

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



TESIS DE GRADO
SOFTWARE MÓVIL DE GEOLOCALIZACIÓN PARA LA
BANCA EN LA CIUDAD DE LA PAZ

Para optar al Título de Licenciatura en Informática

MENCIÓN: Ingeniería de Sistemas Informáticos

POSTULANTE: ALEX PEREZ CORIMAYTA
TUTOR: LIC. GROVER ALEX RODRIGUEZ RAMIREZ
REVISOR: LIC. JAVIER REYES PACHECO

LA PAZ – BOLIVIA

2014



**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

A Dios por ser la luz que me guía a cumplir cada objetivo y levantarme de cada caída.

A mi madre y mi padre por la confianza e inspiración que me brindan con su ejemplo de vida.

A mis hermanos por su apoyo, comprensión y cariño.

A mis amigos que a lo largo de mi vida universitaria me brindaron lo mejor de la amistad.

AGRADECIMIENTOS

Agradezco a mi docente tutor Lic. Grover Alex Rodriguez Ramirez, por impartirme sus conocimientos y experiencia profesional, respaldándome durante la elaboración y conclusión de la presente tesis; colaborándome desinteresadamente, aconsejándome con acierto, permitiéndome presentarles este trabajo. También, agradezco infinitamente a mi docente revisor Lic. Javier Reyes Pacheco, por su incondicional apoyo, por sus observaciones y sugerencias que le dieron forma al trabajo.

CONTENIDO

RESUMEN.....	xiii
SUMMARY.....	xiv
CAPÍTULO I.....	1
1 MARCO REFERENCIAL.....	1
1.1 INTRODUCCIÓN.....	1
1.2 ANTECEDENTES.....	2
1.3 OBJETO DE ESTUDIO O INVESTIGACIÓN.....	3
1.4 PLANTEAMIENTO DEL PROBLEMA.....	3
1.4.1 FORMULACIÓN DEL PROBLEMA.....	4
1.5 OBJETIVOS.....	4
1.5.1 OBJETIVO GENERAL.....	4
1.5.2 OBJETIVOS ESPECÍFICOS.....	4
1.6 HIPÓTESIS.....	5
1.7 VARIABLES.....	5
1.7.1 VARIABLE INDEPENDIENTE.....	5
1.7.2 VARIABLE DEPENDIENTE.....	5
1.7.3 VARIABLE MODERANTE.....	6
1.8 JUSTIFICACIÓN.....	6
1.8.1 JUSTIFICACIÓN SOCIAL.....	6
1.8.2 JUSTIFICACIÓN ECONÓMICA.....	6
1.8.3 JUSTIFICACIÓN TÉCNICA.....	6
1.9 LÍMITES Y ALCANCES.....	6
1.10 METODOLOGÍA.....	7
1.10.1 TIPO DE INVESTIGACIÓN.....	7

1.10.2	MÉTODO DE APLICACIÓN.....	8
	CAPÍTULO II.....	9
2	MARCO TEÓRICO.....	9
2.1	INTRODUCCIÓN.....	9
2.2	SISTEMAS DE INFORMACIÓN GEOGRÁFICA	9
2.3	FORMULA DE HAVERSINE.....	11
2.4	METODOLOGÍA DE DESARROLLO MOBILE-D	13
2.4.1	FASES.....	14
2.4.1.1	EXPLORACIÓN	14
2.4.1.2	INICIALIZACIÓN	15
2.4.1.3	PRODUCCIÓN.....	15
2.4.1.4	ESTABILIZACIÓN	15
2.4.1.5	PRUEBAS DEL SISTEMA	16
2.4.2	ELEMENTOS	16
2.4.3	ENFOQUE PRAGMÁTICO.....	17
2.5	GOOGLE MAP MAKER.....	17
2.6	API DE GOOGLE MAPS	18
2.6.1	LIBRERÍA DE AUTOCOMPLETADO DE LUGARES.....	18
2.6.2	LIBRERÍA DE DIBUJO.....	19
2.6.3	SERVICIO DE RUTAS	19
2.7	PHONEGAP	20
2.8	ANDROID	21
2.8.1	ARQUITECTURA.....	22
2.9	EMULADOR RIPPLE.....	24

2.9.1	CAPACIDADES DE RIPPLE	24
CAPÍTULO III.....		26
3	DISEÑO METODOLÓGICO	26
3.1	INTRODUCCIÓN.....	26
3.2	SOFTWARE MÓVIL HÍBRIDO CON PHONEGAP	26
3.2.1	PROCESO DE DESARROLLO	27
3.2.1.1	CODIGO NATIVO	28
3.2.1.2	CODIGO HTML.....	28
3.2.1.3	CODIGO FUENTE APLICACIÓN.....	28
3.2.1.4	SDK TOOLS.....	28
3.2.1.5	COMPILADOR	29
3.2.1.6	PACKAGE	29
3.2.1.7	PACKAGE DISTRIBUIBLE	29
3.2.1.8	MARKETPLACE.....	29
3.3	SERVICIOS BASADOS EN LOCALIZACIÓN.....	29
3.3.1	TIPOS DE ENTIDADES	30
3.3.1.1	DATOS VECTORIALES PUNTUALES	30
3.3.1.2	DATOS RASTER	30
3.3.2	MAPAS DIGITALES	30
3.4	LA GEOLOCALIZACIÓN COMO PROCESO DE UN SOFTWARE MÓVIL HÍBRIDO	30
3.5	DESARROLLO DEL PROTOTIPO	31
3.5.1	EXPLORACIÓN.....	31

3.5.1.1	DEFINICIÓN DE LOS GRUPOS DE INTERES	32
3.5.1.2	ACTORES.....	32
3.5.1.3	ROLES Y TAREAS	32
3.5.1.4	HISTORIAS DE USUARIO.....	33
3.5.1.5	COLECCIÓN DE REQUERIMIENTOS	36
3.5.1.6	REQUERIMIENTOS.....	36
3.5.1.7	ARQUITECTURA.....	37
3.5.2	INICIALIZACIÓN	39
3.5.2.1	LISTA DE TAREAS	39
3.5.2.2	PLAN DE ENTREGAS	40
3.5.2.3	PLANIFICACIÓN DE FASES	41
3.5.2.4	ITERACIONES.....	41
3.5.2.5	DIAGRAMA DE CLASES	43
3.5.2.6	INTERFAZ CON API DE LOCALIZACIÓN DE GOOGLE MAPS ..	43
3.5.2.7	DIAGRAMA DE ACTIVIDADES	44
3.5.2.8	DISEÑO DE LA INTERFAZ DEL ADMINISTRADOR.....	45
3.5.2.9	DISEÑO DE LA INTERFAZ DEL USUARIO	53
3.5.3	PRODUCCIÓN.....	56
3.5.3.1	ESTÁNDARES DE CODIFICACIÓN.....	56
3.5.3.1.1	WORKSHOP DE POST ITERACIÓN.....	56
3.5.3.2	ADICION DE PUNTOS DE SERVICIOS FINANCIEROS.....	56
3.5.3.3	MODIFICACION DE PUNTOS DE SERVICIOS FINANCIEROS ..	57
3.5.3.4	ESTABLECIMIENTO PARÁMETROS DE BÚSQUEDA.....	57
3.5.3.5	ESTABLECIMIENTO DE PUNTOS DE ORIGEN Y DESTINO	57
3.5.3.6	BÚSQUEDA DE PUNTOS DE SERVICIO FINANCIERO	58
3.5.3.7	DISCRIMINACIÓN DE ENTIDADES FINANCIERAS Y TIPOS DE SERVICIO	58
3.5.3.8	TRAZADO DE LA RUTA MINIMA ENTRE DOS PUNTOS	58
3.5.3.9	TEST DE ACEPTACIÓN.....	59

3.5.4	ESTABILIZACIÓN	61
3.5.4.1	WORKSHOP DE POST ITERACIÓN.....	61
3.5.4.2	TEST DE ACEPTACIÓN.....	61
3.5.5	PRUEBAS	62
3.5.5.1	PLAN DE PRUEBAS.....	62
3.5.5.2	PRUEBAS DE ACEPTACIÓN POR ITERACIÓN.....	63
3.5.5.2.1	Resultados Iteración 1	64
3.5.5.2.2	Resultados Iteración 2	64
3.5.5.2.3	Resultados Iteración 3	64
3.5.5.2.4	Resultados Iteración 4	65
3.5.5.2.5	Resultados Iteración 5	65
3.5.5.2.6	Resumen de resultados de iteraciones.....	65
3.5.5.3	RESULTADOS TIEMPOS DE ACCESO.....	66
CAPÍTULO IV.....		67
4	EVALUACIÓN DE RESULTADOS.....	67
4.1	INTRODUCCIÓN.....	67
4.2	EVALUACIÓN DE LOS CASOS DE PRUEBA	67
4.3	PRUEBA DE HIPÓTESIS	67
4.3.1	EVALUACIÓN DE RESULTADOS	68
4.3.2	DEFINICIÓN DE LA HIPÓTESIS H0 Y H1	68
4.3.3	DETERMINACIÓN DE LA REGIÓN CRÍTICA.....	69
4.3.4	CÁLCULO DEL ESTADÍSTICO DE LA PRUEBA.....	70
4.3.5	TOMA DE DECISIÓN.....	71

CAPÍTULO V.....	72
5 CONCLUSIONES Y RECOMENDACIONES	72
5.1 CONCLUSIONES	72
5.2 RECOMENDACIONES.....	73
BILIOGRAFÍA	75
ANEXOS.....	77
ANEXO A.....	78
TABLA DE DISTRIBUCIÓN NORMAL.....	78
ANEXO B.....	79
PRUEBA DE HIPÓTESIS PARA PROPORCIONES	79
DOCUMENTACIÓN.....	82

INDICE DE TABLAS

Tabla 1: Roles y tareas de los usuarios	32
Tabla 2: Historia de usuario	34
Tabla 3: Historia de usuario Actualización de Ubicaciones.....	34
Tabla 4: Historia de usuario despliegue de ubicaciones de Entidades Financieras	35
Tabla 5: Historia de usuario	36
Tabla 6: Lista de tareas	40
Tabla 7: Descripción de Iteraciones.....	42
Tabla 8: Estimación de esfuerzo Procesos principales.....	43
Tabla 9: Test de aceptación de Producción	60
Tabla 10: Test de aceptación de Estabilización	62
Tabla 11: Resultados Iteración 1	64
Tabla 12: Resultados Iteración 2	64
Tabla 13: Resultados Iteración 3	64
Tabla 14: Resultados Iteración 4	65
Tabla 15: Resultados Iteración 5	65
Tabla 16: Resumen de resultados de iteraciones.....	65
Tabla 17: Resultado tiempos de acceso	66
Tabla 18: Promedio tiempos de acceso	66

INDICE DE FIGURAS

Figura 1: Triángulo esférico resuelto por la ley de Haversine	13
Figura 2: Fases y Etapas de Mobile-D.....	14
Figura 3: Características de Google Maps Maker.....	18
Figura 4: Desarrollo de aplicaciones basados en phonegap	20
Figura 5: Características de phonegap	21
Figura 6: Arquitectura de Android	23
Figura 7: Arquitectura de desarrollo de una aplicación híbrida	27
Figura 8: Esquema General de la Geolocalización como proceso de un software móvil híbrido.....	31
Figura 9: Arquitectura del prototipo.....	38
Figura 10: Cronograma de Trabajo.....	40
Figura 11: Esquema de desarrollo del sistema.....	41
Figura 12: Interfaz con API de localización de Google Maps.....	43
Figura 13: Diagrama de actividades	44
Figura 14: Inicio de sesión	45
Figura 15: Ingreso de datos erróneos	45
Figura 16: Pantalla de inicio.....	46
Figura 17: Adicionar Sucursal	46
Figura 18: Seleccionar Entidad Financiera	47
Figura 19: Seleccionar Tipo de Servicio	47
Figura 20: Cargando Nuevo Registro	48

Figura 21: Entidad Financiera Adicionada correctamente	48
Figura 22: Modificar Entidad Financiera	49
Figura 23: Seleccionar Entidad a Modificar	49
Figura 24: Modificar o Eliminar Registro	50
Figura 25: Adicionar usuario	50
Figura 26: Usuario adicionado correctamente	51
Figura 27: Eliminación de usuario.....	51
Figura 28: Usuario eliminado correctamente	52
Figura 29: Interfaz para selección entidad financiera y selección de radio de búsqueda	53
Figura 30: Mapa y panel de Resultados	54
Figura 31: Elección de punto y trazado de ruta	55
Figura 32: Fragmento de código para la implementación de la Fórmula de Haversine.....	59
Figura 33: Región crítica para la hipótesis.....	69
Figura 34: Resultado tabla de la función de distribución normal.....	70
Figura 35: Distribución de Z_0 y Z_c en el gráfico para la toma de decisión.....	71

RESUMEN

El crecimiento de la banca en nuestro medio se hace evidente mediante el crecimiento del número de sucursales y cajeros automáticos de cada una de las entidades financieras. A veces esta situación pasa desapercibida por el usuario ya que no cuenta con información a la mano, y esto hace que el usuario recorra mayores distancias para llegar a una sucursal o cajero automático conocido por el mismo ya con anterioridad, con la posibilidad de que exista el servicio requerido más próximo a su ubicación en ese momento.

El presente trabajo de investigación, ofrece un sistema cuyo proceso principal se basa en la geolocalización de las entidades bancarias, tanto sucursales como cajeros automáticos, que permita al usuario ubicar a la sucursal o cajero automático de la entidad requerida más próxima a su ubicación trazando una o más rutas según el radio de aproximación requerida.

A fin de alcanzar el objetivo propuesto, se desarrolló dos herramientas de software. Una que se aplica en un entorno web destinada a la gestión y administración de ubicaciones de los diversos servicios bancarios de cada una de las entidades. La segunda que se aplica a un entorno móvil destinado al usuario permitiéndole realizar la consulta de ubicaciones referentes a su ubicación actual.

Palabras clave: Georreferenciación, Mobile-D, Phoneyap, GPS, Google Maps.

SUMMARY

The growth of banking in our environment is evident by the growing number of branches and automated for each of the financial institutions ATMs. Sometimes this situation goes unnoticed by the user since no information at hand, and this makes the user scroll farther to reach a branch or ATM known by the same as above, with the possibility of the required service nearest to your current location.

The present research provides a system whose main process is based on the geolocation of banks, both branches and ATMs, which allows the user to locate the branch or ATM of the requested location closer to drawing a or more routes as required radius approximation.

To achieve this objective, two software tools were developed. One that is applied in a web environment dedicated to the management and administration of the various locations of banking services in each of the entities. The second applied to a mobile environment intended for the user making the request allowing locations related to their current location.

Keywords: *Geolocation, Mobile-D, Phonegap, GPS, Google Maps.*

CAPÍTULO I

1 MARCO REFERENCIAL

1.1 INTRODUCCIÓN

El crecimiento de la banca en nuestro medio se hace evidente mediante el crecimiento del número de sucursales y cajeros automáticos de cada una de las entidades financieras. A veces esta situación pasa desapercibida por el usuario ya que no cuenta con información a la mano, y esto hace que el usuario recorra mayores distancias para llegar a una sucursal o cajero automático conocido por el mismo ya con anterioridad, con la posibilidad de que exista el servicio requerido más próximo a su ubicación en ese momento.

En la actualidad la geolocalización o georreferenciación, que es utilizado como un proceso en el desarrollo de Sistemas de Información Geográfica (SIG o GIS, en su acrónimo inglés Geographic Information System), se ha convertido en una poderosa herramienta para obtener ubicaciones según su latitud y longitud. Tal utilidad puede ser utilizada para obtener datos precisos sobre la ubicación de los diversos servicios que nos ofrecen las entidades bancarias.

Un Sistema de Información Geográfica funciona como una base de datos con información geográfica que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, señalando un objeto se conocen sus atributos e, inversamente, preguntando por un registro de la base de datos se puede saber su localización en la cartografía.

Uno de los servicios que causo un salto cualitativo en cuanto a georreferenciación es Google Earth, ya que se ha democratizado el uso de información georreferenciada,

puesto que puede ser utilizado sobre todos los contenidos de tipo social presentes en la actualidad.

Aún en la actualidad algunos de estos servicios son poco exactos al momento de georreferenciar una ubicación, además de que son de difícil uso. Google Maps como referencia ha mostrado un registro moroso de sitios y lugares, además de inexactitud de ubicaciones.

El presente trabajo de investigación, ofrece un Sistema cuyo proceso principal se basa en la geolocalización de servicios que nos brindan las entidades bancarias, tanto sucursales como cajeros automáticos, que permita al usuario ubicar a la sucursal o cajero automático de la entidad requerida más próxima a su ubicación trazando una ruta según el radio de aproximación requerido.

1.2 ANTECEDENTES

El crecimiento de la Banca en nuestro país gracias al desarrollo de los departamentos y por consiguiente sus habitantes, más aún, en el área troncal del país, ha dado paso al posicionamiento geográfico estratégico de nuevas centrales, sucursales y nuevos mecanismos de transacción financiera, como son los cajeros automáticos y sucursales, de cada una de las diferentes entidades bancarias.

Desde la aparición del Canadian Geographical Information System (CGIS) en 1962 hasta la actualidad se han ido implementando numerosas aplicaciones de los SIG en los más variados ámbitos, por lo que ha dejado de ser un campo de geógrafos y planificadores y se ha convertido en una herramienta multidisciplinaria, debido a la aparición en los últimos años de nuevas herramientas cuya facilidad de uso ha extendido y democratizado esta tarea fuera del ámbito técnico existente hasta ahora.(Chang, 2010)

El área en cuanto a servicios referentes a georreferenciación ha ido en crecimiento con constantes avances para mejorar los tiempos de respuesta al usuario. Una de estas empresas es Foursquare que es una red social con más de 2.7 millones de usuarios, una

aplicación multiplataforma excepto la aplicación nativa para iPad y cuenta con su propia API.

Yelp es una empresa que brinda servicios en EEUU, y cuenta con más de 2 millones de usuarios, cuenta con su propia API para el desarrollo y es multiplataforma.

Google Maps de la empresa Google, es el SIG que más usuarios tiene, ya que cuenta con información mundial de fácil acceso por su documentación y servicio de soporte, pero cuenta con imprecisiones y retraso dependiendo al ancho de banda y al hardware del ordenador.

Google Maps tiene documentación aún en fase de desarrollo, es por eso que varias empresas van colaborando al proyecto Open Street Maps como por ejemplo: Apple, Microsoft (Mapas Bing), Yahoo (Mapas yahoo), Foursquare que recientemente cambio de Google Maps a Open Street Maps en su versión pc ya que su versión para móviles aún sigue usando Google Maps.(Duclos, 2010)

Del mismo modo, la masificación y evolución constante de la georreferenciación se ha visto impulsada por el uso mashups (aplicaciones web híbridas) en sitios Web 2.0, permitiendo la localización de contenidos digitales (vídeo, noticias, modelados 3D, etc.) en cartografía digital, dentro de lo que se ha venido a llamar la Información Geográfica Voluntaria.

1.3 OBJETO DE ESTUDIO O INVESTIGACIÓN

Desarrollo de un software centrado en un servicio de georreferenciación con respecto a la ubicación de servicios bancarios en la ciudad de La Paz.

1.4 PLANTEAMIENTO DEL PROBLEMA

Mediante el análisis y la respectiva observación del trabajo y la importancia de las entidades financieras, se determinó la necesidad de implementar un sistema de geolocalización que haga referencia a la información de la ubicación de este tipo de servicio.

1.4.1 FORMULACIÓN DEL PROBLEMA

Los datos sobre las ubicaciones de sucursales y cajeros automáticos se encuentran por separado en cada una de las entidades financieras.

La ciudad de La Paz cuenta con servicios de información inadecuados con respecto a consultas de georreferenciación limitándose a mostrar avenidas, calles, plazas y algunas instituciones.

La falta de información sobre la ubicación del cajero automático más cercano en cualquier lugar a cualquier hora del día y cualquier día de la semana.

El hecho de no conocer la ciudad imposibilita al usuario a tener un fácil acceso a los datos de ubicación de la entidad aun teniendo en mano la dirección.

Planteándose así un problema principal de la investigación que es:

¿CÓMO RECIBIR INFORMACIÓN CONFIABLE Y PRECISA SOBRE LA UBICACIÓN Y LA RUTA HACIA EL SERVICIO DE UNA ENTIDAD FINANCIERA USANDO TECNOLOGÍAS DE GEORREFERENCIACIÓN Y EL USO DE UN DISPOSITIVO MÓVIL?

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

Desarrollar un software móvil utilizando geolocalización que involucre información sobre la ubicación de servicios financieros, y que le brinde al usuario una herramienta confiable y precisa capaz de mostrar la ubicación y trazar la ruta hacia la ubicación del servicio requerido.

1.5.2 OBJETIVOS ESPECÍFICOS

- Desarrollar un módulo que permita ubicar un tipo de servicio de alguna entidad financiera en específico.

- Diseñar una base de datos con los registros de ubicaciones y los detalles de cada uno de los servicios de las entidades financieras.
- Interacción de la aplicación con Google Maps.
- Implementar un módulo visual de trazado de rutas, que permita ubicar al usuario e indicarle las rutas a través del uso trazados de ruta dentro de un mapa.
- Implementar un módulo visual de ubicación de entidades financieras indicando el tipo de servicio dentro de un mapa a través del uso de marcadores.
- Implementar un módulo visual que permita identificar entidades próximas mediante el trazado de un radio respecto a la ubicación del usuario.
- Desarrollar una aplicación web para dispositivos móviles, ya sean SmartPhones o Tablets.

1.6 HIPÓTESIS

H_0 : “LA GEOLOCALIZACIÓN COMO PROCESO DE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA PERMITE RECIBIR INFORMACIÓN CONFIABLE Y PRECISA SOBRE UBICACIONES DE SERVICIOS FINANCIEROS DENTRO DE UN MAPA, Y MEDIANTE EL USO DE UN DISPOSITIVO MÓVIL SE PODRÁ ESTABLECER LA RUTA HACIA UNA DE ESAS POCISIONES”, CON UN GRADO DE CONFIABILIDAD DE AL MENOS 85%, ADEMÁS DE UN NIVEL DE SIGNIFICANCIA A DEL 5%.

1.7 VARIABLES

1.7.1 VARIABLE INDEPENDIENTE

Información confiable y precisa sobre ubicaciones de servicios financieros dentro de un mapa.

1.7.2 VARIABLE DEPENDIENTE

Mediante el uso de un dispositivo móvil se podrá establecer la ruta hacia una de esas posiciones.

1.7.3 VARIABLE MODERANTE

La geolocalización como proceso de un sistema de información geográfica.

1.8 JUSTIFICACIÓN

1.8.1 JUSTIFICACIÓN SOCIAL

La necesidad de implementar este sistema se centra en el ahorro de tiempo y recursos por parte del usuario al momento de buscar la entidad financiera o el cajero automático más cercano referente a su ubicación actual, teniendo a la mano la información de los detalles necesarios que proporciona cada entidad.

1.8.2 JUSTIFICACIÓN ECONÓMICA

El presente trabajo se justifica económicamente ya que el costo económico en cuanto al sistema se refiere solamente al mantenimiento, puesto que el software utilizado para el desarrollo e implementación del sistema es de carácter libre.

1.8.3 JUSTIFICACIÓN TÉCNICA

El presente proyecto se desarrolló a partir del framework de Phonegap, que permite desarrollar aplicaciones para diferentes plataformas móviles (Android, IOS, Symbian, blackberry, webOS), basándose en tecnologías web (HTML, CSS, JavaScript, JQuery, Ajax), con el fin de adaptar la compatibilidad con la mayor cantidad de navegadores web y dispositivos móviles.

1.9 LÍMITES Y ALCANCES

- El software móvil se desarrollará para dispositivos móviles, como SmartPhones y tablets con sistema operativo Android.
- El área de trabajo se limita al sector financiero; centrales y sucursales de entidades bancarias, así como cajeros automáticos.

- La interacción del software con los mapas de Google Maps y Open Street Maps brindará información confiable y actualizada desde cualquier dispositivo móvil con sistema operativo android.
- Se accederá al sistema desde internet.
- El análisis y diseño del presente proyecto se ha delimitado bajo la tecnología y los datos que se tienen en la ciudad de La Paz de tal manera que el proyecto solo alcanzara un desarrollo como prototipo.
- Debido a que el sistema que se presenta plantea un prototipo, la información almacenada dentro de la base de datos solo toma en cuenta un número mínimo representativo en cuanto a ubicaciones de servicios bancarios y no así la totalidad existente dentro de la ciudad de La Paz.

1.10 METODOLOGÍA

1.10.1 TIPO DE INVESTIGACIÓN

Para alcanzar los objetivos propuestos referentes a la presente tesis se procedió a revisar la información centrada en una metodología móvil aplicable que satisfaga los diversos requerimientos específicos que el desarrollo móvil necesita.

Se consultará como se han desarrollado las aplicaciones en el pasado para saber que hay que mejorar o cambiar en cuanto a la aplicación de la metodología en el desarrollo del software móvil.

Se procederá a asignar pasos bien especificados sobre la realización de cada uno de los procesos, documentándose cada avance y cada proceso para que forme parte del manual que se presentará al final del proceso e implementación del software.

Es por ello que se requiere hacer un tipo de estudio aplicado y tecnológico basado en pruebas del objeto de estudio.

1.10.2 MÉTODO DE APLICACIÓN

Las características requeridas para el desarrollo de software móvil (sistema cliente) se adaptan a metodologías ágiles, aunque, la metodología elegida es Mobile-D que viene siendo una mezcla de muchas técnicas, ya que se ha apoyado en otras soluciones bien conocidas y consolidadas: *eXtreme Programming (XP)*, *Crystal Metodologies* y *Rational Unified Process (RUP)*. Es por eso que la metodología apunta a la necesidad de disponer de un ciclo de desarrollo muy rápido, aplicando los principios de programación extrema en lo que se refiere a las prácticas de desarrollo, las metodologías Crystal proporcionan un input muy valioso en términos de la escalabilidad de los métodos y el RUP que es la base para el diseño completo del ciclo de vida.

Mobile-D al combinar los beneficios de las metodologías XP, Crystal y RUP proporciona las siguientes razones para ser la metodología seleccionada en el desarrollo del software móvil:

- Está diseñada para el desarrollo de aplicaciones móviles.
- Es una metodología ágil con ciclos de desarrollo cortos y para equipos pequeños.
- Facilidad para detectar y resolver tempranamente problemas técnicos.
- Se basa en el desarrollo basado en pruebas que es una de las mejores formas de asegurar la calidad.
- Se logra mejores diseños al basarse en el desarrollo basado en pruebas.
- Tiene un enfoque centrado en la satisfacción del usuario final, permitiendo mejorar el producto al realizar iteraciones cortas.
- Las tareas en cada fase están bien detalladas.

CAPÍTULO II

2 MARCO TEÓRICO

2.1 INTRODUCCIÓN

La geolocalización o georreferenciación es el posicionamiento en el que se define la localización de un objeto espacial en un sistema de coordenadas y datos determinados. El proceso de geolocalización se aplica a un SIG que en este caso será Google Maps.

2.2 SISTEMAS DE INFORMACIÓN GEOGRÁFICA

Un Sistema de Información Geográfica (SIG o GIS, en su acrónimo inglés) es una integración organizada de hardware, software, datos geográficos y personal, diseñada para capturar, almacenar, manejar, analizar, modelar y representar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer unas necesidades concretas de información.

El SIG funciona como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, señalando un objeto se conocen sus atributos e, inversamente, preguntando por un registro de la base de datos se puede saber su localización en la cartografía. (Gutiérrez, 2000)

La razón fundamental para utilizar un SIG es la gestión de la información espacial. El sistema permite separar la información en diferentes capas temáticas y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, y facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no podríamos obtener de otra forma.

Hardware

Los ordenadores personales y los dispositivos en telefonía móvil que se ofrecen hoy en cualquier tienda de informática cumplen los requerimientos para la creación y operación de un SIG. Los dispositivos y periféricos opcionales tienen su utilidad principalmente en la entrada y salida de los datos (módem, escáner, GPS, impresora en color, y otros).

Software

Cada programa (o paquete) de SIG es un conjunto de algoritmos para acceder, analizar y sintetizar datos espaciales y sus atributos asociados. Unas funciones son clásicas y las encontramos en prácticamente todos los paquetes de SIG y otras funciones son más particulares. Existen programas que enfocan más en el manejo de la cartografía vectorial y otros para la cartografía ráster. Casi todos utilizan ya ambos formatos, aunque con predominio de uno de ellos. La elección del programa SIG depende de las aplicaciones y análisis que se pretende realizar. No hay ningún líder entre los programas, algunos tienen muy buenas herramientas para el tratamiento de imágenes de satélite y otros incluyen un amplio rango de módulos para el modelado y evaluaciones estadísticas. Los usuarios experimentados utilizan normalmente varios programas distintos, según características de los datos y aplicaciones.

Los precios se encuentran entre coste cero (software libre) hasta varios miles de dólares. Un buen programa para empezar (y para llegar bastante lejos) es Idrisi, un programa creado en la universidad norte-americana Clark University. Tiene su lado fuerte en el análisis y la modelación en el formato ráster y su lado débil en el formato vectorial y la salida de cartografía. Por otra parte permite acceder a la estructura de los datos, lo que facilita el familiarizarse con los principios del SIG y sus procedimientos. (Duclos, 2010)

Datos

La mayoría de la cartografía digital actual proviene de cartografía tradicional que se ha digitalizado en tableta o escaneado y después vectorizado. Una fuente importante de

información son las coordenadas tomadas por un GPS (p.ej. observaciones durante un censo), y otra fuente importante - muy distinta en su estructura a la anterior – son las fotos aéreas e imágenes de satélite. Lo que realmente hace un SIG interesante es la posibilidad de generar nueva cartografía a partir de los mapas iniciales: Un mapa de distancia a la carretera más próxima a partir de un mapa de la red de carreteras, un mapa del índice de erosión a partir de información sobre de tipo de suelo y cubierta vegetal en combinación con precipitaciones y pendiente del terreno, etc.

Personal

El usuario enlaza con todos estos componentes y pone el SIG en funcionamiento. El éxito de la implementación y del diseño del SIG depende en gran medida de sus conocimientos (tanto del SIG como del tema a analizar), del conjunto de métodos, ideas y modelos que aplica en el proyecto, su capacidad de reconocer y resolver problemas que pueden surgir durante el proceso y de su capacidad de adquirir o convertir información al formato digital para integrarla en el SIG.(Chang, 2010)

A menudo son los componentes de Hardware y Software los que más llaman la atención y en que más dinero se invierte. Sin embargo, los puntos cruciales son los Datos y el/los Usuario(s). Por un lado porque la disponibilidad de los datos sobre que se realizan las operaciones, la precisión y actualidad son lo que determina la salida de resultados. Y por parte de los usuarios porque es imposible aprender hasta las profundidades el manejo de un SIG en un fin de semana, leyendo solo el manual del programa.

2.3 FORMULA DE HAVERSINE

La fórmula del semiverseno es una importante ecuación para la navegación astronómica, en cuanto al cálculo distancia de círculo máximo entre dos puntos de un globo sabiendo sus longitud y latitud. Es un caso especial de una fórmula más general de trigonometría esférica, la ley de los semiversenos, que relaciona los lados y los ángulos de "triángulos esféricos".

Estos nombres se derivan del hecho de que suele expresarse en términos de la función haversine, dada por $\text{hav}(\theta) = \frac{1 - \cos(\theta)}{2}$. Las fórmulas también podrían estar escritas en términos de cualquier múltiplo del haversine, como la antigua función verseno (el doble del haversine).

Históricamente, el haversine tuvo, quizás, una ligera ventaja ya que su máximo es "1", por lo que las tablas logarítmicas de sus valores podían acabar con el valor cero. Hoy en día, la forma del haversine también es interesante, ya que no tiene ningún coeficiente delante de la función seno ².

Para cualquier par de puntos sobre una esfera:

$$\text{hav} \frac{d}{R} = \text{hav} \varphi_1 - \varphi_2 + \cos \varphi_1 \cos \varphi_2 \text{hav} \Delta \lambda \quad (\text{Ec. 1})$$

Donde

- *hav* es la función haversine, $\text{hav}(\theta) = \frac{1 - \cos(\theta)}{2}$
- *d* es la distancia entre los dos puntos (a lo largo de un círculo máximo de la esfera, véase distancia esférica),
- *R* es el radio de la esfera,
- φ_1 es la latitud del punto 1,
- φ_2 es la latitud del punto 2, y
- $\Delta \lambda$ es la diferencia de longitud

Tenga en cuenta que el argumento a la función haversine Aquí se supone que debe darse en radiánes. En grados, $\text{hav}(d/R)$ de la fórmula se convertiría en $\text{hav}(180 \cdot d / \pi R)$.

Entonces se puede resolver por *d* ya sea mediante la simple aplicación de la haversine inversa (si está disponible) o mediante el uso de la arcoseno (arcoseno).(Pratt, 2013)

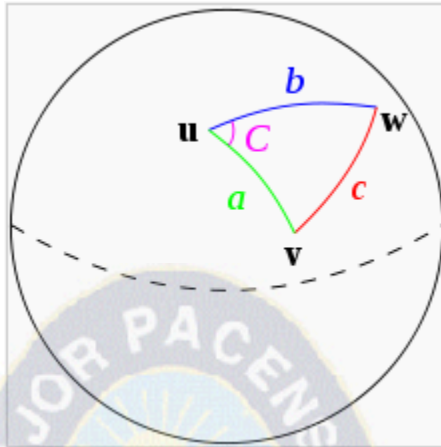


Figura 1: Triángulo esférico resuelto por la ley de Haversine

Fuente: Wikipedia.org

2.4 METODOLOGÍA DE DESARROLLO MOBILE-D

Mobile-D es una metodología para el desarrollo ágil de software, que no solamente está orientada al desarrollo de aplicaciones móviles, también se puede usar en aplicaciones en aplicaciones de seguridad, financieras, de logística y de simulación.

Mobile-D se basa en la programación extrema (XP) para la implementación, Crystal Methodologies para la escalabilidad y en el Proceso Unificado de Desarrollo (RUP) para la cobertura del ciclo de vida.

La metodología se creó en un periodo de intenso crecimiento en el terreno de las aplicaciones móviles. Por tanto, en ese momento no existían demasiados principios de desarrollo a los que acudir. Los autores de Mobile-D apuntan a la necesidad de disponer de un ciclo de desarrollo muy rápido para equipos muy pequeños. De acuerdo con sus suposiciones, Mobile-D está pensado para grupos de no más de 10 desarrolladores colaborando en un mismo espacio físico. Si trabajan con el ciclo de desarrollo propuesto, los proyectos deberían finalizar con el lanzamiento de productos completamente funcionales en menos de diez semanas. (Tanja, 2012)

2.4.1 FASES

El desarrollo siguiendo el enfoque de Mobile-D se compone de 5 fases, cada una de estas fases tiene asociado etapas, tareas y prácticas, como se muestra en la Figura 1.

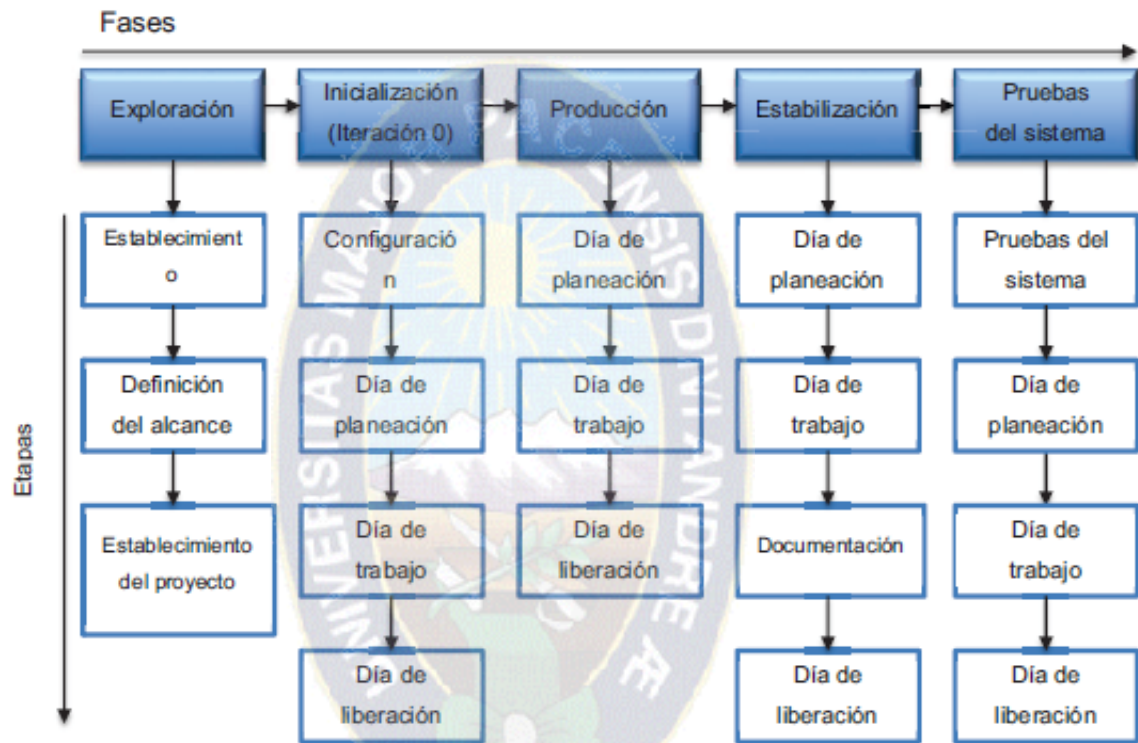


Figura 2: Fases y Etapas de Mobile-D

Fuente: VVT Electronics

2.4.1.1 EXPLORACIÓN

La fase de exploración, siendo ligeramente diferente del resto del proceso de producción, se dedica al establecimiento de un plan de proyecto y los conceptos básicos. Por lo tanto, se puede separar del ciclo principal de desarrollo (aunque no debería obviarse).

Los autores de la metodología ponen además especial atención a la participación de los clientes en esta fase.

2.4.1.2 INICIALIZACIÓN

Durante la fase de inicialización, los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico (incluyendo el entrenamiento del equipo de desarrollo). Los autores de Mobile-D afirman que su contribución al desarrollo ágil se centra fundamentalmente en esta fase, en la investigación de la línea arquitectónica. Esta acción se lleva a cabo durante el día de planificación. Los desarrolladores analizan el conocimiento y los patrones arquitectónicos utilizados en la empresa (extraídos de proyectos anteriores) y los relacionan con el proyecto actual. Se agregan las observaciones, se identifican similitudes y se extraen soluciones viables para su aplicación en el proyecto. Finalmente, la metodología también contempla algunas funcionalidades nucleares que se desarrollan en esta fase, durante el día de trabajo.

2.4.1.3 PRODUCCIÓN

En la fase de "productización" se repite la programación de tres días (planificación, trabajo-liberación) se repite iterativamente hasta implementar todas las funcionalidades.

Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano (de ahí el nombre de esta técnica de *Test-Driven Development, TDD*). Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación.

2.4.1.4 ESTABILIZACIÓN

En la fase de estabilización, se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyecto multi-equipo con diferentes subsistemas desarrollados por equipos distintos. En esta fase, los desarrolladores realizarán tareas similares a las que debían desarrollar en la fase de "productización", aunque en este caso todo el esfuerzo se

dirige a la integración del sistema. Adicionalmente se puede considerar en esta fase la producción de documentación.

2.4.1.5 PRUEBAS DEL SISTEMA

La última fase (prueba y reparación del sistema) tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados.

2.4.2 ELEMENTOS

Existen elementos principales involucrados en las diferentes prácticas en el transcurso del ciclo de desarrollo.

- Ajuste y enfoque de fases: los proyectos se llevan a cabo en iteraciones donde cada una comienza con un día de planificación.
- Línea de arquitectura: éste enfoque es utilizado junto con los patrones de arquitectura y modelado ágil.
- Desarrollo basado en pruebas: el enfoque de pruebas – primero es utilizado junto con casos de prueba automatizadas.
- Integración continua: las prácticas de Software Configuration Manager (SCM) se aplican a través de múltiples medios.
- Programación en pares: la codificación, pruebas y refactorización se lleva a cabo en pares.
- Métricas: pocas métricas se recogen con rigurosidad y se utilizan con fines de mejorar la retroalimentación y el proceso de desarrollo.
- Mejoras en el proceso de software ágil: talleres de post – iteración son utilizados para mejorar continuamente el proceso de desarrollo.
- Cliente externo: el cliente participa en las jornadas de planificación y liberación.
- Enfoque centrado en el usuario: se hace hincapié en la identificación y el cumplimiento de necesidades del usuario final.

Estos elementos son prácticas ya establecidas en metodologías ágiles, con la inclusión de la línea de arquitectura, que se usa para capturar el conocimiento de una organización de soluciones arquitectónicas, tanto de fuentes internas y externas, y usar estas soluciones cuando sea necesario.

2.4.3 ENFOQUE PRAGMÁTICO

Los autores de esta metodología dicen haberlo probado obteniendo una certificación CMMI de nivel 2. Esto parece ser una ventaja comparativa importante frente a otras metodologías, puesto que la contratación de empresas para la externalización del desarrollo de software se rige por la auditoría de los ciclos y técnicas de desarrollo que utilizan (y CMMI es una de las métricas de aseguramiento de calidad más aceptadas en el sector).

Adicionalmente, sus creadores han introducido Mobile-D en numerosos proyectos de desarrollo con clientes reales. La base inicial de 4 casos de estudio se ha desarrollado durante años y afirman sus autores que los ciclos de desarrollo se han actualizado y mejorado a partir de la experiencia obtenida. (Spataru, 2010)

2.5 GOOGLE MAP MAKER

Google Map Maker es un servicio que permite enriquecer o modificar las informaciones y los datos disponibles en Google Maps, al ser un servicio colaborativo da la oportunidad a cada internauta a realizar cuatro tipos posibles de contribución vía Map Maker:

- **Añadir lugares:** Proveer información suplementaria sobre las zonas que le interesan, así como colocar un comercio o establecimiento en el mapa.
- **Modificar la descripción de un lugar para los lugares ya cartografiados:** Esta funcionalidad permite abastecer elementos más precisos de descripción, marcar los límites y los contornos de un edificio por ejemplo.
- **Trazar caminos:** Permite especificar caminos

- **Examinar y comentar las modificaciones aportadas por otros internautas:**
Aunque las contribuciones están abiertas a todos, éstas son controladas y corregidas a priori por otros usuarios, expertos regionales así como un equipo de correctores de Google, una vez validadas, son publicadas y visibles en los servicios de Google Maps.



Figura 3: Características de Google Maps Maker

Fuente: Google Maps Maker

Google Maps Maker ha sido la base que ha permitido construir ciudades enteras dentro de Google Maps, gracias al trabajo colaborativo de internautas.

2.6 API DE GOOGLE MAPS

Google Maps provee a los desarrolladores un API capaz de aprovechar los datos disponibles a través del servicio, en el seno de las propias aplicaciones, permitiendo a los desarrolladores programar dentro de mapas basándose en un conjunto de librerías y servicios proporcionadas por la API de Google Maps como se menciona a continuación.

2.6.1 LIBRERÍA DE AUTOCOMPLETADO DE LUGARES

La función de autocompletado de lugares se vincula a un campo de texto de la página web y controla los caracteres que se introducen en ese campo. Al introducir texto, la

función de autocompletado envía predicciones de lugares a la aplicación en forma de una lista desplegable de elementos que se pueden seleccionar.

Cuando un usuario selecciona un lugar de la lista, el objeto de autocompletado recibe información sobre el lugar seleccionado, como por ejemplo coordenadas, nombre de ubicación etc.

2.6.2 LIBRERÍA DE DIBUJO

La API de Google Maps proporciona un conjunto de clases que permite el dibujo de figuras geométricas como se presentan a continuación:

2.6.2.1 CÍRCULOS

Google Maps incluye la clase circle que permite trazar círculos en los cuales se puede definir colores, grosores y niveles de opacidad personalizados para el borde del círculo (el "trazo"), así como colores y opacidades personalizados para el área que engloba (el "relleno").

2.6.2.2 POLILINEAS

Google Maps incluye la clase polyline que permite trazar caminos, como la creación de rutas de ciclismo, etc.

Las polilíneas se componen de varias líneas conectadas. Una línea consiste en dos puntos: un punto de partida y un punto de terminal. Estos puntos se componen de coordenadas.

2.6.3 SERVICIO DE RUTAS

El servicio de rutas de Google Maps proporciona la clase directions Service, que despliega una ruta completamente modificable, a la cual se le debe especificar un origen y destino mediante coordenadas geográficas, las coordenadas geográficas se deben representar en base a pares ordenados de latitud y longitud, además esta ruta se adapta automáticamente a calles y avenidas permitiendo describir rutas dentro de Google Maps.

2.7 PHONEGAP

PhoneGap es un framework para el desarrollo de aplicaciones móviles producido por Nitobi, y comprado posteriormente por Adobe Systems. Principalmente, PhoneGap permite a los programadores desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3 como se ilustra a continuación en la figura 4.



Figura 4: Desarrollo de aplicaciones basados en phonegap

Fuente: www.pixelovers.com

PhoneGap permite el desarrollo ya sea ejecutando las aplicaciones en nuestro navegador web, sin tener que utilizar un simulador dedicado a esta tarea, y brinda la posibilidad de soportar funciones sobre frameworks como JQuery Mobile, además nos permite acceder a las funcionalidades nativas de los dispositivos móviles utilizando JavaScript. Así, podemos desarrollar toda la lógica de nuestra aplicación en JavaScript y utilizar la API de PhoneGap para acceder a las funcionalidades nativas del dispositivo como se muestra en la figura5.

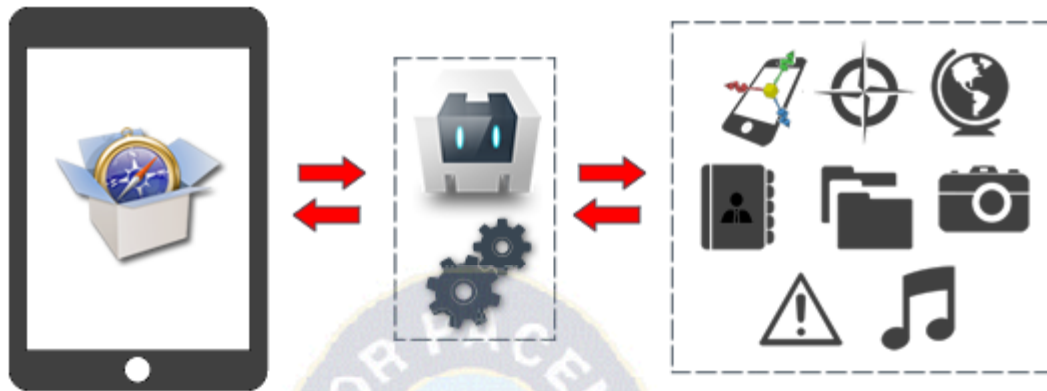


Figura 5: Características de phonegap

Fuente: www.pixelovers.com

Para poder compilar el código generado al código nativo para las diferentes plataformas móviles, PhoneGap nos proporciona el servicio PhoneGap Build que permite subir nuestros archivos con código a un servidor que se encarga de compilar nuestro código al código nativo para las diferentes plataformas móviles.

2.8 ANDROID

Android es una solución completa de software de código libre para teléfonos y dispositivos móviles. Es un paquete que engloban un sistema operativo, un "runtime" de ejecución basado en Java, un conjunto de librerías de bajo y medio nivel y un conjunto inicial de aplicaciones destinadas al usuario final (todas ellas desarrolladas en Java). Android se distribuye bajo una licencia libre permisiva (Apache) que permite la integración con soluciones de código propietario.

Android surge como resultado de la Open Handset Alliance un consorcio de 48 empresas distribuidas por todo el mundo con intereses diversos en la telefonía móvil y un compromiso de comercializar dispositivos móviles con este sistema operativo. El desarrollo viene avalado principalmente por Google (tras la compra de Android Inc. en 2005) y entre las compañías encontramos compañías de software (Ebay, LivingImage...) operadores (Telefónica, Vodafone, T-Mobile...), fabricantes de móviles

(Motorola, Samsung, acer, LG, HTC...) o fabricantes de Hardware (nVidia, Intel o Texas Instruments).(Spataru, 2010)

2.8.1 ARQUITECTURA

Android presenta una arquitectura basada en 4 niveles (figura 10), que detallamos a continuación por orden ascendente:

- **Un kernel Linux** versión 2.6 que sirve como base de la pila de software y se encarga de las funciones más básicas del sistema: gestión de drivers, seguridad, comunicaciones, etc.
- **Una capa de bibliotecas de bajo nivel** en C y C++, como SQLite para persistencia de datos; Open GL ES para gestión de gráficos 3D, con aceleración 3D opcional y Webkit como navegador web embebido y motor de renderizado HTML.
- **Un framework para el desarrollo de aplicaciones**, dividido en subsistemas para gestión del sistema como el "Administrador de paquetes", el "Administrador de telefonía" (para la gestión del hardware del teléfono anfitrión) o el acceso a APIs sofisticadas de geolocalización o mensajería XMPP. Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar el reuso de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Éste mismo mecanismo permite que los componentes sean reemplazados por el usuario. También incluye un sistema de vistas para manejar el interfaz de usuario de las aplicaciones, que incluyendo posibilidad de visualización de mapas o renderizado html directamente en el interfaz gráfico de la aplicación.

- **Aplicaciones:** Las aplicaciones base incluyen un teléfono, cliente de email, programa de envío de SMS, calendario, mapas, navegador, contactos... que pueden a su vez ser usados por otras aplicaciones.

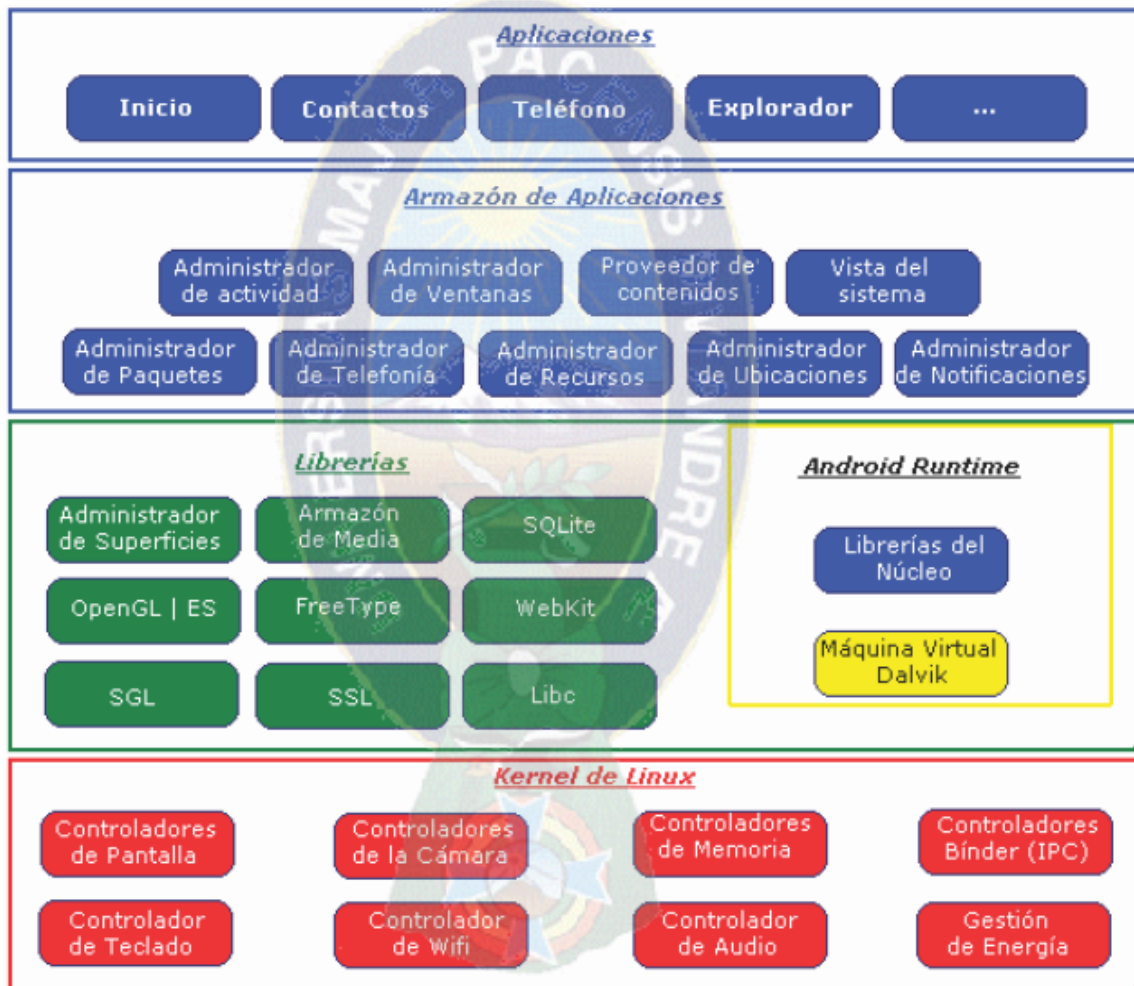


Figura 6: Arquitectura de Android

Fuente: www.wikipedia.org

2.9 EMULADOR RIPPLE

Ripple es un plugin de iOS, Android y Blackberry para Google Chrome que recrea el entorno y sensores de un dispositivo móvil real dentro del navegador web. Tiene un sistema avanzado de simulación para probar aplicaciones basadas en PhoneGap y Webworks de Blackberry.

Con este plugin y el soporte HTML5 de Google Chrome es posible simular, con bastante aproximación a la realidad, prácticamente cualquier aplicación basada en tecnologías web, incluso si utiliza eventos o sensores exclusivos del móvil como uso de botones o movimientos del acelerómetro.

2.9.1 CAPACIDADES DE RIPPLE

A pesar que Chrome comparte el mismo motor de render (Webkit) que iOS, Android y Blackberry y tiene la posibilidad de mostrar elementos gráficos prácticamente en las mismas condiciones que un equipo móvil, tiene algunas restricciones. Los navegadores carecen de algunos eventos y sensores exclusivos de los dispositivos móviles.

Ripple se encarga de simular estas capacidades para acercarse al máximo posible al verdadero entorno móvil algunas de las capacidades que instala en el navegador son:

Devices: Simula la apariencia, tamaño y resolución de pantalla de múltiples equipos y sistemas operativos móviles. También permite cambiar la orientación del equipo en horizontal ó vertical.

Platforms: Muestra las plataformas y versiones disponibles para emulación: PhoneGap (Apache Cordova), Blackberry Webworks y web móvil.

Information: Despliega información importante sobre el documento y modo de emulación actual.

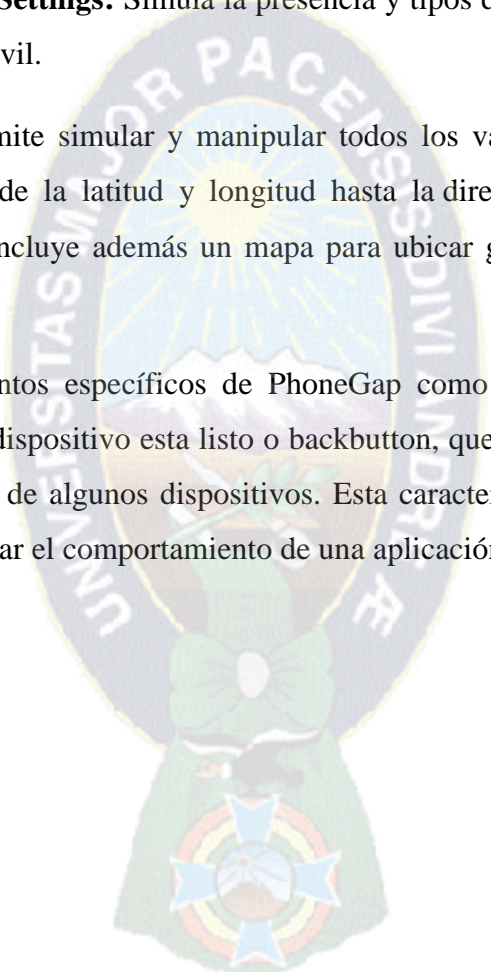
Accelerometer: Muestra un dispositivo virtual en 3D que se puede rotar libremente sobre todos sus ejes. Al manipular el dispositivo virtual se envía directamente la

información a la aplicación para que reaccione de igual manera, simulando así el movimiento real de un dispositivo. La información detallada se muestra en tiempo real para depurar y comprobar los resultados. Incluye también la opción de simular la acción de “sacudida” o “shake”.

Device & Network Settings: Simula la presencia y tipos de conexión a internet propios de un dispositivo móvil.

Geo Location: Permite simular y manipular todos los valores de las coordenadas de geolocalización, desde la latitud y longitud hasta la dirección y velocidad en que se desplaza el móvil. Incluye además un mapa para ubicar gráficamente las coordenadas que se introducen.

Events: Activa eventos específicos de PhoneGap como deviceReady, que señala el momento en que el dispositivo está listo o backButton, que se activa cuando se presiona el botón de regresar de algunos dispositivos. Esta característica es particularmente útil porque permite emular el comportamiento de una aplicación PhoneGap en un dispositivo móvil real.



CAPÍTULO III

3 DISEÑO METODOLÓGICO

3.1 INTRODUCCIÓN

En este apartado se desarrollará el marco estratégico constituido por los métodos, técnicas (procedimientos), e instrumentos que se emplearán en la ejecución del proyecto de investigación para poner a prueba la hipótesis, alcanzar los objetivos de investigación y así dar una respuesta al problema de investigación.

3.2 SOFTWARE MÓVIL HÍBRIDO CON PHONEGAP

En la mayor parte de las ocasiones relacionamos desarrollos móviles con aplicaciones nativas. Este tipo de desarrollos son desarrollos hechos de forma específica para un determinado sistema operativo.

Pero también encontramos las aplicaciones Web móviles. Que no son más que aplicaciones que corren sobre un navegador web. Pero en este caso, sobre el navegador web de nuestro dispositivo móvil.

También existen lo que llamamos aplicaciones híbridas. Este tipo de aplicaciones son una combinación de ambas: Nativas + Web. Y todo apunta a que este tipo de aplicaciones móviles tienden a ser las más utilizadas.

Phonegap es una plataforma híbrida la cual nos permite el desarrollo de aplicaciones móviles usando tecnologías web (HTML5, JavaScript, CSS). Es de Código Abierto (Open source), está basado en estándares W3C Mobile y cuenta con un API JavaScript para acceder a las características de los dispositivos.

3.2.1 PROCESO DE DESARROLLO

El desarrollo de esta aplicación híbrida permitirá el uso de tecnologías multiplataforma como HTML5, JavaScript y CSS, también permite acceder a una buena parte de los dispositivos y sensores del teléfono.

Un buen ejemplo de aplicaciones híbridas es Facebook. Se descarga de la App Store y cuenta con todas las características de una aplicación nativa pero requiere ser actualizada ocasionalmente.

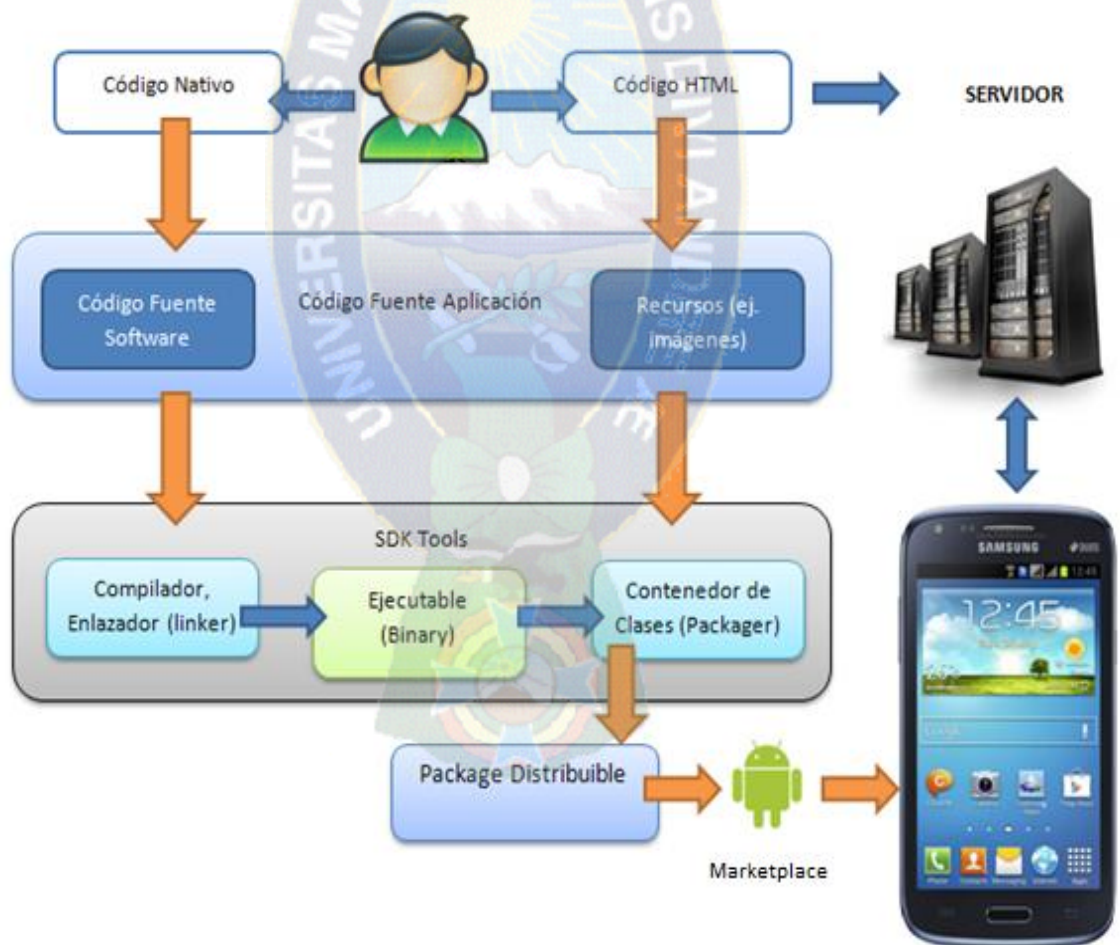


Figura 7: Arquitectura de desarrollo de una aplicación híbrida

Fuente: Elaboración Propia

Phonegap tiene dos objetivos; Primero, permite que un código fuente cualquiera se pueda ejecutar en diversas plataformas. Segundo, Phonegap permite que la aplicación web acceda a los diferentes elementos del teléfono.

3.2.1.1 CODIGO NATIVO

Una aplicación móvil implementada en código nativo es mucho más rápida que las aplicaciones implementadas en PhoneGap. Por otro lado, la funcionalidad disponible para PhoneGap es más limitada que la que proporciona el sistema operativo nativo, por tanto la aplicación desarrollada no contará con muchas de las características de la plataforma nativa. Lo mismo implica que la aplicación puede no adaptarse correctamente a las actualizaciones que se realicen en el sistema operativo nativo

3.2.1.2 CODIGO HTML

El código HTML utilizado para el desarrollo de la aplicación móvil es acompañado por código en JavaScript, para que de esta manera, mediante PhoneGap pueda ajustarse de forma adecuada a la plataforma nativa. Mostrando de esta manera compatibilidad no solo con Android sino con otros sistemas operativos para dispositivos móviles.

3.2.1.3 CODIGO FUENTE APLICACIÓN

El código fuente que se trabajó para el desarrollo del software móvil es único, generándose un programa compatible con todos los posibles sistemas operativos soportados por el entorno de desarrollo.

3.2.1.4 SDK TOOLS

SDK Tools viene siendo el kit de desarrollo de software con el que se procedió a trabajar en el desarrollo del software móvil, permitiéndonos ejecutar un emulador del sistema Android de la versión que sea. Todas las aplicaciones Android se desarrollan en lenguaje Java con este kit.

3.2.1.5 COMPILADOR

PhoneGap Build viene siendo el compilador y complemento perfecto para todo lo que significa PhoneGap también conocido como Apache Córdova. Lo que primero debemos hacer es crear una cuenta, luego PhoneGap Build compila nuestra aplicación híbrida para seis sistemas operativos móviles como son: iOS, Android, BlackBerry OS, Symbian, Web OS y Windows Phone, integrado hace muy poco.

3.2.1.6 PACKAGE

El package aplicado al desarrollo del software móvil es una agrupación de clases afines a Google Maps. Equivale al concepto de librería de Google Maps existente. Una clase puede definirse como perteneciente a un package y puede usar otras clases definidas en ese o en otros packages.

3.2.1.7 PACKAGE DISTRIBUIBLE

El archivo generado por PhoneGap a partir del código fuente, es una aplicación con extensión .apk (*Application PacKage File*) es un paquete para el sistema operativo Android. Este formato es una variante del formato JAR de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android para Smartphones y tablets.

3.2.1.8 MARKETPLACE

Es un sitio a través del cual miles de desarrolladores del mundo entero ofrecen sus productos y millones de usuarios pueden descargar aplicaciones gratuitas o de pago.

3.3 SERVICIOS BASADOS EN LOCALIZACIÓN

Durante el desarrollo de nuestro software móvil nos vimos a la tarea de identificar los servicios basados de geolocalización. Que son aplicaciones que ofrecen un servicio personalizado en tiempo real al usuario, basándose en la localización geográfica de un dispositivo.

Las operaciones básicas que permiten realizar los LBS (Location Based Services) con la información de localización son:

- **Localizar:** determinar la posición a la que el usuario haga referencia.
- **Buscar:** determinar la posición de sucursales y cajeros automáticos.
- **Navegar:** determinar la ruta óptima desde la localización del usuario a punto asignado por el usuario
- **Consultar:** explorar las características de un lugar o una organización
- **Encontrar:** buscar, o explorar servicios, eventos o personas cercanos a la localización del usuario.
- **Visualizar en un mapa:** ver los resultados de la búsqueda y la propia localización en un mapa, interactuar y navegar en él.

3.3.1 TIPOS DE ENTIDADES

3.3.1.1 DATOS VECTORIALES PUNTUALES

Sucursales de Entidades Financieras y Cajeros Automáticos.

3.3.1.2 DATOS RASTER

Imágenes o fotografías aéreas que se sitúan sobre el territorio.

3.3.2 MAPAS DIGITALES

Servicios basados en APIs o integrados en un GIS que nos permiten crear mapas digitales o aplicaciones relativamente complejas basadas en mapas.

3.4 LA GEOLOCALIZACIÓN COMO PROCESO DE UN SOFTWARE MÓVIL HÍBRIDO

La geolocalización como base de un software móvil híbrido tiene la capacidad de almacenar, procesar, presentar y manipular la información obtenida referente a los geo datos; ofreciendo datos precisos a usuarios específicos, en nuestro caso a clientes de las distintas entidades bancarias que prestan sus servicios en la urbe paceña.

A continuación mostramos un esquema general de uso de este tipo de proceso:



Figura 8: Esquema General de la Geolocalización como proceso de un software móvil híbrido

Fuente: Elaboración propia

3.5 DESARROLLO DEL PROTOTIPO

En los siguientes acápite se presenta el desarrollo del software móvil de geolocalización referente a los servicios de las entidades financieras en la ciudad de La Paz con cada una de las fases empleando la metodología Mobile-D.

3.5.1 EXPLORACIÓN

Esta fase permite establecer las acciones iniciales para identificar los requerimientos funcionales y no funcionales que se debe cumplir para lograr el éxito del proyecto. Se define el planteamiento y establecimiento del proyecto para sentar las bases para la implementación controlada del producto en relación con el desarrollo del software y lograr el éxito en las siguientes fases del proyecto, preparando y verificando todos los posibles problemas críticos de desarrollo y sean mitigados en el tiempo.

3.5.1.1 DEFINICIÓN DE LOS GRUPOS DE INTERES

- Clientes de entidades financieras que realizan sus diversos movimientos financieros dentro de la ciudad de La Paz, ya que necesitan contar con información al alcance de la mano sobre la ubicación de algún servicio requerido de una entidad financiera en particular y la ruta más corta hacia ese servicio.
- Personas que no conocen la topografía, ni las calles de la ciudad de La Paz, y necesitan información sobre su ubicación actual y la ubicación de sucursales o cajeros automáticos de determinada entidad financiera de la cual es cliente.

3.5.1.2 ACTORES

Los actores del sistema involucrados con la construcción del sistema de información de geolocalización de servicios financieros son principalmente el usuario y el administrador.

3.5.1.3 ROLES Y TAREAS

Para el desarrollo del sistema se han logrado identificar a los siguientes usuarios:

NOMBRE	DESCRIPCIÓN	STAKEHOLDER
Administrador	Es el responsable de guardar y mantener de forma actualizada las ubicaciones de los distintos servicios financieros que serán publicadas para que los usuarios finales puedan utilizarlos.	Administrador
Usuario final	Es el usuario que requiere los servicios de información.	Usuario

Tabla 1: Roles y tareas de los usuarios

Fuente: Elaboración propia

3.5.1.4 HISTORIAS DE USUARIO

Siguiendo con la fase de exploración, es necesario realizar una recopilación e identificación clara de los requerimientos funcionales con los que se deben cumplir al finalizar el desarrollo del sistema software y así tener una visión general de lo que se desea crear.

Para esta actividad se realizaron entrevistas dirigidas a los usuarios generando así las siguientes historias de usuario (ver tablas 2 - 5).

ADMINISTRADOR

Historia de usuario		1/Febrero/2014
Historia de usuario N° 001		
Nombre: Información		Prioridad: Alta
Descripción	<p>Deseo mostrar las ubicaciones de los cajeros o sucursales de las distintas entidades financieras ubicadas en la ciudad de La Paz, según información actualizada, recolectada de los sitios web de las mismas.</p> <p>Esta información se debe mostrar de manera rápida utilizando un mapa georreferenciado que muestre las ubicaciones de estas entidades financieras y de sus servicios (cajeros automáticos o sucursales).</p>	
Estimación	Se estima desarrollar la aplicación en un periodo no mayor a dos meses.	
Prueba de aceptación		
	El sistema debe mostrar de forma gráfica usando un mapa georreferenciado la ruta mínima de un punto A y un punto B, que hacen referencia a la ubicación del usuario y la ubicación del servicio financiero	

	elegido.
--	----------

Tabla 2: Historia de usuario

Fuente: Elaboración propia

ADMINISTRADOR

Historia de usuario		1/Febrero/2014
Historia de usuario N° 002: Administración de Ubicaciones		
Nombre: Información		Prioridad: Alta
Descripción	El constante cambio sobre la demanda de servicios financieros por parte de usuarios dentro de la ciudad, las mismas entidades financieras van ampliando sus puntos de servicio, así como la reubicación de los mismos, para el mejor servicio al usuario, es por eso que la base de datos debe estar en constante actualización y es necesario la administración de la misma.	
Estimación	Seis días de trabajo	
Prueba de aceptación		
Descripción	El sistema debe ofrecer al administrador la posibilidad de gestionar ubicaciones de cajeros automáticos y sucursales de las distintas entidades financieras.	

Tabla 3: Historia de usuario Actualización de Ubicaciones

Fuente: Elaboración propia

ADMINISTRADOR

Historia de usuario		1/Febrero/2014
Historia de usuario N° 003: Despliegue Ubicaciones de Entidades Financieras.		
Nombre: Consultas		Prioridad: Media
Descripción	Para lograr una mayor aceptación, es necesario disponer de medios gráficos basado en el Google Maps que permitan captar la atención y necesidad de los usuarios. Este proceso debe ser automáticamente realizado por el sistema utilizando funciones y librerías del Google Maps.	
Estimación	Cinco días de trabajo	
Prueba de aceptación		
Descripción	El sistema debe tener un sistema de administración de íconos, rutas y puntos fijos establecidos en la aplicación para permitir el manejo sencillo	

Tabla 4: Historia de usuario despliegue de ubicaciones de Entidades Financieras

Fuente: Elaboración propia

USUARIO

Historia de usuario		1/Febrero/2014
Historia de usuario N° 005		
Nombre: Control y consultas		Prioridad: Media
Descripción	Deseo que la aplicación me muestre una entidad financiera requerida, con	

	un servicio requerido más próximo a la ubicación que deseo.
Estimación	Cinco días para el desarrollo
Prueba de aceptación	
Descripción	El sistema debe tener un módulo de búsqueda de ubicaciones con una interface de manejo amigable.

Tabla 5: Historia de usuario

Fuente: Elaboración propia

3.5.1.5 COLECCIÓN DE REQUERIMIENTOS

Es una tarea en la cual los requerimientos para el producto son establecidos en nivel apropiado, los cuales pueden ser funcionales o no funcionales. El objetivo es producir la definición inicial general del producto, propósito y funcionalidad.

3.5.1.6 REQUERIMIENTOS

3.5.1.6.1 FUNCIONALES

- La aplicación debe ser capaz de mostrar todos los cajeros o sucursales de una determinada entidad financiera definida por el usuario que se encuentran dentro de un radio de distancia referente a la posición del usuario o al punto al cual el usuario desea referirse.
- La aplicación debe ser capaz clasificar tipos de servicio, ya sean estos cajeros o sucursales.
- La aplicación debe ser capaz de señalar la entidad financiera requerida por el usuario.

- El módulo de administrador deberá permitir gestionar: adicionar, eliminar y modificar las diferentes ubicaciones de los servicios financieros dentro de la base de datos.
- El módulo administrador deberá permitir administrar: adicionar, eliminar y modificar otros usuarios que tengan acceso al control de la base de datos.
- La base de datos deberá contar con un mínimo representativo de ubicaciones de servicios financieros registrados en función de las entidades financieras que brindan sus servicios dentro del centro de la ciudad.
- La aplicación desarrollada debe ajustarse de manera automática a diferentes dispositivos móviles inteligentes.

3.5.1.6.2 NO FUNCIONALES

- El administrador del sistema deberá ser autenticado mediante algoritmos de seguridad.
- El módulo administrador desplegará mensajes de confirmación acorde a las acciones de adición, eliminación o modificación de las rutas dentro almacenadas en la base de datos.
- El administrador deberá validar todos los campos antes de adicionar, eliminar o modificar información de la ubicación de un servicio financiero dentro de la base de datos mediante el uso de funciones georreferenciadas de uso libre.
- La base de datos que almacena las rutas debe estar en permanente actualización.

3.5.1.7 ARQUITECTURA

Mediante la identificación de identificación de los requerimientos funcionales, se establecieron dos subsistemas que fueron implementados, para cumplir con los objetivos del presente proyecto.

Su arquitectura se muestra a continuación en la Figura 10.

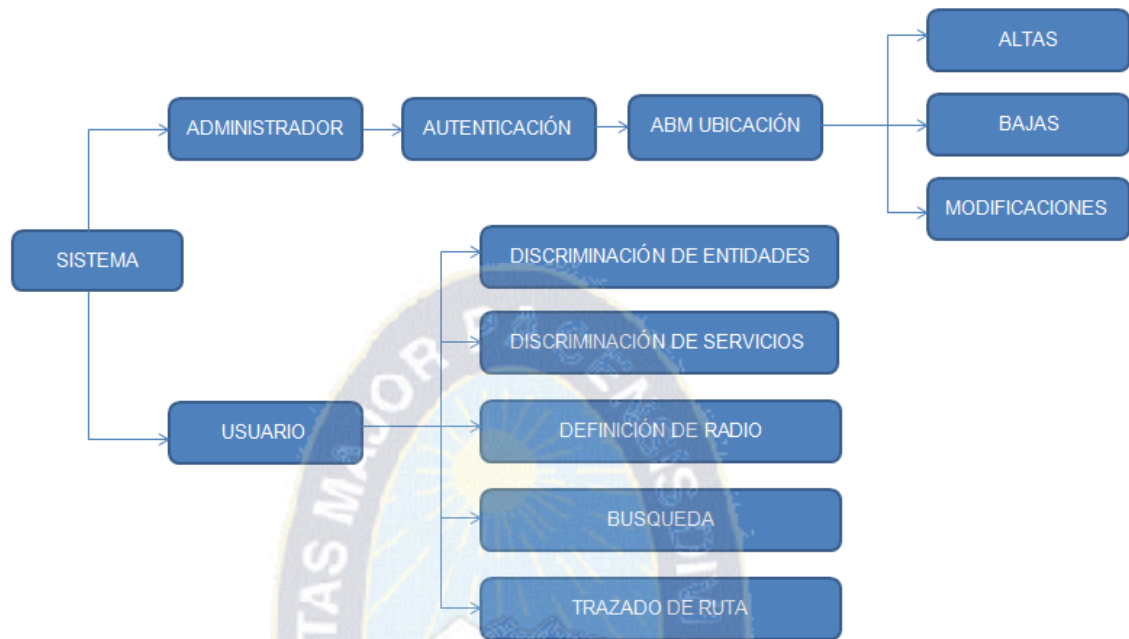


Figura 9: Arquitectura del prototipo

Fuente: Elaboración propia

Básicamente el prototipo presenta dos módulos plenamente identificados: Administración, destinado a la gestión de los datos del sistema y Usuarios, orientada al uso de la aplicación con interfaces gráficas.

- **Administración:** Esta opción del sistema permite la administración de las ubicaciones de los servicios (sucursales o cajeros automáticos) de las distintas entidades financieras. Esta opción está habilitada solamente para aquellos usuarios autorizados que conozcan la contraseña de acceso al sistema.
- **Usuario:** Es la parte del sistema destinado a los usuarios, quienes a partir de la interfaz gráfica que tiene la aplicación, éste puede realizar búsquedas de entidades financieras y tipos de servicio (sucursales o cajeros automáticos), puede consultar por un determinado punto geográfico concreto (una ubicación dentro de la ciudad), o desde su ubicación actual, ajustando la distancia mínima

con la cual se desean realizar las búsquedas mediante la asignación de un radio de búsqueda respecto a la ubicación requerida.

Adicionalmente este módulo contiene algoritmos de búsqueda basados en técnicas matemáticas, como lo es la geometría. Así como el módulo de trazado de ruta que utiliza la fórmula de Harvesine para la aproximación esférica de la distancia entre dos puntos de la superficie terrestre.

3.5.2 INICIALIZACIÓN

3.5.2.1 LISTA DE TAREAS

Para llevar con éxito el desarrollo del prototipo, se definió una serie de tareas que fueron desarrolladas cronológicamente en base a prioridades establecidas, como se muestra a continuación en la tabla 6.

TAREA	PRIORIDAD
Historias de usuario	1
Diseño de interfaces de usuario	2
Diseño de la base de datos	3
Diagrama de clases	4
Gestión de usuarios	5
Gestión de puntos de servicio	6
Establecimiento parámetros de búsqueda	7
Búsqueda de puntos de servicio	8
Discriminación de puntos de servicio	9
Trazado de ruta	10

Pruebas de aceptación	11
-----------------------	----

Tabla 6: Lista de tareas
Fuente: Elaboración propia

3.5.2.2 PLAN DE ENTREGAS

A continuación se muestra el plan de entregas definido en la figura 12.

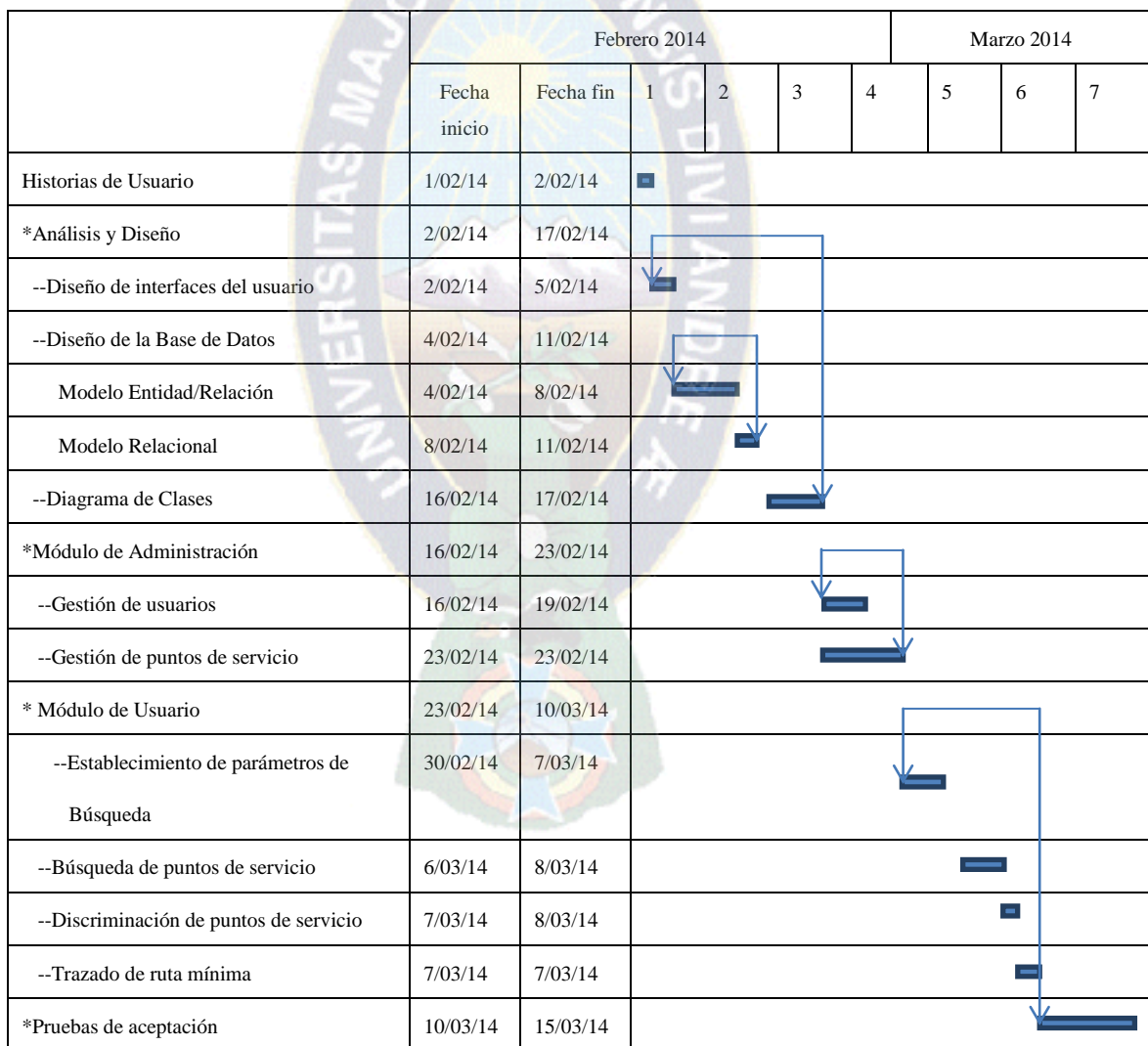


Figura 10: Cronograma de Trabajo

Fuente: Elaboración propia

3.5.2.3 PLANIFICACIÓN DE FASES

Para el desarrollo de la aplicación, se planifica realizar por lo menos tres iteraciones ejecutadas con un periodo de dos semanas y media cada una según los criterios de prioridad establecidas para cada una de las tareas que se definieron en la tabla 6.



Figura 11: Esquema de desarrollo del sistema

Fuente: elaboración propia

3.5.2.4 ITERACIONES

FASE	ITERACIÓN	DESCRIPCIÓN
Exploración		
Inicialización	Iteración 0	Establecimiento del proyecto, desarrollo de tareas basadas por prioridades.
Producción	Iteración 1	Implementación de módulos. Refinamiento y actualización de historias de usuario. Refinamiento de interfaces. Generación y ejecución de pruebas de aceptación.
Estabilización	Iteración 2	Refactorización de módulos. Refinamiento de interfaces. Generación y ejecución de pruebas de aceptación.

Pruebas del sistema	Iteración pruebas del sistema	Se realiza la evaluación de las pruebas y se realiza el análisis de los resultados.
---------------------	-------------------------------	---

Tabla 7: Descripción de Iteraciones

Fuente: Elaboración propia

Del plan de Entregas, las tareas 1, 2, 3 y 4 deben ser elaboradas en la primera iteración con una planificación de elaboración de unas 2 semanas.

Las tareas 5, 6 serán realizadas en la iteración 2. La Planificación de entregas de los productos es de aproximadamente 2 semanas, las actividades se realizan de forma paralela.

Finalmente las tareas 7, 8, 9,10, y 11 se realizaran en la tercera iteración de forma paralela secuencial con duración concreta de 2 semanas.

La Tabla 8, muestra la estimación de esfuerzo de tareas por cada una de las actividades principales.

TAREA	DIAS
Historias de usuario	1
Diseño de interfaces de usuario	2
Diseño de la base de datos	4
Diagrama de clases	4
Gestión de usuarios	3
Gestión de puntos de servicio	5
Establecimiento parámetros de búsqueda	5
Búsqueda de puntos de servicio	5
Discriminación de puntos de servicio	1

Trazado de ruta	1
Pruebas de aceptación	5

Tabla 8: Estimación de esfuerzo Procesos principales

Fuente: Elaboración propia

3.5.2.5 DIAGRAMA DE CLASES

3.5.2.6 INTERFAZ CON API DE LOCALIZACIÓN DE GOOGLE MAPS

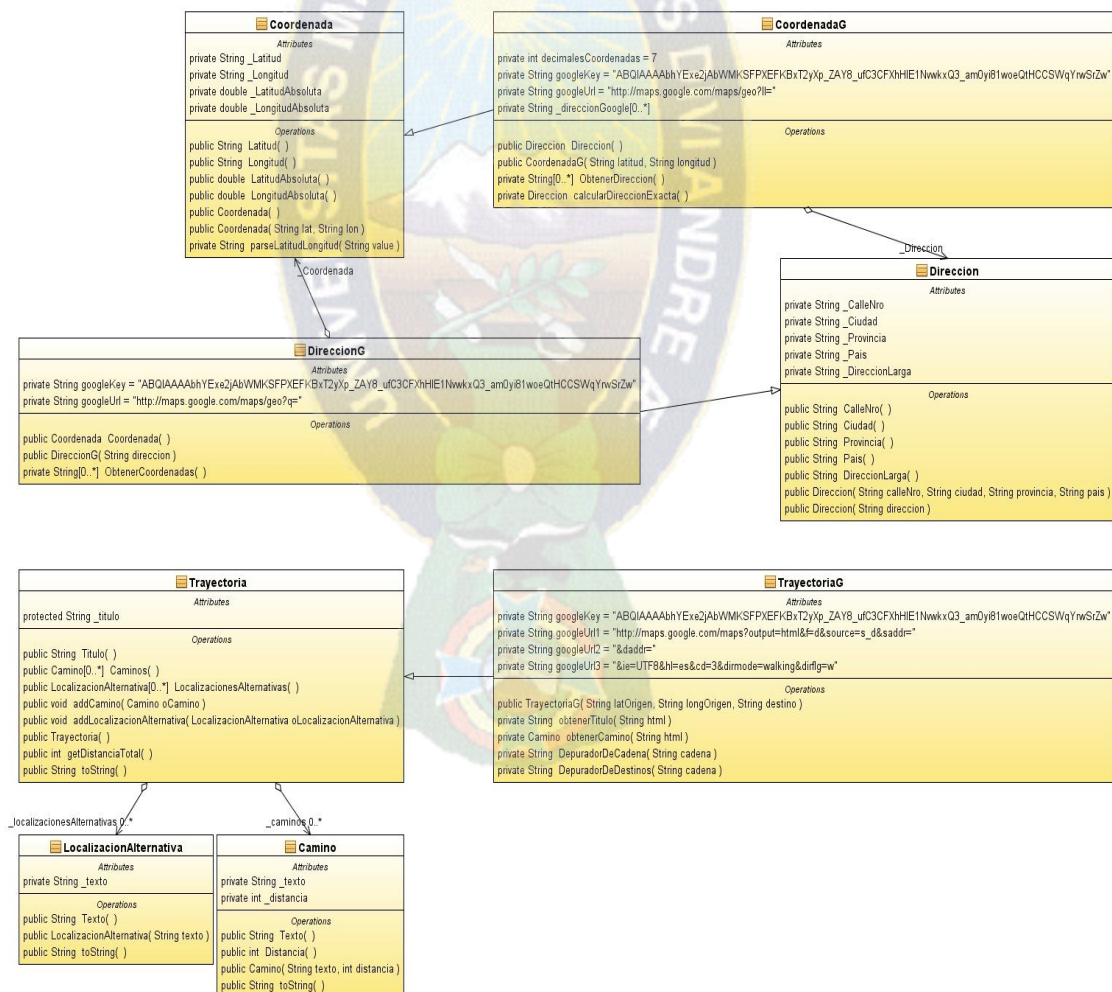


Figura 12: Interfaz con API de localización de Google Maps

Fuente: developers.google.com

3.5.2.7 DIAGRAMA DE ACTIVIDADES

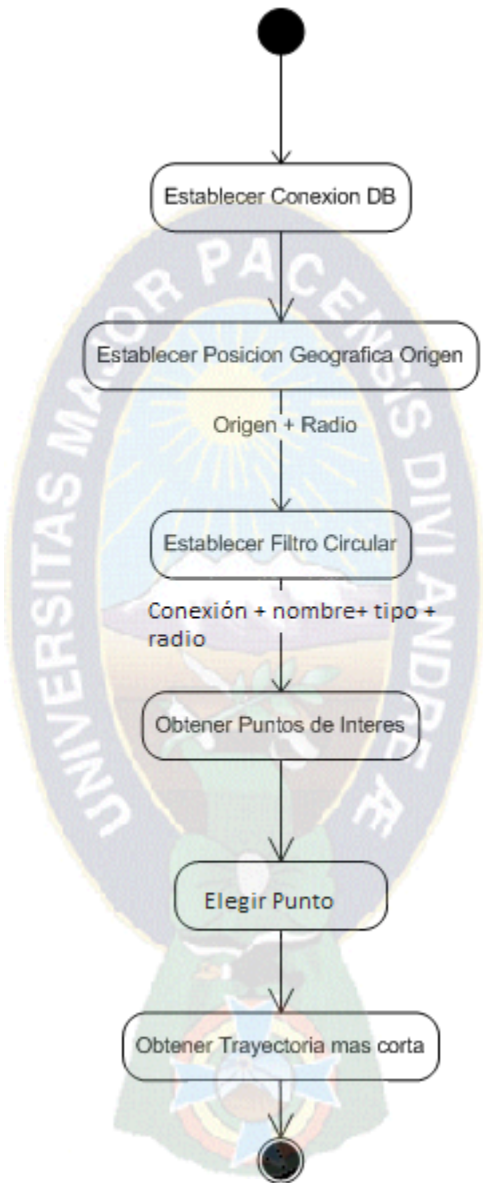


Figura 13: Diagrama de actividades

Fuente: elaboración propia

3.5.2.8 DISEÑO DE LA INTERFAZ DEL ADMINISTRADOR

A continuación se muestra el diseño de interfaces para el administrador, que permite realizar la gestión de las ubicaciones de los puntos de servicio de las entidades financieras.

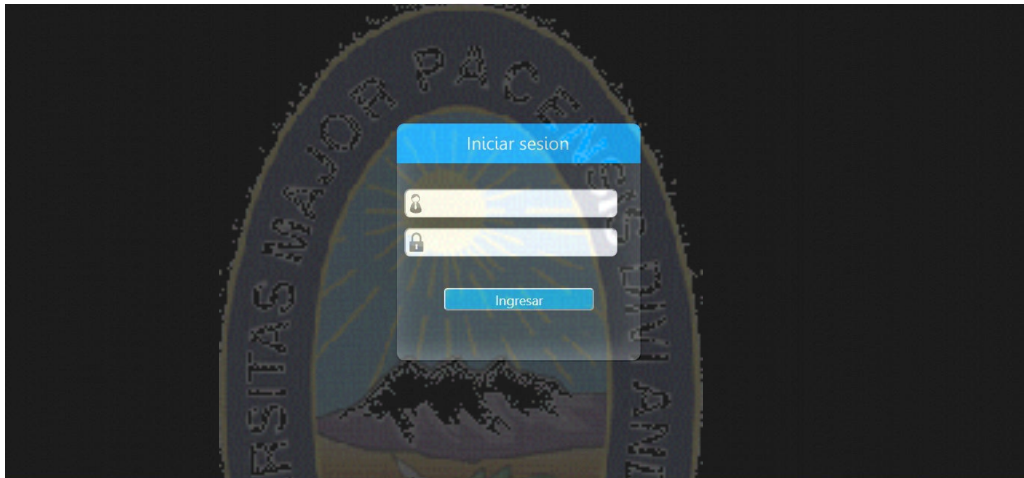


Figura 14: Inicio de sesión

Fuente: Elaboración propia

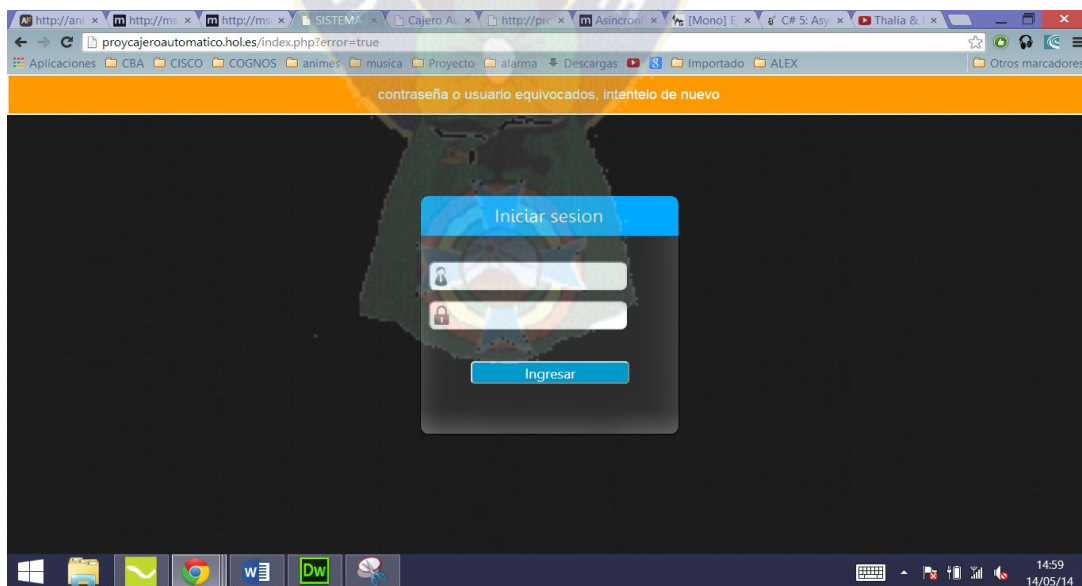


Figura 15: Ingreso de datos erróneos

Fuente: Elaboración propia

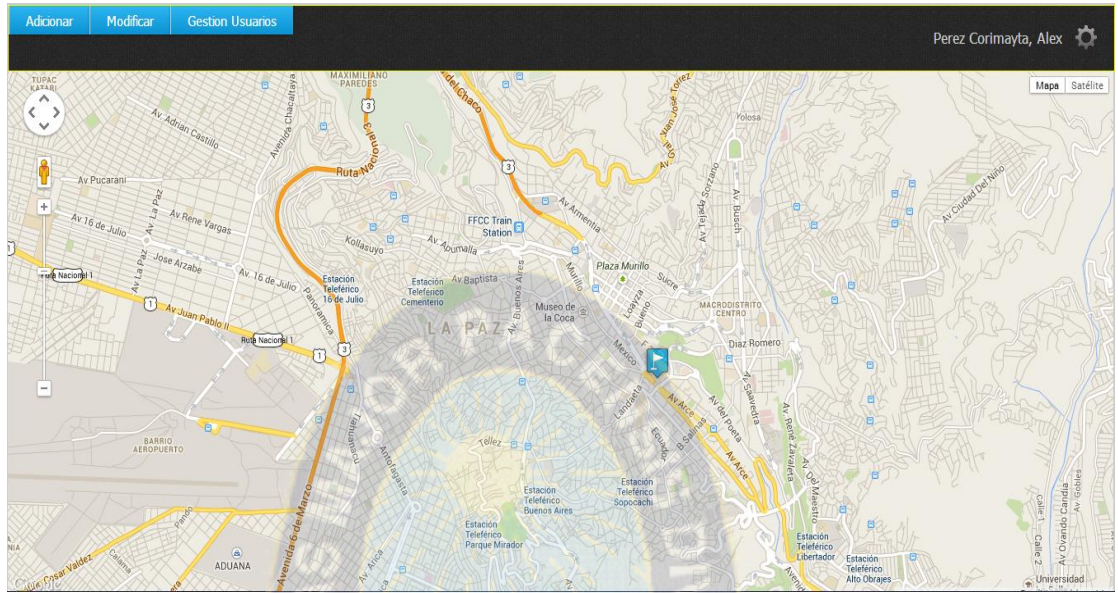


Figura 16: Pantalla de inicio
Fuente: Elaboración propia

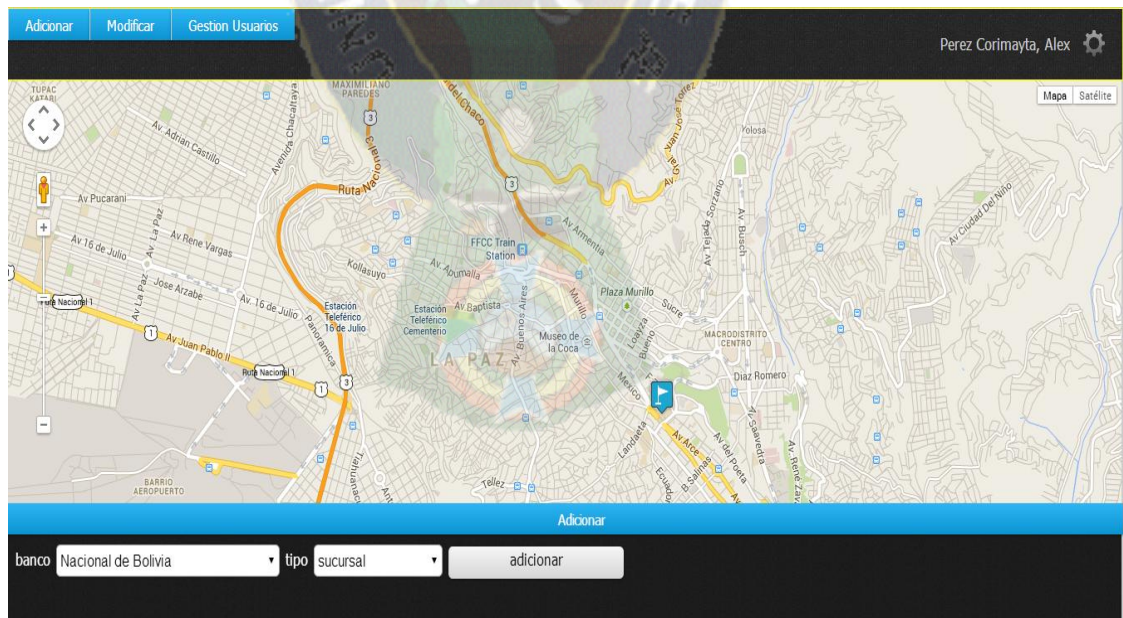


Figura 17: Adicionar Sucursal
Fuente: Elaboración propia

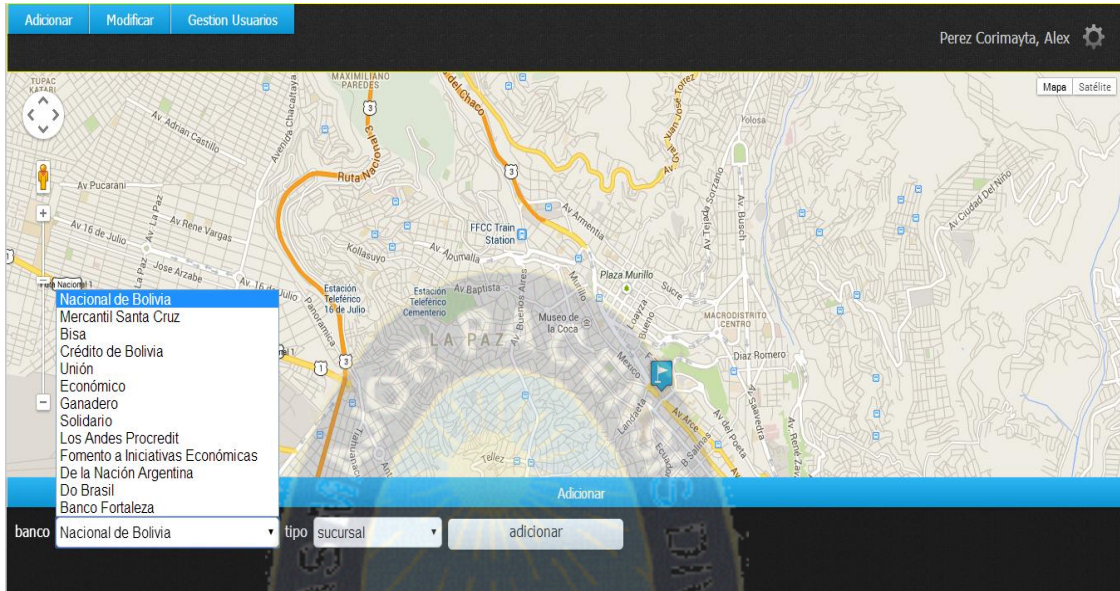


Figura 18: Seleccionar Entidad Financiera

Fuente: Elaboración propia

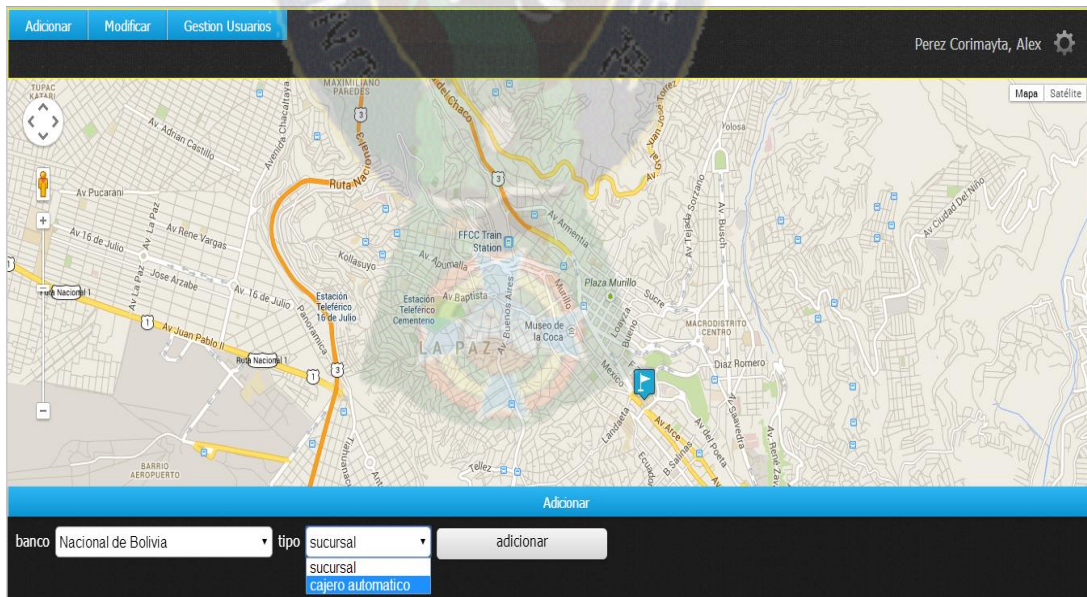


Figura 19: Seleccionar Tipo de Servicio

Fuente: Elaboración propia

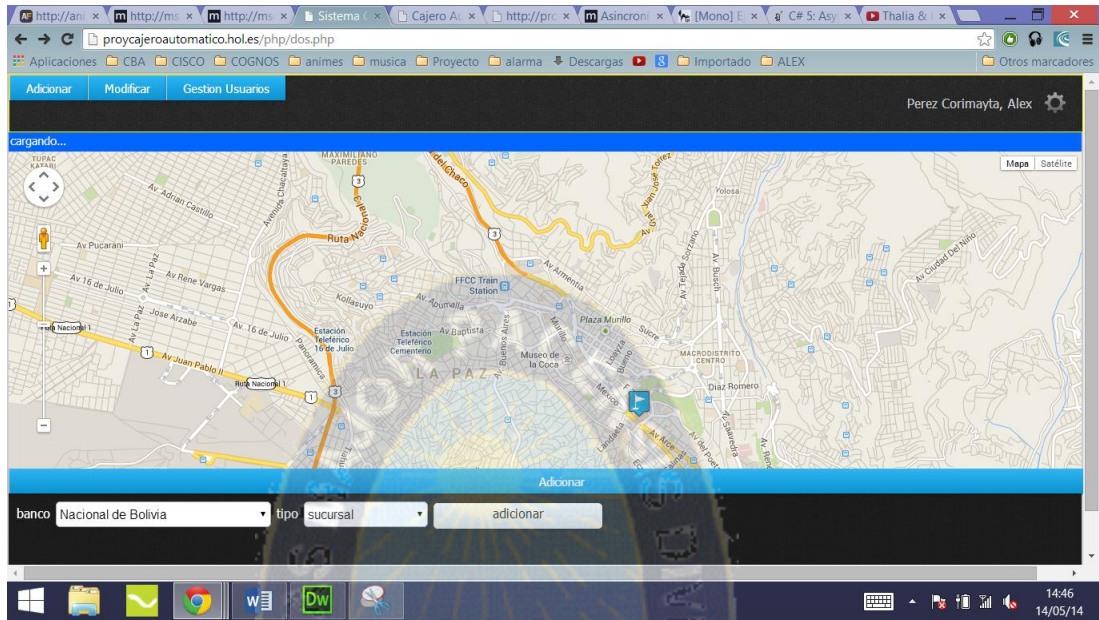


Figura 20: Cargando Nuevo Registro

Fuente: Elaboración propia

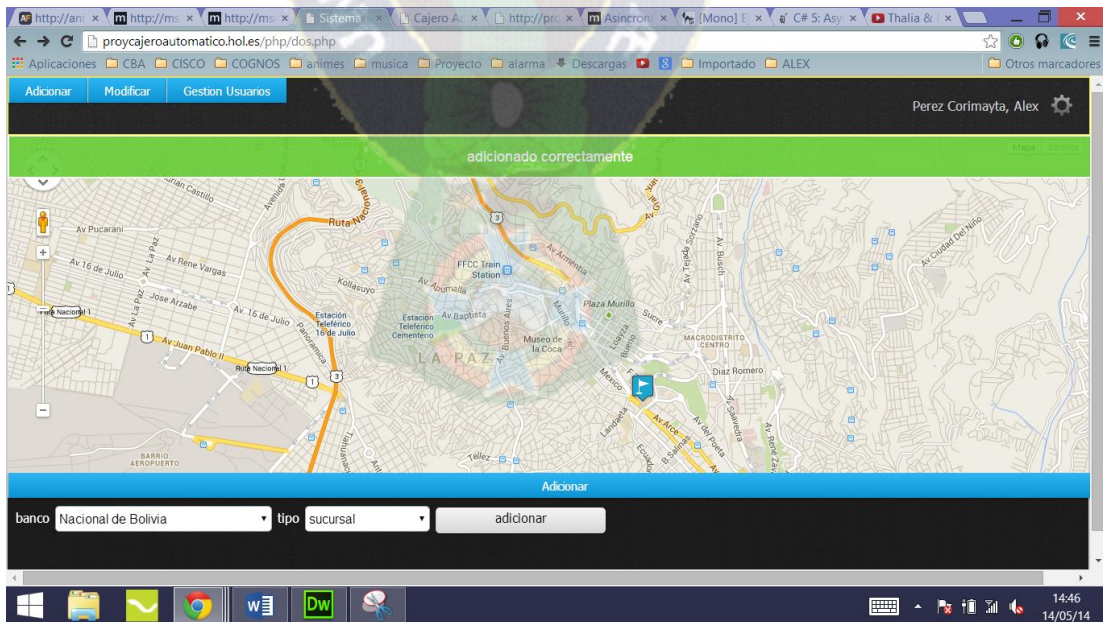


Figura 21: Entidad Financiera Adicionada correctamente

Fuente: Elaboración propia

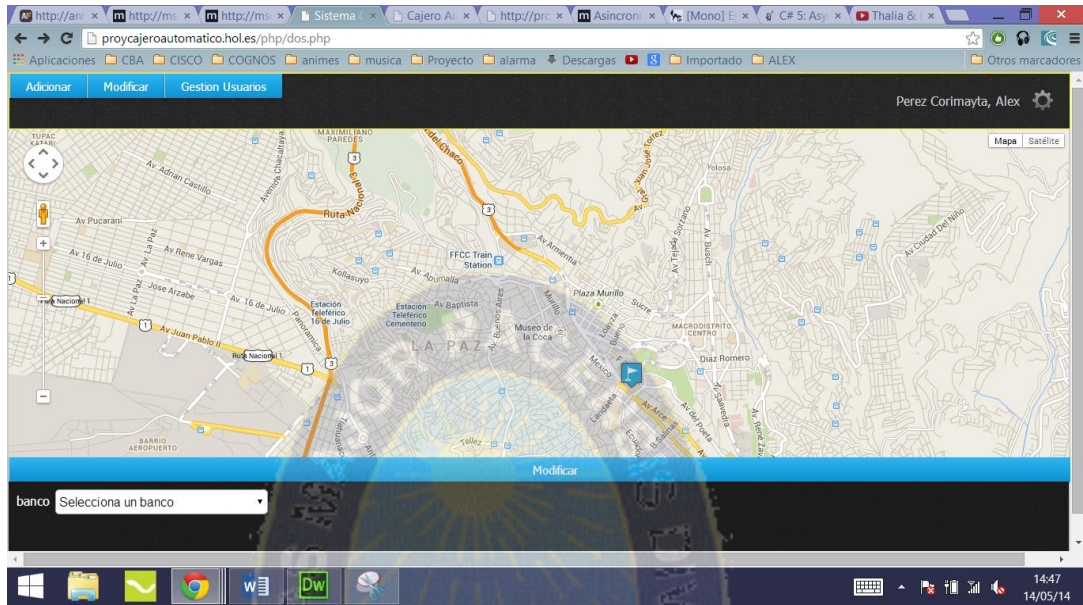


Figura 22: Modificar Entidad Financiera

Fuente: Elaboración propia

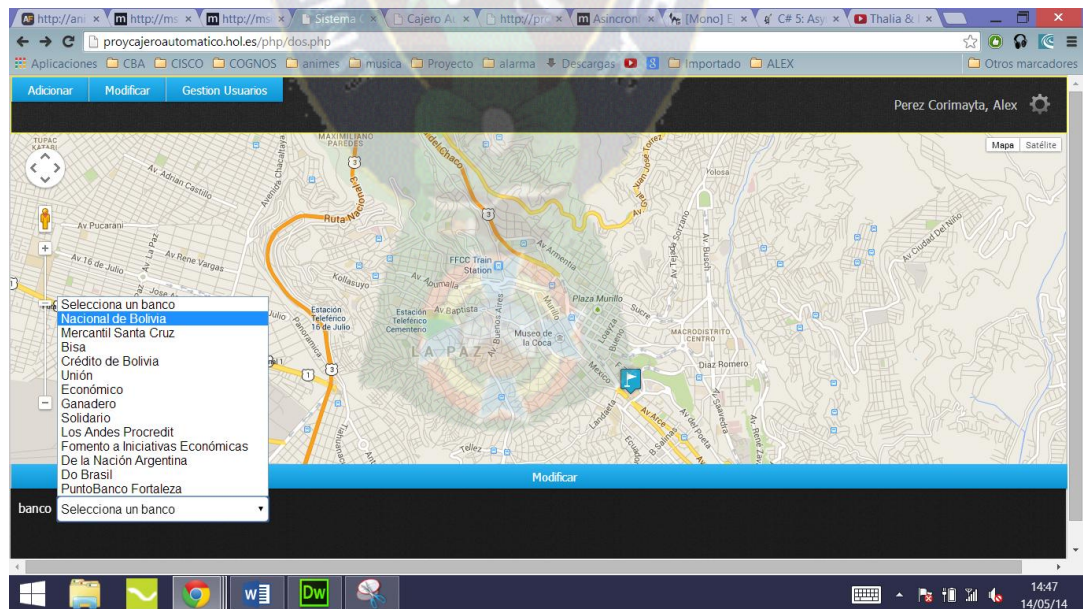


Figura 23: Seleccionar Entidad a Modificar

Fuente: Elaboración propia

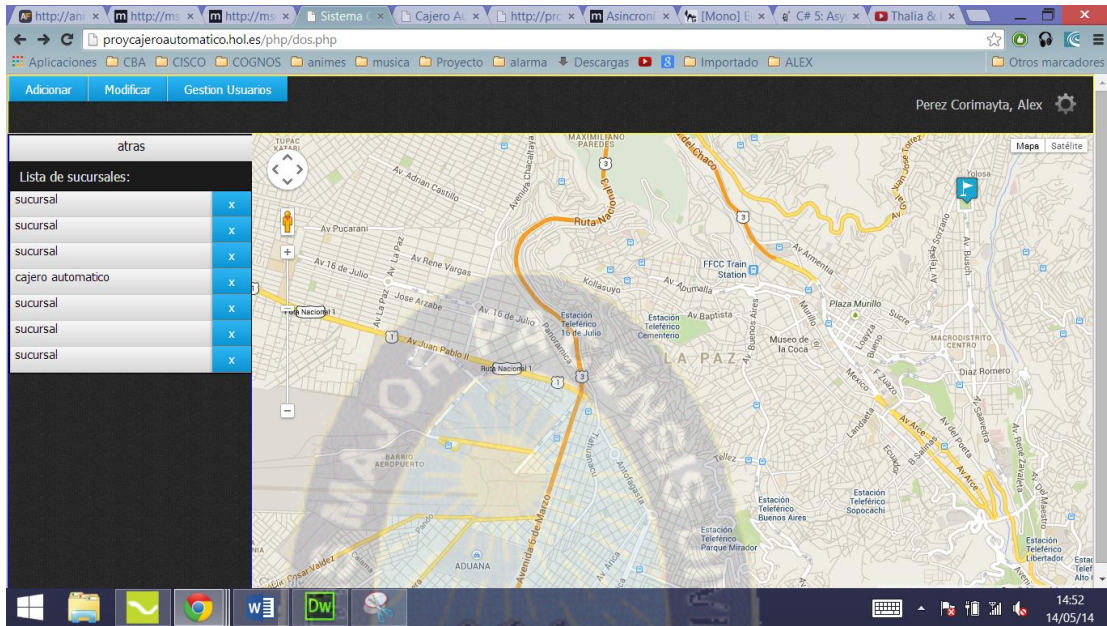


Figura 24: Modificar o Eliminar Registro

Fuente: Elaboración propia

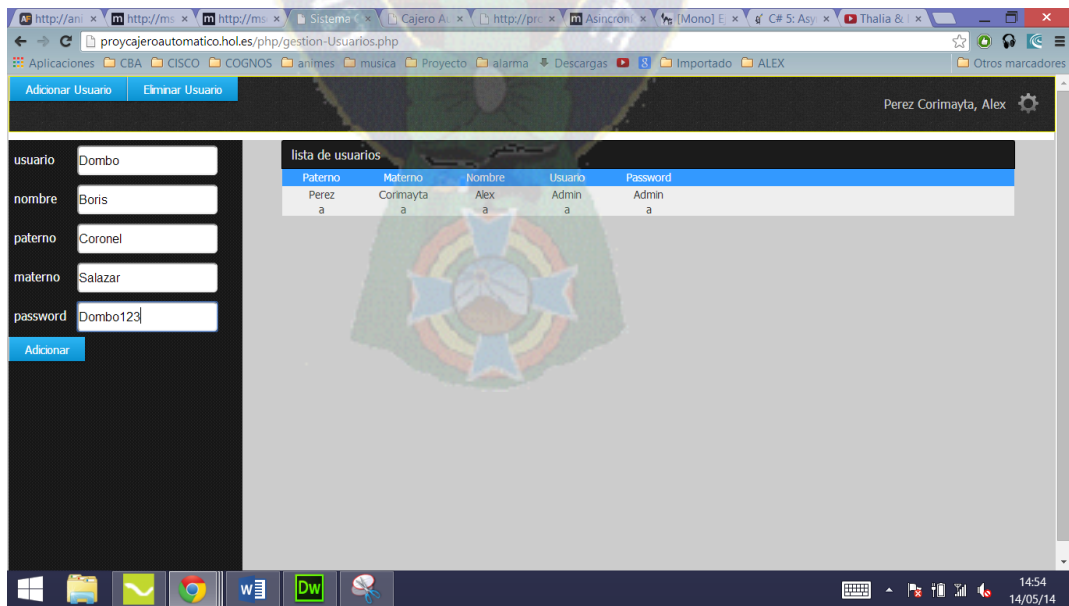


Figura 25: Adicionar usuario

Fuente: Elaboración propia

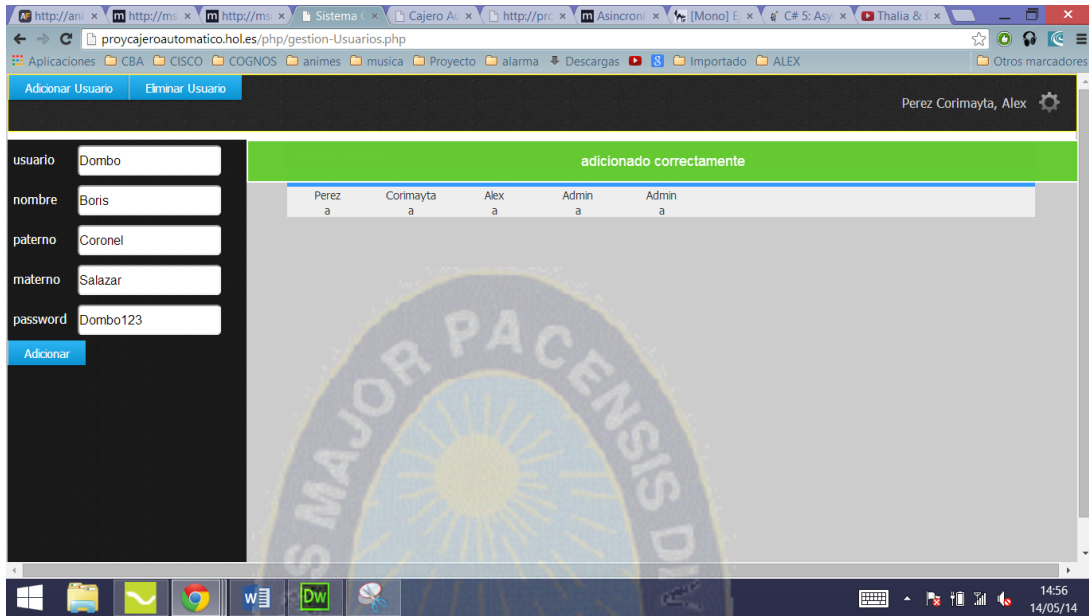


Figura 26: Usuario adicionado correctamente
Fuente: Elaboración propia

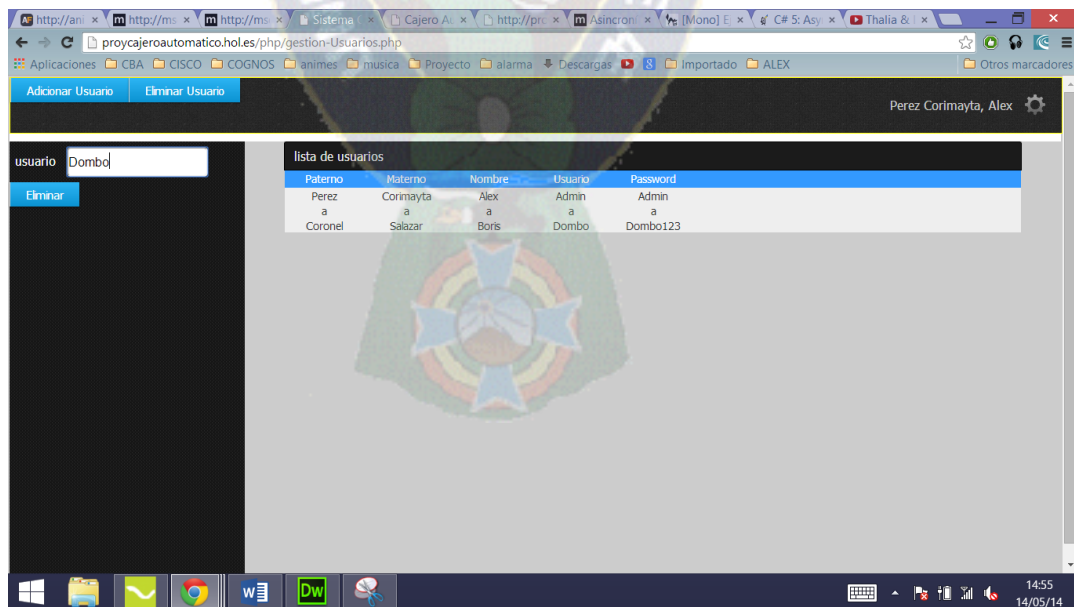


Figura 27: Eliminación de usuario
Fuente: Elaboración propia

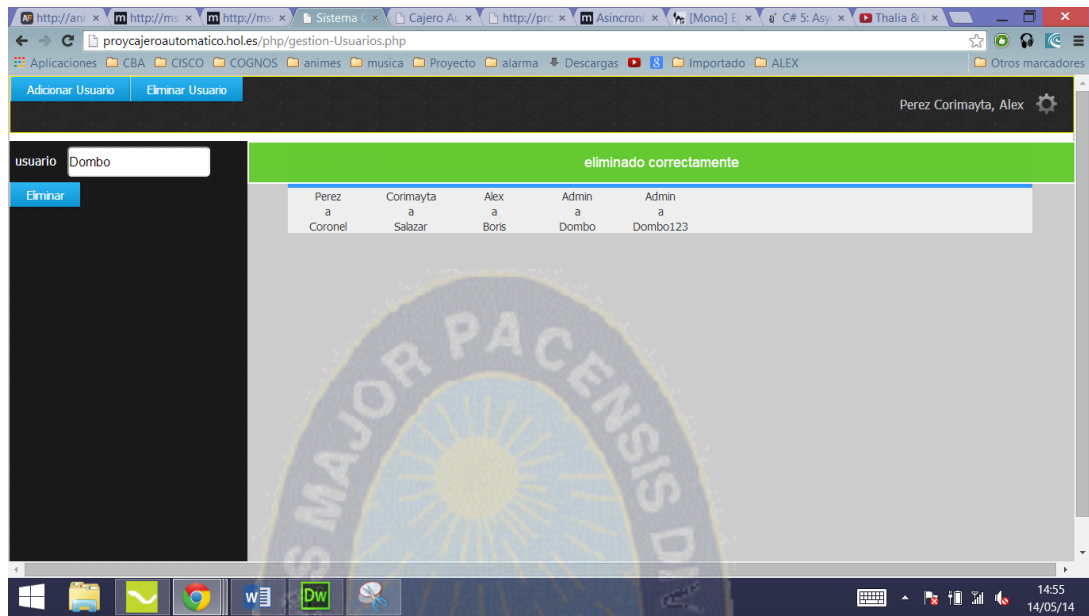


Figura 28: Usuario eliminado correctamente

Fuente: Elaboración propia

3.5.2.9 DISEÑO DE LA INTERFAZ DEL USUARIO

A continuación se muestra el diseño de interfaces para el usuario, que permite definir la interface de interacción entre el usuario y el software móvil.

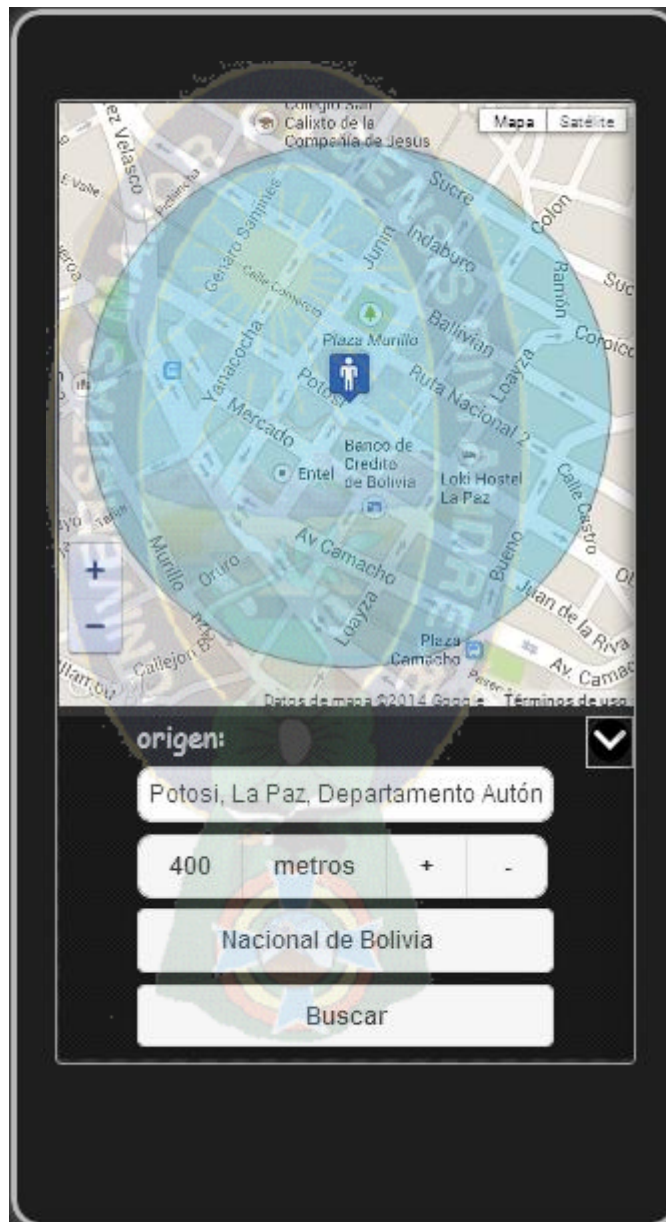


Figura 29: Interfaz para selección entidad financiera y selección de radio de búsqueda

Fuente: Elaboracion propia

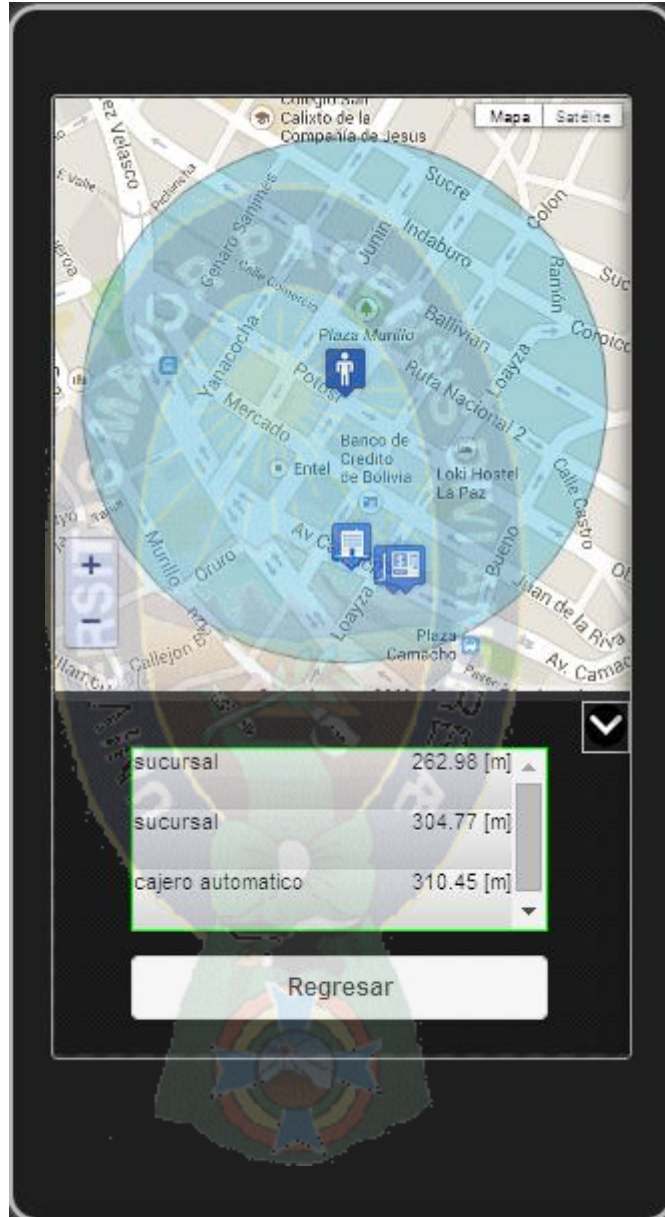


Figura 30: Mapa y panel de Resultados

Fuente: Elaboracion propia

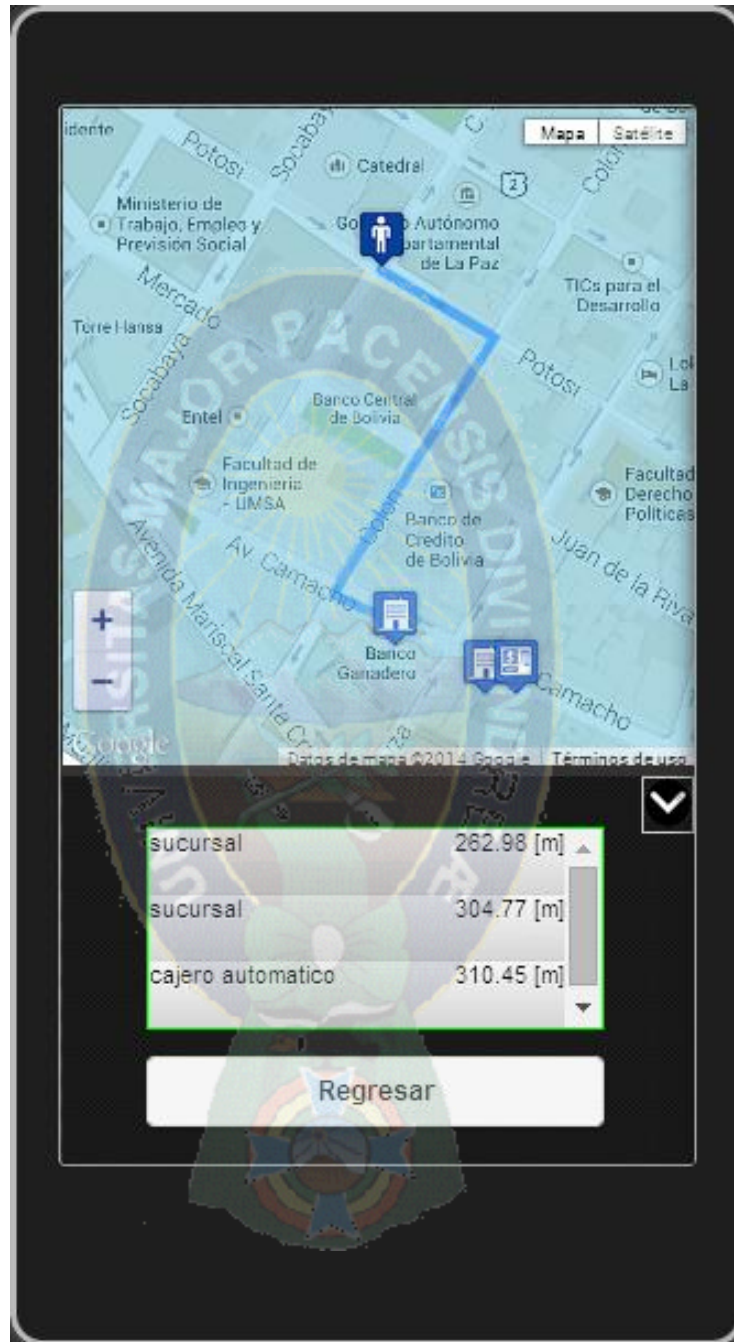


Figura 31: Elección de punto y trazado de ruta

Fuente: Elaboracion propia

3.5.3 PRODUCCIÓN

En esta sección se implementa la funcionalidad en el producto aplicando un ciclo de desarrollo iterativo e incremental; además se asegura la calidad del producto, haciendo uso de pruebas para comprobar que el sistema implementa las funcionalidades de forma correcta.

3.5.3.1 ESTÁNDARES DE CODIFICACIÓN

3.5.3.1.1 WORKSHOP DE POST ITERACIÓN

- Mejoras: Se debe mejorar la obtención de los datos de las entidades financieras, al igual que los resultados de las búsquedas.
- Fortalezas: El software móvil funciona con normalidad. Los datos mostrados están referidos a la ubicación del usuario o a la ubicación a la que el mismo desee referirse y el resultado de la búsqueda según los parámetros introducidos por el usuario.

Las altas, bajas y modificaciones de puntos de servicio financiero por parte del administrador funcionan con normalidad.

Las altas, bajas y modificaciones de usuarios por parte del administrador funcionan con normalidad.

El trazado de la ruta mínima de los puntos seleccionados por el usuario funciona con normalidad.

- Debilidades: El intercambio de información con el servidor es lenta por el uso de internet mediante tecnología 2G, pero con el uso de tecnologías 3G o superiores la interacción con el servidor cuenta de relativa normalidad.

3.5.3.2 ADICION DE PUNTOS DE SERVICIOS FINANCIEROS

Creación del panel de adición de ubicaciones, donde puede ingresar los datos de un nuevo punto de servicio financiero, especificando la entidad financiera a la que se hace referencia, el tipo de servicio (cajero automático o sucursal) y la ubicación, que se realiza arrastrando el punto a la ubicación (calle o avenida) dentro de Google Maps.

3.5.3.3 MODIFICACION DE PUNTOS DE SERVICIOS FINANCIEROS

Creación del panel de modificación que le permite al administrador realizar la modificación de un punto de servicio financiero ubicado dentro de la base de datos, el administrador indica la entidad financiera y el tipo de servicio, visualizando el panel de resultados con todos aquellos puntos que coinciden con el ingreso de esos parámetros, eligiendo posteriormente aquel punto de servicio financiero el cual desea modificar.

3.5.3.4 ESTABLECIMIENTO PARÁMETROS DE BÚSQUEDA

Este módulo permite al usuario establecer los siguientes parámetros de búsqueda; elegir a la entidad financiera, definir el tipo de servicio requerido, ya sea un cajero automático o una sucursal, definir un determinado radio de búsqueda. Mediante el cual el usuario podrá visualizar todos aquellos puntos a su alrededor los cuales le brinden ese servicio.

3.5.3.5 ESTABLECIMIENTO DE PUNTOS DE ORIGEN Y DESTINO

Este módulo permite al usuario establecer el punto de origen, este puede ser el punto en el cual se encuentre en el momento utilizando GPS o algún otro punto al cual desee referirse y el punto de destino que se seleccionará de todos los puntos visualizados en el radio de búsqueda.

Para el establecimiento de un punto de origen distinto a la ubicación del usuario, se hace uso del método de Autocomplete de la clase places de Google Maps que permite realizar la búsqueda de un determinado lugar (establecimientos, ubicaciones geográficas o puntos de interés destacados) al hacer uso de la librería **Google Places**, facilitando la búsqueda al contar con la función de autocompletado y sugiriendo al usuario una lista de lugares coincidentes con la entrada de texto.

Una vez seleccionado un lugar de la lista de sugerencias, el marcador correspondiente a la caja de texto se ubica automáticamente en el lugar seleccionado.

3.5.3.6 BÚSQUEDA DE PUNTOS DE SERVICIO FINANCIERO

El módulo de búsqueda de rutas se encarga de realizar la búsqueda de todos aquellos puntos que cumplen las condiciones de los parámetros de búsqueda ingresados por el usuario.

Para empezar se establece una circunferencia de radio r y con punto de origen en (h, k) a través de la ecuación de la circunferencia (Ec. 2).

$$x - h^2 + y - k^2 = r^2 \quad (\text{Ec. 2})$$

La circunferencia establecida permitirá identificar aquellos puntos de servicio financiero que estén ubicados dentro de un radio r .

3.5.3.7 DISCRIMINACIÓN DE ENTIDADES FINANCIERAS Y TIPOS DE SERVICIO

La discriminación de entidades financieras y tipos de servicio que ofrecen las mismas, permite clasificar los puntos de servicio según el criterio de los parámetros de búsqueda y elección de puntos ingresados por el usuario para así poder determinar aquella ruta que representa un menor recorrido de entre los puntos de origen y de destino.

3.5.3.8 TRAZADO DE LA RUTA MINIMA ENTRE DOS PUNTOS

Este módulo permite realizar el trazado de la ruta mínima entre dos puntos (el de origen y el de destino) dentro de un mapa, a través de una instancia de la clase polilyne, el cual tiene el método setpath() que permite almacenar al vector de coordenadas, esto implementando la fórmula de haversine en la consulta a la base de datos.

```

// FORMULA DE HAVERSINE
$sql="SELECT tipo,lat,lng, ( 6371 * ACOS(
                                COS( RADIANS('".$lat."') )
                                * COS(RADIANS( lat ) )
                                * COS(RADIANS( lng )
                                - RADIANS('".$lng."') )
                                + SIN( RADIANS('".$lat."') )
                                * SIN(RADIANS( lat ) )
                                )
                                ) AS distancia
FROM PUNTO_ATENCION
WHERE PUNTO_ATENCION.id_Banco = ( SELECT id_Banco
                                FROM BANCO
                                WHERE BANCO.nombre = '".$banco.'')
HAVING distancia < '".$radio.'"
ORDER BY distancia ASC";

```

Figura 32: Fragmento de código para la implementación de la Fórmula de Haversine

Fuente: Elaboracion propia

3.5.3.9 TEST DE ACEPTACIÓN

Hoja de Prueba de Aceptación	
Test ID	1
Historia	Funcionalidad Software Móvil –Web
Fecha Escrita	02/02/2014
Fecha Corrida	7/03/2014
Paso/Defecto	Paso

<p>Descripción</p>	<ol style="list-style-type: none"> 1 Modificación de puntos de servicios financieros 2 Establecimiento de parámetros de búsqueda 3 Establecimiento de puntos de origen y de destino 4 Búsqueda de puntos de servicio financiero 5 Discriminación de entidades financieras y tipos de servicio 6 Trazado de la ruta mínima entre dos puntos
<p>Resultados esperados</p>	<ol style="list-style-type: none"> 1 Ingreso de datos por medio de la plataforma web del administrador a la base de datos del servidor. 2 Actualización de la base de datos mediante la plataforma web del administrador 3 Correcta captura de parámetros de búsqueda y posterior envío 4 Correcta captura de coordenadas de puntos definidos para su posterior envío 5 Inspección visual del panel de resultados y comprobación de la base de datos 6 Inspección visual referente a la ruta mínima y comprobación de resultados mediante la revisión de distancias recorridas

Tabla 9: Test de aceptación de Producción

Fuente: Elaboración propia

3.5.4 ESTABILIZACIÓN

En esta sección se procede a estabilizar la correcta funcionalidad del prototipo, se continúa aplicando el desarrollo del ciclo iterativo e incremental asegurándose la calidad del producto.

3.5.4.1 WORKSHOP DE POST ITERACIÓN

- Mejoras: mejora en los estilos de la interfaz gráfica tanto en la aplicación móvil como en la plataforma web dando una mejor apariencia y presentación al usuario final y al administrador respectivamente.
- Fortalezas: La aplicación funciona de manera adecuada contando con una conexión a internet.
- Debilidades: La aplicación funciona solamente si se cuenta con una conexión a internet.

3.5.4.2 TEST DE ACEPTACIÓN

Hoja de Prueba de Aceptación	
Test ID	2
Historia	Correcciones Software Móvil –Web
Fecha Escrita	07/03/2014
Fecha Corrida	10/03/2014
Paso/Defecto	Paso
Descripción	<ol style="list-style-type: none">1 Ocultar paneles de control2 Lista desplegable de resultados3 Despliegue de mensajes de confirmación4 Validación de campos

Resultados esperados	1 Inspección visual 2 Inspección visual 3 Inspección visual 4 Inspección visual
-----------------------------	--

Tabla 10: Test de aceptación de Estabilización

Fuente: Elaboración propia

3.5.5 PRUEBAS

3.5.5.1 PLAN DE PRUEBAS

Para cada pantalla se prueba lo siguiente:

- Datos válidos
- Valores límite
- Datos inválidos
- El diseño debe ser como esta en la documentación
- Los enlaces entre pantallas tanto del software móvil como de la plataforma web deben funcionar tal como se las describe en la documentación.

Para las pruebas de tiempo de carga se tomo en cuenta los siguientes criterios:

- Se tomaron en cuenta los tres requerimientos funcionales más importantes (despliegue del panel de resultados, despliegue de los puntos en el mapa, trazado de la ruta mínima hacia el punto elegido).
- Se midió el tiempo de respuesta

Para las pruebas de tiempo de acceso se tomó en cuenta los siguientes criterios:

- Se tomó en cuenta el requerimiento funcional más importante para la comparación (trazado de la ruta mínima)

- Se considera tiempo de acceso al tiempo desde que un usuario abre una aplicación y recibe totalmente la información deseada.
- Para el acceso a la plataforma web se describen dos escenarios:
 - 1) El usuario accede al sitio mediante un buscador como Google, Bing, etc.
 - 2) El usuario accede al sitio escribiendo directamente la dirección en la barra direcciones del buscador.
- Se realizaron un determinado número de mediciones y entre cada medición se cerró la aplicación.

El Plan de Pruebas de Aceptación describe los pasos que se deben seguir para verificar que el sistema construido satisface los requerimientos.

El Plan de Pruebas de Aceptación es uno de los planes de prueba detallados y corresponde al nivel de pruebas de aceptación del sistema o de la solución. Este plan describe clara y completamente como realizar las pruebas.

Las pruebas de aceptación, involucran al usuario final y pretenden comprobar que la solución cumple con el modelo de negocio para el que fue desarrollado. Detección de defectos del producto entregado y planes de acción para corrección de los mismos.

3.5.5.2 PRUEBAS DE ACEPTACIÓN POR ITERACIÓN

Las pruebas de aceptación por iteración, se realizan a los requerimientos funcionales, y a los no funcionales como facilidad de uso, recuperación, eficiencia, entre otros; y se pretende lograr: corrección, vale decir, carencia de ambigüedad; completitud, es decir, especificación completa y clara del problema; y por último pero no menos importante, consistencia, quiere decir, que no haya requisitos contradictorios.

3.5.5.2.1 Resultados Iteración 1

	NÚMERO DE PRUEBAS	PORCENTAJE
Pruebas aceptadas	14	88%
Pruebas reprobadas	2	12%
TOTAL	16	100%
Pruebas Corregidas	2	100%

Tabla 11: Resultados Iteración 1

Fuente: Elaboración propia

3.5.5.2.2 Resultados Iteración 2

	NÚMERO DE PRUEBAS	PORCENTAJE
Pruebas aceptadas	13	81%
Pruebas reprobadas	3	19%
TOTAL	16	100%
Pruebas Corregidas	3	100%

Tabla 12: Resultados Iteración 2

Fuente: Elaboración propia

3.5.5.2.3 Resultados Iteración 3

	NÚMERO DE PRUEBAS	PORCENTAJE
Pruebas aceptadas	3	75%
Pruebas reprobadas	1	25%
TOTAL	4	100%
Pruebas Corregidas	1	100%

Tabla 13: Resultados Iteración 3

Fuente: Elaboración propia

3.5.5.2.4 Resultados Iteración 4

	NÚMERO DE PRUEBAS	PORCENTAJE
Pruebas aceptadas	5	84%
Pruebas reprobadas	1	16%
TOTAL	6	100%
Pruebas Corregidas	1	100%

Tabla 14: Resultados Iteración 4

Fuente: Elaboración propia

3.5.5.2.5 Resultados Iteración 5

	NÚMERO DE PRUEBAS	PORCENTAJE
Pruebas aceptadas	4	80%
Pruebas reprobadas	1	20%
TOTAL	5	100%
Pruebas Corregidas	1	100%

Tabla 15: Resultados Iteración 5

Fuente: Elaboración propia

3.5.5.2.6 Resumen de resultados de iteraciones

	NÚMERO DE PRUEBAS	PORCENTAJE
Pruebas aceptadas	39	83%
Pruebas reprobadas	8	17%
TOTAL	47	100%
Pruebas Corregidas	8	100%

Tabla 16: Resumen de resultados de iteraciones

Fuente: Elaboración propia

3.5.5.3 RESULTADOS TIEMPOS DE ACCESO

En base a las mediciones tomadas de los tiempos de acceso a la aplicación móvil y al acceso de un sitio web desde un navegador tradicional, en la siguiente tabla se muestra la comparación del promedio de las mediciones obtenidas al acceder a las funcionalidades de la aplicación móvil y del sitio web.

	Aplicación Móvil	Sitio Web
Tiempo 1	5	4
Tiempo 2	4	5
Tiempo 3	4	5
Tiempo 4	3	6
Tiempo 5	3	4
Tiempo 6	5	2
Tiempo 7	2	5
Tiempo 8	3	2
Promedio	3.625	4.125

Tabla 17: Resultado tiempos de acceso

Fuente: Elaboración propia

	Aplicación Móvil	Sitio Web	Diferencia
Promedio	3.625	4.125	0.5

Tabla 18: Promedio tiempos de acceso

Fuente: Elaboración propia

Las pruebas de tiempos de acceso se realizaron desde un teléfono Samsung Galaxy Core con versión sistema operativo android 4.0.1 con tecnología 3G. Para el sitio web se utilizó el navegador Internet Explorer 11.

CAPÍTULO IV

4 EVALUACIÓN DE RESULTADOS

4.1 INTRODUCCIÓN

En los siguientes acápite se presenta el desarrollo de las pruebas y el análisis del cumplimiento de los requerimientos, dando lugar a la prueba de la hipótesis planteada y por tanto el cumplimiento de los objetivos planteados.

4.2 EVALUACIÓN DE LOS CASOS DE PRUEBA

En este punto recordaremos el resumen de los resultados obtenidos mediante las pruebas de aceptación por iteración desarrolladas en el anterior capítulo, siendo estos los siguientes:

Pruebas totales:	47
Pruebas aceptadas:	39
Pruebas reprobadas:	8
Porcentaje de éxitos:	83%
Porcentaje de fracasos:	17%

Como se observa en los resultados, el porcentaje de éxitos es menor al 83% esperado al momento de plantearse la Hipótesis, pero para comprobar de manera cuantificable se este valor se asemeja al valor esperado, se realiza una prueba de hipótesis estadística.

4.3 PRUEBA DE HIPÓTESIS

Una hipótesis es una suposición que se establece como base de una investigación que puede confirmar o negar su validez, su función principal es demarcar el problema

que se va a investigar considerando componentes tales como lugar, características de los sujetos, tiempo, etc.

4.3.1 EVALUACIÓN DE RESULTADOS

Para determinar si el porcentaje de éxitos obtenido en las pruebas puede ser considerado cercano al 85% de nivel de confianza esperado, se hará uso de una PRUEBA DE HIPÓTESIS PARA PROPORCIONES, la cual se encuentra detallada en el anexo B.

Las variables usadas en dicha prueba serán las mismas mencionadas en la evaluación de casos de prueba:

$$p_0 = 85$$

$$q_0 = 15$$

$$p = 83$$

$$n = 47$$

N es el número total de sujetos en el espacio muestral y no se conoce. Además que se tomará un nivel de significancia α del 5%.

4.3.2 DEFINICIÓN DE LA HIPÓTESIS H_0 Y H_1

Recordando la hipótesis planteada:

H_0 : LA GEOLOCALIZACIÓN COMO PROCESO DE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA PERMITE RECIBIR INFORMACIÓN CONFIABLE Y PRECISA SOBRE UBICACIONES DE SERVICIOS FINANCIEROS DENTRO DE UN MAPA, Y MEDIANTE EL USO DE UN DISPOSITIVO MÓVIL SE PODRÁ ESTABLECER LA RUTA HACIA UNA DE ESAS POSICIONES.

H_1 : RECHAZA H_0 .

En este caso se espera que el porcentaje de éxitos sea igual o mayor a 85%:

$$H_0: p \geq p_0 H_1: p < p_0$$

Es decir:

$$H_0: p \geq 0.85 H_1: p < 0.85$$

HIPÓTESIS NULA H_0 : SE AFIRMA EL VALOR CERCANO AL 85% DE EFECTIVIDAD.

HIPÓTESIS ALTERNA H_1 : AFIRMACION DE QUE SE ACEPTARA SI LOS DATOS DE PRUEBA PROPORCIONAN EVIDENCIA DE QUE LA HIPÓTESIS NULA ES FALSA.

4.3.3 DETERMINACIÓN DE LA REGIÓN CRÍTICA

La región crítica para la hipótesis planteada, es la siguiente:

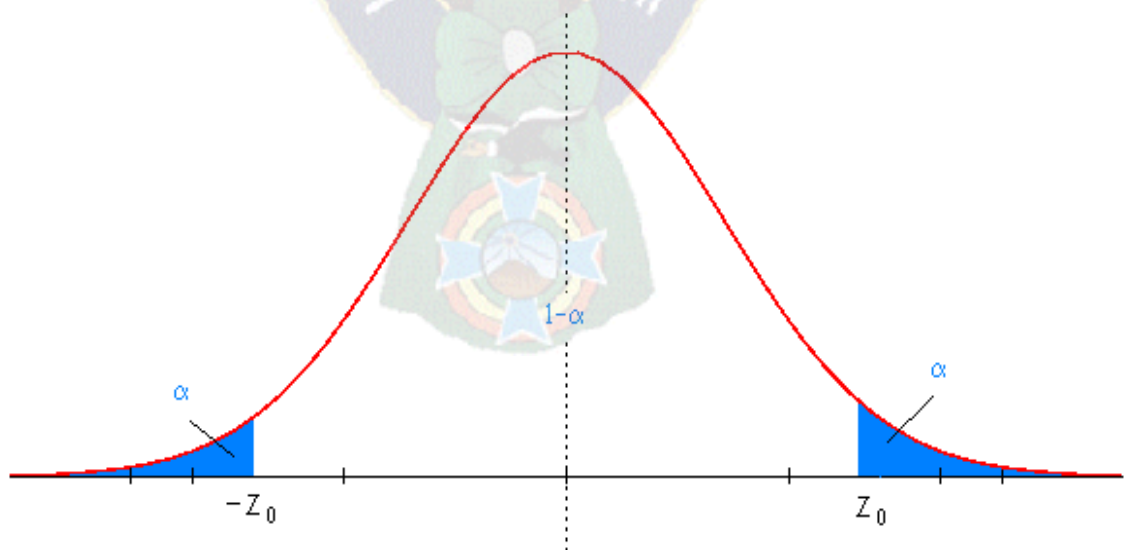


Figura 33: Región crítica para la hipótesis

Fuente: Elaboracion propia

Como n se refiere en este caso al número de pruebas, en este caso 47, el punto crítico a usar es $-Z_0$ y se determina mediante:

$$-Z_0 = -Z_{1-\alpha} = -Z_{1-0.05}$$

$$-Z_0 = -Z_{0.05}$$

Este valor se halla de la tabla de la función de distribución normal, la cual se encuentra en el anexo A. Para obtener el valor de z se elige de la tabla mencionada el valor más cercano a 0,95; el cual está ubicado en la fila 1,6 y columna 0,04.

z	...	0,04
...		↓
1,6	→	0,94950

Figura 34: Resultado tabla de la función de distribución normal

Fuente: Elaboración propia

El valor de z se obtiene sumando ambos valores:

$$-Z_{0,95} = -1.6 + 0.04 = -1,64$$

4.3.4 CÁLCULO DEL ESTADÍSTICO DE LA PRUEBA

Como se conoce el número total de individuos en el espacio muestral, el valor del estadístico de la prueba se obtiene mediante la fórmula:

$$Z_c = \frac{p - p_0}{\frac{p_0 q_0}{n}} = \frac{0.83 - 0.85}{\frac{0.85 \cdot 0.15}{47}} = -0.384$$

4.3.5 TOMA DE DECISIÓN

Al comparar el valor de los estadísticos Z_0 y Z_c , se observa que el valor del estadístico de la prueba no se encuentra dentro de la región crítica, por lo tanto se acepta la hipótesis nula H_0 .

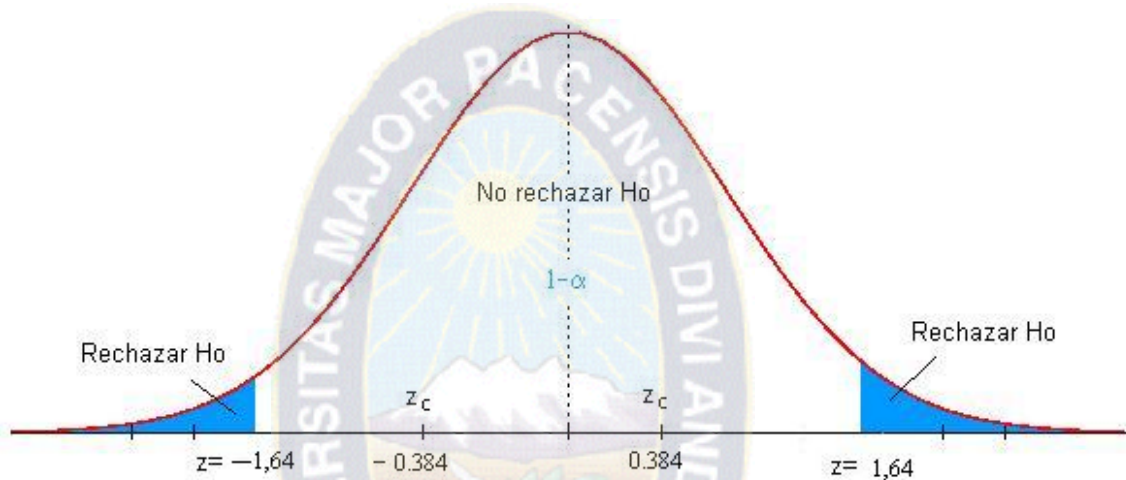


Figura 35: Distribución de Z_0 y Z_c en el gráfico para la toma de decisión

Fuente: Elaboración propia

El promedio de éxito del prototipo al momento de reconocer las muestras se acerca al 85%. Por tanto como no se rechaza H_0 se podría concluir y afirmar la hipótesis H_0 : "LA GEOLOCALIZACIÓN COMO PROCESO DE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA PERMITE DESPLEGAR UBICACIONES DE SERVICIOS FINANCIEROS DENTRO DE UN MAPA, EN EL CUAL MEDIANTE EL USO DE UN DISPOSITIVO MÓVIL SE PODRÁ ESTABLECER EL RECORRIDO ÓPTIMO HACIA UNA DE ESAS POSICIONES".

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Se pudo alcanzar el objetivo principal de estudio: Desarrollar un software móvil utilizando georreferenciación que involucre información sobre la ubicación de servicios financieros y que le brinde al usuario una herramienta confiable y precisa capaz de mostrar la ruta hacia la ubicación del servicio requerido.

Mediante el uso de la metodología Mobile-D se desarrolló la aplicación web-móvil cumpliendo los siguientes objetivos específicos planteados en el primer capítulo:

- Se desarrollo el módulo que permite filtrar las ubicaciones de una entidad financiera en específico dentro del mapa.
- Se diseño una base de datos con coordenadas señalando el nombre y el tipo de servicio que representa.
- Se uso el API de Google Maps para acceder a los mapas y librerías; que a su vez permitieron implementar el módulo visual de trazado de rutas, el uso de marcadores para identificar las ubicaciones tanto de sucursales como cajeros automáticos, el módulo visual de trazado de radio de búsqueda.
- Se uso PhoneGap para el desarrollo de la aplicación móvil compatible con sistemas operativos móviles.

Mediante el análisis de resultados obtenidos en el capítulo cuatro, se concluye la solución del problema principal del CÓMO RECIBIR INFORMACIÓN CONFIABLE Y PRECISA SOBRE LA UBICACIÓN Y LA RUTA HACIA EL SERVICIO DE UNA ENTIDAD FINANCIERA USANDO TECNOLOGÍAS DE GEORREFERENCIACIÓN Y EL USO DE UN DISPOSITIVO MÓVIL.

Al crear una aplicación para dispositivos móviles se incrementa la usabilidad de las personas, además de ser en parte, una importante herramienta para personas que no viven en nuestra ciudad.

En los resultados se observa una descarga lenta de la información, esto se debe a las limitaciones de conectividad de nuestro medio. Sin embargo, la aplicación es de gran utilidad y susceptible a convertirse en un producto comercializable.

A pesar de contar con documentación sobre la metodología de desarrollo no se encontró mucha información o plantillas sobre la realización del plan de iteraciones, por lo que se debió recurrir a información del tema en extreme programming por ser una de las metodologías base de Mobile-D.

El desarrollo basado en pruebas permitió detectar y corregir errores en una temprana etapa del desarrollo lo que permite no arrastrar errores en otros disminuyendo el impacto de la corrección de error, permitiendo que el software móvil tenga alta calidad.

5.2 RECOMENDACIONES

Junto con la información de las distintas entidades financieras que tienen sus actividades en la ciudad de La Paz, la aplicación podría extenderse a otro tipo de rubros e incluso ciudades.

Al desarrollar aplicaciones móviles se debe tener en cuenta que los usuarios son del tipo ocasional, es decir no están constantemente usando las aplicaciones todo el tiempo sino por intervalos cortos, por lo cual se debe permitir al usuario acceder a las funcionalidades más importantes de forma más rápida e intuitiva.

No existen muchas metodologías orientadas al desarrollo de aplicaciones web-móviles y podría ser un campo en el que se planteen nuevas soluciones que agilicen el desarrollo de software sin muchos formalismos como lo hacen las metodologías en la actualidad.

Se recomienda continuar con la implementación de la aplicación compatible a otros sistemas operativos móviles considerando la flexibilidad de los servicios web.

Adquirir nuevos conocimientos sobre el desarrollo de software móvil híbrido y el uso de nuevas herramientas para agilizar los procesos de desarrollo e incentivar a los estudiantes de la carrera al desarrollo de soluciones adaptables a los problemas y necesidades de la actualidad.



BILIOGRAFÍA

- Amo, F. A., & Normand, L. M. (2005). *Introducción a la ingeniería del software*. Delta Publicaciones, 2005.
- Chang, K.-T. (2010). *Introduction to Geographic Information Systems*. McGraw-Hill.
- Duclos, .. C. (2010). *Product Formuns Google*. Obtenido de *Product Formuns Google*: <http://productforums.google.com/forum/#!topic/maps-es/S0wHzdsSj0>.
- Fernández, G. (2002). *Introducción a Extreme Programming*. Recuperado el 12 de Septiembre de 2013, de <http://www.um.edu.ar/>
- Gutiérrez, .. J. (2000). *SIG: Sistemas de Información Geográfica*. Puebla: Síntesis.
- Laurent Debrauwer, F. V. (2009). *UML 2 Iniciacion, ejemplos y ejercicios corregidos*. Ediciones ENI, 2009.
- Lopez, C. P. (2000). *Tecnicas de Muestreo Estadístico*. Mexico: Alfaomega Grupo, Editor, S.A de C.V.
- Mancha, U. d. (Octubre de 2013). *rabajo-XP.pdf*. Obtenido de <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>
- Murphy, C., & Yates, J. (2009). *The International Organization for Standardization (ISO): global governance through voluntary consensus*. Taylor & Francis, 2009.
- Paredes, J. (8 de Febrero de 2013). En Bolivia, 320 mil personas usan internet banda ancha. *La Razón*.

Pratt, M. (2013). *La Formula de Haversine*.

Pressman, R. S. (2008). *Ingeniería de Software. Sexta edición*. Mc Graw Hil.

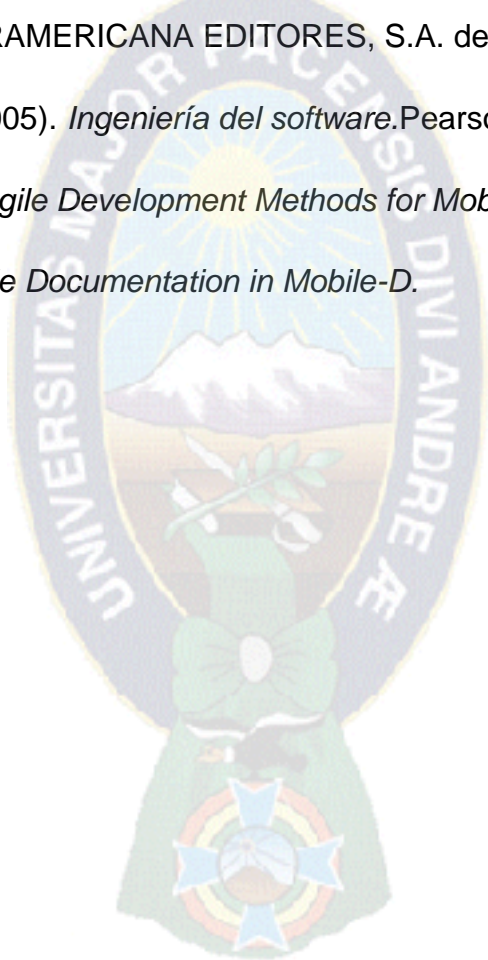
Salkind, N. J. (1998). *Métodos de investigación*. Pearson Educación, 1998.

Sampieri, R. H. (1998). *Metodologia de la Investigacion*. Mexico: McGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V.

Sommerville, I. (2005). *Ingeniería del software*. Pearson Educación, 2005.

Spataru. (2010). *Agile Development Methods for Mobile Applications*.

Tanja. (2012). *Agile Documentation in Mobile-D*.





ANEXOS

ANEXO A

TABLA DE DISTRIBUCIÓN NORMAL

z	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
0,0	0,5000	0,5040	0,5080	0,5120	0,5160	0,5199	0,5239	0,5279	0,5319	0,5359
0,1	0,5398	0,5438	0,5478	0,5517	0,5557	0,5596	0,5636	0,5675	0,5714	0,5753
0,2	0,5793	0,5832	0,5871	0,5910	0,5948	0,5987	0,6026	0,6064	0,6103	0,6141
0,3	0,6179	0,6217	0,6255	0,6293	0,6331	0,6368	0,6406	0,6443	0,6480	0,6517
0,4	0,6554	0,6591	0,6628	0,6664	0,6700	0,6736	0,6772	0,6808	0,6844	0,6879
0,5	0,6915	0,6950	0,6985	0,7019	0,7054	0,7088	0,7123	0,7157	0,7190	0,7224
0,6	0,7257	0,7291	0,7324	0,7357	0,7389	0,7422	0,7454	0,7486	0,7517	0,7549
0,7	0,7580	0,7611	0,7642	0,7673	0,7703	0,7734	0,7764	0,7794	0,7823	0,7852
0,8	0,7881	0,7910	0,7939	0,7967	0,7995	0,8023	0,8051	0,8078	0,8106	0,8133
0,9	0,8159	0,8186	0,8212	0,8238	0,8264	0,8289	0,8315	0,8340	0,8365	0,8389
1,0	0,8413	0,8438	0,8461	0,8485	0,8508	0,8531	0,8554	0,8577	0,8599	0,8621
1,1	0,8643	0,8665	0,8686	0,8708	0,8729	0,8749	0,8770	0,8790	0,8810	0,8830
1,2	0,8849	0,8869	0,8888	0,8907	0,8925	0,8944	0,8962	0,8980	0,8997	0,9015
1,3	0,9032	0,9049	0,9066	0,9082	0,9099	0,9115	0,9131	0,9147	0,9162	0,9177
1,4	0,9192	0,9207	0,9222	0,9236	0,9251	0,9265	0,9279	0,9292	0,9306	0,9319
1,5	0,9332	0,9345	0,9357	0,9370	0,9382	0,9394	0,9406	0,9418	0,9429	0,9441
1,6	0,9452	0,9463	0,9474	0,9484	0,9495	0,9505	0,9515	0,9525	0,9535	0,9545
1,7	0,9554	0,9561	0,9573	0,9582	0,9591	0,9599	0,9608	0,9616	0,9625	0,9633
1,8	0,9641	0,9649	0,9656	0,9664	0,9671	0,9678	0,9686	0,9693	0,9699	0,9706
1,9	0,9713	0,9719	0,9726	0,9732	0,9738	0,9744	0,9750	0,9756	0,9761	0,9767
2,0	0,9772	0,9778	0,9783	0,9788	0,9793	0,9798	0,9803	0,9808	0,9812	0,9817
2,1	0,9821	0,9826	0,9830	0,9834	0,9838	0,9842	0,9846	0,9850	0,9854	0,9857
2,2	0,9861	0,9864	0,9868	0,9871	0,9875	0,9878	0,9881	0,9884	0,9887	0,9890
2,3	0,9893	0,9896	0,9898	0,9901	0,9901	0,9906	0,9909	0,9911	0,9913	0,9916
2,4	0,9918	0,9920	0,9922	0,9925	0,9927	0,9929	0,9931	0,9932	0,9934	0,9936
2,5	0,9938	0,9940	0,9941	0,9943	0,9945	0,9946	0,9948	0,9949	0,9951	0,9952
2,6	0,9953	0,9954	0,9956	0,9957	0,9959	0,9960	0,9961	0,9962	0,9963	0,9964
2,7	0,9965	0,9966	0,9967	0,9968	0,9969	0,9970	0,9971	0,9972	0,9973	0,9974

Tabla 20: Tabla de distribución normal

Fuente: <http://www.vadenumeros.es/>

ANEXO B

PRUEBA DE HIPÓTESIS PARA PROPORCIONES

En el campo de la estadística, para comprobar la validez de una afirmación de carácter cuantitativo, se realiza un proceso de prueba de hipótesis estadística.

Frecuentemente se desea estimar la proporción de elementos que tienen una característica determinada, en tal caso, las observaciones son de naturaleza cualitativa. Cuando se analiza información cualitativa y se está interesado en verificar un supuesto acerca de la proporción poblacional de elementos que tienen determinada característica, es útil trabajar con la prueba de hipótesis para la proporción.

En este caso la prueba de hipótesis para una proporción define que una hipótesis es una afirmación, que puede ser de dos tipos:

- Hipótesis nula; denotada por H_0 , es la afirmación a priori que se pretende demostrar luego de haberse realizado el estudio.
- Hipótesis alterna; denotada por H_1 , es el opuesto de la hipótesis nula, por lo que representa que la afirmación hecha a priori no pudo ser demostrada.

Ambas hipótesis son afirmaciones totalmente opuestas. Una prueba de hipótesis consiste en determinar cual de las dos afirmaciones es correcta, por ello se realiza para decidir si la hipótesis H_0 es correcta, caso contrario, se acepta H_1 .

MÉTODOS

Podemos usar cualquiera de los siguientes:

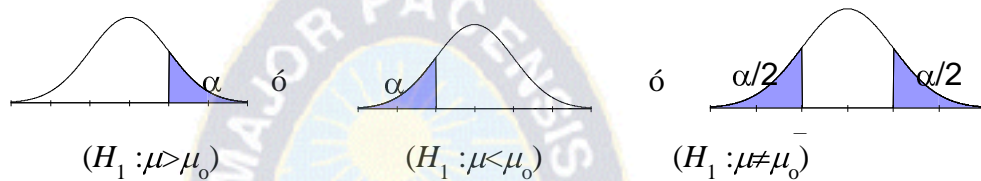
1. Método de la región de rechazo (Método 1) ó
2. Método del valor P (Método 2)

A. MÉTODO DE LA REGIÓN DE RECHAZO (MÉTODO 1)

Digamos que p_0 es la proporción aceptada o reclamada.

Paso 1 Establezca las hipótesis. $H_0 : p = p_0$
 $H_1 : p > p_0$ ó
 $p < p_0$ ó
 $p \neq p_0$

Paso 2 Use el nivel de significancia (α) y dibuje la región de rechazo en la curva normal estándar (curva z).



Paso 3 Calcule el valor z para la proporción muestral $\left(\bar{p} = \frac{x}{n}\right)$ usando la fórmula

$$Z = \frac{\bar{p} - p_0}{\sigma_p}, \quad \sigma_p = \sqrt{\frac{p_0(1-p_0)}{n}}$$

Paso 4 Dibuje este valor de z en el diagrama de la región de rechazo (Paso 2).

Paso 5 Si el valor z cae dentro de la región de rechazo (sombreada), entonces rechace H_0 . Si cae fuera de la región sombreada, entonces no rechace H_0 .

Paso 6 Escriba la conclusión de la prueba.

B. MÉTODO DEL VALOR P (MÉTODO 2)

Dejemos que p_0 sea la proporción aceptada o reclamada.

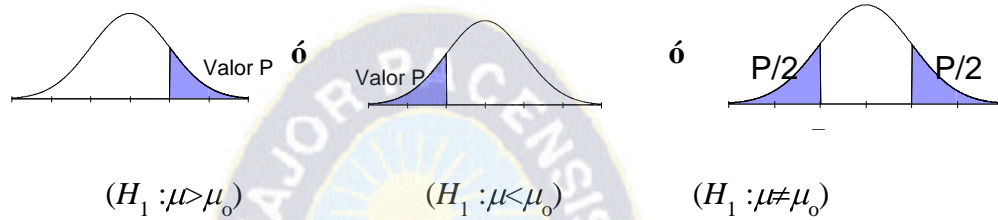
Paso 1 Establezca las hipótesis: $H_0 : p = p_0$
 $H_1 : p > p_0$ ó
 $p < p_0$ ó
 $p \neq p_0$

Paso 2 Calcule el valor z para la proporción muestral $\left(\bar{p} = \frac{x}{n}\right)$ usando la

fórmula:

$$Z = \frac{\bar{p} - p_0}{\sigma_p}, \text{ donde } \sigma_p = \sqrt{\frac{p_0(1-p_0)}{n}}.$$

Paso 3 Usando la hipótesis alterna dibuja la región bajo la curva z que representa los valores extremos.



Paso 4 El valor P = al área de la cola sombreada (s) en el Paso 3.

Paso 5 Si el valor $P < \alpha$, entonces rechaza H_0
Si el valor $P \geq \alpha$, entonces no rechaces H_0 .

Paso 6 Escribe la conclusión de la prueba.



DOCUMENTACIÓN