web el encargado de ejecutarlo.

Algunas de las características de su enorme popularidad son:

- Es un lenguaje libre. Puede descargarse de <http://www.php.net>.
- Tiene una extensa documentación oficial en varios idiomas (disponible libremente en <http://www.php.net>).
- Existen multitud de extensiones: para conectar con bases de datos, para manejo de sockets, para generar documentos PDF, para generar dinámicamente páginas en Flash, etc.
- Al ejecutarse en el servidor, los programas PHP lo pueden usar todo tipo de máquinas con todo tipo de sistemas operativos.

- En caso de que un cliente falle (por error hardware, virus, etc) se puede seguir usando el sistema desde otro cualquiera que tenga un navegador web con conexión al servidor.

#### **2.7.4. HTML5**

El HTML5 (HyperText Markup Language, versión 5) es la quinta revisión del lenguaje de programación “básico” de la World Wide Web, el HTML. Esta nueva versión pretende remplazar al actual (X)HTML, corrigiendo problemas con los que los desarrolladores web se encuentran, así como rediseñar el código actualizándolo a nuevas necesidades que demanda la web de hoy en día.

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas APIs (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto que (como veremos más adelante) es extremadamente significativo.

Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones. Con la variedad de dispositivos para acceder a Internet y la diversidad de interfaces disponibles para interactuar con la web, un aspecto básico como la estructura se vuelve parte vital del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como los fundamentos de un edificio.

#### **2.7.5. Javascript**

Javascript es un lenguaje interpretado usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos Javascript, fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores

más exitosos fue transformar el código Javascript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje Javascript como la mejor opción para la web.

Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, Javascript fue expandido en relación con portabilidad e integración. A la vez, interfaces de programación de aplicaciones (APIs) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales. Estas nuevas APIs (como Web Storage, Canvas, y otras) son interfaces para librerías incluidas en navegadores. La idea es hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web.

## **2.8. GESTORES DE BASES DE DATOS**

### **2.8.1. Definición de Base de Datos**

Las bases de datos son un componente imprescindible en cualquier sistema informático, por tal razón debe de llevarse en paralelo la experiencia en desarrollo de aplicaciones en una plataforma de software y el desarrollo y modelado de base de datos, por lo cual, uno de los aportes más importantes y aplicables que ha traído la Informática a las actividades diarias de cualquier organización ha sido el concepto de Base de Datos, por cuanto supone la disciplina en la organización de los datos de un empresa, por lo que podemos definir las bases de datos como: “ Fondo común de información almacenada en una computadora para que cualquier persona o programa autorizado pueda acceder a ella, independientemente de su procedencia y del uso que haga”.

Como podemos observar en la definición de base de datos, nos presenta la conceptualización como una estructura de información, pero no habla de cómo se crea, se destruye y se dota de contenido. El sistema que suministra al usuario las herramientas que le permitan manipular y efectuar las operaciones mencionadas forma parte de los sistemas de gestión de base de datos.

## **2.8.2. Sistemas de Gestión de Base de Datos (SGBD)**

Es un software o conjunto de programas que permite crear y mantener una base de datos. El SGBD actúa como interfaz entre los programas de aplicación (Usuarios) y el sistema operativo. El objetivo principal de un SGBD es proporcionar un entorno eficiente a la hora de almacenar y recuperar la información de la base de datos.

Este software facilita el proceso de definir, construir y manipular bases de datos para diversas aplicaciones.

De esta forma podemos identificar las siguientes instrucciones fundamentales que debe proveer un sistema de gestión de base de datos (SGBD).

- Crear, modificar, eliminar obtener la estructura asociada al esquema lógico de una base de datos.
- Instanciar datos operativos en una base de datos, modificar dichas instancias, eliminarlas recuperarlas bajo diferentes criterios de búsqueda.

Comprendido el concepto de base de datos y sistema de gestión de base de datos se puede afirmar con claridad que se necesitará de estas dos herramientas para el almacenamiento y tratamiento de la información para la geolocalización de cajeros automáticos y agencias bancarias. Evaluando alternativas de software libre, se buscaron SGBD que se ajusten a cualquier necesidad que involucre factores técnicos y económicos, en el argumento monetario se podrá obtener una muy buena reducción de costos ya que el software libre por lo general es muy económico y en muchas ocasiones tiende a cero, para este caso, MySQL y SQLite se consiguen de manera gratuita en cualquier distribución de Linux o en la red.

### **2.8.2.1. MySQL**

Es un sistema de administración de base de datos relacionales (SGBDR) rápido, robusto y fácil de usar. Se adapta bien a la administración de datos en un entorno de red, especialmente en arquitectura cliente/servidor. Se proporciona con muchas herramientas y es compatible con muchos lenguajes de programación. Es el más

célebre SGBDR del mundo Open Source, en particular gracias a su compatibilidad con el servidor de páginas Web Apache y el lenguaje de páginas Web dinámicas PHP. Las razones para escoger MySQL como solución para la administración de datos:

- *Coste:* El coste de MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- *Asistencia:* MySQL AB, ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.
- *Velocidad:* MySQL es mucho más rápido que la mayor parte de sus rivales.
- *Funcionalidad:* MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, Compatibilidad para la mayor parte SQL ANSI, volcado online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación. Así mismo, se desarrolla y actualiza de forma mucho más rápida que muchos de sus rivales, por lo que prácticamente todas las funciones estándar de MySQL todavía no están en fase de desarrollo.
- *Portabilidad:* MySQL se ejecuta en la inmensa mayoría de sistemas operativos y l mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- *Facilidad de uso:* MySQL resulta fácil de utilizar y de administrar. Gran parte de las viejas bases de datos presentan problemas por utilizar sistemas obsoletos, lo que complica innecesariamente las tareas de administración.

#### **2.8.2.2. SQLite**

SQLite es una fuente abierta de base de datos relacional integrada. Originalmente lanzado en 2000, fue diseñado para proporcionar una manera conveniente la gestión de base de datos para aplicaciones sin el sistema dedicado de base de datos relacionales. Hoy en día existe una gran variedad de productos de base de datos



relacionales en el mercado diseñado específicamente para sus usos incrustados, subproductos tales como Sybase SQL Anywhere, InterSystems Caché, Pervasive PSQL, y el motor de Microsoft Jet. Algunos fabricantes han adaptado sus grandes bases de datos para crear variantes incorporadas. Ejemplos de estos IBM's DB2 Everyplace, 10g de Oracle, y el motor de Microsoft SQL Server Desktop. La fuente abierta de bases de datos MySQL y Firebird Ambas versiones ofrecen también integración. De todos estos productos, sólo dos son de código abierto y libres de derechos de licencia: Firebird y SQLite. De estos dos últimos, sólo uno está diseñado exclusivamente para su uso como una base de datos integrada: SQLite.

### **Características sobresalientes de SQLite**

SQLite utiliza el lenguaje de programación ANSI C. Es mantenible fácilmente y razonablemente rápido. Tiene características como el ser agradable, identificable, y digno de elogio, razones para la cual Android lo ha adoptado como SGBD principal y misma que se utilizará para almacenar y procesar los datos de geolocalización en el dispositivo móvil.

Otras características propias de SQLite que conviene mencionar son las siguientes:

*Cero Configuración:* No es necesario realizar ningún paso de instalación o configuración independiente para inicializar el software de gestión de la base de datos SQLite antes de usarlo.

*Embebido:* No es necesario un proceso de servidor independiente dedicado a SQLite. La librería SQLite es integrable en sus propias aplicaciones, en las cuales no es necesario incluir algún sistema de comunicación entre procesos para interactuar con SQLite.

*Interfaz de aplicación:* SQLite proveer un entorno SQL de aplicaciones C para manipular bases de datos. Proporciona un conjunto de Interfaz de programación de aplicaciones (API) para SQL dinámico, se puede montar sentencias SQL sobre la marcha y pasar hacia abajo en la interfaz para su ejecución.

*Soporte transaccional:* Admite las propiedades básicas de transacciones, es decir, atomicidad, consistencia, aislamiento y durabilidad (ACID). No se requieren acciones de los usuarios o los administradores de bases de datos sobre una caída del sistema o fallo de red para recuperar bases de datos.

*Multiproceso:* es una biblioteca de subprocesos y muchos hilos, con un proceso de aplicación se puede acceder a la misma base de datos o diferentes al mismo tiempo. SQLite se encarga de la concurrencia de hilos a nivel de base de datos.

*Personalización:* SQLite proporciona un marco adecuado en el que se puede definir y utilizar funciones de SQL hechas a medida.

*Unicode:* SQLite admite la codificación UTF-8 y UTF-16 basado en estándares para texto Unicode.

*Requerimiento de memoria:* Aunque SQLite puede utilizar una cantidad ilimitada de pila y un montón de espacios, puede ser hecho para funcionar con un mínimo de espacio de pila de 4 Kb y aproximadamente 100 Kb de almacenamiento dinámico. Esta característica es muy eficaz para pequeños dispositivos móviles que están restringidas con una pequeña cantidad de la memoria principal.

*Multiplataforma:* SQLite se ejecuta en Linux, Windows, Mac OS, OS/2 OpenBSD, y otros sistemas operativos. También se ejecuta en sistemas operativos embebidos, como Android, Symbian, Palm, Vxworks.

## **2.9. METODOLOGÍA Y TÉCNICA DE DESARROLLO**

### **2.9.1. Metodología para el Desarrollo de Aplicaciones Móviles (MDAM)**

La metodología propuesta para el desarrollo de aplicaciones para móviles se fundamenta en la experiencia de investigaciones previas en aplicaciones móviles, la evaluación del potencial de éxito para servicios de tercera generación denominada 6 M, la ingeniería de software educativo con modelado orientado por objetos (ISE-OO), y principalmente en los valores de las metodologías ágiles.



De la ISE-OO se hereda el enfoque de los micromundos más utilizados en los servicios móviles interactivos son: Mundo, Escenarios, Personajes y Roles, Argumento e Historia, Variables Compensatorias, Variables de Control, Variables de Resultado, Zonas de Comunicación, Ambientación-Characterización, Recuperación de Estados Anteriores, Manejo de Información del Usuario, Mecanismos para Análisis de Desempeño, Ampliación de las Posibilidades del Micromundo, Personalización del Ambiente y, Soporte a la Comunicación en Grupo, entre otros (Gomez, Galvis y Mariño, 1998).

De las metodologías ágiles se heredan los conceptos inmersos en los cuatro postulados o manifiesto ágil (Beck et al., 2001).

- Desarrollar software que funciona, más que conseguir buena documentación.
- La respuesta ante el cambio, es más importante que el seguimiento de un plan.
- Colaboración con el cliente sobre negociación contractual.
- Individuos e interacciones sobre procesos y herramientas.

De las 6 M's se extrae la concepción de que las aplicaciones móviles deben garantizar el cumplimiento de las necesidades de los usuarios. La 6 M's debe su nombre a los seis atributos que se miden para evaluar el éxito del servicio propuesto son: Movement, Moment, Me, Multi-user, Money and Machines (Ahonen, Barret y Golding, 2002).

- *Movement* (Movimiento): un servicio móvil debe ser “móvil” por naturaleza, la ubicación debe ser una parte integral del servicio.
- *Moment* (Momento): un servicio que cuente con este atributo debe estar disponible en cualquier instante de tiempo en que el usuario desee usar dicho servicio.
- *Me* (Yo): se refiere al nivel de personalización de un servicio.
- *Multi-user* (Multiusuario): Busca extenderse dentro de la comunidad, que el servicio sea interactivo y que pueda utilizarse por múltiples usuarios de manera simultánea.
- *Money* (Dinero): como cualquier acción comercial, un servicio móvil tiene un fin lucrativo, ya sea para el operador, para el proveedor del servicio o para el usuario.
- *Machines* (Maquinas): la tecnología (terminal o redes) siempre es el factor que posibilita o limita; el atributo máquina busca añadir potencia a los dispositivos de

última generación que cada vez tienen mayor prestaciones a nivel de hardware y software.

### 2.9.2. Fases de la Metodología para el Desarrollo de aplicaciones Móviles (MDAM)

La metodología se encuentra enmarcada en cinco fases como se muestra en la figura 2.9, denominadas: análisis, diseño, desarrollo, pruebas de funcionamiento y entrega. A continuación se describe cada una de las actividades que intervienen en el desarrollo de la propuesta.



Figura 2.13: Etapas de la Metodología para el Desarrollo de Aplicaciones Móviles

Fuente: Elaboración propia

#### 1) FASE DE ANÁLISIS

En esta fase se analizan las peticiones o requerimientos de las personas o entidad para la cual se desarrolla el servicio móvil “Cliente”, el propósito es definir las características del mundo o entorno de la aplicación. Se realizan tres tareas: obtener requerimientos, clasificar los requerimientos y personalizar el servicio.

- **Obtener requerimientos:** se sugiere hacer una serie de entrevistas al cliente, para que manifieste los síntomas del problema o necesidades que se pretenden solucionar con las tecnologías móviles, o simplemente, para que señale las características que debe tener la aplicación.
- **Clasificar los requerimientos:** una vez identificados los requerimientos que debe tener el software, se procede a clasificarlos. Dichos requerimientos se pueden clasificar en entorno, mundo, funcionales y no funcionales.
  - **El entorno:** se refiere a todo lo que rodea al servicio. Por ejemplo, las características técnicas del dispositivo móvil del cliente, el sistema operativo subyacente (móvil y servidores), la tecnología utilizada para la transferencia de información, el Sistema Manejador de Base de Datos, *Data Base Management System* (DBMS), si se requiere, el formato de archivos y, otros módulos tecnológicos utilizados para el servicio.
  - **El mundo:** es la forma cómo interactúan el usuario y la aplicación. Aquí se encuentran los requerimientos de la Interfaz Gráfica de Usuario, *Graphical User Interface* (IGU), la forma en que el software va a generar los datos de salida, el formato de los datos y los demás requerimientos que involucren la comunicación hombre-máquina, considerando la gama tecnológica de los teléfonos móviles de los usuarios a la que va dirigida el servicio.
  - **Los requerimientos funcionales:** son todos aquellos que demandan una función dentro del sistema. Se deben definir claramente cada una de las tareas que debe realizar la aplicación.
  - **Los requerimientos no funcionales:** son la estabilidad, la portabilidad, el rendimiento, el tiempo de salida al mercado y, el costo, entre otros.
- **Personalizar el servicio:** adicionalmente se deben analizar aspectos de la cotidianidad del cliente como preferencias, costumbres y particularidades del usuario, con el propósito de garantizar la aceptación del servicio.

## 2) FASE DE DISEÑO

El objetivo de esta etapa es plasmar el pensamiento de la solución mediante diagramas o esquemas, considerando la mejor alternativa al integrar aspectos técnicos, funcionales, sociales y económicos. A esta fase se retorna si no se obtiene lo deseado en la etapa prueba de funcionamiento.

Se realizan cuatro actividades en esta fase: definir el escenario, estructurar el software, definir tiempos y asignar recursos.

- **Definir el escenario:** las aplicaciones móviles se pueden diseñar para ejecutarse en diferentes escenarios, dependiendo del sistema de conexión y sincronización con el servidor o aplicación central; el proceso de sincronización se realiza para insertar, modificar o borrar información. Entre los diferentes escenarios se encuentran los siguientes:
  - **Desconectado:** los procesos se realizan en el dispositivo móvil desconectado, después de terminar el proceso, si se requiere, puede conectarse con una aplicación central mediante el proceso de sincronización.
  - **Semiconectado:** los procesos pueden ejecutarse en el dispositivo móvil desconectado, pero se requiere establecer conexión en algún momento para terminar el proceso, al sincronizar la información con el servidor o aplicación central. En los escenarios desconectado y semiconectado se recomienda utilizar los protocolos y tecnologías que se ajusten al servicio y capacidades tecnológicas del dispositivo. Algunos son: *Media Transfer Protocol (MTP)*, *Near Field Communication (NFC)*, *SlowSync*, *FastSync*, *SyncML*, entre otros.
  - **Conectado:** el dispositivo debe estar siempre conectado con la aplicación central o servidor para su correcto funcionamiento, no se almacenan datos o archivos en el móvil, la sincronización se realiza mediante la validación de formularios, usualmente se utiliza el Protocolo de Transferencia de Hipertexto (*Hypertext Transfer Protocol, HTTP*).

- **Estructurar el software:** se deben utilizar algunos diagramas de Modelado de Lenguaje Unificado, *Unified Modeling Language* (UML), según las necesidades del proyecto, modelando el sistema desde varias perspectivas.

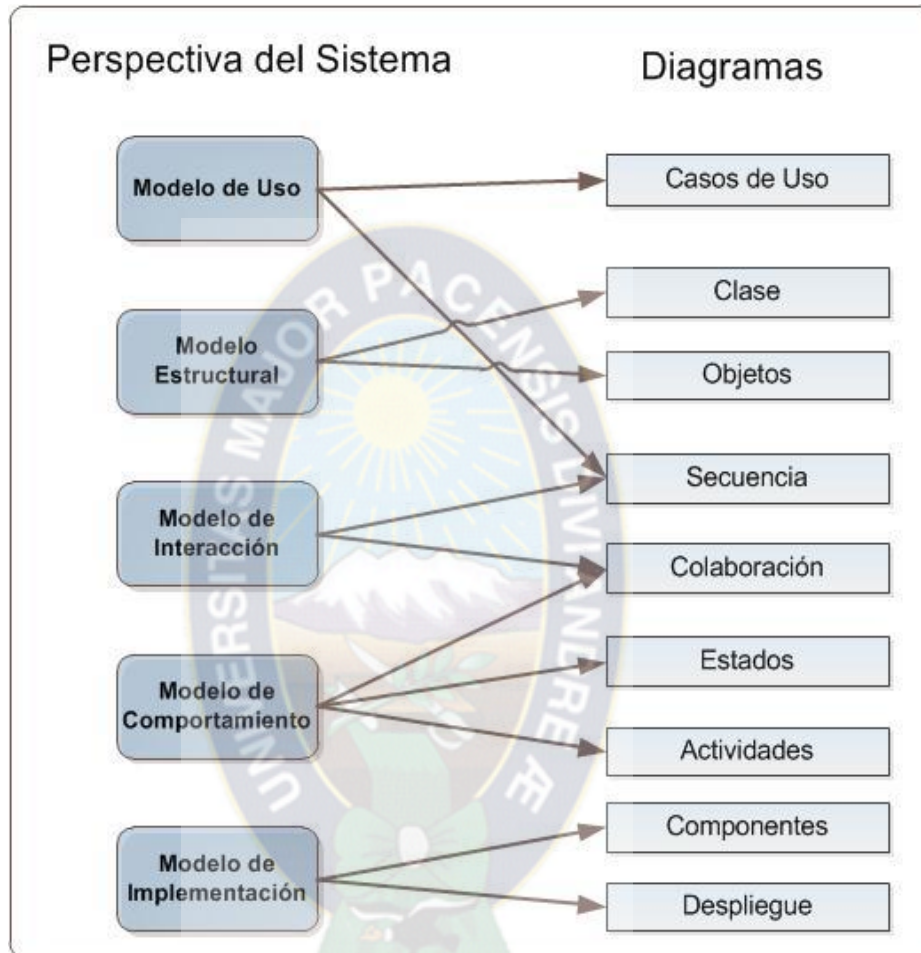


Figura 2.14: Posibles diagramas para el desarrollo de aplicaciones móviles

Fuente: Elaboración Propia

Se sugiere traducir los requerimientos obtenidos de la etapa anterior en un diagrama que describa en forma objetiva el servicio por implementar. Además, definir un patrón de diseño para flexibilizar, modular y reutilizar lo desarrollado; la selección del patrón de diseño debe estar acorde con el escenario del servicio. Algunos patrones que se ajustan a los escenarios de las aplicaciones móviles son: modelo vista de controlador, diseño de capas, entre otros.



- **Definir tiempos:** se establecen los plazos para cada una de las actividades restantes, con el objetivo de terminar la aplicación a tiempo para su salida al mercado. Se debe tener en cuenta el diseño computacional del software realizado en la tarea anterior y, las características volátiles y dinámicas de los servicios móviles.
- **Asignar recursos:** se asignan recursos para realizar cada actividad y alcanzar los objetivos propuestos, se deben considerar recursos humanos, financieros y tecnológicos. Además, se deben seleccionar las herramientas para el desarrollo.

### 3) FASE DE DESARROLLO

El objetivo de esta fase es implementar el diseño en un proyecto de software. En esta etapa se realizan las siguientes actividades:

- **Codificar:** se escribe en el lenguaje de programación seleccionado, cada una de las partes definidas en los diagramas realizados en la etapa de diseño.
- **Pruebas unitarias:** se verifica el funcionamiento de la aplicación. En primer lugar, se comprueba la correcta operación de cada elemento desarrollado (objeto, clase, actividad, documento, entre otros en forma individual); posteriormente, se pone en funcionamiento el conjunto de elementos, comprobando la interrelación entre ellos. Se ejecuta y se observan los resultados obtenidos, para compararlos con los esperados.
- **Documentar el código:** a medida que se codifica y se prueba cada elemento, se redacta la pequeña documentación sobre lo desarrollado.
- **Codificar ayudas:** además del manual de instalación y de usuario, deben existir una serie de ayudas que informen de manera didáctica lo que puede hacer el usuario con la aplicación. Estas ayudas deben ser codificadas en el mismo lenguaje de programación e integrada en la interfaz de aplicación para visualizarlas en el móvil.

### 4) FASE DE PRUEBA DE FUNCIONAMIENTO

El objetivo de esta fase es verificar el funcionamiento de la aplicación en diferentes escenarios y condiciones; para esto se realizan las siguientes tareas:

- **Emulación y simulación:** se realizan pruebas simulando el escenario y emulando el dispositivo móvil, explorando todas las utilidades y funciones de la aplicación, introduciendo diferentes datos, inclusive erróneos, para medir la funcionalidad y el nivel de robustez del software. Si se encuentran algunas fallas, se debe regresar a la etapa de codificación en la fase de desarrollo para solucionar los problemas, si las pruebas son satisfactorias se procede a la etapa de pruebas con dispositivos reales.
- **Dispositivos reales:** deben hacerse pruebas de campo en equipos reales para medir el desempeño y el rendimiento del aplicativo. Si se encuentran fallas en el tiempo de ejecución, si el software no cumple con los requerimientos especificados, o si el cliente solicita un cambio de última hora, hay que regresar a la fase de diseño para reestructurar y solucionar el inconveniente presentado.
- **Análisis de las 6 M's:** para valorar el potencial de éxito del servicio, se sugiere buscar un grupo de expertos en el campo del desarrollo móvil que utilicen el método de evaluación de las 6 M's, y califiquen la presencia de los seis atributos en la aplicación desarrollada.

Cualquier servicio que brinde un gran valor en cualquiera de las 6 M's tiene un buen potencial para el éxito como servicio móvil. Si la evaluación de las 6 M's del servicio es satisfactoria, se debe rediseñar el servicio fortaleciendo los atributos mencionados.

## 5) FASE DE ENTREGA

Terminada la depuración de la aplicación y atendidos todos los requerimientos de última hora del cliente se da por finalizada la aplicación y se procede a la entrega del ejecutable, el código fuente, la documentación y el manual del sistema.

- **Manuales:** el objetivo es el entrenamiento; una aplicación móvil debe constar de un manual del sistema móvil debe constar de un manual del sistema donde se indique el proceso de instalación, la atención a posibles fallas en el tiempo de ejecución y, las especificaciones técnicas mínimas de hardware y software que requiere el equipo, para el funcionamiento adecuado del aplicativo desarrollado.



- **Distribución:** se define el canal de comercialización de la aplicación, con el propósito de adecuar la aplicación al medio de distribución. A continuación se mencionan algunos de los canales de distribución existentes.

Las tiendas físicas u *outlets*, especializadas o no, corresponden a las tiendas que venden dispositivos y servicios de telecomunicaciones, normalmente operadores o marcas como Samsung, Apple y otros.

Los portales de operadores o desarrolladores de servicios, ofrecen un catálogo amplio de aplicaciones y ventas vía *Web Site* desde el PC, que luego son instaladas en el móvil.

Las *Applications Stores*, son las tiendas online de los fabricantes de dispositivos o de sistemas operativos.

### 2.9.3. Lenguaje Unificado de Modelado (UML)

En todas las disciplinas de la ingeniería se hace evidente la importancia de los modelos ya que describen el aspecto y la conducta de “algo”. Ese “algo” puede existir, estar en estado de desarrollo o estar, todavía, en un estado de planeación. Es en este momento cuando los diseñadores del modelo deben investigar los requerimientos pueden incluir áreas tales como funcionalidad, performance y confiabilidad. Además, a menudo, el modelo es dividido en un número de vistas, cada una de las cuales describe un aspecto específico del producto o sistema en construcción.

El modelado sirve no solamente para los grandes sistemas, aun en aplicaciones de pequeño tamaño se obtienen beneficios de modelado, sin embargo es un hecho que entre más grande y más complejo es el sistema, más importante es el papel que juega el modelado por simple razón: “El hombre hace modelos de sistemas complejos porque no puede entenderlos en su totalidad”.

UML, es una técnica para la especificación de sistemas en todas sus fases. Nació en 1994 cubriendo los aspectos principales de todos los métodos de diseño antecesores y, precisamente, los padres de UML son Grady Booch, autor del método Booch; James Rumbaugh, autor del método OMT e Ivar Jacobson, autor los métodos OOSE y Objectory. La versión 1.0 de UML fue liberada en Enero 1997.

#### **2.9.3.1. Beneficios de UML**

- Mejores tiempos totales de desarrollo (50% o más).
- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta utilización y minimización de costos.

#### **2.9.3.2. UML ¿Método o Lenguaje de Modelado?**

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de modelado. Un método es una manera explícita de estructurar el pensamiento y las acciones de cada individuo. Además, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que el lenguaje de modelado carece de estas instrucciones. Los métodos contienen modelos y esos modelos son utilizados para describir algo y comunicar los resultados del uso del método.

Un modelo es expresado en un lenguaje de modelado. Un lenguaje de modelado consiste de vistas, diagramas, elementos de modelos, los símbolos utilizados en los modelos y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos. Las reglas son sintácticas, semánticas y pragmáticas.

### **2.10. MÉTRICAS DE CALIDAD DEL SOFTWARE**

Se detallarán los criterios de calidad de ISO 9126, respecto a las Métricas Internas de la calidad del Producto de Software.

Presenta seis métricas de evaluación de software: Métricas de Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad y Portabilidad, las que a su vez se dividen en otros atributos que luego deben también ser evaluados por separado.

Para la evaluación del presente proyecto se considera las normas internas relacionadas a las siguientes métricas.

- Métricas de Funcionalidad
- Métricas de Usabilidad
- Métricas de Portabilidad
- Métricas de Mantenibilidad

#### **2.10.1. Funcionalidad**

Métrica para medir que si las funciones satisfagan necesidades declaradas o implícitas [ISO 9126, 1991], la que a su vez indica que se deben considerar los siguientes atributos:

- Adecuación
- Exactitud
- Interoperabilidad
- Seguridad
- Conformidad de la funcionalidad

Nombre:	Compleitud de implementación funcional
Propósito:	Qué tan completa está la implementación funcional.
Métodos de aplicación	Contar las funciones faltantes detectadas en la evaluación y comparar con el número de funciones descritas en la especificación de requisitos.

Medición, fórmula:	$X = 1 - A/B$ A: Número de funciones faltantes B: Número de funciones descritas en la especificación de requisitos.
Interpretación:	$0 \leq X \leq 1$
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ A: count B: count
Fuente de medición:	Especificación de requisitos Diseño Código fuente Informe de revisión
ISO/IEC 12207 SLCP:	Validación Revisión conjunta

Tabla 2.3: Métricas de Adecuidad

Fuente: [www.iso9126.com](http://www.iso9126.com)

### 2.10.2. Usabilidad

Se define la usabilidad como la capacidad de un producto software de ser comprendido, aprendido, usado y de ser atractivo para el usuario, en condiciones específicas de uno [ISO 9126]. Se debe tener en cuenta que la usabilidad no depende sólo del producto, sino también del usuario, para ello se define las siguientes métricas:

- Entendibilidad
- Aprendibilidad
- Operatibilidad
- Atractivo
- Conformidad de la usabilidad

Nombre:	Funciones Evidentes
Propósito:	Qué proporción de las funciones del sistema son evidentes al usuario.
Método de aplicación	Contar las funciones evidentes al usuario y comparar con el número total de funciones.
Medición, fórmula	$X = A/B$ A: Número de funciones (o tipos de funciones) evidentes al usuario B: Total de funciones (o tipos de funciones)
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, mejor
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ A: count B: count
Fuente de medición:	Especificación de requisitos Diseño Informe de revisión
ISO/IEC 12207 SLCP:	Verificación Revisión conjunta
Audiencia:	Requeridores Desarrolladores

Tabla 2.4: Métricas de Entendibilidad

Fuente: [www.iso9126.com](http://www.iso9126.com)

### 2.10.3. Portabilidad

Es la capacidad de un sistema para ser transferido de un entorno o plataforma a otro. [ISO 9126, 1991]. Las métricas relacionadas a la transportabilidad que define ISO 9126 se presentan a continuación:

- Adaptabilidad
- Instalabilidad
- Coexistencia
- Reempazabilidad
- Conformidad de la portabilidad

Nombre:	Conformidad de Portabilidad
Propósito:	Qué tan conforme es la portabilidad del producto con regulaciones, estándares y convenciones aplicables.
Método de aplicación	Contar los artículos encontrados que requieren conformidad y comparar con el número de artículos en la especificación que requieren conformidad.
Medición, fórmula	$X = A/B$ A: Número de artículos de conformidad. B: Total de artículos que requieren conformidad
Interpretación:	$0 \leq X \leq 1$ Entre más cercano a 1, más completa.
Tipo de escala:	Absoluta
Tipo de medida:	$X = \text{count}/\text{count}$ A: count B: count
Fuente de medición:	Especificación de conformidad y estándares, convenciones y regulaciones relacionadas. Diseño, Código fuente, Informe de revisión.
ISO/IEC 12207 SLCP:	Verificación Revisión conjunta
Audiencia:	Requeridores Desarrolladores

Tabla 2.5: Métricas de Portabilidad

Fuente: [www.iso9126.com](http://www.iso9126.com)



#### 2.10.4. Mantenibilidad

Este atributo mide la capacidad de un sistema para resistir ataques (tanto accidentales como intencionados) contra su seguridad. Para medir la integridad, se tienen que definir dos atributos adicionales: amenaza y seguridad.

- **Amenaza:** Es la probabilidad de que un ataque de un tipo determinado ocurra en un tiempo determinado.
- **Seguridad:** Es la probabilidad de que se pueda repeler el ataque de un tipo determinado Gilb [Gil, 1988]. La integridad del sistema se puede definir como:

$$Integridad = \sum [(1 - amenaza) * (1 - seguridad)]$$

Donde se suman la amenaza y la seguridad para cada tipo de ataque.





## **CAPÍTULO 3**

# **MARCO APLICATIVO**

---

## MARCO APLICATIVO

### 3.1. INTRODUCCIÓN

En este capítulo se inicia el análisis del desarrollo de la aplicación siguiendo la “Metodología para el Desarrollo de Aplicaciones Móviles” (MDAM) la cual fue incluida en cada una de sus cinco fases de desarrollo que se emplearan en la ejecución del proyecto de investigación.

### 3.2. ANÁLISIS METODOLÓGICO

La ingeniería de software, definida como el estudio de los principios y metodológicos para el desarrollo y mantenimiento de la aplicación móvil, nace ante la necesidad de utilizar una serie de operaciones, técnicas, herramientas y soporte documental a la hora de desarrollar la aplicación.

Dichos procedimientos pretenden guiar a los desarrolladores al crear nuevas aplicaciones para dispositivos móviles, pero los requisitos de cada uno son tan variados y cambiantes que ha dado lugar a un sinnúmero de metodologías.

### 3.3. METODOLOGÍA DE DESARROLLO

Para el desarrollo del presente proyecto, se ha planteado la “*Metodología para el Desarrollo de Aplicaciones Móviles*” (MDAM), dicha metodología fue descrita de manera detallada en el capítulo anterior.

Las fases que intervienen en esta metodología son las siguientes:

- Fase de Análisis
- Fase de Diseño
- Fase de Desarrollo
- Fase de Pruebas de funcionamiento
- Fase de Entrega

### 3.4. FASE DE ANALISIS

En esta fase se analizan las peticiones o requerimientos de los usuarios de las entidades financieras “Cliente”, el propósito es definir las características del mundo o entorno de la aplicación. Se realizan tres tareas: obtener requerimientos, clasificar los requerimientos y personalizar el servicio.

#### 3.4.1. Definición de Actores

Los actores del sistema involucrados en el desarrollo del presente proyecto de georreferenciación de cajeros automáticos y agencias bancarias, son principalmente el usuario y el administrador.

ACTOR	DESCRIPCIÓN
Administrador	Es el directo responsable de agregar y mantener actualizada las sucursales de una entidad financiera.
Usuario (terminal móvil)	Es el usuario que requiere los de información de las sucursales bancarias y cajeros automáticos.

Tabla 3.1: Definición de Actores

Fuente: Elaboración propia

#### 3.4.2. Requerimientos

Describiremos el prototipo a implementar a través de sus requerimientos. A estos los distinguiremos entre Funcionales y No funcionales. También necesitaremos una distinción entre las funcionalidades del cliente móvil, y las funcionalidades del administrador.

- ***Requerimientos Funcionales:*** Son los que definen el comportamiento interno del sistema: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran como los casos de uso serán llevados a la práctica.

- **Requerimientos No funcionales:** Especifican criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos.

Esta lista de requerimientos definirá luego los Casos de uso.

### 3.4.2.1. Requerimientos Funcionales

REQUERIMIENTO	DESCRIPCION	TIPO
<b>R1</b>	Ejecución de la Aplicación	Evidente
<b>R2</b>	Conexión con el servidor y la Base de Datos	Oculto
<b>R3</b>	Pantalla de Carga (Splascreen)	Evidente
<b>R4</b>	Carga del Mapa y los Scripts	Oculto
<b>R5</b>	Posicionamiento en la ubicación actual	Evidente
<b>R6</b>	Posicionamiento de los marcadores de la zona	Evidente
<b>R7</b>	Consultar en la base de datos sobre las sucursales de los cajeros automáticos y agencias bancarias de su preferencia.	Oculto
<b>R8</b>	Ver información de un marcador	Evidente
<b>R9</b>	Indicaciones a una dirección	Evidente
<b>R10</b>	Autenticación de los Administradores	Oculto
<b>R11</b>	Insertar en la base de datos de un nuevo marcador.	Oculto
<b>R12</b>	Actualizar información	Oculto
<b>R13</b>	Eliminar de la base de datos de un marcador que ya no existe.	Oculto

Tabla 3.2: Requerimientos Funcionales

Fuente: Elaboración propia

### 3.4.2.2. Requerimientos No Funcionales

REQUERIMIENTO	NOMBRE	DESCRIPCION
<b>RN1</b>	Tiempo de Respuesta	El sistema debe optimizar el tiempo de respuesta de conexión para el intercambio de datos con el servidor.
<b>RN2</b>	Aspectos de Interfaz Gráfica	El sistema debe contar con una interfaz gráfica amigable, previsible y fácil de utilizar.
<b>RN3</b>	Tolerancia a Fallas	El sistema debe poder recuperarse ante un error para evitar el cierre forzoso del mismo.
<b>RN4</b>	Abstracción de Hardware	El sistema debería ser soportado por la mayor gama de dispositivos con Android como sea posible.

Tabla 3.3: Requerimientos No Funcionales

Fuente: Elaboración propia

## 3.5. FASE DE DISEÑO

El objetivo de esta etapa es plasmar el pensamiento de la solución mediante diagramas o esquemas, considerando la mejor alternativa al integrar aspectos técnicos, funcionales, sociales y económicos. A esta fase se retorna si no se obtiene lo deseado en la etapa prueba de funcionamiento.

### 3.5.1. Estructura del Sistema

Para empezar a definir el diseño de la aplicación, es necesario hacer una vista rápida al montaje de toda la infraestructura. En este punto se puede ver globalmente como está diseñada la solución. La solución propuesta permite que el servidor que guarda la



información de las sucursales de las entidades financieras, y las preferencias de los usuarios. A continuación se muestra un esquema de la infraestructura.

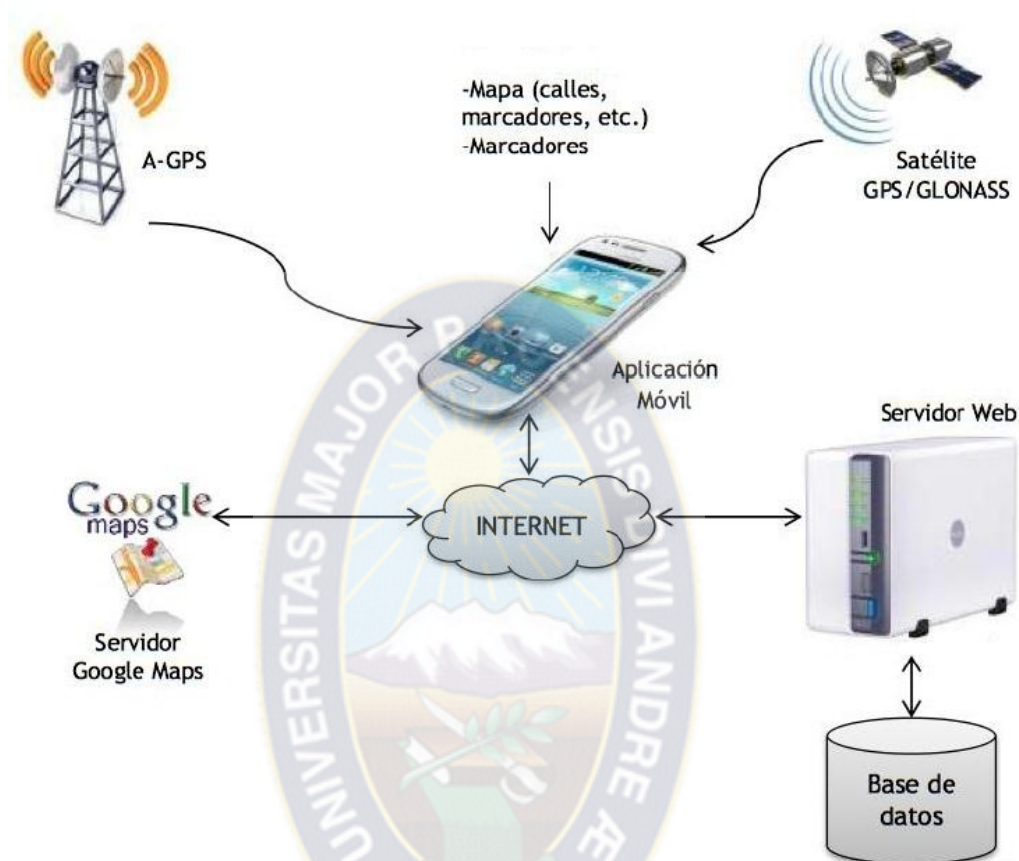


Figura 3.1: Estructura del Sistema

Fuente: Elaboración propia

Al ejecutarse la aplicación se solicita la posición mediante GPS, GLONASS o A-GPS de acuerdo a las características del móvil, una vez obtenida, se envía al servidor web junto con el radio en kilómetros para verificar sucursales cercanos a la posición y devuelve los marcadores al usuario sobreponiendo la información sobre la pantalla e interactuando cada vez que el móvil cambia de dirección.

El sistema requiere como entrada la ubicación del móvil con tres parámetros: latitud y longitud, la cual es devuelta por el GPS, GLONASS o A-GPS, y un radio de alcance en kilómetros. Los marcadores recibidos se muestran en tiempo real en la pantalla del dispositivo tomando en cuenta su ubicación geográfica mientras la aplicación esté activa.

### 3.5.2. Diseño de Navegación

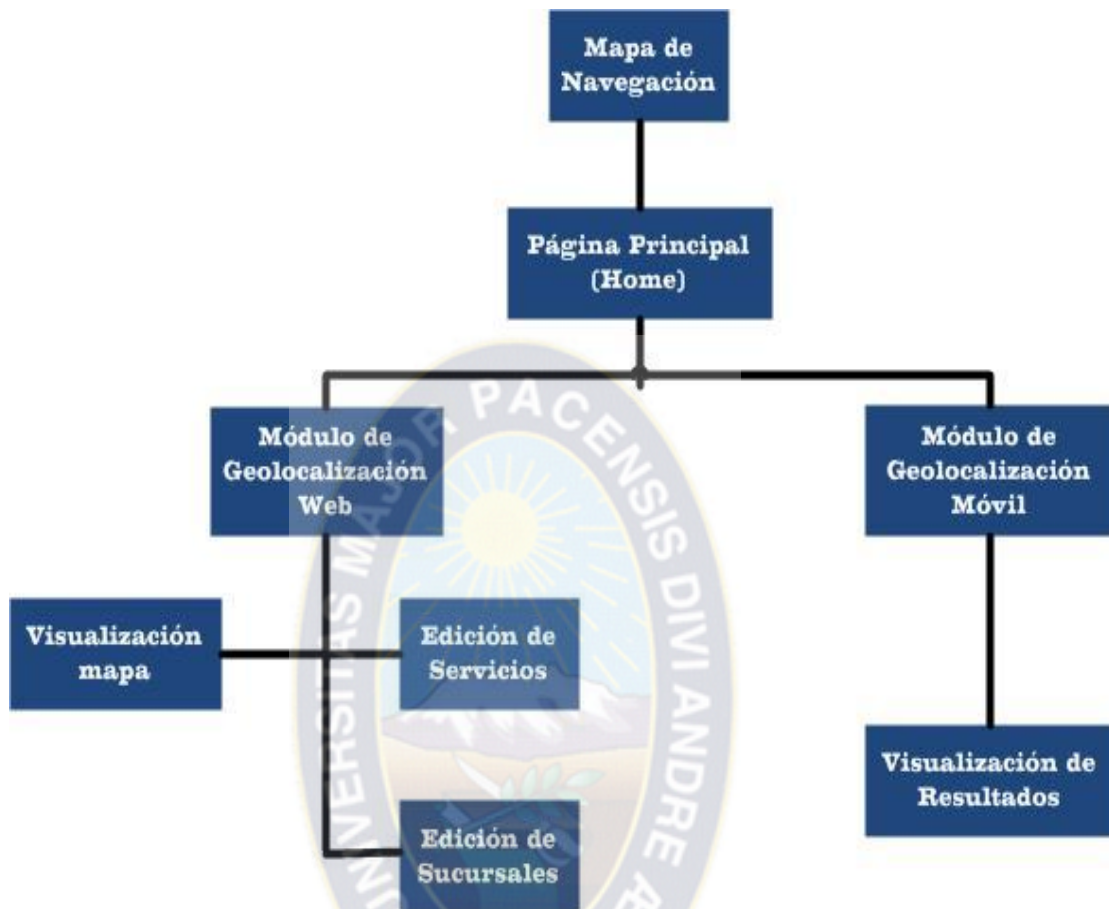


Figura 3.2: Mapa de Navegación

Fuente: Elaboración propia

### 3.5.3. Diagramas de Casos de Uso y Especificaciones

Como tal los casos de uso son un poderoso concepto que ayuda a un analista a comprender la forma en que un sistema deberá comportarse, le ayuda a obtener los requerimientos desde el punto de vista del usuario. Y esta se potencia al usar UML para visualizarlos de tal manera que se puede abstraer aun mayor información de estos.

Entre las ventajas que estos diagramas brindan, es la posibilidad de observar los confines entre la aplicación y el mundo exterior, mostrando más detalladamente la interacción con el sistema.

### Caso de Uso “Ingreso a la aplicación”

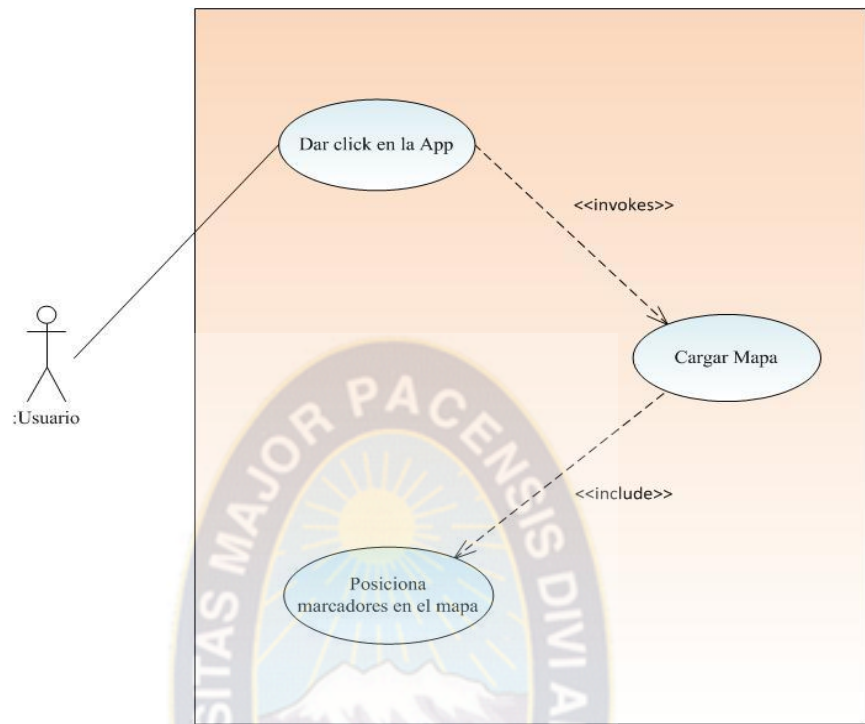


Figura 3.3: Caso de uso “ingreso a la aplicación”

Fuente: Elaboración propia

### Especificación de caso de uso “Ingreso a la aplicación”

CASO DE USO	Ingreso a la Aplicación	
Actores	Usuario	
Propósito	Ingresar a la móvil desde el menú de aplicaciones.	
Descripción	El usuario desde el menú de aplicaciones de Android inicia la aplicación, lo cual hace que se cargue el mapa desde el servidor de google maps, para luego posicionar los marcadores de los cajeros automáticos y agencias bancarias que se obtuvieron de la base de datos.	
Tipo	Primario y esencial.	
Referencias Cruzadas	R1, R2, R3, R4, R5, R6	

Tabla 3.4: Especificación caso de uso “ingreso a la aplicación”

Fuente: Elaboración propia

**Caso de Uso “Búsqueda de sucursales”**

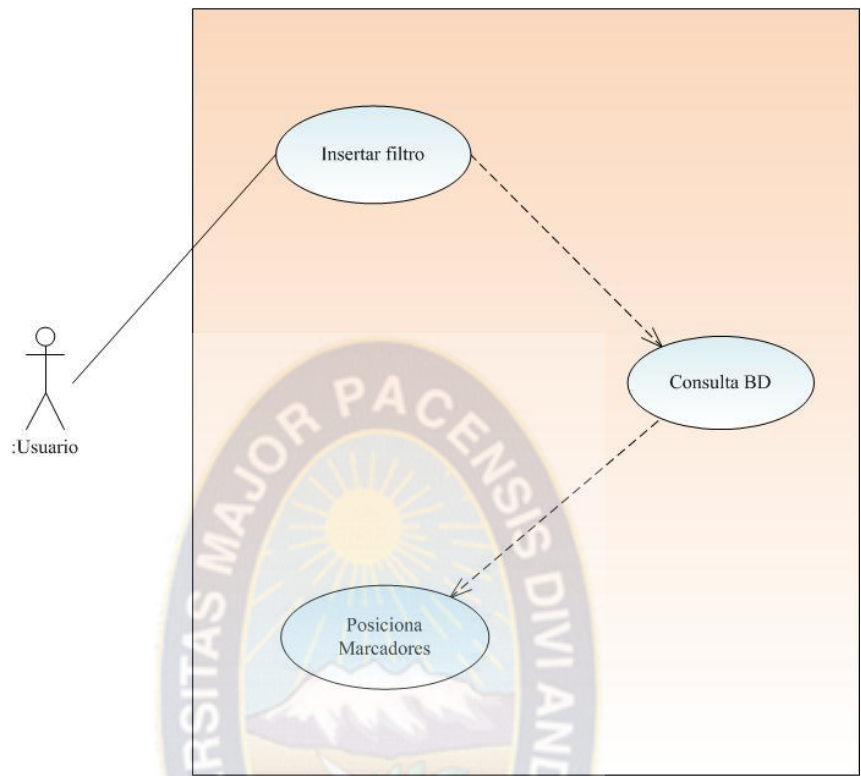


Figura 3.4: Caso de uso “búsqueda de sucursales”

Fuente: Elaboración propia

**Especificación de caso de uso “búsqueda de sucursales”**

CASO DE USO	Búsqueda de sucursales	
Actores	Usuario	
Propósito	Buscar sucursales de cajeros automáticos y agencias bancarias en etiquetas.	
Descripción	El usuario realiza una búsqueda de los cajeros automáticos y agencias bancarias que están a su alrededor o en una zona específica, para lo cual usa unas etiquetas predefinidas para realizar un filtrado.	
Tipo	Secundario.	
Referencias Cruzadas	R7, R6	

Tabla 3.5: Especificación caso de uso “búsqueda de sucursales”

Fuente Elaboración propia

**Caso de uso “Ver más”**

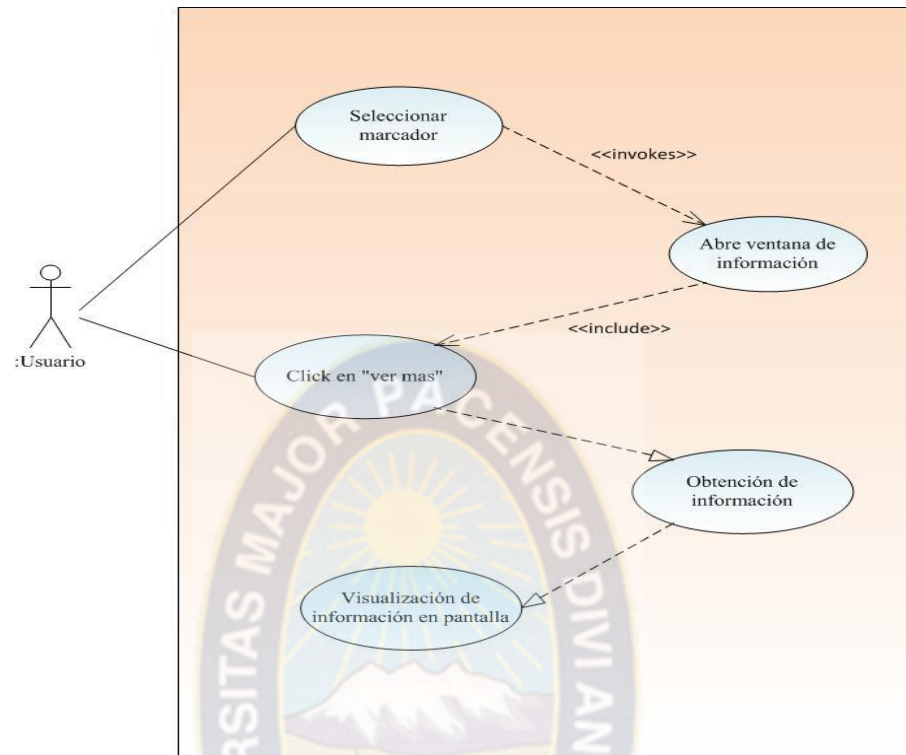


Figura 3.5: Caso de uso “ver más”

Fuente: Elaboración propia

**Especificación de caso de uso “ver más”**

CASO DE USO	Ver más	
Actores	Usuario	
Propósito	Ver la información detallada del cajero automático o agencia bancaria de su preferencia.	
Descripción	El usuario luego de realizar la búsqueda de marcadores, selecciona uno de su interés para ver su información básica, la cual es mostrada y brinda la opción “Ver más” con la cual se despliega la información detallada del cajero automático o agencia bancaria.	
Tipo	Secundario.	
Referencias Cruzadas	R8	

Tabla 3.6: Especificación de uso “Ver más”

Fuente: Elaboración propia

**Caso de uso para la “Ingreso a la plataforma de administración”**

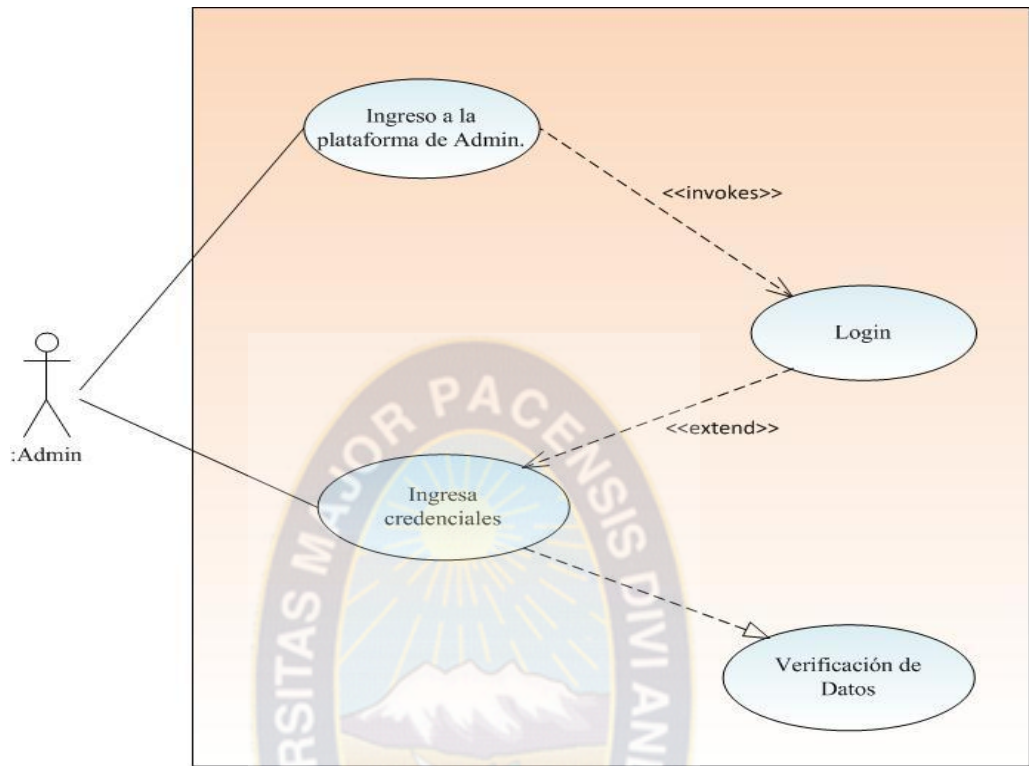


Figura 3.6: Caso de uso “ingreso a la plataforma de administración”

Fuente Elaboración propia

**Especificación de caso de uso “ingreso a la plataforma de administración”**

CASO DE USO	Ingreso a la plataforma de administración
Actores	Administrador.
Propósito	Entrar a la plataforma de administrador.
Descripción	El administrador ingresa a la plataforma web, donde como primera medida se debe identificar usando las credenciales de administrador (Username, Password) y si estas son correctas se accede a la plataforma de administración.
Tipo	Primario y esencial.
Referencias Cruzadas	R10

Tabla 3.7: Especificación caso de uso “ingreso administrador”

Fuente Elaboración propia



**Caso de uso “agregar sucursal”**

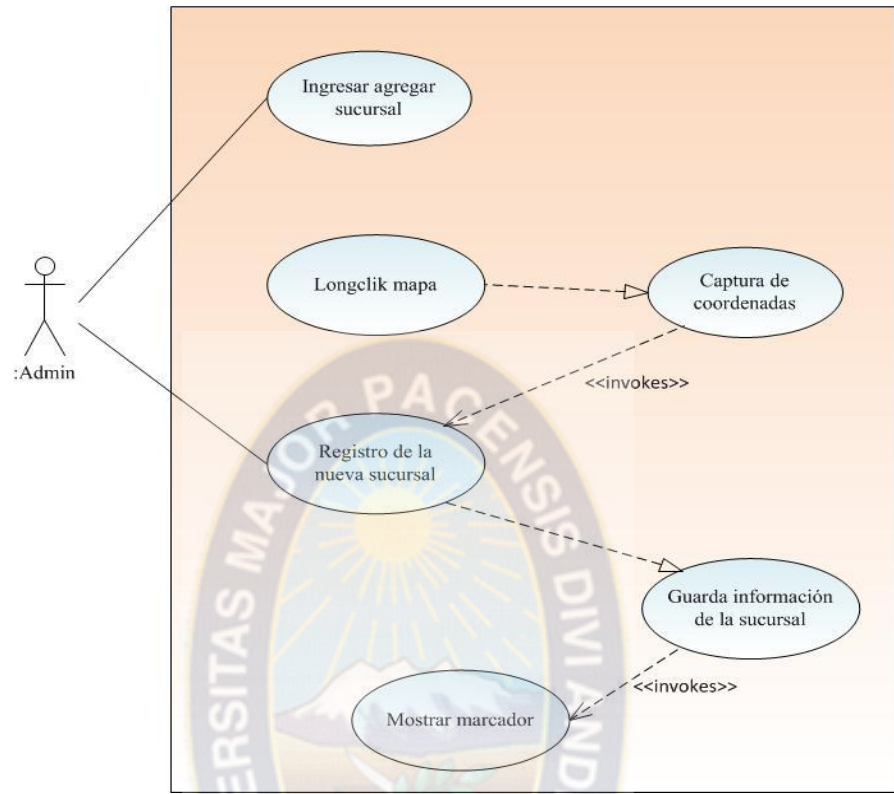


Figura 3.7: Caso de uso “agregar sucursal”  
Fuente: Elaboración propia

**Especificación de caso de uso “agregar sucursal”**

CASO DE USO	Agregar sucursal	
Actores	Administrador.	
Propósito	Agregar una nueva sucursal, se refiere a un nuevo cajero automático o agencia bancaria.	
Descripción	Desde la plataforma de administración el administrador, ingresa a la opción de “Agregar sucursal”, donde por medio de un “longclick” en la ubicación deseada, se llena un formulario para vincular la información de la sucursal.	
Tipo	Secundario.	
Referencias Cruzadas	R11	

Tabla 3.8: Especificación del caso de uso “agregar sucursal”  
Fuente: Elaboración propia

### Caso de uso “actualizar información”

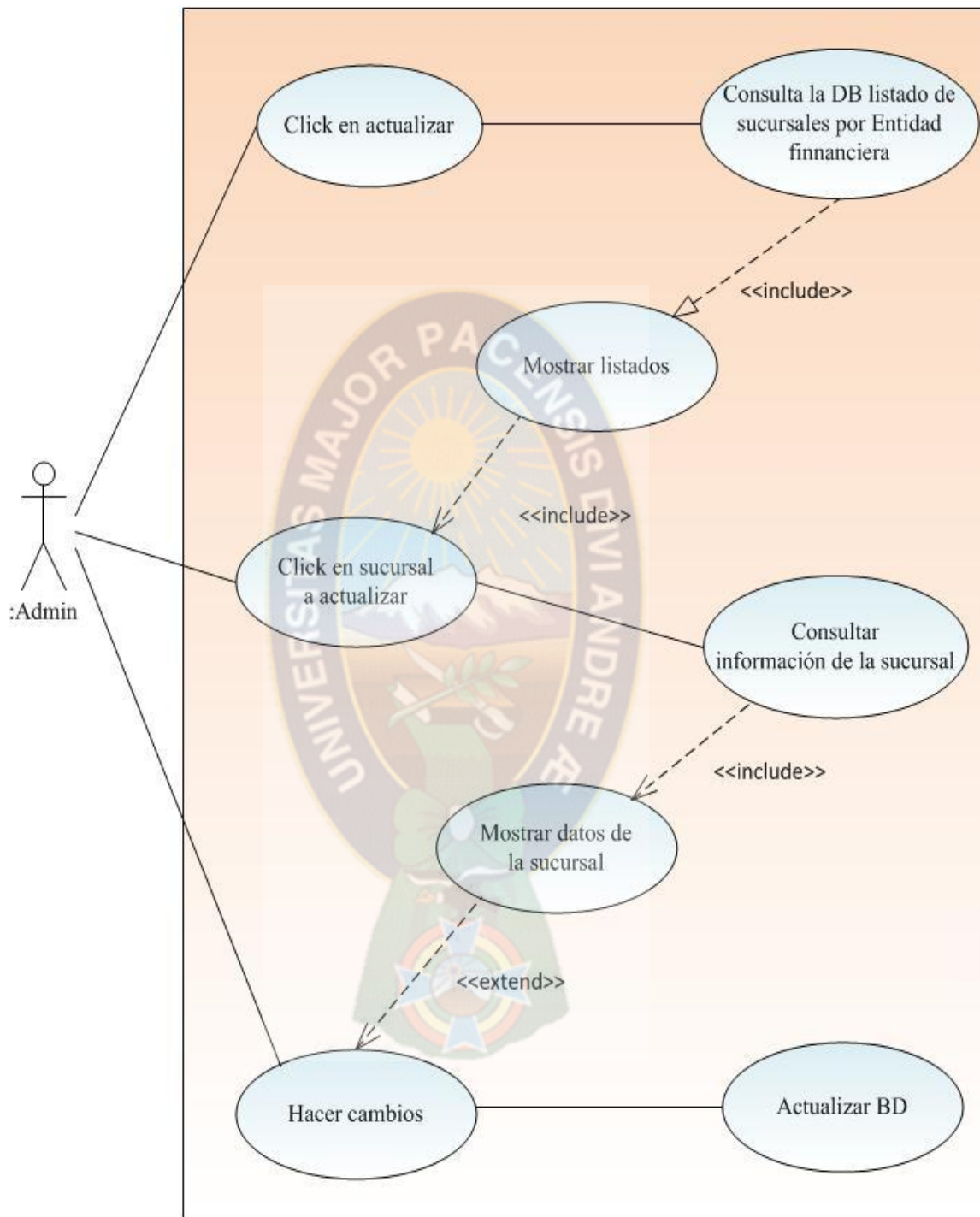


Figura 3.8: Caso de uso “actualizar información”

Fuente: Elaboración propia

### Especificación de caso de uso “actualizar información”

CASO DE USO	Actualizar información
Actores	Administrador.
Propósito	Actualizar o agregar información de las sucursales.
Descripción	El administrador desde el panel de administración, ingresa a la opción “Actualizar información”, luego procede a elegir dentro de la lista de las entidades financieras, al que desea actualizar la información elemental del lugar seleccionado, y el administrador procede a realizar los cambios para luego enviar esta información a la base de datos.
Tipo	Secundario.
Referencias Cruzadas	R12

Tabla 3.9: Especificación caso de uso “actualizar información”

Fuente: Elaboración propia

**Caso de uso “eliminar sucursal”**

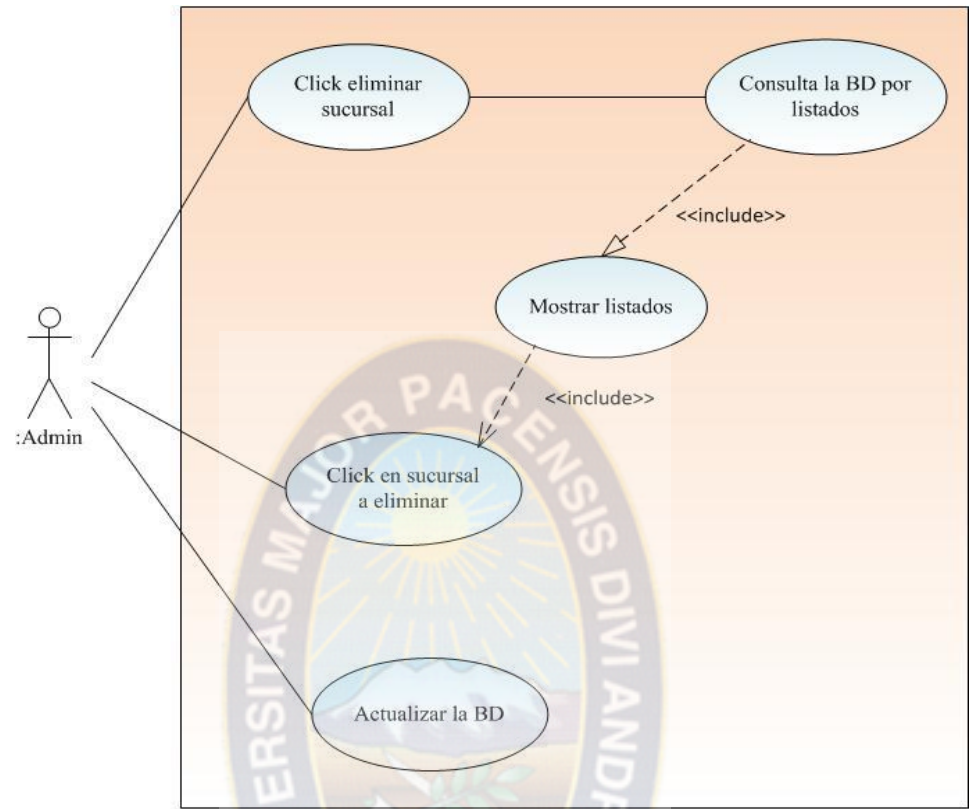


Figura 3.9: Caso de uso “eliminar sucursal”

Fuente: Elaboración propia

**Especificación de caso de uso “eliminar sucursal”**

CASO DE USO	Eliminar sucursal	
Actores	Administrador.	
Propósito	Eliminar sucursales.	
Descripción	El administrador desde el panel de administración, ingresa a la opción “Eliminar sucursal”, luego procede a elegir dentro de la lista de sucursales pertenecientes a entidades financieras, al que desea eliminar el marcador y la información de la base de datos.	
Tipo	Secundario.	
Referencias Cruzadas	R13	

Tabla 3.10: Especificación caso de uso “eliminar marcador”

Fuente: Elaboración propia

### 3.5.4. Diagramas de secuencia

El diagrama de secuencia UML muestra la mecánica de la interacción y la comunicación entre los objetos al transcurrir el tiempo, lo que brinda una importante dimensión al modelado, el tiempo; la idea primordial es que la ejecución en el sistema se realiza en una secuencia establecida y que esta secuencia se toma su tiempo de ir del principio al fin.

#### Secuencia “ingreso a la aplicación”

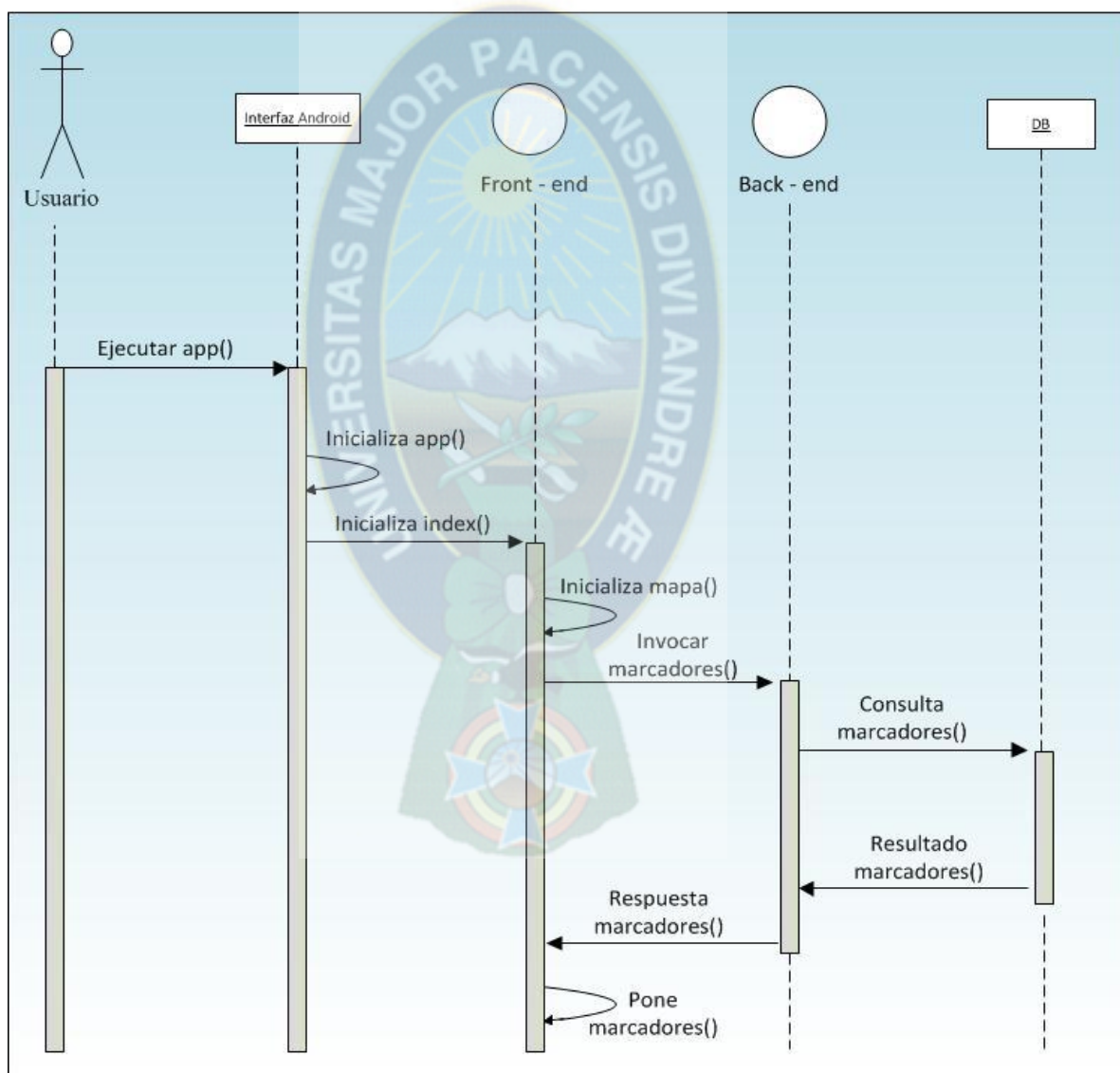


Figura 3.10: Secuencia “ingreso a la aplicación”

Fuente: Elaboración propia

### Secuencia “búsqueda de sucursales”

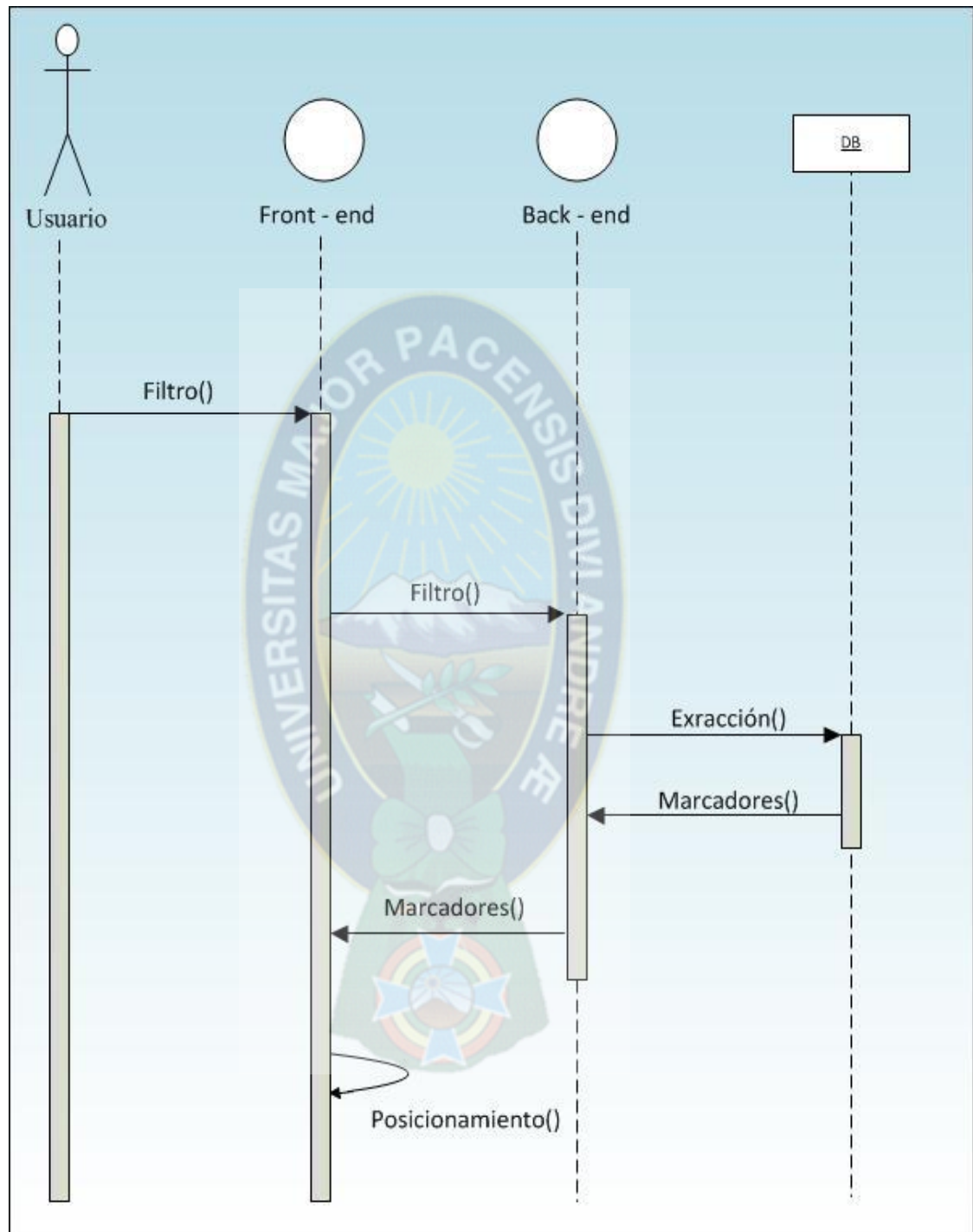


Figura 3.11: Secuencia “búsqueda de sucursales”

Fuente: Elaboración propia



### Secuencia “ver más”

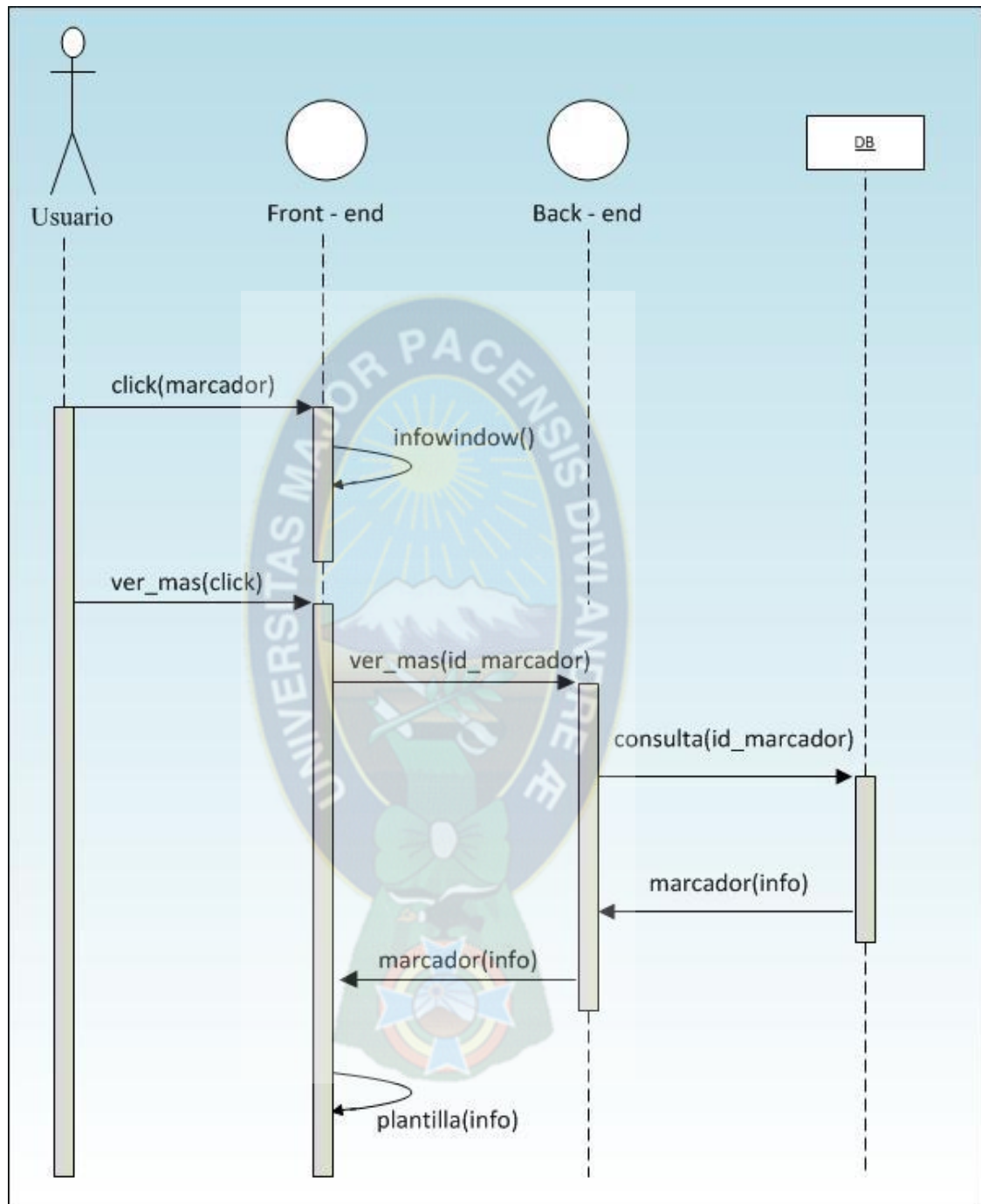


Figura 3.12: Secuencia “ver más”

Fuente: Elaboración propia

### Secuencia “ingreso administrador”

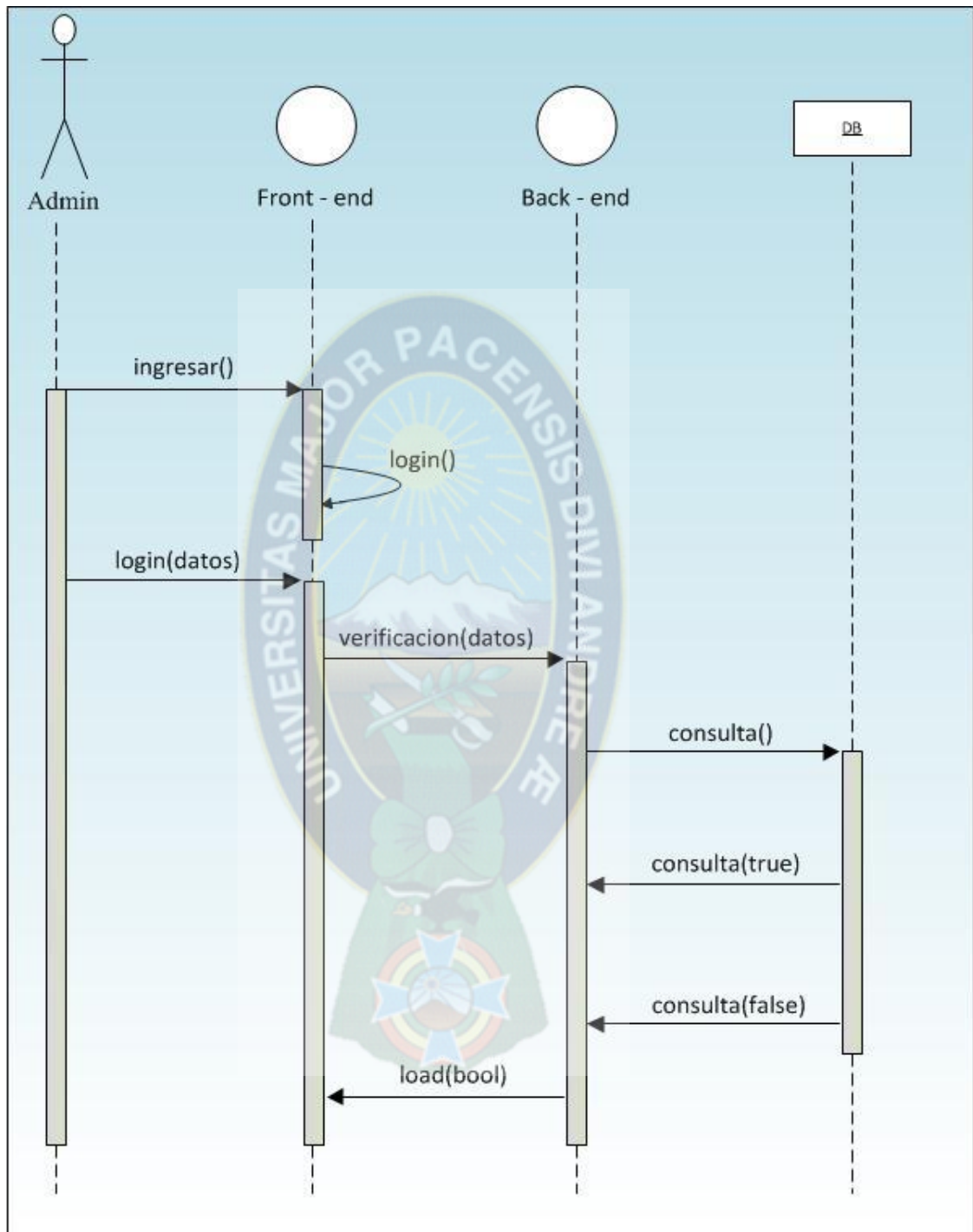


Figura 3.13: Secuencia “ingreso administrador”

Fuente: Elaboración propia

### Secuencia “agregar sucursal”

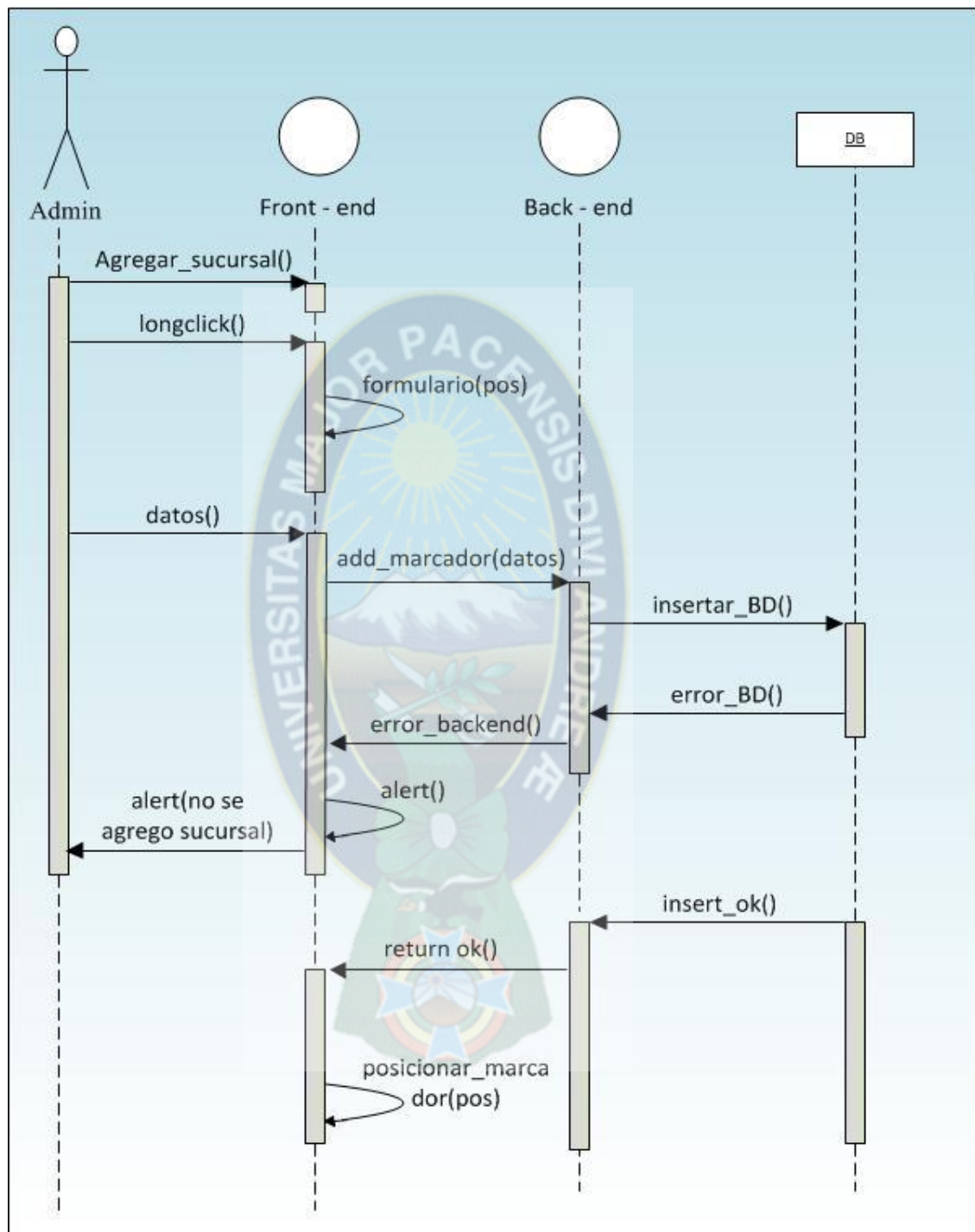


Figura 3.14: Secuencia “agregar sucursal”

Fuente: Elaboración propia

### 3.5.5. Diagrama de clases

El diagrama de relación entre las clases que tendrá la aplicación móvil será el siguiente:

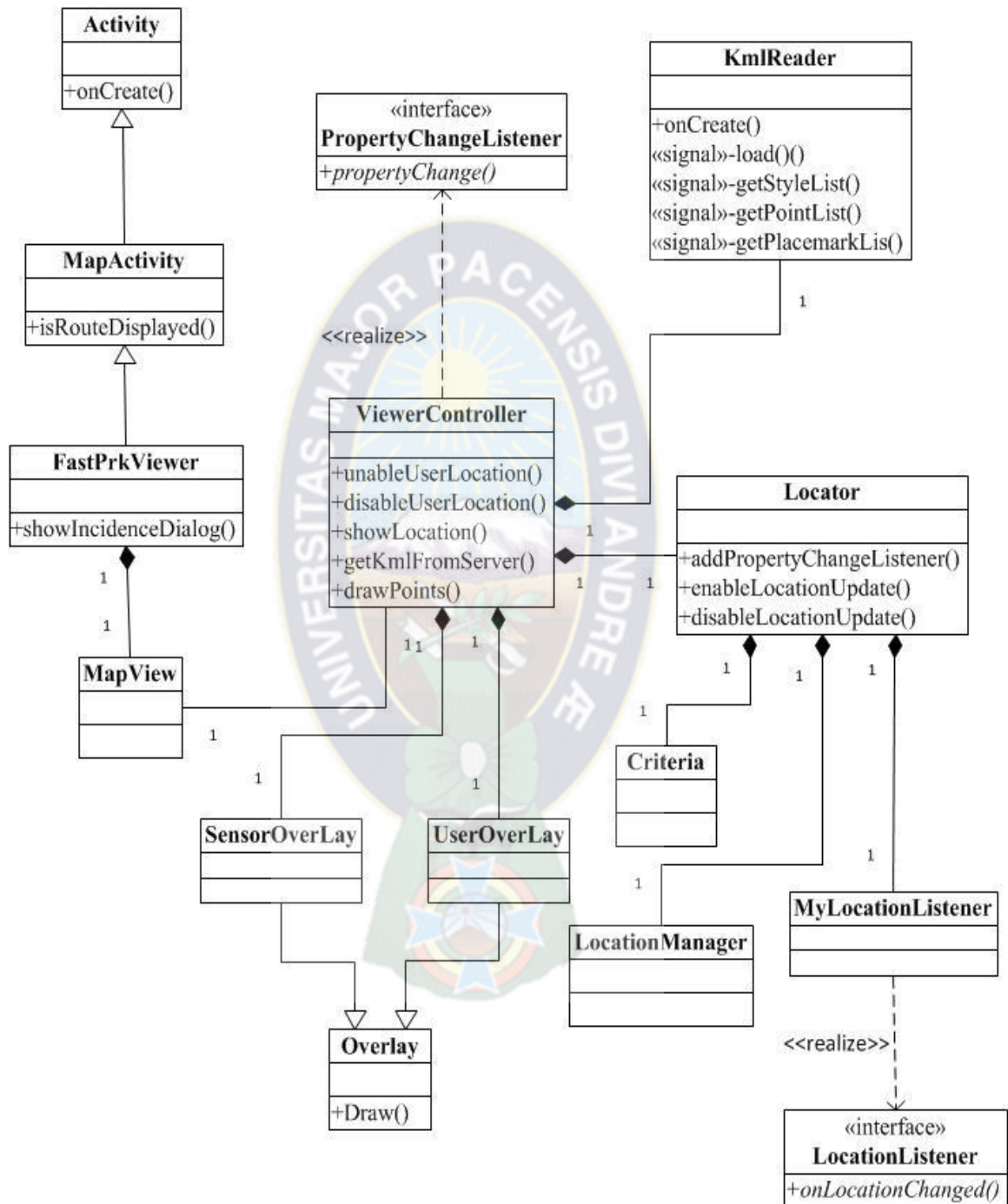


Figura 3.15: Diagrama de clases de la interfaz de la aplicación

Fuente: Elaboración propia

A continuación se detalla el cometido de cada una de estas clases:

- **Activity:** Es el bloque básico de la aplicación en Android. Una actividad usa una serie de funciones para interactuar con su entorno. Por lo general sobrecargará el método "onCreate" el cual se utiliza comúnmente para construir todo el interfaz gráfico.
- **MapActivity:** Es una clase básica de Android que se extiende de Activity en la cual se puede incluir un mapa. Esta clase maneja el ciclo de vida de la aplicación y gestiona los servicios en background para poder mostrar el mapa por pantalla. Es necesario que MapActivity contenga un atributo de la clase MapView.
- **FastPrkViewer:** Clase básica de la aplicación. Esta clase extiende de MapActivity. Se encarga de cargar el interfaz gráfico definido en el fichero main.xml e instanciar el resto de clases necesarias para la aplicación.
- **MapView:** Clase original de Android. Esta clase será la que permita el manejo del mapa, es decir, será la clase utilizada para mostrar por pantalla una sucursal.
- **ViewerController:** Clase que contiene la lógica para el manejo del mapa y las distintas fuentes de localizaciones. Esta clase se implementa de la interfaz PropertyChangeListener. Exactamente esta clase se encuentra a la escucha de cambios en la localización del usuario a través de la clase Locator. Cada vez que reciba un cambio este será representado en el mapa con la nueva localización.
- **OverLay:** Clase original de Android utilizada para situar una imagen sobre el mapa.
- **SensorOverLay:** Clase que extiende de OverLay. Se utiliza para situar sobre el mapa los iconos que indiquen la localización de un sensor de aparcamiento.
- **UserOverLay:** Clase que extiende de OverLay. Se utiliza para situar sobre el mapa el icono que represente la localización del usuario del móvil.
- **KmlReader:** Clase que se encarga de leer y realizar un parser sobre un fichero kml. Con su método load() se encarga de realizar un recorrido por el fichero kml, registrando los "estilos", <Style>, de los puntos y buscando las etiquetas <Placemark>



que son las que definen un punto. A su vez se asegura que estas etiquetas contengan realmente la etiqueta <Point>. Con su método `getPointList` se obtiene un listado de puntos con sus coordenadas y la url donde se encuentra la imagen que representará el overlay.

- **Locator:** Clase que contiene la información para obtener las coordenadas de localización del usuario con los métodos disponibles en el móvil. Además contendrá un listado de listeners de tipo `PropertyChangeListener`. Cuando se detecte un cambio de localización se lo notificará a todos los listeners suscritos, en este caso será la clase `ViewerController`.
- **Criteria:** Clase original de Android. Se utiliza para definir una serie de requisitos para la elección de un proveedor de localizaciones. Estos requisitos pueden ser una mayor o menor precisión en la localización, consumo de batería, etc.
- **LocationManager:** Clase original de Android utilizada para manejar los servicios basados en localización. Aquí se puede definir qué proveedor de localización se quiere utilizar como el GPS, red, etc, o bien utilizar la clase anterior mencionada, `Criteria`, para buscar el mejor proveedor de localización que se adapte a los requisitos que se le han impuesto.
- **MyLocationListener:** Esta clase extiende del interfaz `LocationListener` de Android, la cual, entre otros métodos, obliga a implementar el método `onLocationChanged()`. Con este método se conseguirá actualizar la posición del usuario.

### 3.5.6. Estructura de la Base de Datos

Dentro de la estructura de datos a utilizar en la aplicación se decidió emplear una sola base de datos que contenga 4 tablas en total.

- Tabla de sucursales
- Tabla de tipo de sucursales
- Tabla de provincias
- Tabla de localidades



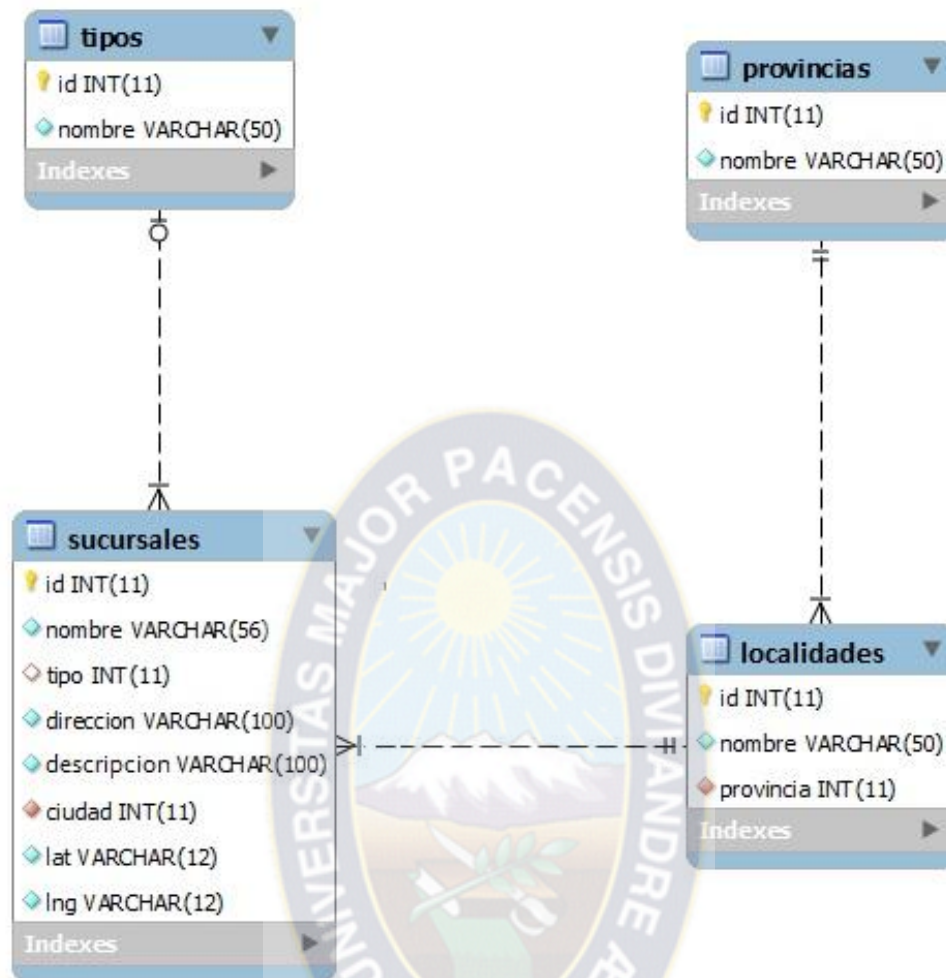


Figura 3.16: Esquema de la Base de datos

Fuente: Elaboración propia

- **Provincias:** En esta tabla almacenaremos las provincias en las que se tengan presencia de cajeros automáticos y agencias de banco.
- **localidades:** En esta tabla almacenaremos las localidades. Además del nombre de la localidad, tendremos un campo que la relacionará con la tabla de provincias.
- **tipos:** Esta tabla nos sirve para almacenar el tipo de sucursal, ("1" Cajero Automático), ("2" Agencia de Banco).

- **Sucursales:** Sin duda la tabla más obvia. Aquí almacenaremos toda la información sobre la sucursal como el nombre, la dirección, descripción, las coordenadas, etc. Además tendremos un campo que la relacione con localidades y otro que la relacione con tipos.

Así pues, tenemos que las relaciones son las siguientes:

- Una provincia puede tener varias localidades.
- Una localidad puede tener varias sucursales.
- Un tipo de sucursal puede pertenecer a varios de éstos.

Como vemos, todas las relaciones son 1-n por lo que, por lo que, no necesitaremos ninguna tabla intermedia para almacenar únicamente relaciones.

### 3.5.7. Diseño de Interfaces

Para el diseño de interfaces graficas se tomó los siguientes tres puntos fundamentales.

- *Estética Visual:* En las aplicaciones actuales la estética resulta ser un aspecto determinante. Se ha intentado dar un aspecto de botones que conducen al uso de las principales funcionalidades de la aplicación combinadas con menús contextuales que proporcionen efectos visuales atractivos.
- *Comodidad de uso:* una aplicación destinada a dispositivos móviles debe estar pensada para que el usuario pueda llegar fácilmente a una funcionalidad intentando que realice el menor número de acciones posible.
- *Independencia del dispositivo:* En el mundo de las aplicaciones móviles no existe un estándar de pantalla de visualización. A este hecho hay que añadirle el tamaño variable de las pantallas de los dispositivos. Por esta razón se utilizará técnicas combinadas de adaptación de partes comunes al dispositivo y partes derivadas a través de las plantillas XML que proporcionan la vista de la Aplicación en el entorno Android.

### 3.5.8. Interfaz del usuario



Figura 3.17: Pantalla de inicio de la aplicación

Fuente: Elaboración propia





Figura 3.18: Guía visual de eventos de configuración

Fuente: Elaboración propia



Figura 3.19: Guía visual de eventos de búsqueda y créditos

Fuente: Elaboración propia

### 3.6. FASE DE DESARROLLO

Se ha decidido hacer una breve descripción de los pasos realizados para la implementación del aplicativo, que permita entender cómo se puede emular la creación de la aplicación móvil y el geo-portal web.

En primera instancia se debe configurar el entorno (portal web) en el cual vamos a hacer la instalación del módulo de geolocalización, para la administración y gestión de las sucursales financieras aplicando técnicas de desarrollo, en este caso tal como se había manifestado en la definición de características tecnológicas todas las herramientas.

Consiguientemente se implementará la aplicación móvil en Android Eclipse por módulos, las cuales se probarán cada etapa.

Finalmente se integraran todos los módulos de geolocalización tanto de la aplicación móvil y el servicio web.

#### 3.6.1. Implementación Web

Se muestra los fragmentos principales de código que permitirá la gestión de los puntos de servicios financieros, en donde el administrador podrá agregar, actualizar y eliminar las sucursales almacenando las coordenadas geográficas y la información necesaria en la base de datos. El código completo en HTML5 lo puede observar en el (**Anexo A**).

##### - Cargado de Google Maps Api V3

```
19 <link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>
20 <link href="css/accordion.css" rel="stylesheet" type="text/css" />
21
22 <script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?sensor=false"></script>
23 <script type="text/javascript" src="http://code.jquery.com/jquery-2.0.3.min.js"></script>
24
25 <!--ARCHIVOS JAVASCRIPT DE BOOTSTRAP-->
26 <script type="text/javascript" src="js/bootstrap.min.js"></script>
27
```



- **Punto inicial del mapa**

```
$(document).on("ready", function(){
    var formulario = $("#formulario");
    //COORDENADAS INICIALES -16.499262796508606, -68.13510907460613
    //VARIABLE PARA PUNTO INICIAL
    var punto = new google.maps.LatLng(-16.499262796508606, -68.13510907460613);
    //VARIABLE PARA CONFIGURACION INICIAL
    var config = {
        zoom:16,
        center:punto,
        mapTypeId: google.maps.MapTypeId.HYBRID
    };

    mapa = new google.maps.Map($("#mapa")[0],config);
    google.maps.event.addListener(mapa,"click", function(event){
```

- **Obtención de Coordenadas**

```
//Coordenadas
var coordenadas=event.latLng.toString();

coordenadas = coordenadas.replace("(", "");
coordenadas = coordenadas.replace(")", "");

//coordenadas por separado
var lista = coordenadas.split(",");

//Variable para dirección, punto o coordenada
var direccion = new google.maps.LatLng(lista[0],lista[1]);
```

- **Agregando Marcadores**

```
//variable para marcador
var marcador = new google.maps.Marker({

    position:direccion, //la posicion del nuevo marcador
    map: mapa,          //en que mapa se ubicara el marcador
    animation:google.maps.Animation.DROP, //como aparecera el marcador en el mapa
    draggable:false //no permitir el arrastre del marcador
});
```



- Pasar las coordenadas al formulario

```
//PASAR LAS COORDENADAS AL FORMULARIO
formulario.find("input[name='cx']").val(lista[0]);
formulario.find("input[name='cy']").val(lista[1]);
//UBICAR EL FOCO EN EL CAMPO TITULO
formulario.find("input[name='titulo']").focus();

//GUARDAR EL MARCADOR EN EL ARRAY
marcadores_nuevos.push(marcador);

//agregar evento click al marcador
google.maps.event.addListener(marcador, "click", function(){
});
```

- Diagramas de clases php

Utilizaremos las siguientes clases php, detalladas en el siguiente diagrama de clases:

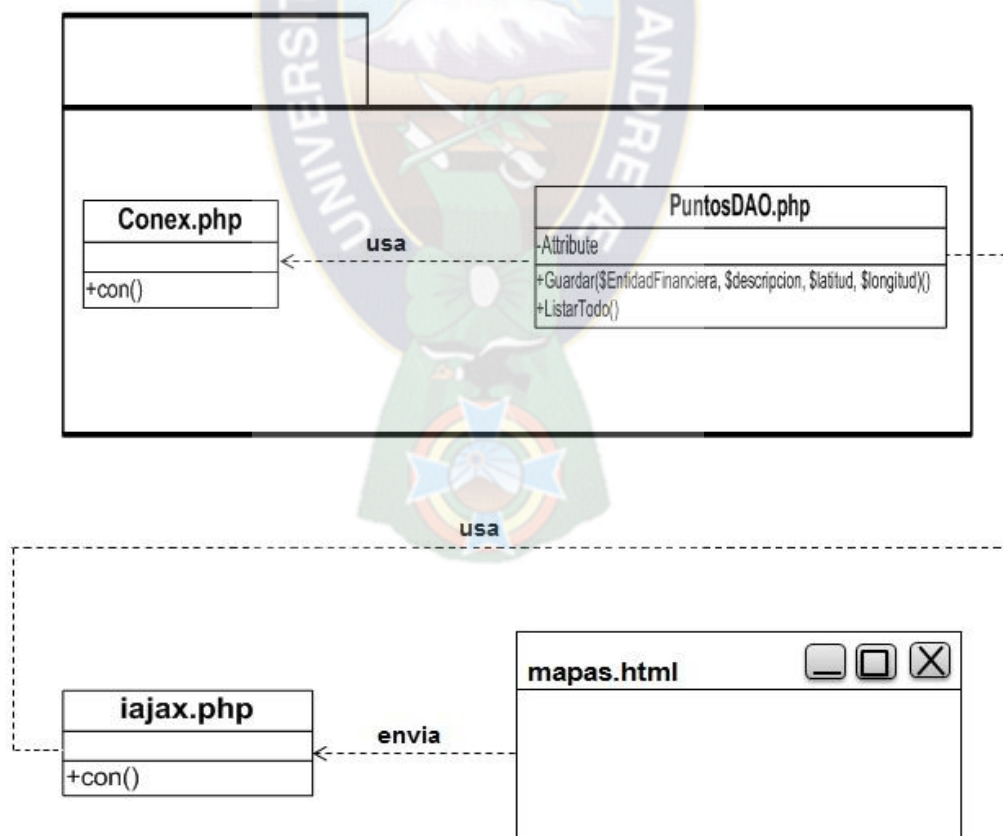


Figura 3.20: Diagrama de clases servicio web

Fuente: Elaboración propia

- Código conex.php

```

1  <?php
2  class conex //CLASE PARA CONEXION A BASE DE DATOS
3  {
4      public static function con()
5      {
6          $conexion = mysql_connect("localhost", "root", "");
7          mysql_select_db("mapas");
8          mysql_query("SET NAMES 'utf8'");
9          if(!$conexion)
10         {
11             return FALSE;
12         }
13         else
14         {
15             return $conexion;
16         }
17     }
18 }
19
20
21 ?>

```

- Código puntosDAO.php

```

<?php
include_once 'conex.php'; //INCLUIR CONEXION DE BASE DE DATOS
class puntosDao
{
    private $r;
    public function _construct()
    {
        $this->r = array();
    }
    public function grabar($titulo, $tipos, $descripcion, $cx, $cy) //METODO PARA
    GUARDAR A LA BD
    {
        $con = conex::con();
        $titulo = mysql_real_escape_string($titulo);
        $tipos = mysql_real_escape_string($tipos);
        $descripcion = mysql_real_escape_string($descripcion);
        $cx = mysql_real_escape_string($cx);
        $cy = mysql_real_escape_string($cy);
        $q = "insert into puntos (Titulo, tipos, descripcion, cx,
cy)." . "values('".addslashes($titulo)."','".addslashes($tipos)."','".addslashes($descripcion)."','
".addslashes($cx)."','".addslashes($cy)."')";
        $rpta = mysql_query($q, $con);
        mysql_close($con);
    }
}

```

```

        if($rpta==1)
        {
            return TRUE;
        }
        else
        {
            return FALSE;
        }
    }

    public function listar_todo()
    {
        $q = "select *from puntos";
        $con = conex::con();
        $rpta = mysql_query($q,$con);
        mysql_close($con);
        while($fila = mysql_fetch_assoc($rpta))
        {
            $this->r[] = $fila;
        }
        return $this->r;
    }

    public function borrar($IdPunto)
    {
        $con = conex::con();
        $IdPunto = mysql_real_escape_string($IdPunto);
        $q = "delete from puntos where IdPunto=".(int)$IdPunto;
        $rpta = mysql_query($q, $con);
        mysql_close($con);
        if($rpta= =1)
        {
            return TRUE;
        }
        else
        {
            return FALSE;
        }
    }

}
?>

```

#### - Código iajax.php

<?php

#Agregar un retardo de 2 segundo para poder apreciar los mensajes.

```

sleep(2);
header('content-type: application/json; charset=utf-8');           //HEADER PARA JSON
include_once 'php/puntosDAO.php';
$ac = isset($_POST["tipo"])?$_POST["tipo"]:"x"; //PARAMETRO PARA
        DETERMINAR LA ACCION
SWITCH($ac)
{
    case "grabar":
        $p = new puntosDao();
        $exito = $p->grabar($_POST["titulo"],$_POST["tipos"],
            $_POST["descripcion"], $_POST["cx"], $_POST["cy"]);
        if($exito)
        {
            $r["estado"] = "ok";
            $r["mensaje"] = "Se Guardo Correctamente";
        }
        else
        {
            $r["estado"] = "Error";
            $r["mensaje"] = "Error al Guardar";
        }
        break;
    case "listar":
        $p = new puntosDao();
        $resultados = $p->listar_todo();
        if(sizeof($resultados)>0)
        {
            $r["estado"] = "ok";
            $r["mensaje"] = $resultados;
        }
        else
        {
            $r["estado"] = "Error";
            $r["mensaje"] = "No hay registros";
        }
        break;
    case "borrar":
        $p = new puntosDao();
        $resultados = $p->borrar($_POST["id"]);
        if($resultados)
        {
            $r["estado"] = "ok";
            $r["mensaje"] = "Borrado Correctamente";
        }
        else
        {
            $r["estado"] = "error";

```

```

        $r["mensaje"] = "Error al borrar";
    }
    break;
default:
    $r["estado"] = "Error";
    $r["mensaje"] = "datos no validos";
    break;
}
echo json_encode($r);      //IMPRIMIR JSON
?>

```

### 3.6.2. Implementación Aplicación Android

#### Módulo de Geolocalización

Para poder utilizar los servicios de geolocalización, debemos solicitar permiso en el manifest para acceder a estos servicios. Se solicita por separado permiso para el servicio de localización.

```

<uses-permission android:name=
"android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name=
"android.permission.ACCESS_COARSE_LOCATION"/>

```

---

Para acceder a los servicios de geolocalización en Android tenemos la clase LocationManager. Esta clase no se debe instanciar directamente, sino que obtendremos una instancia como un servicio del sistema de la siguiente forma:

```

LocationManager manager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);

```

---

Para obtener una localización deberemos especificar el proveedor que queramos utilizar. Los principales proveedores disponibles en los dispositivos son el GPS (LocationManager.GPS\_PROVIDER) y la red 3G o WiFi

(LocationManager.NETWORK\_PROVIDER). Podemos obtener información sobre estos proveedores con:

```

LocationProvider proveedor = manager
.getProvider(LocationManager.GPS_PROVIDER);

```

---

La clase `LocationProvider` nos proporciona información sobre las características del proveedor, como su precisión, consumo, o datos que nos proporciona. Es también posible obtener la lista de todos los proveedores disponibles en nuestro móvil con `getProviders`, u obtener un proveedor basándonos en ciertos criterios como la precisión que necesitamos, el consumo de energía, o si es capaz de obtener datos como la altitud a la que estamos o la velocidad a la que nos movemos. Estos criterios se especifican en un objeto de la clase `Criteria` que se le pasa como parámetro al método `getProviders`.

Para obtener la última localización registrada por un proveedor llamaremos al siguiente método:

```
Location posicion = manager  
.getLastKnownLocation(LocationManager.GPS_PROVIDER);
```

---

El objeto `Location` obtenido incluye toda la información sobre nuestra posición, entre la que se encuentra la latitud y longitud.

Con esta llamada obtenemos la última posición que se registró, pero no se actualiza dicha posición. A continuación veremos cómo solicitar que se realice una nueva lectura de la posición en la que estamos.

#### - **Actualización de la posición**

Para poder recibir actualizaciones de nuestra posición deberemos definir un listener de clase `LocationListener`:

```
class ListenerPosicion implements LocationListener {  
    public void onLocationChanged(Location location) {  
        // Recibe nueva posición.  
    }  
    public void onProviderDisabled(String provider){  
        // El proveedor ha sido desconectado.  
    }  
    public void onProviderEnabled(String provider){  
        // El proveedor ha sido conectado.  
    }  
    public void onStatusChanged(String provider,
```

---



```
int status, Bundle extras){  
    // Cambio en el estado del proveedor.  
}  
};
```

---

Una vez definido el listener, podemos solicitar actualizaciones de la siguiente forma:

```
ListenerPosicion listener = new ListenerPosicion();  
long tiempo = 5000; // 5 segundos  
float distancia = 10; // 10 metros  
manager.requestLocationUpdates(  
    LocationManager.GPS_PROVIDER,  
    tiempo, distancia, listenerPosicion);
```

---

Podemos observar que cuando pedimos las actualizaciones, además del proveedor y del listener, debemos especificar el intervalo mínimo de tiempo (en milisegundos) que debe transcurrir entre dos lecturas consecutivas, y el umbral de distancia mínima que debe variar nuestra posición para considerar que ha habido un cambio de posición y notificar la nueva lectura.

Una vez hayamos terminado de utilizar el servicio de geolocalización, deberemos detener las actualizaciones para reducir el consumo de batería. Para ello eliminamos el listener de la siguiente forma:

```
manager.removeUpdates(listener);
```

---

## - **Alertas de Proximidad**

En Android podemos definir una serie de alertas que se disparan cuando nos acercamos a una determinada posición. Recibiremos los avisos de proximidad mediante intents. Por ello, primero debemos crearnos un Intent propio:

```
Intent intent = new Intent(codigo);  
PendingIntent pi = PendingIntent.getBroadcast(this, -1, intent, 0);
```

---

Para programar las alertas de proximidad deberemos especificar la latitud y longitud, y el radio de la zona de proximidad en metros. Además podemos poner una caducidad a las alertas (si ponemos -1 no habrá caducidad):

```
double latitud = -16.501621647935778;  
double longitud = -68.13198900205272;  
float radio = 500f;  
long caducidad = -1;  
manager.addProximityAlert(latitud, longitud, radio,  
caducidad, pi);
```

---

## - Geocodificación

El geocoder nos permite realizar transformaciones entre una dirección y las coordenadas en las que está. Podemos obtener el objeto Geocoder con el que realizar estas transformaciones de la siguiente forma:

```
Geocoder geocoder = new Geocoder(this, Locale.getDefault());
```

---

Podemos obtener la dirección a partir de unas coordenadas (latitud y longitud):

```
List<Address> direcciones = geocoder  
.getFromLocation(latitud, longitud, maxResults);
```

---

También podemos obtener las coordenadas correspondientes a una determinada dirección:

```
List<Address> coordenadas = geocoder  
.getFromLocationName(direccion, maxResults);
```

---

## Módulo de mapas

La forma más habitual de presentar la información obtenida por el GPS es mediante un mapa. Vamos a ver cómo podemos integrar los mapas de Google en nuestra aplicación, y mostrar en ellos nuestra posición y una serie de puntos de interés.

La API de mapas no está incluida en el SDK básico de Android, sino que se encuentra entre las librerías de Google. Por este motivo para poder utilizar dicha librería deberemos declararla en el AndroidManifest.xml, dentro de la etiqueta application:

```
<application android:icon="@drawable/icon"
android:label="@string/app_name">
<uses-library android:name="com.google.android.maps" />
...
<activity android:name=".MapasActivity"
android:label="@string/app_name">
<intent-filter></intent-filter>
</activity>
</application>
```

---

Para mostrar un mapa, deberemos crear una actividad que herede de MapActivity, en lugar de Activity, y dentro de ella introduciremos una vista de tipo MapView. Podemos introducir esta vista en nuestro layout de la siguiente forma:

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.maps.MapView
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/mvMapa"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:apiKey="vzqSGwNXea1F0KDyc1HvvV6c0JfRQW-mYelyt6Q"
android:clickable="true"
/>
```

---

### 3.7. FASE DE PRUEBAS DE FUNCIONAMIENTO

#### 3.7.1. Visualización de interfaces

El diseño de la aplicación para el ambiente web como para la aplicación móvil, tomó como referencia las artes gráficas tanto del portal web actual, como los lineamientos generales de las entidades financieras.

En cuanto al mapa proveniente de Google Maps v3, se utilizó su estética por defecto, y en lo que tiene que ver con el cuadro de herramientas configurable que se visualiza la característica que permite variar el zoom dentro del mapa.

- **Pantalla de inicio de sesión del administrador**



Figura 3.21: Pantalla de inicio de sesión del administrador

Fuente: Elaboración propia

- **Pantalla de error de acceso**

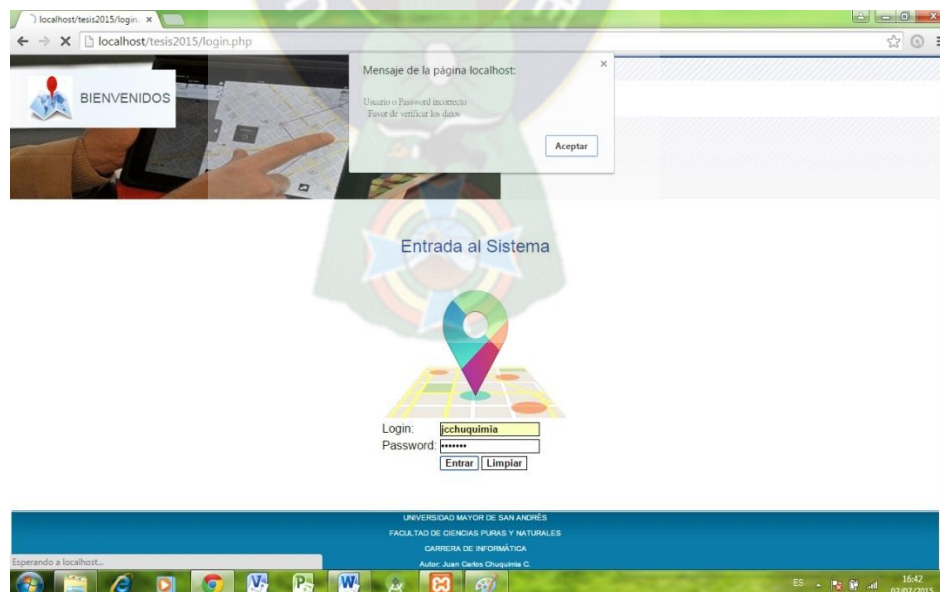


Figura 3.22: Pantalla de error de acceso (usuario y/o incorrectos)

Fuente: Elaboración propia



- **Pantalla con vista satelital**



Figura 3.23: Pantalla con vista satelital

Fuente: Elaboración propia

- **Pantalla con vista de mapa**



Figura 3.24: Pantalla con vista de mapa

Fuente: Elaboración propia

- **Pantalla de agregar nuevas sucursales**

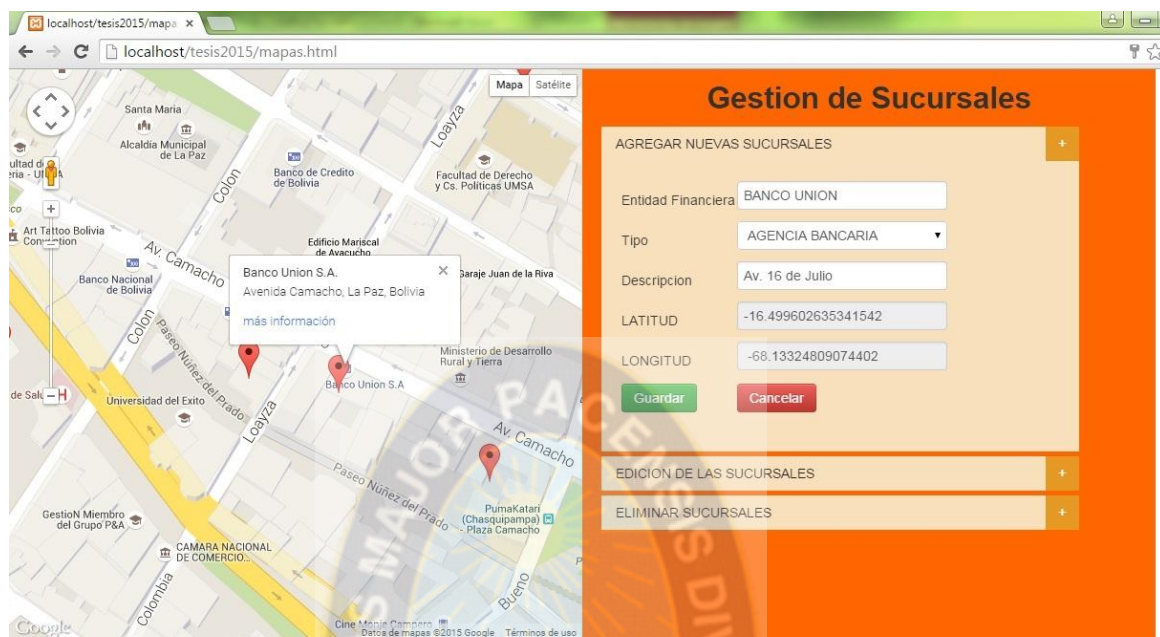


Figura 3.25: Pantalla de agregar nuevas sucursales

Fuente: Elaboración propia

- **Pantalla de edición de sucursales**

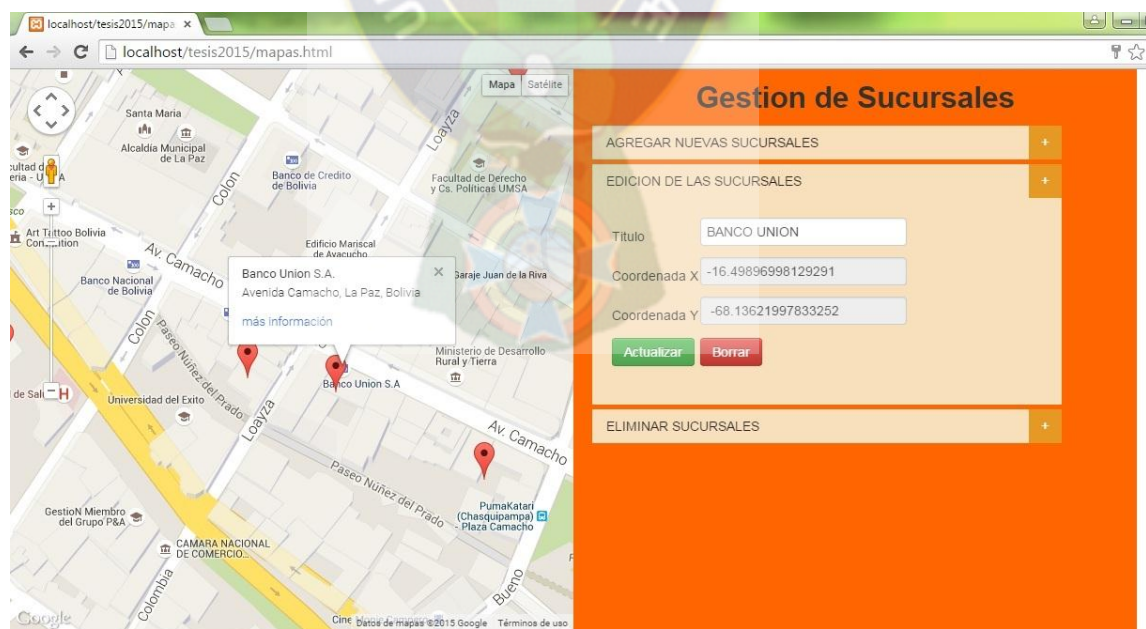


Figura 3.26: Pantalla de edición de sucursales

Fuente: Elaboración propia



- **Pantalla de trazado de ruta**

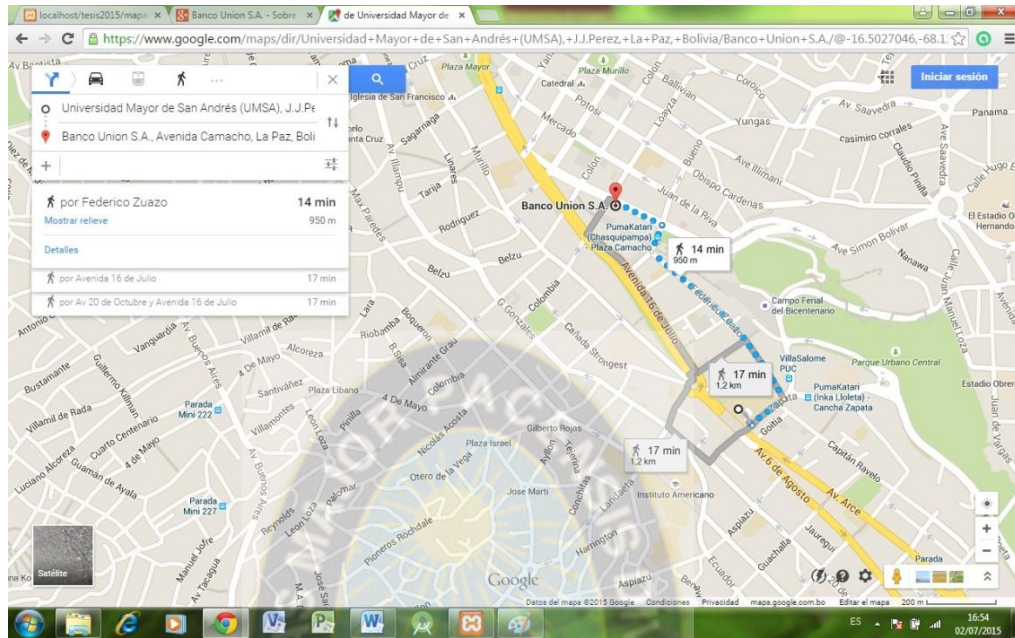


Figura 3.27: Pantalla de trazado de ruta

Fuente: Elaboración propia

- **Pantalla de navegación de la ruta**

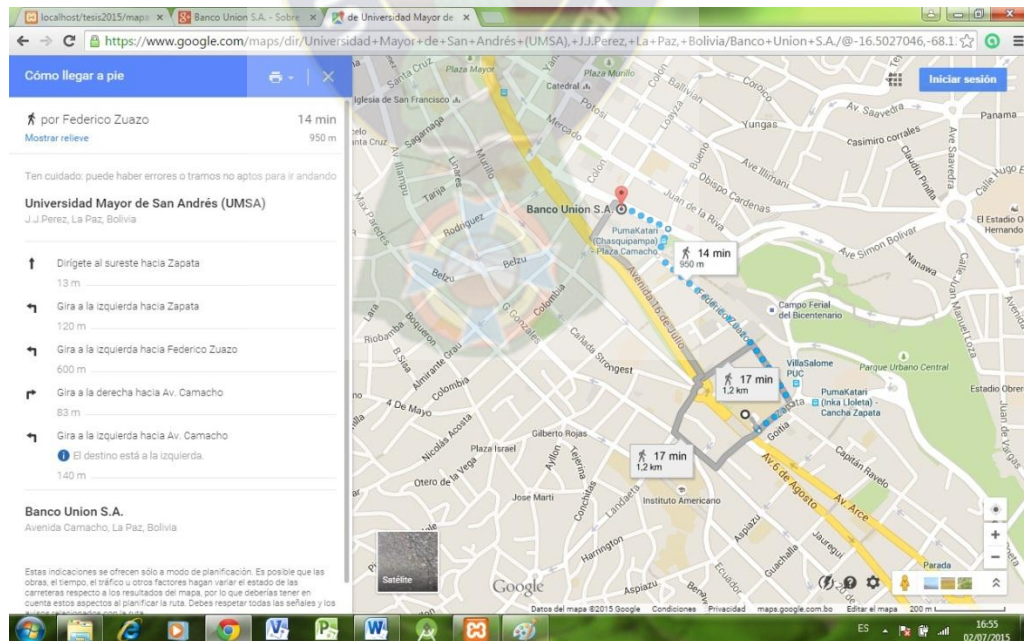


Figura 3.28: Pantalla de navegación de la ruta

Fuente: Elaboración propia

### 3.8. FASE DE ENTREGA

La aplicación de geroreferenciación de cajeros automáticos y agencias bancarias, es una aplicación sencilla de utilizar, y sobre todo, no le interferirá en su día a día y no de provocará una reducción de la batería de su dispositivo. Esta aplicación le facilitará encontrar los servicios financieros en todo el territorio nacional.

### 3.9. CALIDAD DEL SOFTWARE

Para establecer la calidad de la aplicación móvil, se utilizó la métrica de software de Puntos Función lo que nos permite establecer un conjunto de características que deben ser evaluadas.

La evaluación de la calidad de la aplicación no es una tarea sencilla por lo que generalmente se consideran factores subjetivas, sin embargo se puede considerar características y atributos deseables.

- **Flexibilidad**

La flexibilidad es el costo de modificación del producto cuando cambian sus especificaciones, y puede ser calculada a partir de la siguiente fórmula:

$$1 - 0.05(\text{número medio de días} - \text{hombre por cambio})$$

$$\text{Flexibilidad} = 1 - 0.05(2 - 1)$$

$$\text{Flexibilidad} = 0.95 * 100$$

$$\text{Flexibilidad} = 95\%$$

(Significa que en un 95% es flexible el Sistema)

- **Facilidad de Mantenimiento**

Es el esfuerzo necesario para localizar y arreglar un error en el programa y está dada por la siguiente ecuación matemática:

$$\text{facilidad de mantenimiento} =$$

$$1 - 0.1 * (\text{número de días} - \text{hombre por corrección})$$

$$\text{facilidad de mantenimiento} = 1 - 0.1(2 - 1 \text{ persona por corrección})$$

$$\text{facilidad de mantenimiento} = 0.9 * 100$$

$$\text{facilidad de mantenimiento} = 90\%$$

(Significa que en un 90% es fácil de mantener)



## **CAPÍTULO 4**

# **PRUEBA DE HIPÓTESIS**

---

## PRUEBA DE HIPÓTESIS

El método estadístico para comprobar o rechazar la hipótesis, que se ha planteado previamente, se aplicó el método estadístico de la chi-cuadrada ( $X^2$ ).

### 4.1. PLANTEAMIENTO DE LA HIPOTESIS NULA Y ALTERNATIVA

Recordemos que la hipótesis planteada es la siguiente:

“La implementación del sistema Georreferenciación de cajeros automáticos y agencias bancarias mediante aplicación android, permite al usuario obtener la ubicación geográfica de manera ágil y precisa, las sucursales bancarias y cajeros automáticos”.

#### 4.1.1. Hipótesis Nula ( $H_0$ )

**$H_0$ :** La implementación del sistema Georreferenciación de cajeros automáticos y agencias bancarias mediante aplicación android, No responde a las necesidades de obtener la ubicación geográfica de manera ágil y precisa las sucursales bancarias y cajeros automáticos.

#### 4.1.2. Hipótesis Alternativa ( $H_1$ )

**$H_1$ :** La implementación del sistema Georreferenciación de cajeros automáticos y agencias bancarias mediante aplicación android, responde a las necesidades de obtener la ubicación geográfica de manera ágil y precisa las sucursales bancarias y cajeros automáticos.

### 4.2. ESTADISTICO DE PRUEBA

Se utiliza la distribución Chi-Cuadrada, verificando que tipo de relación tiene una muestra aleatoria de tamaño  $n$  y su opinión acerca de la aplicación, si los resultados obtenidos son dependientes o independientes, es decir, que si estas fueran dependientes, las variables están

relacionadas, por otra parte si los resultados muestran que son independientes es posible afirmar que no existe relación entre ellas.

Para ello se procedió a realizar un sondeo de opinión, a los usuarios que tuvieron la experiencia de utilizar la aplicación de Georreferenciación de cajeros automáticos y agencias bancarias, que respondieron a las siguientes preguntas:

A. ¿Qué tanto responde a las necesidades de obtener la ubicación geográfica de manera ágil y precisa, las sucursales bancarias y cajeros automáticos?

1 - Nada

2 – Poco

3 - Mucho

B. ¿Cómo calificaría la experiencia de usar la aplicación?

1 – Nada útil

2 – Poco útil

3 – Muy útil

#### 4.2.1. Frecuencias Observadas

Se tomó una muestra aleatoria de tamaño  $n = 50$ , cuyos datos clasificados proporcionan la siguiente tabla:

| B                  | A         |           |          | TOTAL     |
|--------------------|-----------|-----------|----------|-----------|
|                    | Mucho     | Poco      | Nada     |           |
| Muy útil           | 21        | 2         | 0        | <b>23</b> |
| Moderadamente útil | 13        | 6         | 1        | <b>20</b> |
| Nada útil          | 0         | 5         | 2        | <b>7</b>  |
| <b>TOTAL</b>       | <b>34</b> | <b>13</b> | <b>3</b> | <b>50</b> |

Tabla 4.1: Tabla de contingencia de frecuencias observadas

Fuente: Elaboración propia



#### 4.2.2. Tabla de contingencia para las probabilidades

|                    | A             |               |               |               |
|--------------------|---------------|---------------|---------------|---------------|
| B                  | Mucho         | Poco          | Nada          | Prob marg Y   |
| Muy útil           | 0,4200        | 0,0400        | 0,0000        | <b>0,4600</b> |
| Moderadamente útil | 0,2600        | 0,1200        | 0,0200        | <b>0,4000</b> |
| Nada útil          | 0,0000        | 0,1000        | 0,0400        | <b>0,1400</b> |
| Prob marg X        | <b>0,6800</b> | <b>0,2600</b> | <b>0,0600</b> | <b>1,0000</b> |

Tabla 4.2: Tabla de contingencia para las probabilidades

Fuente: Elaboración propia

#### 4.2.3. Tabla de Frecuencias Esperadas

|                    | A            |              |             |              |
|--------------------|--------------|--------------|-------------|--------------|
| B                  | Mucho        | Poco         | Nada        |              |
| Muy útil           | 15,64        | 5,98         | 1,38        | <b>23,00</b> |
| Moderadamente útil | 13,60        | 5,20         | 1,20        | <b>20,00</b> |
| Nada útil          | 4,76         | 1,82         | 0,42        | <b>7,00</b>  |
|                    | <b>34,00</b> | <b>13,00</b> | <b>3,00</b> | <b>50,00</b> |

Tabla 4.3: Tabla de Frecuencias esperadas

Fuente: Elaboración propia

Tomando como población de estudio, los usuarios de la muestra, se verifica si existe correlación entre el nivel de utilidad de la aplicación, y si este responde a las necesidades de obtener la ubicación geográfica de manera ágil y precisa, las sucursales bancarias y cajeros automáticos.



#### 4.2.4. Nivel de Significación

Se tomará un nivel de significancia del 5% ( $\alpha = 0,05$ )

#### 4.2.5. Nivel de Confianza

Se contrastan las variables a un nivel de confianza de  $1 - \alpha = 0,95$ .

#### 4.2.6. Estadístico Chi cuadrado

$$X^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

Donde:

$O_i$ : Frecuencia Observada

$E_i$ : Frecuencia Esperada

$k$ : Número de Filas \* Número de Columnas, de la tabla de contingencia

#### 4.2.7. Valor Crítico

***Distribución Chi cuadrado con  $(3 - 1) * (3 - 1)gl = 4gl$ .***

| Nivel de Significancia | Valores críticos según nivel de significancia |
|------------------------|---|
| 0.05                   | 9.488   |

Dichos valores se obtuvieron de la tabla de Chi cuadrado tomando en cuenta los grados de libertad y el nivel de significancia.

### TABLA: Distribución $\chi^2$ Ji-Cuadrada

$$Y \sim \chi_{\nu}^2$$

siendo  $\nu$  los grados de libertad.

$$p = P(Y \leq y) = \int_0^y \phi_{\chi^2}(u) du = 1 - \alpha$$

donde

$$\phi_{\chi^2}(u) = \frac{1}{2^{\nu/2} \Gamma(\nu/2)} u^{\nu/2-1} e^{-u/2}$$

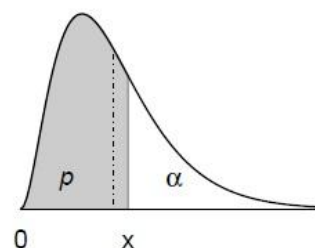


Tabla 2. Valores críticos  $\chi_{(\alpha;\nu)}^2$  de la distribución  $\chi_{\nu}^2$  Ji-Cuadrada.

|       | 0.005 | 0.01  | 0.025 | 0.05  | 0.1   | $p$      |        | 0.90   | 0.95   | 0.975  | 0.99 | 0.995 |
|-------|-------|-------|-------|-------|-------|----------|--------|--------|--------|--------|------|-------|
|       |       |       |       |       |       | $\alpha$ |        |        |        |        |      |       |
| $\nu$ | 0.995 | 0.99  | 0.975 | 0.95  | 0.90  | 0.10     | 0.05   | 0.025  | 0.01   | 0.005  |      |       |
| 1     | 0.000 | 0.000 | 0.001 | 0.004 | 0.016 | 2.706    | 3.841  | 5.024  | 6.635  | 7.879  |      |       |
| 2     | 0.010 | 0.020 | 0.051 | 0.103 | 0.211 | 4.605    | 5.991  | 7.378  | 9.210  | 10.597 |      |       |
| 3     | 0.072 | 0.115 | 0.216 | 0.352 | 0.584 | 6.251    | 7.815  | 9.348  | 11.345 | 12.838 |      |       |
| 4     | 0.207 | 0.297 | 0.484 | 0.711 | 1.064 | 7.779    | 9.488  | 11.143 | 13.277 | 14.860 |      |       |
| 5     | 0.412 | 0.554 | 0.831 | 1.145 | 1.610 | 9.236    | 11.070 | 12.833 | 15.086 | 16.750 |      |       |
| 6     | 0.676 | 0.872 | 1.237 | 1.635 | 2.204 | 10.645   | 12.592 | 14.449 | 16.812 | 18.548 |      |       |

Figura 4.1: Tabla de Chi cuadrado

Fuente: Distribución Chi cuadrado

#### 4.2.8. Valor Empírico

Los valores empíricos se hallan con la siguiente ecuación:  $(O_i - E_i)^2/E_i$

|                              |       |       | TOTAL  |
|------------------------------|-------|-------|--------|
| 1,837                        | 2,649 | 1,380 | 5,866  |
| 0,026                        | 0,123 | 0,033 | 0,183  |
| 4,760                        | 5,556 | 5,944 | 16,260 |
| VALOR EMPÍRICO ( $\chi^2$ )= |       |       | 22.309 |

Tabla 4.4: Tabla de valor empírico

Fuente: Elaboración propia

### 4.3. DECISION

| Nivel se significancia | Valores Críticos según nivel de significancia | Valor empírico | Decisiones |
|------------------------|---|----------------|------------|
| 0,05                   | 9,488   | 22,309         | Rechazo Ho |

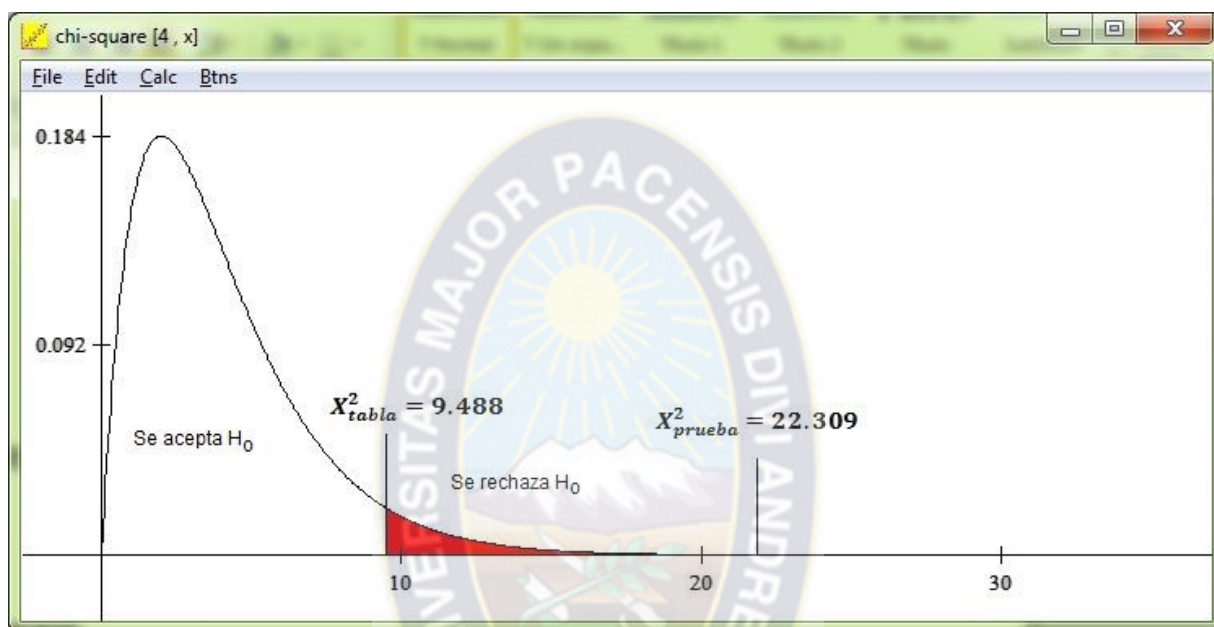


Figura 4.2: Gráfica de decisión en Winstats

Fuente: Elaboración Propia

### 4.4. CONCLUSIÓN

Para aceptar la hipótesis nula el valor de Chi-Cuadrado ( $X^2$ ), pudo haber estado en el rango hasta 9,488.

***Como  $X^2 > 9.488$  entonces rechazamos  $H_0$***

Por lo tanto, se rechaza la hipótesis nula y se toma la hipótesis alternativa como válida.

**H<sub>1</sub>** : La implementación del sistema Georreferenciación de cajeros automáticos y agencias bancarias mediante aplicación android, responde a las necesidades de obtener la ubicación geográfica de manera ágil y precisa las sucursales bancarias y cajeros automáticos.



## **CAPÍTULO 5**

# **CONCLUSIONES Y RECOMENDACIONES**

---

# CONCLUSIONES Y RECOMENDACIONES

## 5.1. CONCLUSIONES

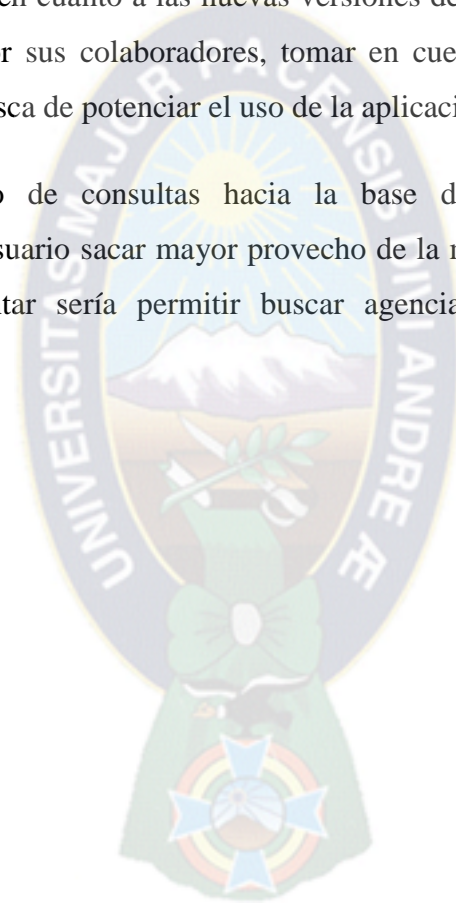
- Con el desarrollo e implementación de la aplicación móvil de Georreferenciación de cajeros automáticos y agencias bancarias, se ha podido cumplir el objetivo de manera satisfactoria, poniendo a prueba en diferentes dispositivos móviles, de los cuales se pudo obtener resultados favorables.
- Con la aplicación, los clientes de las entidades financieras podrán obtener información sobre la ubicación geográfica, de los cajeros automáticos y/o agencias bancarias desde cualquier localidad del territorio nacional donde estén localizadas estas sucursales, las 24 hrs. de manera confiable y precisa.
- Las actuales aplicaciones para el manejo de mapas digitales tienen gran cantidad de herramientas que dificultan su uso y el manejo de la información, con la implementación de este prototipo el manejo de la información de agencias bancarias y cajeros es intuitivo por parte de los administradores, además que su uso por parte de los clientes del banco es amigable.
- Con el auge de nuevas tecnologías, iniciativas como Google Maps han extendido el uso de la información geográfica digital, ha abierto un nuevo camino añadiendo el componente social en torno a los mapas.

## 5.2. RECOMENDACIONES

- El presente prototipo podría ampliar sus horizontes para albergar información georreferenciada de diferentes entidades financieras presentes en el mercado boliviano como es el caso de las cooperativas de ahorro y crédito.
- A partir de este proyecto, se puede reutilizar la base tecnológica empleada y el código fuente, para realizar una gran cantidad de emprendimientos que requieran contar entre

sus servicios con brindar información georreferenciada, como podría ser el caso de un buscador de negocios, una guía gastronómica, una guía turística, etc.

- Socializar este tipo de proyectos en congresos, conferencias, bibliotecas virtuales, foros, etc. Con el objetivo de crear una comunidad con una base tecnológica confiable que pueda ser de provecho en otros emprendimientos cuya línea de desarrollo sean los mapas digitales.
- Permanecer alerta en cuanto a las nuevas versiones de las API's ya que siempre se la está mejorando por sus colaboradores, tomar en cuenta las mejoras y ofrecerlas al usuario final en busca de potenciar el uso de la aplicación.
- Ampliar el rango de consultas hacia la base de datos desde la aplicación, permitiéndole al usuario sacar mayor provecho de la misma, una de esas capacidades extra a implementar sería permitir buscar agencias disponibles por horario de atención.







## **BIBLIOGRAFÍA**

---

## REFERENCIAS BIBLIOGRAFICAS

- Letham, L. (2001). GPS fácil uso del Sistema de Posicionamiento Global. (3ra ed.). Settle, WA: Mountaineers.
- Rogers, R., Lombardo, J., Mednieks, Z., & Meike, B. (2009). Android application development: Programming with the Google SDK. O'Reilly Media, Inc.
- Schmuller, J. (2000). Aprendiendo UML en 24 horas. Pearson education.
- Sommerville, Ian. (2005). Ingeniería del Software. Séptima edición. Pearson Education.
- Amaro, J. E. (2012). El gran libro de programación avanzada con Android. Marcombo.
- Nelson Eduardo Hernández Germán (2006), “Programación De Aplicaciones Para Dispositivos Móviles”, Universidad De El Salvador, Ciudad Universitaria.
- Arroyo Vázquez N. (Dic. 2011). Información en el móvil.
- Girones, J.T. (2011). El Gran Libro de Android.
- Pons O., Martín N., Medina J.M., Acid S. y Vila M. A. (2005). Introducción a las bases de datos, el modelo relacional.
- Cobo Yera, Angel. (2008). Diseño y programación de base de datos.
- Thibaud, Cyril. (2006). MySQL 5: Instalación, implementación, administración, programación.
- Alarcon, V. (2006). Desarrollo de Sistemas de información una metodología basada en el modelado. (1ra ed.). Barcelona: UPC.

## REFERENCIAS ELECTRONICAS

- Georreferenciacion  
<http://es.wikipedia.org/wiki/Georreferenciaci%C3%B3n>
- Introducción A Android.  
<http://developer.android.com/index.html>
- Android Open Handset Alliance. [en línea]  
[http://www.openhandsetalliance.com/android\\_overview.html](http://www.openhandsetalliance.com/android_overview.html)
- GOOGLE DEVELOPERS. Google Maps Javascript API V3 Reference.  
<https://developers.google.com/maps/documentation/javascript/reference?hl=es>
- Historia de Android.

<http://materiageek.com/2009/11/android-cumple-2-anos-breverecorrido-por-su-corta-historia>

- Android: The Open Source Cell Phone.

<http://googleopensource.blogspot.com/2008/10/android-open-source-cell-phone.html>

- Android Debug Bridge.

<http://developer.android.com/guide/developing/tools/adb.html>

- Android Managing Virtual Devices.

<http://developer.android.com/guide/developing/devices/index.html>

- Web services.

<http://www.w3.org/TR/ws-arch/#whatis>.

- MONOGRAFIAS. Cálculo del tamaño de la muestra.

<http://www.monografias.com/trabajos87/calculo-del-tamano-muestra/calculo-del-tamano-muestra.shtml#frmulapara>





## ANEXOS

---

```
<script type="text/css"></script>  
<script type="text/css"/>  
http://maps.googleapis.com/maps/api/js?sensor=false  
http://code.jquery.com/jquery-2.0.3.min.js">http://code.jquery.com/jquery-2.0.3.min.js</script>  
<STRAP-->  
<bootstrap.min.js"></script>
```

```
<!DOCTYPE html>
<head>
<style>
#mapa {
    width:600px;
    height:600px;
    float:left;
    background: #69F;
}
#infor {
    width:600px;
    height:600px;
    float:left;
    background:#F60;
}
</style>
<link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>
<link href="css/accordion.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?sensor=false"></script>
<script type="text/javascript" src="http://code.jquery.com/jquery-2.0.3.min.js"></script>
ARCHIVOS JAVASCRIPT DE BOOTSTRAP-->
<script type="text/javascript" src="js/bootstrap.min.js"></script>
<script>
    var marcadores_nuevos = [];
    var marcadores_bd =[];
    var mapa = null;
    function quitar_marcadores(lista)
    {
        for(i in lista)
        {
            lista[i].setMap(null);
        }
    }

    $(document).on("ready", function(){
        var formulario = $("#formulario");
        var punto = new google.maps.LatLng(-16.499262796508606, -68.13510907460613);
        var config = {
            zoom:16,
            center:punto,
            mapTypeId: google.maps.MapTypeId.HYBRID
        };
        mapa = new google.maps.Map($("#mapa")[0],config);
        google.maps.event.addListener(mapa,"click", function(event){
            var coordenadas=event.latLng.toString();

            coordenadas = coordenadas.replace("(",",");
            coordenadas = coordenadas.replace(")",",");
            //coordenadas por separado
            var lista = coordenadas.split(",");
            var direccion = new google.maps.LatLng(lista[0],lista[1]);
            //variable para marcador
            var marcador = new google.maps.Marker({
                <!--titulo:prompt("Nombre de la Sucursal"),-->
                position:direccion,           //la posicion del nuevo marcador
                map: mapa,                    //en que mapa se ubicara el marcador
                animation:google.maps.Animation.DROP,           //como aparecera el marcador en el mapa
                draggable:false           //no permitir el arrastre del marcador
            });
            //PASAR LAS COORDENADAS AL FORMULARIO
            formulario.find("input[name='cx']").val(lista[0]);
            formulario.find("input[name='cy']").val(lista[1]);
            //UBICAR EL FOCO EN EL CAMPO TITULO
            formulario.find("input[name='titulo']").focus();
        });
    });
</script>
</body>
</html>
```

```

//GUARDAR EL MARCADOR EN EL ARRAY
marcadores_nuevos.push(marcador);
var evento click al marcador
google.maps.event.addListener(marcador, "click", function(){
//mostrar una alerta al hacer click
<!--alert("marcador.titulo");-->
});
quitar_marcadores(marcadores_nuevos);
//UBICAR EL MARCADOR EN EL MAPA
marcador.setMap(mapa);
});
$("#btn_grabar").on("click", function(){
var f=$("#formulario");
if(f.find("input[name='titulo']").val().trim()== "")
{
    alert("Falta titulo");
    return false;
}
var f=$("#formulario");
if(f.find("input[name='cx']").val().trim()== "")
{
    alert("Falta Coordenada X");
    return false;
}
var f=$("#formulario");
if(f.find("input[name='cy']").val().trim()== "")
{
    alert("Falta Coordenada Y");
    return false;
}
if(f.hasClass("busy"))
{
    return false;
}
f.addClass("busy");
var loader_grabar = $("#loader_grabar");
$.ajax({
    type:"POST",
    url:"iajax.php",
    dataType:"JSON",
    data:f.serialize()+"&tipo=grabar",
    success:function(data){
        if(data.estado=="ok")
        {
            loader_grabar.removeClass("label-warning").addClass("label-success").text("Se Guardo
Correctamente").delay(4000).slideUp();
            listar();
        }
        else
        {
            alert(data.mensaje);
        }
    },
    beforeSend:function(){
        //Notificar al usuario mientras que se procesa su solicitud
        loader_grabar.removeClass("label-success").addClass("label label-warning").text("Procesando...").slideDown();
    },
    complete:function(){
        f.removeClass("busy");
        f[0].reset();
        <!--loader_grabar.removeClass("label-warning").addClass("label-success").text("Se Guardo
Correctamente").delay(4000).slideUp();-->
    }
});
return false;
});
//BORRAR
$("#btn_borrar").on("click", function(){
//CONFIRMAR ACCION AL USUARIO
if(confirm("Esta seguro?")==false)

```



```

{
    return;
}
var formulario_edicion = $("#formulario_edicion");
$.ajax({
    type:"POST",
    url:"iajax.php",
    data:formulario_edicion.serialize() + "&tipo=borrar",
    dataType:"JSON",
    success:function(data){
        //SABER CUANDO SE BORRO CORRECTAMENTE
        if(data.estado=="ok")
        {
            // MOSTRAR EL MENSAJE
            alert(data.mensaje);
            quitar_marcadores(marcadores_nuevos);
            //LIMPIAR EL FORMULARIO
            formulario_edicion[0].reset();
            //LISTAR OTRA VEZ LOS MARCADORES
            listar();
        }
        else
        {
            //ERROR AL BORRAR
            alert(data.mensaje);
        }
    },
    beforeSend:function(){

    },
    complete:function(){

    }
});
});
listar();
//FUERA DE READY JQUERY
//FUNCTION PARA RECUPERAR PUNTOS DE LA BD
function listar()
{
    quitar_marcadores(marcadores_bd);
    var formulario_edicion = $("#formulario_edicion");
    $.ajax({
        type:"POST",
        url:"iajax.php",
        dataType:"JSON",
        data:"&tipo=listar",
        success:function(data){
            if(data.estado=="ok")
            {
                $.each(data.mensaje, function(i, item){
                    //OBTENER LAS COORDENADAS DEL PUNTO
                    var posi = new google.maps.LatLng(item.cx, item.cy);
                    //CARGAR LAS PROPIEDADES AL MARCADOR
                    var marca = new google.maps.Marker({
                        idMarcador:item.IdPunto,
                        position:posi,
                        titulo: item.Titulo,
                        tipos: item.tipos,
                        cx:item.cx,      //esas coordenadas vienen de la BD
                        cy:item.cy      //esas coordenadas vienen de la BD
                    });
                    google.maps.event.addListener(marca, "click", function(){
                        $("#collapseTwo").collapse("show");
                        $("#collapseOne").collapse("hide");
                        formulario_edicion.find("input[name='id']").val(marca.idMarcador);
                        formulario_edicion.find("input[name='titulo']").val(marca.titulo).focus();
                        formulario_edicion.find("input[name='cx']").val(marca.cx);
                        formulario_edicion.find("input[name='cy']").val(marca.cy);
                    });
                });
            }
        }
    });
}

```

```

    });
    //AGREGAR EL MARCADOR A LA VARIABLE MARCADORES_BD
    marcadores_bd.push(marca);
    //UBICAR EL MARCADOR EN EL MAPA
    marca.setMap(mapa);
  });
}
else
{
  alert("No Existen Puntos en la BD");
}
},
beforeSend:function(){
},
complete:function(){
}
});
}
//PLANTILLA AJAX
</script>
</head>
<body>
<div id="mapa">
<!--<h2><center>Sucursales Bancarias</center></h2>-->
</div>
<div id="infor">
<h2>
<center>
    Gestion de Sucursales
  </center>
</h2>
<div class="container-fluid">
<div class="accordion" id="accordion2">
<div class="accordion-group">
<div class="accordion-heading"><a class="accordion-toggle" data-toggle="collapse" data-parent="#accordion2"
href="#collapseOne"> AGREGAR NUEVAS SUCURSALES </a> </div>
<div id="collapseOne" class="accordion-body collapse" style="height: 0px;">
<div class="accordion-inner">
<form id="formulario">
<table>
<tr>
<td>Entidad Financiera</td>
<td><input type="text" class="form-control" name="titulo" autocomplete="off"/></td>
</tr>
<tr>
<td>
<td class="tdatos">Tipo</td>
<td><select name="tipos">
<option value="CA">CAJERO AUTOMATICO </option>
<option value="AB">AGENCIA BANCARIA </option>
</select></td>
</tr>
<tr>
<td>
<td>Descripcion</td>
<td><input type="text" class="form-control" name="descripcion" autocomplete="off"/></td>
</tr>
<tr>
<td>
<td>LATITUD</td>
<td><input type="text" class="form-control" readonly name="cx" autocomplete="off"/></td>
</tr>
<tr>
<td>
<td>LONGITUD</td>
<td><input type="text" class="form-control" readonly name="cy" autocomplete="off"/></td>
</tr>
<tr>
<td></td>
<td><span id="loader_grabar" class=""></span></td>
</tr>
<tr>
<td><button type="button" id="btn_grabar" class="btn btn-success btn-sm">Guardar</button></td>

```

```

        <td><button type="button" class="btn btn-danger btn-sm">Cancelar</button></td>
    </tr>
</table>
</form>
</div>
</div>
</div>
<div class="accordion-group">
    <div class="accordion-heading"> <a class="accordion-toggle" data-toggle="collapse" data-parent="#accordion2"
href="#collapseTwo"> EDICION DE LAS SUCURSALES </a> </div>
    <div id="collapseTwo" class="accordion-body collapse">
        <div class="accordion-inner">
            <form id="formulario_edicion">
                <input type="hidden" name="id"/>
                <table>
                    <tr>
                        <td>Titulo</td>
                        <td><input type="text" class="form-control" name="titulo" autocomplete="off"/></td>
                    </tr>
                    <tr>
                        <td class="tdatos">Tipo</td>
                        <td><select name="tipos">
                            <option value="CA">CAJERO AUTOMATICO </option>
                            <option value="AB">AGENCIA BANCARIA </option>
                        </select></td>
                    </tr>
                    <tr>
                        <td>Descripcion</td>
                        <td><input type="text" class="form-control" name="descripcion" autocomplete="off"/></td>
                    </tr>
                    <tr>
                        <td>LATITUD</td>
                        <td><input type="text" class="form-control" readonly name="ex" autocomplete="off"/></td>
                    </tr>
                    <tr>
                        <td>LONGITUD</td>
                        <td><input type="text" class="form-control" readonly name="cy" autocomplete="off"/></td>
                    </tr>
                    <tr>
                        <td></td>
                        <td><span id="formulario_edicion" class=""></span></td>
                    </tr>
                    <tr>
                        <td><button type="button" id="btn_actualizar" class="btn btn-success btn-sm">Actualizar</button></td>
                        <td><button type="button" id="btn_borrar" class="btn btn-danger btn-sm">Borrar</button></td>
                    </tr>
                </table>
            </form>
        </div>
    </div>
</div>
<div class="accordion-group">
    <div class="accordion-heading"> <a class="accordion-toggle" data-toggle="collapse" data-parent="#accordion2"
href="#collapseThree"> LISTADO DE LOS REGISTROS </a> </div>
    <div id="collapseThree" class="accordion-body collapse">
        <div class="accordion-inner"> <a href="lista.php"
            target="popup"
            onclick="window.open('','popup','width = 700, height = 500')"></div>
        <td><center>
            <button type="button">LISTAR LAS SUCURSALES Y CAJEROS</button>
        </center></td>
    </div>
</div>
</div>
</div>
</div>
</body>
</html>

```

## ANEXO B

### CODIGO APLICACIÓN MOVIL ANDROID

#### MainActivity.java

```
package com.desarrollolibre.googlemapsv1;

import android.os.Bundle;
import android.view.Menu;

public class MainActivity extends android.support.v4.app.FragmentActivity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

#### AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.desarrollolibre.googlemapsv1"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <permission
        android:name="com.desarrollolibre.googlemapsv1.permission.MAPS_RECEIVE"
        android:protectionLevel="signature" />
    <permission
        android:name="com.desarrollolibre.googlemapsv1.permission.C2D_MESSAGE"
        android:protectionLevel="signature" />

    <uses-permission
        android:name="com.desarrollolibre.googlemapsv1.permission.C2D_MESSAGE" />
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
    <uses-permission
        android:name="com.desarrollolibre.googlemapsv1.permission.MAPS_RECEIVE" />
    <uses-permission
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

```

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

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

<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name="com.desarrollolibre.googlemaps1.MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <meta-data
        android:name="com.google.android.gms.version"
        android:value="4323000" />
    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="" />
</application>

</manifest>

menu.xml

<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/opcion1" android:title="Configuracion"></item>
    <item android:id="@+id/opcion2" android:title="Ver en"></item>
    <item android:id="@+id/opcion3" android:title="Acerca de"></item>
    <item android:id="@+id/opcion3" android:title="Ayuda"></item>
    <item android:id="@+id/opcion3" android:title="salir"></item>
</menu>

```



# DOCUMENTOS

---