



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“APLICACIÓN MULTIMEDIA INTERACTIVA QUE PERMITE AL
USUARIO, MEDIANTE SU DISPOSITIVO MÓVIL ANDROID V4.0
EN ADELANTE, LOCALIZAR EL CAJERO AUTOMÁTICO
BANCARIO MÁS CERCANO A SU UBICACIÓN
PROPORCIONADA DENTRO DE GUAYAQUIL”**

INFORME DE PROYECTO DE GRADUACIÓN

Previa a la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES

ORIENTACIÓN SISTEMAS MULTIMEDIA

Presentado por:

María Beatriz Gómez Baño
Andrés Alberto Gómez Baño

Guayaquil – Ecuador
2015

AGRADECIMIENTO

Agradecemos a Dios, por ser nuestro guía e inspiración en la vida. A nuestra familia por todo su apoyo incondicional durante la etapa universitaria. A nuestros maestros por ser la fuente de experiencia para el aprendizaje.

María Beatriz y Andrés

DEDICATORIA

A Dios por ser mi guía durante mi camino estudiantil, a mis padres, y a todos mis seres queridos por brindarme todo su apoyo.

María Beatriz

DEDICATORIA

A Dios por bendecirme y darme sabiduría para poder culminar mi carrera universitaria. A mi familia por ser fuente de apoyo y motivación.

Andrés

TRIBUNAL DE SUSTENTACIÓN

Ph.D. Sixto García.

**SUBDECANO
SUBROGANTE DE LA FIEC**

Ph.D. Xavier Ochoa.

**DIRECTOR DEL PROYECTO
DE GRADUACIÓN**

Ph.D. Sixto García.

MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Informe, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)

María Beatriz Gómez Baño

Andrés Alberto Gómez Baño

RESUMEN

Hoy en día el uso de aplicaciones móviles se ha convertido en un factor muy importante para los usuarios de teléfonos o tabletas inteligentes, muchas de estas aplicaciones se descargan de forma gratuita desde la plataforma de distribución digital para dispositivos móviles con sistema operativo Android: “Google Play”.

Dicha plataforma, permite descargar aplicaciones que abarcan desde juegos, redes sociales, música, libros, noticieros, carteleras de cine, cuentas bancarias hasta aplicaciones de personalización del dispositivo, la intención de esta es brindarle facilidades a los usuarios en su vida diaria, simplificando las acciones que ejecutan al realizar sus actividades cotidianas.

La Banca Móvil es considerada muy útil por sus consumidores, por lo cual el sector Bancario del país se mantiene en constante innovación para cumplir todo tipo de exigencias. Actualmente los Bancos Nacionales están incorporando dentro de sus aplicaciones móviles la búsqueda de Cajeros Automáticos por GPS.

Con la implementación de “UbicATM” se propone enlazar en una sola aplicación la ubicación de los Cajeros Automáticos de las Instituciones Bancarias de la ciudad de Guayaquil, por medio de esta aplicación se simplificaría la búsqueda de Cajeros Automáticos.

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE SUSTENTACIÓN.....	V
DECLARACIÓN EXPRESA.....	VI
RESUMEN.....	VII
ÍNDICE GENERAL.....	VIII
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS.....	XIV
ABREVIATURAS.....	XVII
INTRODUCCIÓN	XVIII
CAPÍTULO 1	1
1. ANTECEDENTES Y JUSTIFICACIÓN	1
1.1. DESCRIPCIÓN DEL PROBLEMA.....	2
1.2. SOLUCIÓN PROPUESTA.....	4
1.3. JUSTIFICACIÓN DEL PROYECTO	7
1.4. OBJETIVOS.....	8
1.4.1. Objetivos Generales	8
1.4.2. Objetivos Específicos	9
1.5. ALCANCE DEL PROYECTO	9
1.6. METODOLOGÍA.....	11
1.6.1. Determinar la ubicación de los Cajeros Automáticos.....	11
1.6.2. Configurar el Servidor Web	12
1.6.3. Diseñar y Desarrollar la Aplicación.....	12
1.6.4. Someter a Pruebas la Aplicación.....	13
CAPÍTULO 2.....	15
2. MARCO TEÓRICO Y ANÁLISIS DE LA SOLUCIÓN	15
2.1. REQUERIMIENTOS	15
2.2. ANDROID	19

2.2.1. Versiones de Android	20
2.2.2. Android SDK	26
2.3. GOOGLE APP ENGINE.....	27
2.4. FRAMEWORK.....	28
2.5. BASE DE DATOS	28
2.6. SERVICIOS WEB.....	28
2.7. APACHE JMETER.....	29
2.8. ANÁLISIS DE LA SOLUCIÓN.....	29
CAPÍTULO 3.....	51
3. DISEÑO DE LA SOLUCIÓN.....	51
3.1. UBICACIONES DE LOS CAJEROS AUTOMÁTICOS.....	52
3.2. ESQUEMA DEL SISTEMA WEB.....	53
3.3. ARQUITECTURA DEL SISTEMA WEB	54
3.4. MODELO DE BASE DE DATOS	57
3.5. ALGORITMO DE BÚSQUEDA DE PUNTOS CERCANOS	59
3.6. ESQUEMA DE LA APLICACIÓN ANDROID	61
3.7. ARQUITECTURA DE LA APLICACIÓN ANDROID.....	63
3.8. ARQUITECTURA DE DEPENDENCIA	66
3.9. DISEÑO DE PRUEBAS	69
CAPÍTULO 4.....	72
4. IMPLEMENTACIÓN DE LA SOLUCIÓN.....	72
4.1. DESARROLLO DEL SISTEMA WEB.....	72
4.2. DESARROLLO DE LA APLICACIÓN MÓVIL.....	82
4.3. PUBLICACIÓN DE LA APLICACIÓN WEB	87
CAPÍTULO 5.....	93
5. PRUEBAS EXPERIMENTALES Y ANÁLISIS DE RESULTADOS.....	93
5.1. PRUEBAS DE FUNCIONALIDAD	94
5.2. PRUEBAS DE ESTRÉS.....	94
5.3. REPORTE DE PRUEBAS DE FUNCIONALIDAD	98
5.3.1. Reporte de Pruebas Funcionales del Sistema Web.....	98

5.3.2. Reporte de Pruebas Funcionales de la Aplicación Android	106
5.4. REPORTE DE PRUEBAS DE ESTRÉS.....	109
CONCLUSIONES Y RECOMENDACIONES	114
GLOSARIO.....	118
BIBLIOGRAFÍA.....	119
ANEXOS.....	122
ANEXO 1	123
Pruebas de Funcionalidad del Sistema Web.....	123
ANEXO 2	142
Pruebas de Funcionalidad de la Aplicación Android	142

ÍNDICE DE FIGURAS

Figura 1.1 – ¿Dónde está el ATM más cercano?	3
Figura 1.2 – Aplicaciones Independientes VS UbiATM.	5
Figura 1.3 – Módulos de la Aplicación Móvil.	6
Figura 1.4 – Apache JMeter.....	14
Figura 2.1 – Arquitectura del SO Android.	20
Figura 2.2 – Casos de Uso del Sistema Web.....	33
Figura 0.1 - Casos de Uso de la Aplicación Android.	46
Figura 3.1 – Ubicación de los ATM.	53
Figura 3.2 - Esquema del Sistema Web.....	53
Figura 3.3 – Arquitectura del Sistema Web.....	54
Figura 3.4 – Modelo de Base de Datos.....	57
Figura 3.5 – Representación Semiversen.	60
Figura 3.6 – Esquema de la Aplicación Android.....	61
Figura 3.7 – Arquitectura de la Aplicación Android.....	63
Figura 3.8 – Arquitectura de dependencia para el servicio web.	67
Figura 3.9 – Arquitectura de dependencia para la aplicación móvil.....	68
Figura 3.10 – Ejecución de los Casos de Pruebas.....	69
Figura 3.11 – Porcentaje de Aceptación de Pruebas.	71
Figura 4.1 – Botón “Crear proyecto” en App Engine.	73
Figura 4.2 – Creación de un nuevo Proyecto en App Engine.....	73
Figura 4.3 – APIs y autenticación.	74

Figura 4.4 – APIs de Google Maps.	74
Figura 4.5 – Botón “Habilitar API”.	75
Figura 4.6 – Cuota Google Maps.	75
Figura 4.7 – Descarga de Codeigniter.	76
Figura 4.8 – Icono del archivo comprimido CodeIgniter.	76
Figura 4.9 – Modelo Vista Controlador.....	77
Figura 4.10 – config.php.	78
Figura 4.11 – Configuración del parámetro base_url	78
Figura 4.12 – Localización del archivo database.php.....	79
Figura 4.13 – Parámetros de conexión a la Base de Datos.	79
Figura 4.14 – Implementación de la fórmula Semiversenos.....	80
Figura 4.15 – Filtrado de la información por Banco o Redes de Cobro.	81
Figura 4.16 – Resultados Relevantes.	81
Figura 4.17 – Algoritmo búsqueda ATMs cercanos.....	81
Figura 4.18 – Permisos de la Aplicación.	83
Figura 4.19 – Obtención de Coordenadas Geográficas.	84
Figura 4.20 – Última ubicación obtenida por GPS.....	84
Figura 4.21 – Última ubicación por Wifi.....	85
Figura 4.22 – Creación requerimiento Bancos.	86
Figura 4.23 – Realizar llamada Http.....	86
Figura 4.24 – Procesar respuesta Bancos.	86
Figura 4.25 – Crear requerimiento Cajero.....	87
Figura 4.26 – Parámetros requerimiento Cajero.	87
Figura 4.27 – Descargar el Google APP Engine SDK.....	88

Figura 4.28 – Archivo para ejecutar Google APP Engine.....	88
Figura4.29 – Welcome to the Google App Engine.	89
Figura4.30 – Google App Engine Launcher.	89
Figura 4.31 – Creación del Proyecto.....	90
Figura 4.32 – Creación de nueva aplicación.	91
Figura 4.33 – Subir el sitio web a los servidores de APP Engine.	91
Figura 4.34 – Hacer Deploy de la aplicación a Google	92
Figura 5.1 – Peticiones enviadas simultáneamente.	97
Figura 5.2 – Petición de tipo Http.....	97
Figura 5.3 – Resultados luego de haber realizado las peticiones.....	110
Figura 5.4 – Resultados de la petición de tipo Http.	110
Figura 5.5 – Tiempo de respuesta y el tiempo medio según las pruebas.....	111
Figura 5.6 – Simulación con 200 usuarios simultáneos.....	112
Figura 5.7 – Tiempo de respuesta y tiempo medido en requerimientos/segundo..	113

ÍNDICE DE TABLAS

Tabla 1- Usuarios del Sistema Web.....	32
Tabla 2 – Listado de Casos de Uso del Sistema Web.	34
Tabla 3 – Caso de Uso 1.	35
Tabla 4 – Caso de Uso 2.	36
Tabla 5 – Caso de Uso 3.	37
Tabla 6 – Caso de Uso 4.	38
Tabla 7 – Caso de Uso 5.	39
Tabla 8 – Caso de Uso 6.	40
Tabla 9 – Caso de Uso 7.	41
Tabla 10 – Caso de Uso 8.	42
Tabla 11 – Caso de Uso 9.	43
Tabla 12 – Caso de Uso 10.	44
Tabla 13 – Usuario de la Aplicación Android.....	45
Tabla 14 – Listado de Casos de Uso de la Aplicación Android.	46
Tabla 15 – Caso de Uso 11.	47
Tabla 16 – Caso de Uso 12.	48
Tabla 17 – Caso de Uso 13.	49
Tabla 18 – Caso de Uso 14.	50
Tabla 19 – Ejecución de Pruebas Caso de Uso CU01.	99
Tabla 20 – Ejecución de Pruebas Caso de Uso CU02.	100
Tabla 21 – Ejecución de Pruebas Caso de Uso CU03.	100
Tabla 22 – Ejecución de Pruebas Caso de Uso CU04.	101

Tabla 23 – Ejecución de Pruebas Caso de Uso CU05.....	102
Tabla 24 – Ejecución de Pruebas Caso de Uso CU06.....	103
Tabla 25 – Ejecución de Pruebas Caso de Uso CU07.....	103
Tabla 26 – Ejecución de Pruebas Caso de Uso CU08.....	104
Tabla 27 – Ejecución de Pruebas Caso de Uso CU09.....	105
Tabla 28 – Ejecución de Pruebas Caso de Uso CU10.....	105
Tabla 29 – Ejecución de Pruebas Caso de Uso CU11.....	107
Tabla 30 – Ejecución de Pruebas Caso de Uso CU12.....	107
Tabla 31 – Ejecución de Pruebas Caso de Uso CU13.....	108
Tabla 32 – Ejecución de Pruebas Caso de Uso CU14.....	109
Tabla 33 – Diseño de Casos de Pruebas, Caso de Uso: CU01 Ingresar Banco....	123
Tabla 34 – Diseño de Casos de Pruebas, Caso de Uso: CU02 Consultar Banco.	126
Tabla 35 – Diseño de Casos de Pruebas, Caso de Uso: CU03 Editar Banco.	127
Tabla 36 – Diseño de Casos de Pruebas, Caso de Uso: CU04 Eliminar Banco....	130
Tabla 37 – Diseño de Casos de Pruebas, Caso de Uso: CU05 Ingresar Cajero. ...	131
Tabla 38 – Diseño de Casos de Pruebas, Caso de Uso: CU06 Consultar Cajero.	134
Tabla 39 – Diseño de Casos de Pruebas, Caso de Uso: CU07 Editar Cajero.....	136
Tabla 40 – Diseño de Casos de Pruebas, Caso de Uso: CU08 Eliminar Cajero. ...	139
Tabla 41 – Diseño de Casos de Pruebas, Caso de Uso: CU09 Subir Cajero.....	140
Tabla 42 – Diseño de Casos de Pruebas, Caso de Uso: CU10 Ubicar Cajero.....	141
Tabla 43 – Diseño de Casos de Pruebas, Caso de Uso: CU11 Elegir Banco.	142
Tabla 44 – Diseño de Casos de Pruebas, Caso de Uso: CU12 Elegir Redes asociadas.	143

Tabla 45 – Diseño de Casos de Pruebas, Caso de Uso: CU13 Consultar Cajeros cercanos por Banco.....	144
Tabla 46 – Diseño de Casos de Pruebas, Caso de Uso: CU14 Consultar Cajeros cercanos por Red Bancaria	145

ABREVIATURAS

API	Interfaz de Programación de Aplicaciones. (Application Programming Interface)
APP	Aplicación. (Application)
ATM	Cajero Automático. (Automatic Teller Machine)
BD	Base de Datos (Data Base)
GPS	Sistema de Posicionamiento Global (Global Positioning System)
HTTP	Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol)
MVC	Modelo Vista Controlador (Model View Controller)
NFC	Comunicación de Campo Cercano (Near Field Communication)
PC	Computadora Personal. (Personal Computer)
REST	Transferencia de Estado Representacional (Representational State Transfer)
SDK	Kit de Desarrollo de Software (Software Development Kit)
SO	Sistema Operativo (Operating System)
TI	Tecnología de la Información (Information Technology)
URL	Localizador de Recursos Uniforme (Uniform Resource Locator)
V	Versión (Version)
V4.0+	Versión 4.0 en adelante

INTRODUCCIÓN

Hoy en día el uso de teléfonos inteligentes ha tomado un rol importante para la sociedad debido a las prestaciones que brinda, este facilita diversas actividades que normalmente realiza el ser humano que requieren tiempo y dinero. Por esto, el desarrollo y uso de APPs móviles se ha incrementado, ya que a través de ellas se simplifica el acceso e interacción con los servicios de empresas a los usuarios.

El uso de APPs móviles bancarias ha ido tomando ventaja debido a su rapidez y eficiencia inmediata desde un dispositivo móvil. Lo que resulta relativamente más sencillo que interactuar desde una PC o desde el propio banco.

Haciendo una revisión rápida de las APPs móviles bancarias disponibles en la “Google Play Store”, nos damos cuenta que en su mayoría no cuentan con un servicio de localización de ATMs. Por lo cual, es factible realizar una aplicación para usuarios de la banca móvil de la ciudad de Guayaquil que brinde esta función.

El objetivo principal del proyecto es implementar un servicio de localización de ATMs en la ciudad de Guayaquil desde un dispositivo móvil con SO Android V4.0 en adelante, presentándola en una interfaz amigable con el usuario que muestra un mapa de Google Maps previamente descargado y configurado mediante su API para Android, además de una base de datos enlazada donde se encuentra almacenada la información de cada ATM.

En el capítulo uno de este proyecto se describe el estudio que se hace referente a las aplicaciones móviles que brindan las diferentes entidades bancarias, las funcionalidades que estas brindan, y las que aún no se han implementado, para así ofrecer una solución a ese determinado problema.

En el capítulo dos se realiza una introducción al SO Android, sus diferentes versiones, y las características que brinda cada una. Se describe también los casos de uso, siendo estos los diferentes escenarios que pueden presentarse al momento de hacer uso de la aplicación. Además se describen los conceptos de: Google App Engine, Framework, Base de Datos, Servicios Web, Apache JMeter. Finalmente se presenta el análisis de la solución.

El capítulo tres se enfoca en los pasos previos al desarrollo de la aplicación, siendo estos: recopilación de la información de cada ATM para ser ingresado en la base de datos, creación del esquema del sitio web, su arquitectura, el modelo de la base de datos, implementación del algoritmo de búsqueda de puntos cercanos, y el esquema y la arquitectura de la aplicación móvil.

Finamente se describe el diseño de las pruebas que se realizaran al sistema web y la aplicación móvil un vez implementados.

En el capítulo cuatro se describen las instalaciones y configuraciones que se realizaron antes de iniciar con el desarrollo de la aplicación, la obtención del API KEY de Google, la instalación del Framework, y el desarrollo como tal de la aplicación.

El capítulo cinco describe los casos de prueba, necesarios para corregir fallas del sistema, además de las pruebas de estrés y funcionalidad que se realizaron a los servicios que brinda el sistema. Finalmente se muestran las conclusiones y recomendaciones que se hacen una vez terminado el desarrollo del sistema.

CAPÍTULO 1

1. ANTECEDENTES Y JUSTIFICACIÓN

Este capítulo presenta una descripción global del proyecto, así como también la importancia e impacto que puede tener su aplicación en la sociedad, ya que muchas veces las personas se encuentran con la necesidad de realizar retiros de dinero y desconocen la ubicación de algún cajero por lo que se hacen la siguiente pregunta ¿Dónde quedará el Cajero Automático más cercano?

Para responder a esta pregunta se propone una aplicación que nos permita localizar el cajero automático de nuestro banco de preferencia

más cercano, tomando como referencia nuestra ubicación. Debido a que se requiere que dicha aplicación sea móvil nos ayudaremos de la API que nos ofrece el sistema operativo Android para capturar la ubicación real del usuario y estimar la ubicación del cajero más cercano.

Además, en este capítulo se detallan las bases sobre las cuales se justifica el proyecto y el aporte que este brinda a la sociedad mediante el uso de las tecnologías de información. Así mismo se describen los objetivos generales y específicos del proyecto.

1.1. DESCRIPCIÓN DEL PROBLEMA

Hoy en día es muy frecuente encontrar por las calles de Guayaquil a personas interesadas en conocer la ubicación de los cajeros automáticos más cercanos a su localización, del Banco de su preferencia, como se muestra en la Figura 1.1, si bien es cierto algunas entidades Bancarias tienen a disposición del público aplicaciones que permiten localizar sus cajeros, resulta poco eficiente tener que instalar la aplicación de cada una de las entidades bancarias en el celular o tableta.



Figura 1.1 – ¿Dónde está el ATM más cercano?

Debido al avance de la tecnología móvil, se ha vuelto indispensable brindar facilidades a los usuarios de la Banca, por lo que se implementan aplicaciones móviles para la localización de ATMs con el fin de cumplir este objetivo.

El uso de servicios Bancarios cada día es más común en los dispositivos móviles, debido a que ofrece las mismas ventajas de las PCs tradicionales pero con un nivel de portabilidad extremadamente alto. Con la ayuda del API de Google Maps podemos trasladar la gran funcionalidad y utilidad de Google Maps a nuestras aplicaciones para darle facilidad al usuario al momento de buscar los ATMs.

Actualmente en la ciudad de Guayaquil, a pesar de que la mayoría de los Bancos cuentan con una aplicación para teléfonos

inteligentes, el servicio de ATMs sigue siendo imprescindible, ya que por este medio los usuarios tienen apertura a su dinero. Por lo cual, es muy importante saber la ubicación de los ATMS al momento de proceder a realizar transacciones tales como retiros, pago de servicios, consulta de saldos, transferencias bancarias entre otras.

Según las investigaciones realizadas, las aplicaciones móviles de la Banca de la ciudad de Guayaquil, en su gran mayoría, no ofrecen la ubicación geográfica del ATM más cercano al lugar donde se encuentra el usuario o a una coordenada proporcionada por los mismos.

1.2. SOLUCIÓN PROPUESTA

Como parte integral de la solución se plantea la implementación de un sistema dirigido a los usuarios de la Banca que integre las ubicaciones de los cajeros automáticos de la ciudad de Guayaquil, con el fin de presentar de manera gráfica por medio del uso de mapas la ubicación de cada uno de los ATMs.

En la Figura 1.2 se hace referencia al sistema tradicional para ubicar cajeros por cada banco, VS la búsqueda mediante la aplicación UbicATM.



Figura 1.2 – Aplicaciones Independientes VS UbicATM.

Para llevar a cabo este proyecto se ha dividido en 3 módulos principales:

Módulo 1: Geolocalización, es el encargado de buscar y establecer las coordenadas de longitud y latitud de la persona que realiza el requerimiento.

Módulo 2: Recolección de datos, es aquel que se encarga de almacenar la información de cada una de las entidades bancarias tales como la ubicación geográfica de cada uno de sus cajeros automáticos y estimar la distancia más corta entre la posición del usuario y los cajeros.

Módulo 3: visualización de resultados, este permitirá mostrar la ubicación de los cajeros de manera gráfica mediante el uso de mapas.



Figura 1.3 – Módulos de la Aplicación Móvil.

La Figura 1.3 muestra la interacción entre los módulos que conforman la aplicación, donde todo empieza con el requerimiento

del usuario, luego se procede a estimar la ubicación del mismo, estas coordenadas son los valores de entrada al módulo de recolección de datos el cual se encarga de comparar y estimar la ruta más corta entre los cajeros y el usuario, para finalmente con la ayuda de la API de Google Maps visualizar la información en el mapa.

1.3. JUSTIFICACIÓN DEL PROYECTO

Con este proyecto se espera potenciar el uso de las Tecnologías de Información aprovechando los bajos costos que estos representan, ya que es posible utilizar estas herramientas para apoyar con tecnología procesos que llevan a cabo las personas en su diario vivir.

Los resultados generados por el sistema con el uso de la tecnología le brindarán al usuario una herramienta de navegación en los distintos lugares de la ciudad de Guayaquil, permitiéndole al usuario ubicarse y dirigirse al cajero automático de su banco de preferencia más cercano.

Este proyecto fomentará la investigación en ESPOL sobre las TI para que futuras generaciones desarrollen nuevos proyectos

basados en éstas, que contribuyan al mejoramiento de la calidad de vida de las personas.

Gracias a este proyecto, el servicio Bancario móvil podría funcionar de manera más eficiente si se implementa una opción de localización de ATMs que brinde las facilidades de búsquedas por: nombre del Banco, Bancos en general o Red Bancaria, la gran ventaja que tenemos al usar una aplicación de Localización de ATMs es el confort que brinda a los usuarios ya que no tienen que consultar en sitios web o guías telefónicas para encontrar el ATM cercano a la ubicación deseada.

1.4. OBJETIVOS

1.4.1. Objetivos Generales

Desarrollar una Aplicación Multimedia Interactiva que permita al usuario, mediante su dispositivo móvil Android v4.0 en adelante, localizar el Cajero Automático Bancario más cercano a su ubicación proporcionada dentro de Guayaquil.

1.4.2. Objetivos Específicos

- Determinar y almacenar la ubicación de los Cajeros Automáticos.
- Descargar y configurar el API de Google Maps para Android.
- Desarrollar la APP con una interfaz amigable con el usuario.
- Añadir e interactuar con el mapa en la aplicación
- Desarrollar los servicios de búsqueda para la localización de Cajeros Automáticos.
- Mostrar el mapa de la ciudad de Guayaquil, situando la ubicación establecida por el usuario y de los cajeros automáticos más cercanos a este.

1.5. ALCANCE DEL PROYECTO

Este proyecto permitirá buscar la o las ubicaciones de los cajeros automáticos pertenecientes a Bancos ubicados en la ciudad de Guayaquil, quedando por fuera Instituciones de crédito, Cooperativas o Bancos que no cuenten con el servicio de ATMs.

A continuación se presenta el listado de Instituciones Bancarias que fueron consideradas en la aplicación del proyecto:

- Banco de Guayaquil
- Banco del Pacífico
- Banco Bolivariano
- Banco Pichincha
- Banco Solidario
- Banco General Rumiñahui
- Banco Produbanco – Grupo Promerica.
- Banco del Austro
- Banco Procredit
- Banco Machala
- Banco Internacional
- Banco del Bank

Los resultados obtenidos por la aplicación podrán ser filtrados por banco o redes asociadas tales como:

- Banred
- Pacificard
- American Express
- Bankard
- Nexo
- Visa Débito
- Visa
- Plus
- MasterCard

- Cirrus
- Maestro
- Diners Club

1.6. METODOLOGÍA

La primera etapa para la elaboración de este proyecto consiste en la identificación del problema o necesidad seguida de la solución propuesta, luego de esto se procede a revisar las diferentes herramientas tecnológicas que pueden ayudar en la elaboración del mismo, una vez definidos estos puntos se sigue el procedimiento detallado a continuación:

1.6.1. Determinar la ubicación de los Cajeros Automáticos

Para determinar la ubicación de los ATMs de la ciudad de Guayaquil, se procederá primeramente a verificar en la página web de cada Institución Bancaria si existe información de los mismos, tal como su dirección o su ubicación geográfica.

En el caso de las Instituciones que no cuenten con dicha información en su sitio web, acudiremos a las matrices de estos Bancos para consultar las direcciones de sus ATMs.

Una vez obtenida esta información, en el caso de los ATMs que contamos con su dirección, identificaremos una por una en Google Maps, para obtener su latitud y longitud, ya que dichos datos serán ingresados en nuestra BD.

Estos datos servirán para localizar los ATMs dentro del mapa de Google Maps en la aplicación a desarrollar y de esta forma se ayudará al usuario a encontrar el más cercano a su ubicación.

1.6.2. Configurar el Servidor Web

Esta etapa considera la configuración de la cuenta para utilizar los servicios de Cloud Computing ofrecidos por Google bajo la plataforma App Engine, la cual permitirá alojar nuestros servicios y almacenar la información de bancos y cajeros automáticos, para esto se utilizará tecnología php y mysql.

1.6.3. Diseñar y Desarrollar la Aplicación

Esta etapa comprende el diseño de interfaces de comunicación entre la aplicación móvil y los servicios, así como también la implementación de la aplicación móvil para

Android haciendo uso del SDK provisto por Google y el API de Google maps.

La aplicación se conectará con nuestros servicios web para suministrar la información de los cajeros más cercanos y será optimizada para que le sea de fácil uso al usuario. Hay que tener en cuenta que se compilará la APP para ser utilizado por versiones de Android superiores a la V4.0.

1.6.4. Someter a Pruebas la Aplicación

Una vez finalizado el desarrollo de la aplicación esta se someterá a pruebas internas, para esto se llevaran a cabo pruebas cruzadas, donde el resultado de dichas pruebas servirá para la depuración y corrección de defectos que presente la aplicación, este tipo de pruebas constará de dos ciclos.

Una vez finalizadas las pruebas internas se procederá a las pruebas funcionales o pruebas con usuarios en donde se tendrá una retroalimentación más real del funcionamiento de la aplicación y servirá para realizar mejoras o corrección de errores.

Para asegurar la calidad y alta disponibilidad de la aplicación se someterán los componentes principales de la aplicación a pruebas de estrés, las cuales consisten en llevar al sistema al punto más alto de su rendimiento, incluyendo los servicios web que consume la aplicación, estas pruebas serán realizadas con la ayuda de la herramienta Jmeter, ilustrada en la Figura 1.4.



Figura 1.4 – Apache JMeter

CAPÍTULO 2

2. MARCO TEÓRICO Y ANÁLISIS DE LA SOLUCIÓN

En este capítulo se muestran los requerimientos con los que se realizará el proyecto así como una breve introducción al SO Android, sus diferentes versiones y formas de comunicación con servidores de datos.

2.1. REQUERIMIENTOS

Los requerimientos que tomaremos en cuenta en este proyecto, pueden dividirse en:

- Operativos
- Funcionales

- No Funcionales
- Físicos

En los requerimientos operativos destacamos el uso de la versión de Android escogida para la implementación de la Aplicación Móvil llamada “UbicATM”.

En los requerimientos funcionales tenemos el uso de los servidores “Google App Engine” para hospedar el sistema web que utilizará “UbicATM”, en conjunto con el framework, base de datos y servicios web a utilizar para agilizar la implementación y ejecución del sistema.

En los requerimientos funcionales en la aplicación y en el Sistema Web, se tienen:

- El administrador del Sitio Web puede ingresar un banco al sistema.
- El administrador del Sitio Web puede consultar los bancos que se encuentran ingresados al sistema.
- El administrador del Sitio Web puede editar uno o más bancos existentes en el sistema.
- El administrador del Sitio Web puede eliminar uno o más bancos existentes en el sistema.

- El administrador del Sitio Web puede ingresar un nuevo cajero al sistema.
- El administrador del Sitio Web puede consultar los cajeros existentes en el sistema.
- El administrador del Sitio Web puede editar uno o más cajeros existentes en el sistema.
- El administrador del Sitio Web puede eliminar uno o más cajeros existentes en el sistema.
- El administrador del Sitio Web puede ingresar al sistema un listado de cajeros desde un archivo externo.
- El administrador del Sitio Web puede establecer la ubicación dentro del mapa de uno o más cajeros.
- El usuario móvil puede elegir el banco del cual desea consultar los cajeros más cercanos a su ubicación.
- El usuario móvil puede elegir las redes asociadas para las cuales desea consultar los cajeros más cercanos a su ubicación.
- El usuario móvil puede consultar los cajeros cercanos a su ubicación en base al Banco seleccionado.
- El usuario móvil puede consultar los cajeros cercanos a su ubicación en base a la red bancaria seleccionada.

En los requerimientos no funcionales en la aplicación y en el Sistema Web, se tienen:

- La aplicación móvil funciona en Android v 4.0 o superior.
- Para la consulta, ingreso, edición y eliminación de los datos, se hace uso de un sistema web.
- Al ingresar un nuevo Banco al sistema, éste debe contener un nombre, sin caracteres especiales.
- Al ingresar un nuevo Cajero al sistema, éste debe contener un nombre sin caracteres especiales, además de especificar su ubicación en coordenadas geográficas, el Banco y la red bancaria a la que pertenece.
- Para ingresar un nuevo Cajero, previamente debe existir el Banco al cual pertenece.
- La ubicación de un Cajero dentro del mapa se puede modificar.
- El tiempo de respuesta cuando se realicen más de 1000 peticiones, no debe ser mayor de 2 segundos.

En los aspectos físicos, nos enfocamos en el hardware. El proyecto estará conformado por dos laptops, las cuales servirán para el

desarrollo de “UbicATM” y un dispositivo móvil donde se procederá a descargar y probar la APP. Dicho dispositivo cuenta con SO Android V4.0+, servicio de internet y GPS.

2.2. ANDROID

Android es un sistema operativo de software libre basado en Linux, multiplataforma y gratuito [16]. Proporciona la interfaz necesaria para desarrollar APPs que accedan a las funciones del dispositivo móvil por medio de Java, que se compilan en un archivo APK y corren en Dalvik, esta es una máquina virtual diseñada para Android y especializada para todo tipo de dispositivos móviles con poca memoria como teléfonos o tabletas.

Entre sus características importantes se destaca que este SO es completamente libre, es decir que no implica un costo, lo cual lo vuelve bastante popular entre fabricantes y desarrolladores. Es de código abierto para sus usuarios, puede ser descargado, inspeccionado, compilado e incluso cambiado, permitiendo así adaptar el SO a distintos dispositivos.

La arquitectura del SO esta descrita por la figura 2.1:

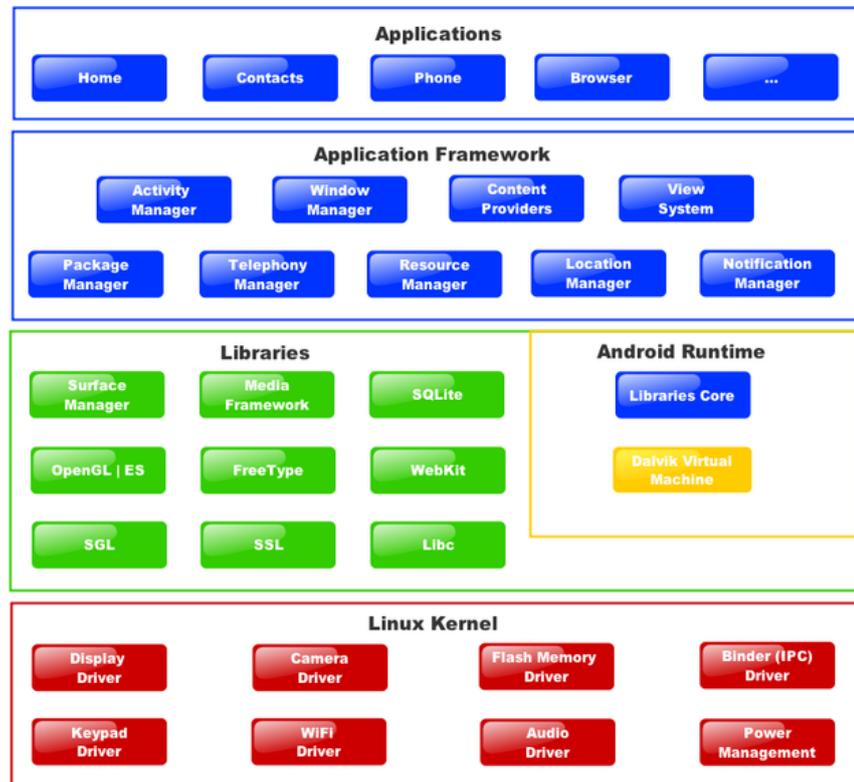


Figura 2.1 – Arquitectura del SO Android.

2.2.1. Versiones de Android

A la fecha del desarrollo de la presente APP, Android cuenta con las siguientes versiones, lanzadas al mercado desde el 2008:

Android 1.0 Apple Pie

Esta primera versión fue lanzada en el 2008, se presenta como un sistema operativo móvil totalmente gratuito y Open Source. Estaba desarrollado sobre el kernel de Linux

2.6. Entre sus características tenemos:

- Menú desplegable de notificaciones
- Widgets de escritorio
- Android Market
- Navegador, Maps, Google Talk, reproductor de YouTube y soporte para cámaras [1].

Android 1.1 Banana Bread

Esta versión de Android lanzada en febrero de 2009, llegó para los dispositivos T-Mobile G1, la cual presentó las siguientes características:

- Soporte para marquesina en diseños de sistemas
- Reseñas al buscar negocios en los mapas
- Guardar archivos adjuntos en los mensajes [2].

Android 1.5 Cupcake

Esta versión de Android lanzada en abril de 2009 tuvo una gran acogida, gracias a una serie de características que mostró, las cuales fueron de gran impacto para sus usuarios.

Entre ellas tenemos:

- Paquete de widgets para la pantalla de inicio; reloj analógico, calendario, reproductor de música, marco de imágenes, y búsqueda.
- Autorrotación
- Teclado virtual en pantalla [3].

Android 1.6 Donut

Presentó mejoras a la versión Cupcake y fue de gran ayuda para los dispositivos móviles que estaban restringidos por su hardware para tener versiones renovadas. Entre sus características tenemos:

- Mejoras en el Android Market
- Pantalla para controlar la batería
- Actualización constante del kernel de Linux
- Motor de texto a voz [4].

Android 2.1 Eclair

Esta versión fue lanzada en octubre de 2009 presentando las siguientes mejoras a su versión antecesora:

- Mejoras en el teclado virtual.
- Galería 3D.
- Soporte para más resoluciones de pantalla.
- Google Maps multitáctil y soporte de capas [5].

Android 2.2 Froyo

Presentada como una mejora a su versión precedente, fue lanzada en Mayo de 2010 con las siguientes características:

- Actualizaciones automáticas.
- Soporta Adobe Flash
- Permite crear un punto de acceso Wifi a partir de nuestra conexión 3G.
- Soporte para HTML5 [6].

Android 2.3 Gingerbread

Lanzada en diciembre de 2010 destacando una mejor velocidad dentro de la interfaz, además de las siguientes características:

- Soporte para pantallas XWGA
- Soporte para telefonía por Volp
- Soporte para Near Field Communications
- Selección de texto con un toque y copiar/pegar [7].

Android 3.2 Honeycomb

En febrero de 2011 se presentó la mejora de la versión anterior de Android, destacando la conectividad USB y el soporte para teclados y gamepads, además de las siguientes características:

- Escritorio 3D con widgets rediseñados
- Sincronización de favoritos en Google Chrome
- Mejor soporte para redes Wi-Fi
- Navegación por pestañas
- Multitarea simplificada [8].

Android 4.0 Ice CreamSandwich

Esta versión fue lanzada en octubre de 2011. Siendo declarado que Android 4.0 era teóricamente compatible con cualquier dispositivo Android 2.3 en producción en ese momento, pero sólo si su procesador y memoria ram lo soportan.

Presentando las siguientes mejoras a su versión antecesora:

- Captura de pantalla integrada
- Desbloqueo facial
- WiFiDirect
- Sincronización automática del navegador con los marcadores de Chrome del usuario.
- Habilidad de acceder a aplicaciones directamente desde la pantalla de bloqueo [9].

Android 4.1 JellyBean

Esta versión fue lanzada en junio de 2012, acompaña al Nexus 7 y cuenta con Google Now[10] además de las siguientes características:

- Crear texto a partir de la voz
- El teclado Android se renueva y se añade la escritura por gestos.
- Diseñado para permitir a los discapacitados visuales, manipular su dispositivo.
- Transferir video por medio de la conexión NFC [11].

Android 4.4 KitKat

Presentó mejoras a la versión JellyBean, entre sus características tenemos:

- Impresión de documentos mediante WiFi
- Detector y contador de pasos
- Crear videos de alta calidad [12].

Android 5.0 Lollipop

Esta versión fue lanzada en junio de 2014, entre sus características tenemos:

- Las notificaciones pueden aparecer en la pantalla de bloqueo
- Conectividad más rápida
- Mejora en el sonido [13].

2.2.2. Android SDK

Es un kit de desarrollo de software, compuesto por herramientas y programas, mediante el cual podemos desarrollar aplicaciones y ejecutarlas en un emulador [14].

El SDK de Android incluye lo siguiente:

- Bibliotecas necesarias
- Depurador

- Emulador
- Documentación pertinente para las interfaces de programación de aplicaciones para Android (API)
- Código fuente de la muestra
- Tutoriales para el SO Android

2.3. GOOGLE APP ENGINE

Google App Engine es una plataforma que permite alojar un sitio web sobre los servidores de “Google”, siendo esto de forma gratuita y permite utilizar hasta 500 MB de almacenamiento permanente y suficiente ancho de banda y CPU para casi cinco millones de visitas mensuales [18].

Entre los lenguajes de programación compatibles con este servicio tenemos: Python, Java, Php y Go, además de tener compatibilidad para bases de datos MySql.

Lo interesante de esta plataforma es que no debemos preocuparnos por configuraciones o rendimiento de servidores ya que Google administra esto y en caso de pasarnos de la cuota se deben pagar costos bajos.

2.4. FRAMEWORK

Un Framework es un conjunto de librerías y funciones que agilizan la implementación de un sistema. Para efecto de nuestra aplicación haremos uso del Framework “Codeigniter” escrito en lenguaje Php.

Entre sus ventajas tenemos:

- Esquema Modelo, Vista y Controlador
- Manejo de Sesiones
- Clases para manejo de base de datos

2.5. BASE DE DATOS

Las Bases de datos son colecciones de información organizadas de tal forma que pueden ser fácilmente accedidas y modificadas [19].

En nuestra aplicación hemos hecho uso del gestor de base de datos MySQL ya que se ajusta a nuestras necesidades y se lo puede usar de forma libre en cualquier proyecto.

2.6. SERVICIOS WEB

Son una tecnología que nos permite la comunicación entre dispositivos electrónicos por medio de una red de datos utilizando protocolos y estándares para intercambiar información.

La arquitectura más utilizada en los últimos años para lograr la comunicación entre dispositivos es conocida como REST, este se

basa totalmente en el protocolo HTTP. Lo que permite ser usada por cualquier cliente o dispositivo.

Los métodos disponibles dentro de la arquitectura REST son los mismos que se utilizan el protocolo HTTP así:

- GET: se lo utilizará para operaciones de consulta
- POST: operaciones de inserción
- DELETE: operaciones de eliminación

2.7. APACHE JMETER

La aplicación Apache JMeter Desktop fue diseñada para llevar a cabo pruebas de comportamiento funcional y medir el rendimiento.

Puede ser usada para medir el rendimiento en recursos estáticos como dinámicos. Se puede utilizar para hacer un análisis gráfico de rendimiento o para probar el comportamiento de Scripts y Objetos de Servidor bajo carga pesada concurrente [15].

2.8. ANÁLISIS DE LA SOLUCIÓN

Para llevar a cabo el proyecto de la mejor manera posible hemos decidido usar las siguientes soluciones:

Se buscará mediante los sitios web de cada Institución Bancaria la ubicación de sus cajeros en la ciudad de Guayaquil, en caso de no existir se procederá a consultar la información en la matriz de cada Institución.

Se desarrollará un sistema web en PHP haciendo uso del Framework "Codeigniter" y base de datos MySQL, desde este sistema se podrá ingresar y consultar la información de bancos y cajeros de Guayaquil.

Se subirá nuestro sistema web a la plataforma "Google App Engine" desde la cual se podrá acceder vía internet al sistema web y sus servicios.

La información recopilada de las Instituciones Bancarias será cargada a nuestro sistema web por medio de archivos de Excel donde se encuentren los datos de cada cajero (Banco, Dirección, Redes Bancarias Asociadas, Ubicación).

Desarrollaremos el algoritmo que nos permita encontrar los cajeros más cercanos a un punto (latitud y longitud) dado, este algoritmo

será utilizado para desarrollar el servicio REST que use nuestra aplicación Android en la consulta de cajeros.

Se implementará la aplicación móvil “UbicATM” sobre la versión 4.0 de Android, sobre la cual se ejecutarán las respectivas búsquedas de los ATMs de la ciudad de Guayaquil, ya sea por Banco o por Red Bancaria.

Para llevar a cabo todo este proceso planteamos Casos de Uso, tanto para el Sistema Web, como para la Aplicación Android.

Casos de Uso del Sistema Web

Los casos de uso son una forma de representar los requerimientos del sistema y describir los pasos o actividades que deben realizarse para llevar a cabo un proceso.

Antes de definir los casos de uso se deben definir a los actores o usuarios que interactúan con el sistema.

Usuarios del sistema

Tabla 1- Usuarios del Sistema Web.

Usuario	Descripción
Administrador	Este usuario podrá administrar todas las opciones del sistema.

Diagrama de Casos de Uso

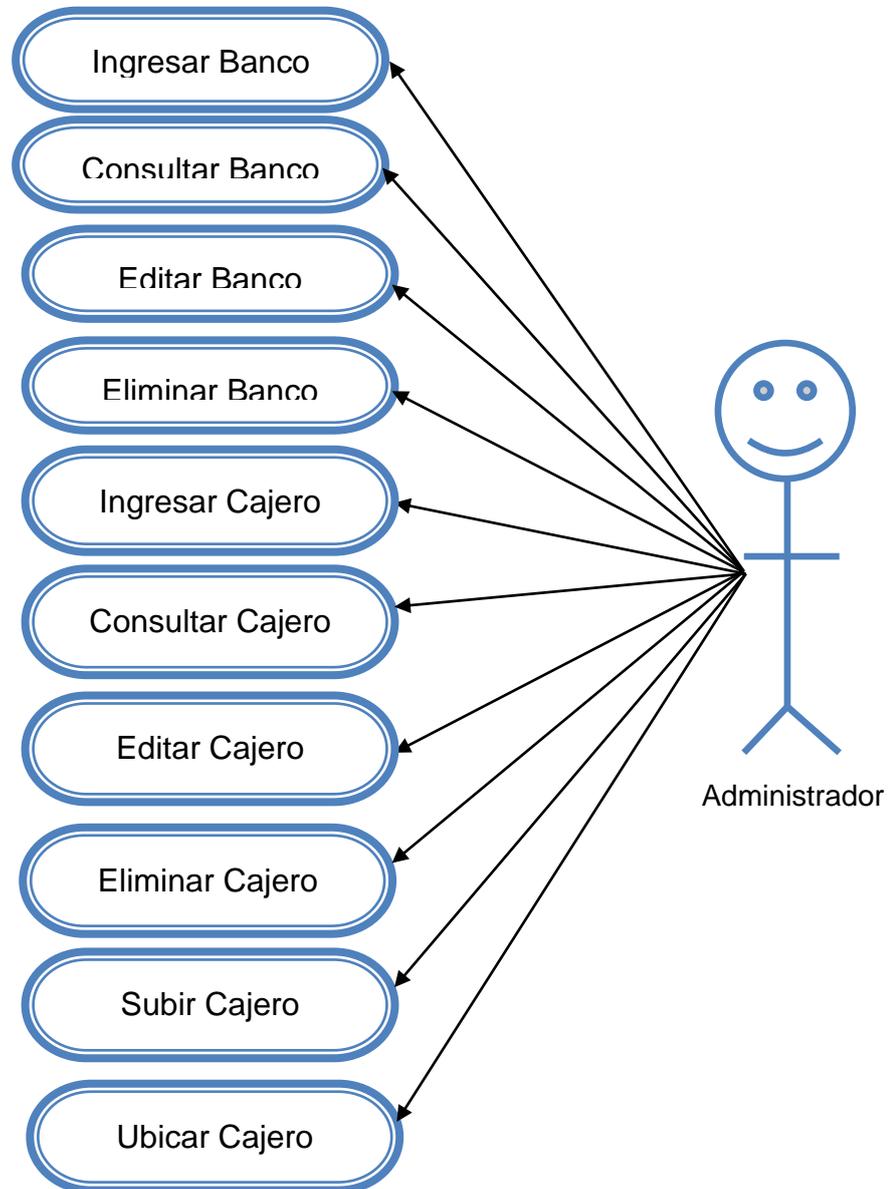


Figura 2.2 – Casos de Uso del Sistema Web.

Listado de Casos de Uso

Tabla 2 – Listado de Casos de Uso del Sistema Web.

ID	Casos de Uso	Administrador
CU01	Ingresar Banco	X
CU02	Consultar Banco	X
CU03	Editar Banco	X
CU04	Eliminar Banco	X
CU05	Ingresar Cajero	X
CU06	Consultar Cajero	X
CU07	Editar Cajero	X
CU08	Eliminar Cajero	X
CU09	Subir Cajero	X
CU10	Ubicar Cajero	X

A continuación se detalla cada caso de uso:

Tabla 3 – Caso de Uso 1.

ID	CU01
Nombre	Ingresar Banco.
Pantalla	
Descripción	<p>Esta opción nos permite Ingresar Banco, es decir crear un nuevo Banco con su nombre y estado, el cual puede ser activo o inactivo. Automáticamente el sistema asignará un ID a cada Banco.</p>
Rol	Administrador.
Pre-Condicion	No aplica.
Acciones	<p>Escoger la opción: Banco.</p> <p>Escoger la opción: Nuevo.</p> <p>Llenar el campo nombre.</p> <p>Seleccionar el estado.</p> <p>Guardar.</p> <p>El sistema almacenará el nuevo Banco.</p>

Tabla 4 – Caso de Uso 2.

ID	CU02																																																				
Nombre	Consultar Banco.																																																				
Pantalla	<p>Consultar Bancos</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Nombre</th> <th>Estado</th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pichincha</td> <td>AC</td> <td> </td> </tr> <tr> <td>2</td> <td>Bolivariano</td> <td>AC</td> <td> </td> </tr> <tr> <td>3</td> <td>Pacífico</td> <td>AC</td> <td> </td> </tr> <tr> <td>4</td> <td>Guayaquil</td> <td>AC</td> <td> </td> </tr> <tr> <td>5</td> <td>Produbanco</td> <td>AC</td> <td> </td> </tr> <tr> <td>7</td> <td>Austro</td> <td>AC</td> <td> </td> </tr> <tr> <td>8</td> <td>ProCredit</td> <td>AC</td> <td> </td> </tr> <tr> <td>9</td> <td>Machala</td> <td>AC</td> <td> </td> </tr> <tr> <td>10</td> <td>Solidario</td> <td>AC</td> <td> </td> </tr> <tr> <td>11</td> <td>Ruminahui</td> <td>AC</td> <td> </td> </tr> <tr> <td>12</td> <td>Banco Del Bank</td> <td>AC</td> <td> </td> </tr> <tr> <td>13</td> <td>Internacional</td> <td>AC</td> <td> </td> </tr> </tbody> </table>	ID	Nombre	Estado		1	Pichincha	AC	 	2	Bolivariano	AC	 	3	Pacífico	AC	 	4	Guayaquil	AC	 	5	Produbanco	AC	 	7	Austro	AC	 	8	ProCredit	AC	 	9	Machala	AC	 	10	Solidario	AC	 	11	Ruminahui	AC	 	12	Banco Del Bank	AC	 	13	Internacional	AC	 
ID	Nombre	Estado																																																			
1	Pichincha	AC	 																																																		
2	Bolivariano	AC	 																																																		
3	Pacífico	AC	 																																																		
4	Guayaquil	AC	 																																																		
5	Produbanco	AC	 																																																		
7	Austro	AC	 																																																		
8	ProCredit	AC	 																																																		
9	Machala	AC	 																																																		
10	Solidario	AC	 																																																		
11	Ruminahui	AC	 																																																		
12	Banco Del Bank	AC	 																																																		
13	Internacional	AC	 																																																		
Descripción	Al seleccionar esta opción, podremos Consultar Bancos , en donde se podrá visualizar la lista de Bancos antes creados, junto con su ID y estado.																																																				
Rol	Administrador.																																																				
Pre-Condicion	Que exista la información de los Bancos.																																																				
Acciones	<p>Escoger la opción: Banco.</p> <p>Escoger la opción: Consultar.</p> <p>El sistema mostrará los Bancos del Sistema.</p>																																																				

Tabla 5 – Caso de Uso 3.

ID	CU03
Nombre	Editar Banco.
Pantalla	
Descripción	Esta opción nos permitirá modificar el nombre o estado del Banco.
Rol	Administrador.
Pre-Condicion	Que exista el Banco.
Acciones	<p>Escoger la opción: Banco.</p> <p>Escoger la opción: Consultar.</p> <p>Escoger el Banco que se desea editar dando clic en el botón .</p> <p>Modificar el campo nombre.</p> <p>Modificar el campo estado.</p> <p>Guardar.</p> <p>El sistema almacenará los cambios del Banco.</p>

Tabla 6 – Caso de Uso 4.

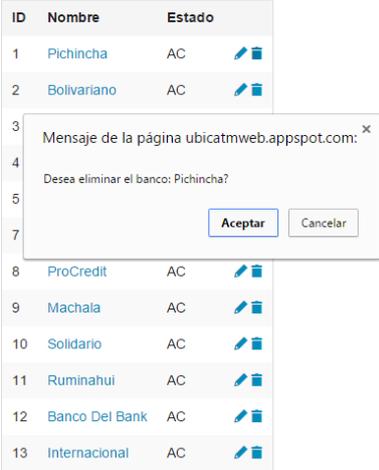
ID	CU04
Nombre	Eliminar Banco.
Pantalla	<p>Consultar Bancos</p>  <p>The screenshot shows a table with columns 'ID', 'Nombre', and 'Estado'. The table lists 13 banks: Pichincha, Bolivariano, ProCredit, Machala, Solidario, Ruminahui, Banco Del Bank, and Internacional. A confirmation dialog box is overlaid on the table, asking 'Desea eliminar el banco: Pichincha?' with 'Aceptar' and 'Cancelar' buttons.</p>
Descripción	Esta opción nos permitirá eliminar un Banco.
Rol	Administrador.
Pre-Condicion	Que exista el Banco.
Acciones	<p>Escoger la opción: Banco.</p> <p>Escoger la opción: Consultar.</p> <p>Escoger el Banco que se desea eliminar dando clic en el botón .</p> <p>Dar clic en el botón Aceptar.</p> <p>El sistema eliminará los datos del Banco.</p>

Tabla 7 – Caso de Uso 5.

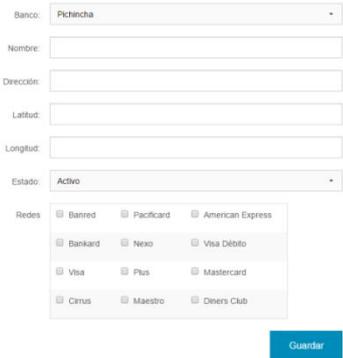
ID	CU05
Nombre	Ingresar Cajero.
Pantalla	<p>Ingresar Cajero</p> 
Descripción	Esta opción nos permite Ingresar Cajero de forma manual.
Rol	Administrador.
Pre-Condicion	Que exista el Banco al que pertenece el Cajero.
Acciones	<p>Escoger la opción: Cajero.</p> <p>Escoger la opción: Nuevo.</p> <p>Seleccionar el Banco al que pertenece el cajero.</p> <p>Llenar el campo nombre.</p> <p>Llenar el campo dirección.</p> <p>Llenar el campo latitud.</p> <p>Llenar el campo longitud.</p> <p>Llenar el campo estado.</p> <p>Guardar.</p> <p>El sistema almacenará los datos del Cajero.</p>

Tabla 8 – Caso de Uso 6.

ID	CU06																										
Nombre	Consultar Cajero.																										
Pantalla	<p>Consultar Cajeros</p> <p>Nombre: <input type="text" value="Guayaquil"/></p> <table border="1"> <thead> <tr> <th>Nombre</th> <th>Dirección</th> </tr> </thead> <tbody> <tr> <td>9 De Octubre 1</td> <td>9 Octubre 1404 Y Machala</td> </tr> <tr> <td>Aeropuerto Internacional</td> <td>Av. De Las Américas, Aeropuerto J. J. De Olmedo</td> </tr> <tr> <td>Ag Boulevard 1</td> <td>9 De Octubre Entre García Avilés Y Ramichaca</td> </tr> <tr> <td>Ag Rio Centro Sur</td> <td>Cc Rocentro Sur, Avenida 25 De Julio</td> </tr> <tr> <td>Ag Nueva Portete 1</td> <td>Portete Y La 15</td> </tr> <tr> <td>Agencia Pascuales</td> <td>Av. Principal De Pascuales</td> </tr> <tr> <td>Aguirre 1</td> <td>Aguirre 623 Y García Avilés</td> </tr> <tr> <td>Aje Ecuador</td> <td>Km. 15.5 Vía A Daule</td> </tr> <tr> <td>AKI 25 De Julio 2</td> <td>Av. 25 De Julio, Santa Mónica, Av. 25 De Julio Y Calle 51C</td> </tr> <tr> <td>AKI Astillero</td> <td>Chile Y San Martín Esquina</td> </tr> <tr> <td>AKI Atarazana</td> <td>Cdla. Atarazana</td> </tr> <tr> <td>AKI Domingo Comin 2</td> <td>Avenida 10Se (Monseñor Domingo Comin) Y Calle 55 Se</td> </tr> </tbody> </table>	Nombre	Dirección	9 De Octubre 1	9 Octubre 1404 Y Machala	Aeropuerto Internacional	Av. De Las Américas, Aeropuerto J. J. De Olmedo	Ag Boulevard 1	9 De Octubre Entre García Avilés Y Ramichaca	Ag Rio Centro Sur	Cc Rocentro Sur, Avenida 25 De Julio	Ag Nueva Portete 1	Portete Y La 15	Agencia Pascuales	Av. Principal De Pascuales	Aguirre 1	Aguirre 623 Y García Avilés	Aje Ecuador	Km. 15.5 Vía A Daule	AKI 25 De Julio 2	Av. 25 De Julio, Santa Mónica, Av. 25 De Julio Y Calle 51C	AKI Astillero	Chile Y San Martín Esquina	AKI Atarazana	Cdla. Atarazana	AKI Domingo Comin 2	Avenida 10Se (Monseñor Domingo Comin) Y Calle 55 Se
Nombre	Dirección																										
9 De Octubre 1	9 Octubre 1404 Y Machala																										
Aeropuerto Internacional	Av. De Las Américas, Aeropuerto J. J. De Olmedo																										
Ag Boulevard 1	9 De Octubre Entre García Avilés Y Ramichaca																										
Ag Rio Centro Sur	Cc Rocentro Sur, Avenida 25 De Julio																										
Ag Nueva Portete 1	Portete Y La 15																										
Agencia Pascuales	Av. Principal De Pascuales																										
Aguirre 1	Aguirre 623 Y García Avilés																										
Aje Ecuador	Km. 15.5 Vía A Daule																										
AKI 25 De Julio 2	Av. 25 De Julio, Santa Mónica, Av. 25 De Julio Y Calle 51C																										
AKI Astillero	Chile Y San Martín Esquina																										
AKI Atarazana	Cdla. Atarazana																										
AKI Domingo Comin 2	Avenida 10Se (Monseñor Domingo Comin) Y Calle 55 Se																										
Descripción	Esta opción nos permite Consultar Cajero , presentándonos todos sus datos.																										
Rol	Administrador.																										
Pre-Condicion	Que exista el Cajero que se desea Consultar																										
Acciones	<p>Escoger la opción: Cajero.</p> <p>Escoger la opción: Consultar.</p> <p>Seleccionar el Banco.</p> <p>El sistema mostrará los datos del Cajero.</p>																										

Tabla 9 – Caso de Uso 7

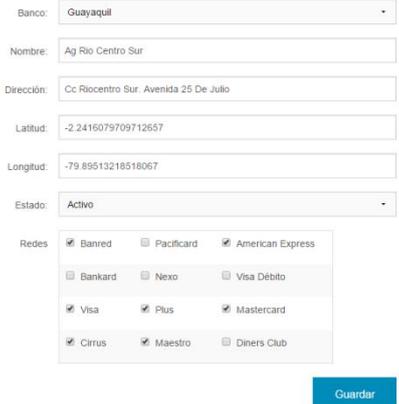
ID	CU07
Nombre	Editar Cajero.
Pantalla	
Descripción	Esta opción nos permite Editar Cajero , permitiéndonos modificar su nombre, dirección, latitud, longitud, estado y redes asociadas.
Rol	Administrador.
Pre-Condicion	Que exista el Cajero que se desea Editar.
Acciones	<p>Escoger la opción: Cajero.</p> <p>Escoger la opción: Consultar.</p> <p>Seleccionar el Banco.</p> <p>Dar clic en Consultar.</p> <p>Escoger el Cajero que se desea editar dando clic en el botón .</p> <p>Modificar el campo nombre, dirección, latitud, longitud, estado o redes asociadas.</p> <p>Guardar.</p> <p>El sistema almacenará los cambios del Cajero.</p>

Tabla 10 – Caso de Uso 8.

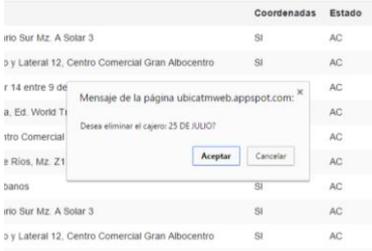
ID	CU08
Nombre	Eliminar Cajero.
Pantalla	
Descripción	Esta opción nos permitirá eliminar un Cajero.
Rol	Administrador.
Pre-Condicion	Que exista el Cajero.
Acciones	<p>Escoger la opción: Cajero.</p> <p>Escoger la opción: Consultar.</p> <p>Escoger el Cajero que se desea eliminar dando clic en el botón .</p> <p>Dar clic en el botón Aceptar.</p> <p>El sistema eliminará los datos del Cajero.</p>

Tabla 11 – Caso de Uso 9.

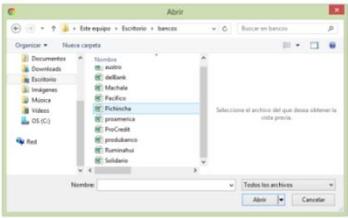
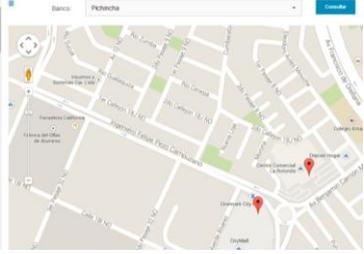
ID	CU09
Nombre	Subir Cajero.
Pantalla	<p>Subir Cajeros</p> 
Descripción	Esta opción nos permitirá subir un archivo con la lista de Cajeros pertenecientes a un Banco.
Rol	Administrador.
Pre-Condicion	Que exista el ID del Banco.
Acciones	<p>Escoger la opción: Cajero.</p> <p>Escoger la opción: Subir.</p> <p>Escoger la opción: Seleccionar archivo.</p> <p>Buscar en nuestro directorio el archivo con la información del Cajero.</p> <p>Dar clic en el botón Abrir.</p> <p>Dar clic en el botón Subir.</p> <p>El sistema agregará los datos del Cajero al Banco indicado.</p>

Tabla 12 – Caso de Uso 10.

ID	CU10
Nombre	Ubicar Cajero.
Pantalla	<p>Consultar Cajeros</p> <p>ALBAN BORJA AV CARLOS JULIO AROSEMENA E RANCOS</p> <p>RIO CENTRO LOS CEBOS Av. del Bombero SIN Km 6.5 Vía a la Costa VIA DAULE KM 5 1/2 VIA DAULE Y AV PRIMERA AGENCIA PETROCOMERCIAL KM 14 15 VIA DAULE</p> <p>Agencia CITY MALL AV BENJAMIN CARRION Y FELPE PESO URDESA VICTOR EMILIO ESTRADA Y LAS LOMAS C C LA ROTONDA AV BENJAMIN CARRION Y 3ERA POLICENTRO AV DEL PERIODISTA Y DR JUAN BAUTISTA ARZUBE</p> 
Descripción	Esta opción nos permitirá ubicar el marcador de cada Cajero en la posición deseada dentro del mapa de Google Maps.
Rol	Administrador.
Pre-Condicion	Que exista el Cajero.
Acciones	<p>Escoger la opción: Mapa.</p> <p>Seleccionar el Banco.</p> <p>Dar clic en Consultar.</p> <p>Dar clic en el botón </p> <p>Seleccionar el Cajero deseado.</p> <p>Mover el marcador a la posición deseada.</p>

Casos de Uso de la Aplicación Android

Como se mencionó anteriormente los casos de uso nos van a permitir representar los requisitos funcionales del sistema, para esto se detallan a continuación los actores o usuarios que interactúan con él.

Usuarios del sistema

Tabla 13 – Usuario de la Aplicación Android.

Usuario	Descripción
Usuario Móvil	Este usuario podrá los cajeros más cercanos a su ubicación.

Diagrama de casos de uso

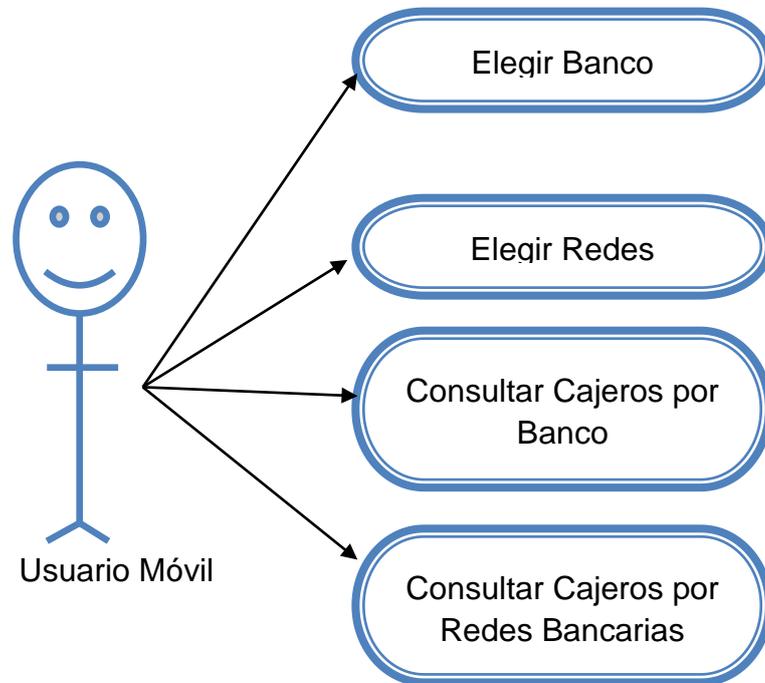


Figura 0.1 - Casos de Uso de la Aplicación Android.

Listado de casos de uso

Tabla 14 – Listado de Casos de Uso de la Aplicación Android.

ID	Casos de Uso	Usuario móvil
CU11	Elegir Banco	X
CU12	Elegir Redes Asociadas	X
CU13	Consultar Cajeros cercanos	X

A continuación se detalla cada caso de uso:

Tabla 15 – Caso de Uso 11.

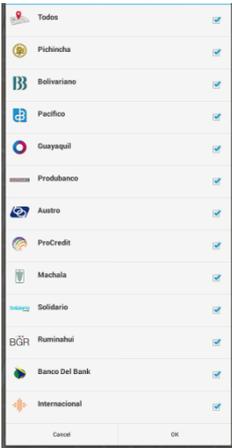
ID	CU11
Nombre	Elegir Banco
Pantalla	 <p>The screenshot shows a vertical list of bank names with checkboxes to their right. The banks listed are: Todos, Pichincha, Bolivariano, Pacifico, Guayaquil, Produbanco, Austro, ProCredit, Machala, Solidario, Raminahui, Banco Del Bank, and Internacional. At the bottom of the list are 'Cancel' and 'OK' buttons.</p>
Descripción	Esta opción permita seleccionar el banco del cual se desea consultar sus cajeros cercanos.
Rol	Usuario móvil
Pre- Condiciones	Ninguna
Acciones	<p>El usuario debe lanzar la aplicación.</p> <p>Escoger en el menú la opción bancos.</p> <p>El sistema le mostrara un listado con los nombre de los bancos disponibles en el sistema.</p>

Tabla 16 – Caso de Uso 12.

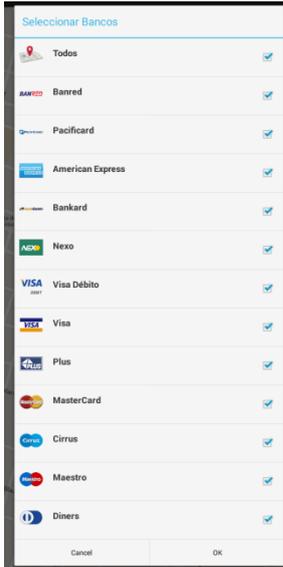
ID	CU12
Nombre	Elegir Redes Asociadas
Pantalla	 <p>The screenshot shows a mobile application screen titled "Seleccionar Bancos". It features a list of bank logos and names, each with a checked checkbox to its right. The banks listed are: Todos, Banred, Pacificard, American Express, Bankard, Nexo, Visa Débito, Visa, Plus, MasterCard, Cirrus, Maestro, and Diners. At the bottom of the screen, there are "Cancel" and "OK" buttons.</p>
Descripción	Esta opción permita seleccionar las redes de la cual se desea consultar sus cajeros cercanos.
Rol	Usuario móvil
Pre- Condiciones	Ninguna
Acciones	<p>El usuario debe lanzar la aplicación.</p> <p>Escoger en el menú la opción Redes.</p> <p>El sistema le mostrara un listado con los nombre de las redes disponibles en el sistema.</p>

Tabla 17 – Caso de Uso 13.

ID	CU13
Nombre	Consultar cajeros cercanos por Banco.
Pantalla	
Descripción	Esta opción permita consultar los cajeros más cercanos.
Rol	Usuario móvil
Pre-Condiciones	Escoger un banco
Acciones	<p>El usuario debe lanzar la aplicación.</p> <p>Escoger en el menú la opción bancos.</p> <p>El sistema le mostrará un listado con los nombres de los bancos disponibles.</p> <p>Escoger el banco de su elección o todos y dar clic en el botón OK.</p> <p>El sistema le mostrará en el mapa los 10 cajeros más cercanos a su posición o puede navegar en el mapa y se cargarán automáticamente los cajeros más cercanos con respecto a la ubicación</p>

Tabla 18 – Caso de Uso 14

ID	CU14
Nombre	Consultar cajeros cercanos por Red Bancaria.
Pantalla	
Descripción	Esta opción permita consultar los cajeros más cercanos.
Rol	Usuario móvil
Pre-Condiciones	Escoger una Red Bancaria
Acciones	<p>El usuario debe lanzar la aplicación.</p> <p>Escoger en el menú la opción Redes Bancarias.</p> <p>El sistema le mostrará un listado con los nombres de las Redes Bancarias disponibles.</p> <p>Escoger la red bancaria de su elección o todos y dar clic en el botón OK.</p> <p>El sistema le mostrará en el mapa los 10 cajeros más cercanos a su posición o puede navegar en el mapa y se cargarán automáticamente los cajeros más cercanos con respecto a la ubicación</p>

CAPÍTULO 3

3. DISEÑO DE LA SOLUCIÓN

Este capítulo trata de la forma en que se desarrolló el proyecto, cómo se recopiló la información de cada ATM de la ciudad de Guayaquil para ser ingresada en nuestras Base de Datos, además del esquema del sitio web incluyendo su diseño lógico, el modelo de la Base de Datos y la descripción y estudio del algoritmo empleado para la búsqueda de los ATMs más cercanos a la ubicación deseada.

3.1. UBICACIONES DE LOS CAJEROS AUTOMÁTICOS

Mediante la recopilación de información de las páginas web de los Bancos de la ciudad de Guayaquil se determinó la ubicación geográfica de los siguientes Bancos:

- Banco de Guayaquil
- Banco del Pacífico
- Banco Bolivariano
- Banco Solidario
- Banco General Rumiñahui
- Banco Internacional
- Banco Produbanco- Grupo Promerica

Existieron bancos en lo que no se logró obtener la información en su sitio web, por lo que se procedió a solicitar ubicación de los ATMs mediante consultas a los directivos en las matrices de dichas instituciones:

- Banco Pichincha
- Banco del Austro
- Banco ProCredit
- Banco Machala
- Banco del Bank

Esta información recopilada de la ubicación de cada uno de los cajeros correspondientes a las entidades financieras antes mencionadas, será nuestra fuente de datos, para la interpretación y visualización de ubicaciones en el mapa, como se muestra en la Figura 3.1 y así la aplicación permitirá orientar al usuario hacia la ruta que debe escoger.



Figura 3.1 – Ubicación de los ATM.

3.2. ESQUEMA DEL SISTEMA WEB

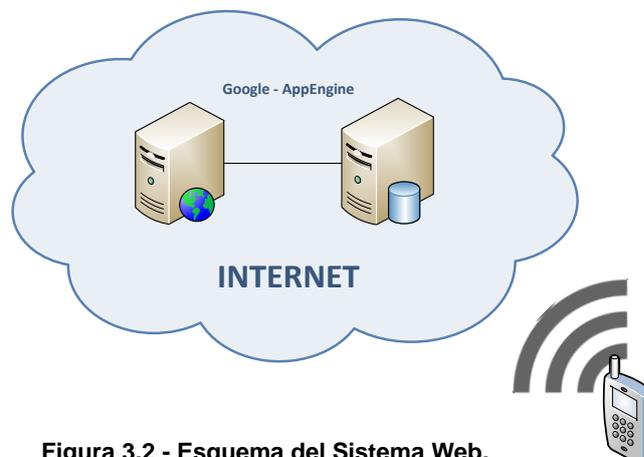


Figura 3.2 - Esquema del Sistema Web.

Como se muestra en la Figura 3.2, se cuenta con una aplicación proveedora de servicios (Consulta de Bancos y Consulta de Redes Bancarias), los cuales serán consumidos por la aplicación móvil desarrollada en Android, estos servicios serán expuestos en el internet por medio de la plataforma App Engine que ofrece Google.

3.3. ARQUITECTURA DEL SISTEMA WEB

Dentro de esta sección definiremos de manera global los distintos componentes del sistema que llevan a cabo alguna tarea de cómputo y la comunicación entre ellos.

En la Figura 3.3 se muestra la arquitectura del sistema Web y sus diferentes capas.

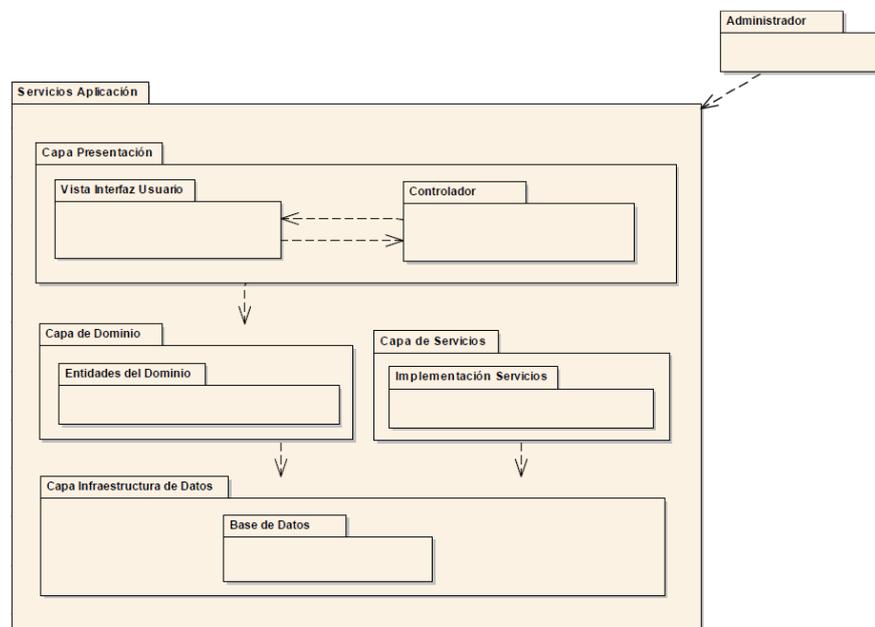


Figura 3.3 – Arquitectura del Sistema Web.

Capa de Presentación

Esta capa tiene la tarea de mostrar la información al usuario e interactuar con él para proveer de información al sistema.

Los componentes de esta capa conforman toda la funcionalidad necesaria para que el administrador interactúe con el sistema. Con el objetivo de mantener separadas la lógica de la aplicación y presentación hemos elegido el patrón de diseño MVC, por lo que esta capa la subdividimos en:

- **Vista de Usuario**

Estos componentes son la parte visual del sistema, contienen lo indispensable para que el Administrador visualice y manipule la información. A esta Sub-Capa pertenecen los controles visuales y componentes que se encarguen de dar formato a los datos.

- **Controladores**

El controlador es el encargado de aislar la lógica y gestión del sistema, de los componentes gráficos. Siendo este el que interpreta la interacción del usuario, interviniendo de manera adecuada tanto sobre la vista como sobre el dominio para provocar cambios de estado en la representación interna de los datos y su visualización [17].

Esto nos ayuda a tener una programación más ordenada evitando tener lógica de la aplicación dentro de las vistas de usuario.

Capa de Dominio

La capa de dominio contiene componentes que se encargan de la lógica negocio, por lo que es la encargada de la adquisición de datos convirtiéndolos en valores significativos para la aplicación.

Para efecto de nuestro sistema la capa de dominio únicamente está compuesta de:

- **Entidades de Dominio**

Esta Sub-Capa contiene los principales elementos sobre los cuales se quiere implementar el sistema, estos representan los objetos del mundo real tales como cajeros o bancos. Y se encarga de su obtención, procesamiento, validación, asociación o cualquier otra tarea referente a la manipulación de estos objetos.

Capa de Servicios

La capa de servicios será la encargada de exponer nuestros procedimientos en la web con el fin de lograr interoperabilidad con la aplicación móvil.

Únicamente consta de:

- **Implementación de Servicios**

Esta Sub-Capa implementa los métodos de servicios e invoca a los componentes del dominio que sean necesarios para su ejecución.

3.4. MODELO DE BASE DE DATOS

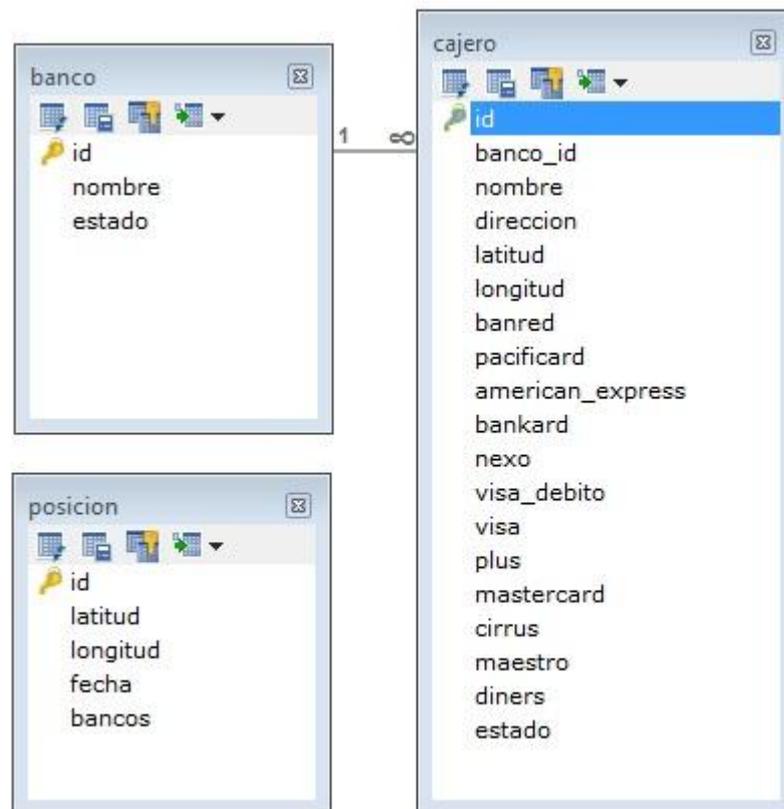


Figura 3.4 – Modelo de Base de Datos.

La base de datos de nuestro sistema web, mostrada en la Figura 3.4, cuenta con las siguientes tablas:

- Banco
- Cajero
- Posición

La tabla Banco será la encargada de almacenar la representación física de nuestra entidad Banco, para nuestra solución sólo necesitaremos el nombre del banco y un estado para indicar si se encuentra activo.

La tabla Cajero será la encargada de almacenar toda la información relacionada con los cajeros automáticos entre los que tenemos: banco al que pertenece, nombre, dirección, posición geográfica (latitud y longitud), redes bancarias asociadas y estado.

La tabla Posición almacenará las consultas que se hacen al sistema, esta información se almacenará con fines de pruebas de funcionalidad, siendo esta, el lugar desde el que se hace la consulta (latitud y longitud), fecha y bancos consultados.

3.5. ALGORITMO DE BÚSQUEDA DE PUNTOS CERCANOS

Nuestro algoritmo de búsqueda debe incluir tres partes de vital importancia que son:

- Fórmula Para Calcular Distancias Geográficas
- Cálculo de distancia entre cajeros y ubicación Actual
- Ordenamiento y filtrado de los resultados

La fórmula que implementaremos en nuestro algoritmo nos permitirá encontrar la distancia entre dos puntos de la tierra (latitud y longitud), para posteriormente aplicar esta fórmula entre cada una de las coordenadas geográficas de los cajeros automáticos y la posición del usuario suministrada por el dispositivo móvil. Estos resultados los ordenaremos descendientemente según la distancia calculada y filtraremos sólo los 10 primeros resultados siendo estos los más relevantes.

Para lograr esto usaremos la fórmula de Semiversenos (Haversine), ilustrada en la Figura 3.5, con el objetivo de encontrar la distancia entre dos puntos de una esfera en este caso del globo terrestre.

La fórmula de Semiversenos nos dice que:

Para dos puntos en una esfera de radio R , con latitudes ϕ_1 y ϕ_2 , diferencia de latitudes $\Delta\phi = \phi_1 - \phi_2$ y longitud de separación $\Delta\lambda$ la distancia entre los dos puntos está definida por:

$$\text{Semiversen}\left(\frac{d}{R}\right) = \text{Semiversen}(\Delta\phi) + \cos(\phi_1) \cos(\phi_2) \text{Semiversen}(\Delta\lambda)$$

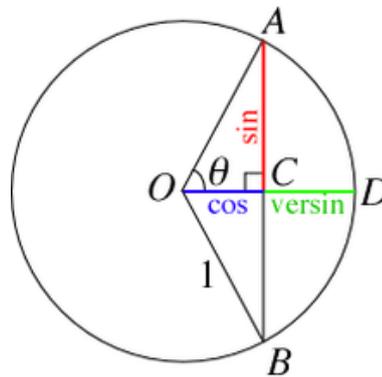


Figura 3.5 – Representación Semiversen.

Planteando nuestra fórmula obtenemos:

R = Radio de la Tierra

Δlat = latitud2 – latitud1

Δlong = longitud2 – longitud1

$$a = \text{sen}^2(\Delta\text{lat}|2) + \cos(\text{lat1}) * \cos(\text{lat2}) * \text{sen}^2(\Delta\text{long}|2)$$

$$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R * c$$

Donde expresado en funciones de BD obtenemos:

$$3956 * 2 * \text{ASIN} \left(\text{SQRT} \left(\text{POWER}(\text{SIN}((\text{orig.lat} - \text{dest.lat}) * \pi / 180 / 2), 2) + \text{COS}(\text{orig.lat} * \pi / 180) * \text{COS}(\text{dest.lat} * \pi / 180) * \text{POWER}(\text{SIN}((\text{orig.lon} - \text{dest.lon}) * \pi / 180 / 2), 2) \right) \right) \text{ as DISTANCIA}$$

Por lo que el algoritmo de búsqueda de cajeros quedaría estructurado de la siguiente forma:

```
cajeros := getListadoCajeros()
```

```
cajerosDistancia := distanciaSemiversenos(cajeros)
```

```
resultado := ordenarFiltrar(cajerosDistancia)
```

```
retornar resultado;
```

3.6. ESQUEMA DE LA APLICACIÓN ANDROID

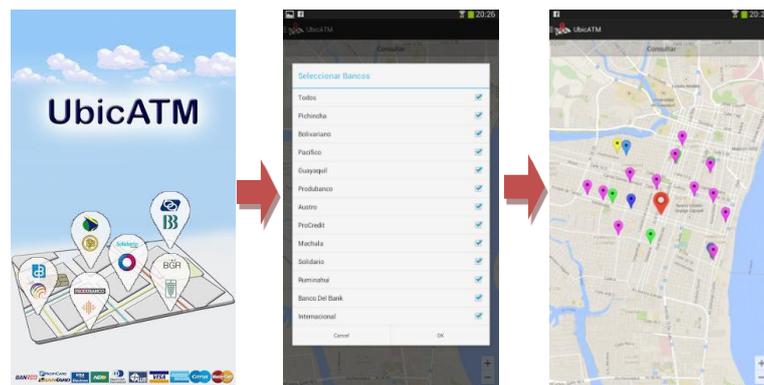


Figura 3.6 – Esquema de la Aplicación Android.

Como se muestra en la figura 3.6, la Aplicación Móvil “UbicATM” consumirá los servicios expuestos por el Sistema Web. Al iniciar la APP, cargará el splash y a su vez el listado de bancos.

Aparecerá por defecto el mapa con la última ubicación consultada y este tratará de obtener automáticamente las coordenadas geográficas de la ubicación actual, además permitirá desplazarnos en el mapa obteniendo una nueva ubicación.

Dentro del menú contamos con las opciones Banco y Redes Bancarias, las cuales nos permitirán realizar la búsqueda a nuestro criterio.

En el caso de Banco podemos filtrar la búsqueda por uno o más bancos, obteniendo como resultado los diez cajeros más cercanos del banco o los bancos seleccionados.

En Redes Bancarias podemos filtrar la búsqueda por una o más redes bancarias, obteniendo como resultado los diez cajeros más cercanos que cumplen con lo requerido.

3.7. ARQUITECTURA DE LA APLICACIÓN ANDROID

Dentro de esta sección definiremos de manera general los distintos componentes de la aplicación que llevan a cabo alguna tarea de cómputo y la comunicación entre ellos.

En la Figura 3.7 se muestra la arquitectura de la aplicación móvil y sus diferentes capas.

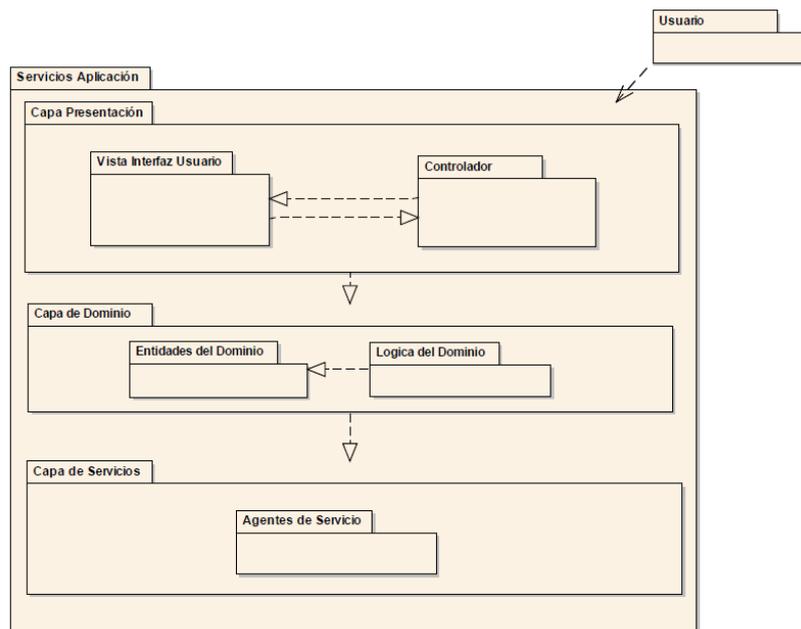


Figura 3.7 – Arquitectura de la Aplicación Android.

Capa de Presentación

Esta capa tiene la tarea de mostrar la información al usuario e interactuar con él para obtener información.

Los componentes de esta capa conforman toda la funcionalidad necesaria para que los usuarios interactúen con el sistema. Con el objetivo de mantener separadas la lógica de la aplicación y presentación hemos elegido el patrón de diseño MVC, por lo que esta capa la subdividimos en:

- **Vista de Usuario**

Estos componentes son la parte visual del sistema, contienen lo necesario para que el Usuario consuma la aplicación. A esta Sub-Capa pertenecen los controles visuales que muestran y obtienen información del Usuario.

- **Controladores**

El controlador es el encargado de aislar la lógica y gestión del sistema, de los componentes gráficos. Siendo este el que interpreta la interacción del usuario, interviniendo de manera adecuada tanto sobre la vista como sobre el dominio para provocar cambios de estado en la representación interna de los datos y su visualización.

Capa de Dominio

La capa de dominio contiene componentes que se encargan de la lógica negocio, por lo que es la encargada de la adquisición de datos, convirtiéndolos en valores significativos para la aplicación.

Para efecto de nuestra aplicación la capa de dominio está compuesta de:

- **Entidades de Dominio**

Esta Sub-Capa contiene los principales elementos sobre los cuales se quiere implementar el sistema, estos representan los objetos del mundo real tales como cajeros o bancos. Y se encarga de su obtención, procesamiento, validación, asociación o cualquier otra tarea referente a la manipulación de estos objetos.

- **Lógica del Dominio**

Esta Sub-Capa contiene los componentes que se utilizarán para la obtención y procesamiento de información adicional, previa la invocación de los Agentes de Servicio.

Capa de Servicios

La capa de servicios será la encargada de consumir los servicios expuestos con el fin de lograr interoperabilidad con el sistema web.

Únicamente consta de:

- **Agentes de Servicios**

Esta Sub-Capa implementa componentes de acceso a datos que abstraen y aíslan los requerimientos de las llamadas a los servicios, además de poder soportar aspectos adicionales como cache y mapeos básicos entre el formato de datos expuesto en los servicios y el formato de datos utilizado por la aplicación.

3.8. ARQUITECTURA DE DEPENDENCIA

Dentro de esta sección se describe como se forma jerárquicamente tanto el servicio web como la aplicación, sus diferentes niveles de dependencia, la forma en que interactúan estos niveles.

Sistema Web

En la Figura 3.8 se muestra desde arriba hacia abajo, como van a depender cada uno de los niveles en el Sistema Web



Figura 3.8 – Arquitectura de dependencia para el servicio web.

Como se indica en la Figura 3.8, en la parte superior se encuentra el Sistema Web como tal, el cual se realizó mediante uso del Framework Codeignitier, mostrado a continuación en la misma Figura.

Este Framework hace uso del lenguaje de programación PHP, por lo que genera dependencia entre estos niveles. A su vez, se utiliza un driver para la conexión desde el lenguaje PHP hacia la capa de la base, que es la Base de Datos.

El App Engine también envía o recibe información desde o hacia la Base de Datos.

Aplicación Móvil

En la Figura 3.9 se muestra de forma gráfica, los niveles de dependencia existentes en el desarrollo de la aplicación móvil.

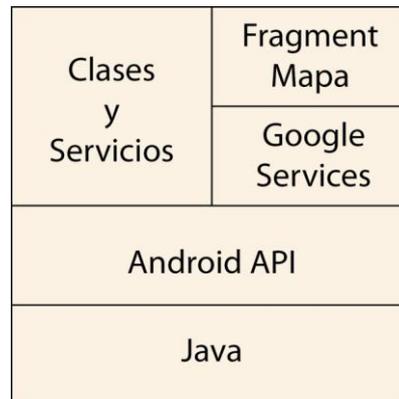


Figura 3.9 – Arquitectura de dependencia para la aplicación móvil.

En la parte superior se muestra el Fragment que contiene al Mapa, el cual es proporcionado por Google Services, para el funcionamiento de la aplicación, las clases y servicios desarrollados, hacen uso del Mapa y los servicios que este proporciona, por lo cual se genera dependencia entre ellos.

Para poder configurar los niveles descritos anteriormente, se hace uso del Android API, es ahí donde se dan los permisos necesarios para el funcionamiento correcto de la aplicación.

El lenguaje de programación usado en la aplicación móvil es JAVA, por lo cual se considera a este como el nivel que está en la base.

3.9. DISEÑO DE PRUEBAS

Con la finalidad de asegurar la calidad de la aplicación se plantea realizar ciertas tareas en las que se van a evaluar los criterios de aceptación de cada funcionalidad, considerando los distintos casos de pruebas detallados a continuación:

- **Caso Positivo:** donde el sistema funciona correctamente, los formularios son llenados correctamente y las consultas son desplegadas sin problemas.
- **Caso Negativo:** donde se trata de llevar el sistema al límite de sus capacidades y empieza a presentar errores en diferentes tareas y solicitudes realizadas por el usuario.

A continuación se presenta en la Figura 3.10 la plantilla utilizada para el registro y ejecución de los casos de pruebas, la cual nos va a servir para emitir un informe del estado y porcentaje de las pruebas.

Diseño Casos de Prueba							Ciclo 1 - Ejecución de pruebas			
ID	Fecha	Precondiciones	Descripción	Pasos	* Resultado esperado	* Tipo Caso de Prueba	Clasificación	Fecha	Resultado	Nº Incidente
TRANSFERENCIAS										

Figura 3.10 – Ejecución de los Casos de Pruebas.

Como podemos observar en la figura 3.10 tenemos los siguientes campos:

ID: representa el identificador del caso.

Fecha: fecha de realización del caso de prueba.

Precondiciones: condiciones necesarias para poder ejecutar el caso de prueba.

Descripción: especificación concisa del caso de prueba.

Pasos: procedimiento a seguir para la ejecución de las pruebas.

Resultado esperado: descripción de los resultados esperados.

Tipo de caso de prueba: identificador del caso de prueba.

Clasificación: puede tener valores como positivo o negativo.

Ciclo – Fecha: Fecha de ejecución del caso de prueba.

Resultado: puede tener valores como: falló, no aplica, aceptado y no probado.

Nº incidente: código del incidente en caso de que exista.

Finalmente después de la ejecución de las pruebas podremos obtener el porcentaje de aceptación de las mismas mediante la tabla mostrada en la Figura 3.11.

Ciclo 1 – Ejecución de Pruebas	Total de Casos Diseñados	
	Pasaron	
	Pasaron regresión	
	Fallaron	
	No probado	
	No Aplica	
	Status	
	%Ejecución del Ciclo	
	%Aprobación	

Figura 3.11 – Porcentaje de Aceptación de Pruebas.

Como se mencionó en el capítulo uno, se realizará pruebas de estrés en donde se llevará al sistema al punto más alto de su rendimiento mediante el cual se establecerá los umbrales sobre los cuales la aplicación opera con normalidad, mediante una representación estadística de los resultados.

CAPÍTULO 4

4. IMPLEMENTACIÓN DE LA SOLUCIÓN

4.1. DESARROLLO DEL SISTEMA WEB

Dentro de esta sección se procede a detallar las configuraciones y algoritmos de mayor jerarquía que se utiliza en el desarrollo del sistema web. Por lo que se describen los siguientes acontecimientos:

- Creación del proyecto en App Engine y obtención de APIs de Google
- Instalación y Configuración del Framework Codeigniter
- Implementación del Algoritmo de puntos cercanos

Creación del proyecto en App Engine y obtención de APIs de Google

En esta sección se explica cómo crear un proyecto en los servidores de Google App Engine para posteriormente hacer un *deploy* del código y ejecutar el sistema web. Además se explicará cómo obtener las API Key necesarias.

El procedimiento es el siguiente:

Ingresar a la página de la consola de desarrolladores de Google y hacer clic en el botón “Crear Proyecto”, mostrado en la figura 4.1.

Un botón rectangular con un fondo azul y el texto "Crear proyecto" en blanco.

Figura 4.1 – Botón “Crear proyecto” en App Engine.

A continuación la página pedirá que se ingrese el nombre del proyecto y se le asignará un identificador. Este identificador seguido de .appspot.com será posteriormente el anfitrión de la URL del sitio web, como se muestra en la Figura 4.2.

Un formulario web con el título "Nuevo proyecto". Tiene dos campos de entrada de texto: "Nombre del proyecto" con el valor "Ubicatm" y "ID del proyecto" con el valor "ubicatmweb". Debajo de los campos hay un enlace "Mostrar las opciones avanzadas...". Al final del formulario hay dos botones: "Crear" (azul) y "Cancelar" (gris).

Figura 4.2 – Creación de un nuevo Proyecto en App Engine.

Con esto se ha creado el nuevo proyecto y el sistema redirecciona a una nueva página mostrando estadísticas de uso del sitio en la parte derecha y opciones de configuración a la izquierda de la pantalla. De estas opciones de configuración se busca la que dice “APIs y autenticación” y se elige la opción APIs, como se muestra en la Figura 4.3.



Figura 4.3 – APIs y autenticación.

Al seleccionar esta opción se cargarán todas las posibles APIs que Google brinda a los desarrolladores para que hagan uso de su plataforma, en este caso se eligen las de Google MapsAndroid y Javascript para la aplicación móvil y el sistema web respectivamente, tal como se indica en la Figura 4.4.



Figura 4.4 – APIs de Google Maps.

Posteriormente, en los dos casos el sistema redirigirá a una nueva página donde se debe dar clic en el botón habilitar, mostrado en la Figura 4.5.



Figura 4.5 – Botón “Habilitar API”.

Esto permitirá hacer uso del Api de Google Maps en las aplicaciones deseadas, y presentará un resumen del uso y límites diarios, tal como se muestra en la Figura 4.6.

Información general	Uso	Cuotas
Estado de facturación		
Esta API está limitada por la cuota total que se muestra a continuación.		
Apply for higher quota 		
: resumen de la cuota		
Diaria (a partir de las 00:00 PST, hora del Pacífico)		
Cuota gratuita	25.000 solicitudes por día	
Cuota total	1.000.000 solicitudes por día	
Restante	1.000.000 solicitudes por día 100 % del total	

Figura 4.6 – Cuota Google Maps.

Instalación y Configuración del Framework Codeigniter

En esta sección se describirán los pasos para realizar una correcta instalación y configuración del Framework Codeigniter. Con este Framework se desarrollará y ejecutará el sistema web de “UbicATM” por lo que es de vital importancia para este propósito.

El procedimiento a seguir es el siguiente:

Descargar el Framework de la página oficial, dando clic en el botón Download como se muestra en la figura 4.7.

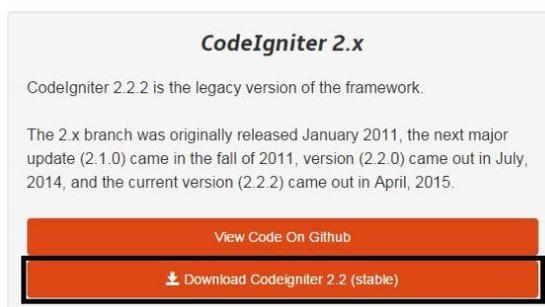


Figura 4.7 – Descarga de Codeigniter.

Una vez descargado, se procede a descomprimir el archivo haciendo clic en el ícono mostrado en la Figura 4.8.



Figura 4.8 – Icono del archivo comprimido Codeigniter.

Una vez descomprimidos los archivos, aparece un listado de directorios. Estos directorios contienen librerías y convenciones que forman parte esencial del Framework, si se accede a la carpeta “application” se puede notar que entre sus sub-carpetas existen tres llamadas: controllers, models, views; como se muestra en la Figura 4.9.



Figura 4.9 – Modelo Vista Controlador.

Dentro de estos directorios se coloca respectivamente los Controladores, Modelos y Vistas siguiendo el patrón de diseño MVC.

El siguiente paso es configurar Codeigniter para que reconozca la ruta base del sitio web, por lo que se debe ubicar al archivo

config.php dentro de la carpeta application\config, como en la Figura 4.10.

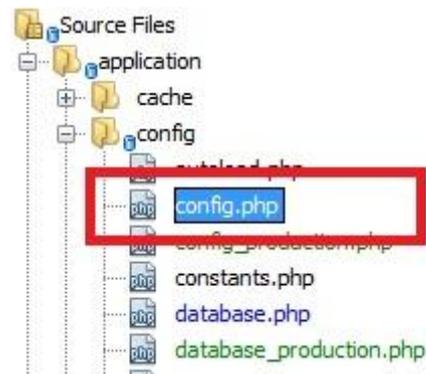


Figura 4.10 – config.php.

Dentro del archivo config.php se modifica el parámetro base_url con la url del sitio de la aplicación que se está desarrollando, como se muestra en la Figura 4.11.

```
$config['base_url'] = 'http://ubicatmweb.appspot.com';
```

Figura 4.11 – Configuración del parámetro base_url
Para que funcione con el nuevo sitio.

A continuación se modifican los parámetros de conexión a la base de datos en el archivo database.php, dentro de application\config, como se muestra en la Figura 4.12

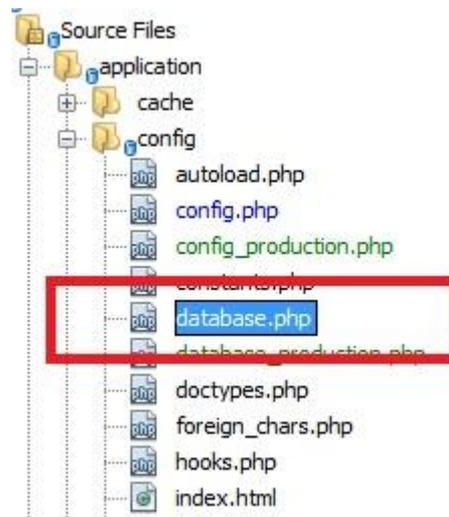


Figura 4.12 – Localización del archivo database.php.

Se modificarán los parámetros *hostname*, *username*, *password*, *database* y *socket* para trabajar con el motor de base de datos, en la Figura 4.13 se muestra cómo quedará la configuración para este caso específico.

```
$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = 'admin3021512&';
$db['default']['database'] = 'ubicatm';
$db['default']['socket'] = '/cloudsql/ubicatmweb:ubicatm';
$db['default']['dbdriver'] = 'mysqli';
```

Figura 4.13 – Parámetros de conexión a la Base de Datos.

Implementación del Algoritmo de puntos cercanos

En esta sección se mostrará de qué forma el algoritmo de búsqueda de puntos cercanos fue elaborado y las partes que contiene.

En este caso particular, el algoritmo se implementa en la capa de infraestructura de datos, por lo que al ser esta la última capa de la estructura, su lenguaje es SQL, de esta manera se brindan los datos depurados a las capas superiores listos para ser expuestos por los servicios y haciendo ligera la comunicación entre base de datos y aplicación.

Al observar el algoritmo podemos notar que consta de 3 partes muy importantes que son:

- Implementación de la fórmula de Semiversenos, mostrada en la Figura 4.14.

```

111.045 * DEGREES (ACOS (COS (RADIANS (?)
    * COS (RADIANS (latitud)
    * COS (RADIANS (?) - RADIANS (longitud))
    + SIN (RADIANS (?)
    * SIN (RADIANS (latitud)))) AS distancia
  
```

Figura 4.14 – Implementación de la fórmula Semiversenos.

- Filtrado de la información por bancos o redes de cobro, mostrado en la Figura 4.15.

```

caj.latitud IS NOT NULL AND caj.longitud IS NOT NULL AND
(? is null or banco_id in (".$coordenada['bancos'].")) AND
(? is null or banco_id in (".$coordenada['bancos'].")) AND
(? is null or banred = 1) AND
(? is null or pacificard = 1) AND
(? is null or american_express = 1) AND
(? is null or bankard = 1) AND
(? is null or nexo = 1) AND
(? is null or visa_debito = 1) AND
(? is null or visa = 1) AND
(? is null or plus = 1) AND
(? is null or mastercard = 1) AND
(? is null or cirrus = 1) AND
(? is null or maestro = 1) AND
(? is null or diners = 1) AND
(? is null or estado = 1)

```

Figura 4.15 – Filtrado de la información por Banco o Redes de Cobro.

- Ordenamiento y obtención de los 10 resultados más relevantes, mostrado en la Figura 4.16.

```

ORDER BY distancia
LIMIT 10

```

Figura 4.16 – Resultados Relevantes.

Al unir todo y realizarlo por medio del Framework Codeigniter quedaría como en la Figura 4.17.

```

function getCercanos($parametros)
{
    $sql=" SELECT caj.id,ban.id idBanco,ban.nombre nombreBanco,caj.nombre,caj.direccion,caj.latitud, caj.longitud,
banred, pacificard, american_express, bankard, nexo, visa_debito, visa, plus, mastercard, cirrus, maestro, diners
111.045* DEGREES(ACOS(COS(RADIANS(?))
* COS(RADIANS(latitud))
* COS(RADIANS(?) - RADIANS(longitud))
+ SIN(RADIANS(?))
* SIN(RADIANS(latitud)))) AS distancia
FROM cajero caj,
banco ban
WHERE caj.banco_id = ban.id AND
caj.latitud IS NOT NULL AND caj.longitud IS NOT NULL AND (? is null or banco_id in (?)) AND
(? is null or banred = 1) AND
(? is null or pacificard = 1) AND
(? is null or american_express = 1) AND
(? is null or bankard = 1) AND
(? is null or nexo = 1) AND
(? is null or visa_debito = 1) AND
(? is null or visa = 1) AND
(? is null or plus = 1) AND
(? is null or mastercard = 1) AND
(? is null or cirrus = 1) AND
(? is null or maestro = 1) AND
(? is null or diners = 1) AND
(? is null or estado = 1)

ORDER BY distancia
LIMIT 25";
$query=$this->db->query($sql,$parametros);
$data = $query->result();
return $data;
}

```

Figura 4.17 – Algoritmo búsqueda ATMs cercanos.

En esta implementación se puede observar claramente la forma como el Framework ayuda a ejecutar un *query* en pocas líneas de código.

4.2. DESARROLLO DE LA APLICACIÓN MÓVIL

Dentro de esta sección se procede a detallar las configuraciones y algoritmos de mayor jerarquía que se utiliza en el desarrollo de esta aplicación Android. Por lo que se describe los siguientes acontecimientos:

- Permisos de la aplicación
- Obtención de Coordenadas Geográficas
- Conexión con Servicios Web

Permisos de la aplicación

Los permisos en Android se los configuran dentro del archivo AndroidManifest.xml, por medio de la etiqueta `<uses-permission/>`, Para este caso se hace uso de:

INTERNET: Permite a la aplicación abrir comunicaciones de red. Se usará para comunicarse con los servicios del sistema web con el fin de obtener bancos y cajeros.

ACCES_FINE_LOCATION: Permite a una aplicación precisar la localización desde fuentes como: GPS, redes móviles o wifi. Será

usado para obtener la ubicación geográfica del usuario al momento de abrir la aplicación.

ACCESS_NETWORK_STATE: Permite a la aplicación acceder a información sobre las redes. Se usará para comprobar si posee acceso a alguna red inalámbrica.

WRITE_EXTERNAL_STORAGE: Permite a la aplicación escribir en almacenamiento externo a la aplicación. Será usado para guardar temporalmente las imágenes de los bancos en cache.

READ_GSERVICES: Permite a la aplicación usar los servicios de Google. En este caso se usará para poder presentar el mapa.

En la Figura 4.18 se muestra la configuración de los permisos dentro del código.

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

Figura 4.18 – Permisos de la Aplicación.

Obtención de Coordenadas Geográficas

En esta sección se muestra cómo esta aplicación Android hace uso de los servicios de geo localización del dispositivo móvil para obtener la ubicación geográfica del usuario al momento de usar la aplicación.

Lo primero que se debe realizar es obtener el objeto locationManager del sistema, por medio de este objeto se procede a verificar si el dispositivo tiene el GPS habilitado o la ubicación por medio de redes móviles o wifi, como se muestra en la Figura 4.19.

```
locationManager = (LocationManager) mContext.getSystemService(LOCATION_SERVICE);
isGPSEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
isNetworkEnabled = locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
```

Figura 4.19 – Obtención de Coordenadas Geográficas.

Si el dispositivo posee el GPS habilitado, se procede a llamar a la función requestLocationUpdates, la misma que mantendrá actualizada a la aplicación con las coordenadas geográficas del usuario ejecutando el callback onLocationChanged y enviándole como parámetro la ubicación.

Mientras la función requestLocationUpdates trata de obtener las coordenadas actuales, temporalmente se devuelve la última ubicación obtenido por el proveedor GPS, como se muestra en la Figura 4.20.

```
if (isGPSEnabled)
{
    if (location == null)
    {
        locationManager.requestLocationUpdates(
            locationManager.GPS_PROVIDER,
            MIN_TIME_BW_UPDATES,
            MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
        location = locationManager.getLastKnownLocation(locationManager.GPS_PROVIDER);
        if (location != null)
        {
            latitude = location.getLatitude();
            longitude = location.getLongitude();
        }
    }
}
```

Figura 4.20 – Última ubicación obtenida por GPS.

De igual forma, si el controlador GPS no está habilitado, la ubicación por redes inalámbricas tratará de hacer el mismo procedimiento como se muestra en la Figura 4.21.

```
else if (isNetworkEnabled)
{
    locationManager.requestLocationUpdates(
        locationManager.NETWORK_PROVIDER,
        MIN_TIME_BW_UPDATES,
        MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
    if (locationManager != null)
    {
        location = locationManager.getLastKnownLocation(locationManager.NETWORK_PROVIDER);
        if (location != null)
        {
            latitude = location.getLatitude();
            longitude = location.getLongitude();
        }
    }
}
```

Figura 4.21 – Última ubicación por Wifi.

Conexión con Servicios Web

Debido a que los servicios publicados en el sistema web son REST la comunicación desde la aplicación se la realiza por medio del protocolo Http, intercambiando contenido en formato JSON.

Los pasos a seguir para obtener el listado de cajeros es el siguiente:

Crear un cliente http y especificarle la URL de consulta; en este caso para una mejor organización del código se mantienen estas rutas dentro de los recursos de la aplicación. Adicionalmente se especifica un encabezado indicando que la comunicación se la realizará por medio de JSON, como se muestra en la Figura 4.22.

```

HttpClient httpClient = new DefaultHttpClient();
Resources res = context.getResources();
String url = res.getString(R.string.server_url) + res.getString(R.string.service_bancos);
HttpGet get = new HttpGet(url);
get.setHeader("content-type", "application/json");
ArrayList<Banco> bancos = null;

```

Figura 4.22 – Creación requerimiento Bancos.

Posteriormente se ejecuta la llamada Http, y se obtiene la respuesta, en primera instancia como una cadena de caracteres para inmediatamente ser convertida en formato JSON, como se muestra en la Figura 4.23.

```

HttpResponse resp = httpClient.execute(get);
String respStr = EntityUtils.toString(resp.getEntity());
JSONArray respJSON = new JSONArray(respStr);

```

Figura 4.23 – Realizar llamada Http.

Finalmente se recorre el arreglo de Bancos JSON y se llena la lista de objetos de tipo Banco, tal como se muestra en la Figura 4.24.

```

bancos = new ArrayList<Banco>();
for(int i=0; i<respJSON.length(); i++)
{
    JSONObject obj = respJSON.getJSONObject(i);
    Banco banco = new Banco();
    banco.id = obj.getInt("id");
    banco.nombre = obj.getString("nombre");
    bancos.add(banco);
}
return bancos;

```

Figura 4.24 – Procesar respuesta Bancos.

De igual manera se realiza el mismo procedimiento para la consulta de Cajeros Cercanos, cambiando el tipo de dato del objeto

devuelto, el método de consulta a POST y la URL de consulta, como se muestra en la Figura 4.25.

```
HttpClient httpClient = new DefaultHttpClient();
Resources res = context.getResources();
String url = res.getString(R.string.server_url) + res.getString(R.string.service_cajeros);
HttpPost post = new HttpPost(url);
post.setHeader("content-type", "application/json");
ArrayList<Cajero> cajeros = null;
```

Figura 4.25 – Crear requerimiento Cajero.

Posteriormente se agregan los parámetros necesarios para la consulta, para este caso son: latitud y longitud desde la que se desean adquirir los cajeros cercanos y el listado de bancos que se desea filtrar, como se muestra en la Figura 4.26.

```
JSONObject dato = new JSONObject();
dato.put("latitud", String.valueOf(puntoCentral.latitud));
dato.put("longitud", String.valueOf(puntoCentral.longitud));
String bancos = bancosElegidos.toString();
dato.put("bancos", bancos.substring(1, bancos.length()-1));
StringEntity entity = new StringEntity(dato.toString());
post.setEntity(entity);
```

Figura 4.26 – Parámetros requerimiento Cajero.

4.3. PUBLICACIÓN DE LA APLICACIÓN WEB

En esta sección se revisarán los requisitos y pasos a realizar para poder publicar este sistema web en los servidores de Google App Engine. Cabe recordar que este sistema web es el que manejará y procesará toda la información de bancos y cajeros por lo que este paso es de suma importancia para el funcionamiento de esta aplicación.

Requisitos

Para poder publicar esta aplicación en los servidores de Google App Engine, previamente se debe haber descargado e instalado el programa “Google App EngineLauncher”, este programa es el que permitirá hacer pruebas locales y realizar el deploy del sitio web.

Para obtenerlo se debe ingresar al sitio oficial de Google y descargar la versión para php que es la que se necesita, como se muestra en la Figura 4.27.

Download the Google App Engine SDK

By downloading, you agree to be bound by the [Terms](#) that govern use of the App Engine SDK.



Figura 4.27 – Descargar el Google APP Engine SDK.

Se guarda el archivo, y se procede a ejecutarlo haciendo clic en el botón mostrado en la Figura 4.28



Figura 4.28 – Archivo para ejecutar Google APP Engine.

A continuación aparecerá la ventana de instalación y se continúa haciendo clic en el botón “Next”, como se muestra en la Figura 4.29



Figura4.29 – Welcome to the Google App Engine.

Finalmente para ingresar a la aplicación se debe acceder por medio del ícono mostrado en la Figura 4.30

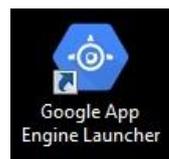


Figura4.30 – Google App Engine Launcher.

Creación del proyecto

Una vez ingresado al programa se debe crear una nueva aplicación por medio de la opción “Create New Application” dentro del menú “File”, tal como se muestra en la Figura 4.31.

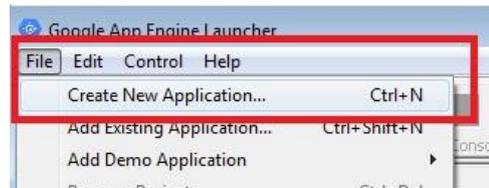


Figura 4.31 – Creación del Proyecto.

Esto mostrará una ventana donde se deben ingresar 3 campos de vital importancia:

Application Name: Se ingresa el Id de la aplicación que previamente se creó en la página de la consola de desarrolladores de Google.

Parent Directory: Se busca la ruta de los archivos del proyecto en Codeigniter.

Runtime: Se selecciona PHP ya que es el lenguaje en que se desarrolló el sistema web.

En la Figura 4.32 se muestra la configuración para este caso específico.

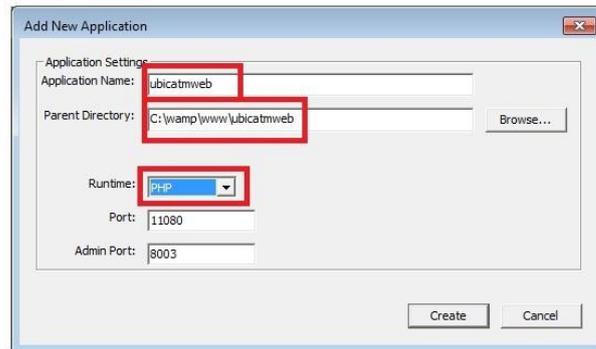


Figura 4.32 – Creación de nueva aplicación.

Esto hará que el sitio web que se está desarrollando aparezca en el listado de aplicaciones del Google App Launcher. Luego se la secciona y se da clic sobre el botón “Deploy” para proceder a subir el sitio a los servidores de Google App Engine, como se muestra en la Figura 4.33.



Figura 4.33 – Subir el sitio web a los servidores de APP Engine.

Finalmente ingresamos el usuario y contraseña, creados previamente, de Google, para proceder con el deploy, como se muestra en la Figura 4.34.



Figura 4.34 – Hacer Deploy de la aplicación a Google

CAPÍTULO 5

5. PRUEBAS EXPERIMENTALES Y ANÁLISIS DE RESULTADOS

Para las pruebas de funcionalidad se realizaron distintos Casos de Prueba basados en los Casos de Uso, tanto del Sistema Web como de la Aplicación Android, se procedió a validar cada caso con los distintos escenarios que pueden tender ambas aplicaciones, para de esta manera detectar y corregir todo tipo de fallas, asegurando así que dichos sistemas funcionen de manera rápida y eficiente.

Mientras que para las pruebas de estrés se hizo uso de la herramienta JMeter, la cual permite analizar y medir el desempeño de los servicios que consume la aplicación móvil, expuestos en el sitio web.

5.1. PRUEBAS DE FUNCIONALIDAD

Las Pruebas de Funcionalidad son pruebas basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el sistema web y la APP móvil. Mediante las Pruebas de Funcionalidad aseguraremos el trabajo apropiado de las características y restricciones que el sistema web y la aplicación móvil deben tener. Tomando en consideración la interacción que existe entre el sistema y sus usuarios.

Para llevar a cabo este análisis, se hará uso de los Casos de Uso, los cuales describen las actividades que deberán realizarse para ejecutar un proceso en el sistema.

5.2. PRUEBAS DE ESTRÉS

Para comprobar el funcionamiento adecuado de la aplicación se realizaron pruebas de estrés sobre el sistema web, de esta forma se busca garantizar que el sistema responderá adecuadamente para un gran número de usuarios.

Haremos uso de las pruebas de estrés para evaluar el desempeño de los servicios web que ofrece nuestro sistema para esto será necesario realizar un alto número de requerimientos a los servicios e interpretar los resultados de: tiempo de respuesta, rendimiento, entre otras variables estadísticas.

Software Utilizado

Para realizar estas pruebas se hizo uso de la herramienta JMeter la cual es una herramienta de código abierto hecha en JAVA diseñada para pruebas de saturación. Esta herramienta nos devolverá, según la configuración dada, los datos antes mencionados de una manera visual y sencilla de interpretar.

Tipos de Pruebas

Con la ejecución de las pruebas se busca definir:

Tiempo de Respuesta y porcentaje de error para una pequeña cantidad de usuarios

Tiempo de Respuesta y porcentaje de error para una gran cantidad de usuarios

De esta forma tendremos noción del rendimiento de nuestra aplicación y podremos definir con certeza si es aceptable o no para el usuario final.

Configuraciones JMeter

Primeramente definiremos los parámetros con los que será configurado el JMeter para lograr nuestros objetivos:

Usuarios

Dentro del JMeter debemos establecer el **número de hilos** a ejecutarse, estos equivalen a la cantidad de usuarios que realizan la petición. Posteriormente se procede a establecer el **Periodo de subida**, el cual indica el rango de tiempo en que los usuarios deben hacer la petición, en nuestro caso como queremos que sean simultaneas le pondremos un segundo. Finalmente se establece el **Contador de Bucle**, el cual indica el número de veces que cada hilo ejecutará la prueba.

Como se muestra en la Figura 5.1, podemos ver un ejemplo de 100 usuarios enviado peticiones simultáneamente por 30 veces. En total se procesarían 3000 peticiones.

Propiedades de Hilo

Número de Hilos: 100

Periodo de Subida (en segundos): 1

Contador del bucle: Sin fin 30

Figura 5.1 – Peticiones enviadas simultáneamente.

Requerimiento

Como segundo paso procedemos a hacer una petición del tipo **HTTP**, en la cual se especifica el **nombre del servidor o la IP** del sitio al que se le harán las pruebas, además de los parámetros necesarios para realizar la petición. Esto se muestra en la Figura 5.2.

Petición HTTP

Nombre: Petición HTTP

Comentarios:

Servidor Web

Nombre de Servidor o IP: ubicatmweb.appspot.com Puerto: Timeout (milisegundos)

Conexión: Respuesta:

Petición HTTP

Implementación HTTP: Protocolo: Método: POST Codificación del contenido:

Ruta: /index.php/servicios/cajero/cercanos/

Redirigir Automáticamente Seguir Redirecciones Utilizar KeepAlive Usar 'multipartform-data' para HTTP POST Cabeceras compatibles con navegadores

Parameters Body Data

Enviar Parámetros Con la Petición:

Nombre:	Valor	¿Codificar?	¿Incluir Equals?
bancos	-1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
latitud	-2.2125853	<input type="checkbox"/>	<input checked="" type="checkbox"/>
longitud	-79.9024072	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Detail Añadir Add from Clipboard Borrar Up Down

Figura 5.2 – Petición de tipo Http

Para este caso se hacen pruebas con los servicios de:

- Cajeros Cercanos
- Listado Bancos

5.3. REPORTE DE PRUEBAS DE FUNCIONALIDAD

5.3.1. Reporte de Pruebas Funcionales del Sistema Web

Los Casos de Uso seleccionados para esta prueba fueron:

- Ingresar Banco
- Consultar Banco
- Editar Banco
- Eliminar Banco
- Ingresar Cajero
- Consultar Cajero
- Editar Cajero
- Eliminar Cajero
- Subir Cajero
- Ubicar Cajero

Se contemplaron los diversos escenarios que pueden presentarse para cada caso de uso.

En el Anexo 1 se muestra cada uno de los Diseños de Casos de Pruebas, donde se especifica para cada prueba, las precondiciones, la descripción, los pasos a realizar, los resultados esperados, el tipo de caso de prueba, la clasificación y resultado, además de la fecha en que se realizó la prueba.

Para el Caso de Uso CU01: Ingresar Banco, como se muestra en la Tabla 33 del Anexo1, se diseñaron tres pruebas donde se clasifican una como positiva y dos como negativas, logrando un resultado Aceptado. De esta manera se obtienen los datos que se detallan en la Tabla 19 alcanzando una aprobación del 100%.

Tabla 19 – Ejecución de Pruebas Caso de Uso CU01.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU01: Ingresar Banco	Total de Casos Diseñados	3
		Pasaron	3
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU02: Consultar Banco, como se muestra en la Tabla 34 del Anexo1, se diseñó una prueba clasificada como positiva, logrando un resultado Aceptado. De esta manera se obtienen los datos que se detallan en la Tabla 20 alcanzando una aprobación del 100%.

Tabla 20 – Ejecución de Pruebas Caso de Uso CU02.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU02: Consultar Banco	Total de Casos Diseñados	1
		Pasaron	1
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU03: Editar Banco, como se muestra en la Tabla 35 del Anexo1, se diseñaron tres pruebas donde se clasifican una como positiva y dos como negativas, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 21 alcanzando una aprobación del 100%.

Tabla 21 – Ejecución de Pruebas Caso de Uso CU03.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU03: Editar Banco	Total de Casos Diseñados	3
		Pasaron	3
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU04: Eliminar Banco, como se muestra en la Tabla 36 del Anexo1, se diseñó una prueba clasificada como positiva, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 22 alcanzando una aprobación del 100%.

Tabla 22 – Ejecución de Pruebas Caso de Uso CU04.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU04: Eliminar Banco	Total de Casos Diseñados	1
		Pasaron	1
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU05: Ingresar Cajero, como se muestra en la Tabla 37 del Anexo1, se diseñaron tres pruebas donde se clasifican una como positiva y dos como negativas, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 23 alcanzando una aprobación del 100%.

Tabla 23 – Ejecución de Pruebas Caso de Uso CU05.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU05: Ingresar Cajero	Total de Casos Diseñados	3
		Pasaron	3
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU06: Consultar Cajero, como se muestra en la Tabla 38 del Anexo1, se diseñaron dos pruebas donde se clasifican como positivas, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 24 alcanzando una aprobación del 100%.

Tabla 24 – Ejecución de Pruebas Caso de Uso CU06.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU06: Consultar Cajero	Total de Casos Diseñados	2
		Pasaron	2
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU07: Editar Cajero, como se muestra en la Tabla 39 del Anexo1, se diseñaron tres pruebas donde se clasifican una como positiva y dos como negativas, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 25 alcanzando una aprobación del 100%.

Tabla 25 – Ejecución de Pruebas Caso de Uso CU07.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU07: Editar Cajero	Total de Casos Diseñados	3
		Pasaron	3
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU08: Eliminar Cajero, como se muestra en la Tabla 40 del Anexo1, se diseñó una prueba clasificada como positiva, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 26 alcanzando una aprobación del 100%.

Tabla 26 – Ejecución de Pruebas Caso de Uso CU08.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU08: Eliminar Cajero	Total de Casos Diseñados	1
		Pasaron	1
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU09: Subir Cajero, como se muestra en la Tabla 41 del Anexo1, se diseñó una prueba clasificada como positiva, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 27 alcanzando una aprobación del 100%.

Tabla 27 – Ejecución de Pruebas Caso de Uso CU09.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU09: Subir Cajero	Total de Casos Diseñados	1
		Pasaron	1
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU10: Ubicar Cajero, como se muestra en la Tabla 42 del Anexo1, se diseñó una prueba clasificada como positiva, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 28 alcanzando una aprobación del 100%.

Tabla 28 – Ejecución de Pruebas Caso de Uso CU10.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU10: Ubicar Cajero	Total de Casos Diseñados	1
		Pasaron	1
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	NotStarted
		%Ejecución del Ciclo	100%
		%Aprobación	100%

5.3.2. Reporte de Pruebas Funcionales de la Aplicación

Android

Los Casos de Uso seleccionados para esta prueba fueron:

- Elegir Banco
- Elegir Redes Asociadas
- Consultar Cajero Cercano por Banco
- Consultar Cajero Cercano por Red Bancaria

Se contemplaron los diversos escenarios que pueden presentarse para cada caso de uso.

En el Anexo 2 se muestra cada uno de los Diseños de Casos de Pruebas, donde se especifica para cada prueba, las precondiciones, la descripción, los pasos a realizar, los resultados esperados, el tipo de caso de prueba, la clasificación y resultado, además de la fecha en que se realizó la prueba.

Para el Caso de Uso CU11: Elegir Banco, como se muestra en la Tabla 43 del Anexo2, se diseñó una prueba clasificada como positiva, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 29 alcanzando una aprobación del 100%.

Tabla 29 – Ejecución de Pruebas Caso de Uso CU11.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU11: Elegir Banco	Total de Casos Diseñados	1
		Pasaron	1
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU12: Elegir Redes Asociadas, como se muestra en la Tabla 44 del Anexo2, se diseñó una prueba clasificada como positiva, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 30 alcanzando una aprobación del 100%.

Tabla 30 – Ejecución de Pruebas Caso de Uso CU12.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU12: Elegir Redes Asociadas	Total de Casos Diseñados	1
		Pasaron	1
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	Not Started
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU13: Consultar Cajero cercano por Banco, como se muestra en la Tabla 45 del Anexo2, se diseñó una prueba clasificada como positiva, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 31 alcanzando una aprobación del 100%.

Tabla 31 – Ejecución de Pruebas Caso de Uso CU13.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU13: Consultar Cajero cercano por Banco	Total de Casos Diseñados	1
		Pasaron	1
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	NotStarted
		%Ejecución del Ciclo	100%
		%Aprobación	100%

Para el Caso de Uso CU14: Consultar Cajero cercano por Red Bancaria, como se muestra en la Tabla 46 del Anexo2, se diseñó una prueba clasificada como positiva, logrando un resultado Aceptado. De esta manera obtenemos los datos que se detallan en la Tabla 32 alcanzando una aprobación del 100%.

Tabla 32 – Ejecución de Pruebas Caso de Uso CU14.

Ciclo 1 – Ejecución de Pruebas	Caso de Uso CU14: Consultar Cajero cercano por Red Bancaria	Total de Casos Diseñados	1
		Pasaron	1
		Pasaron regresión	No Aplica
		Fallaron	0
		No probado	0
		No Aplica	0
		Status	NotStarted
		%Ejecución del Ciclo	100%
		%Aprobación	100%

5.4. REPORTE DE PRUEBAS DE ESTRÉS

Prueba Cajeros Cercanos

El Servicio de cajeros cercanos es de vital importancia para nuestra aplicación ya que este es el que nos devuelve la posición geográfica de los 10 cajeros más cercanos al usuario, su dirección y redes de cobro asociadas. Siendo este el más utilizado dentro de la aplicación.

Probamos los siguientes escenarios:

Pequeña Cantidad de Usuarios

Para este requerimiento hemos simulado 10 usuarios simultáneos repitiendo el requerimiento 5 veces por lo que nuestro número de muestras será 50.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec
Peticion Cajer.	50	613	490	1066	1126	1269	308	1269	0,00%	12,7/sec	36,5
Total	50	613	490	1066	1126	1269	308	1269	0,00%	12,7/sec	36,5

Figura 5.3 – Resultados luego de haber realizado las peticiones.

Como podemos observar en la Figura 5.3, el sistema es muy rápido para este caso obteniendo una Media de 513 milisegundos y un porcentaje de error del 0%. Dando a notar que necesitamos muchos más requerimientos para poder llegar a sus límites.

Gran cantidad de Usuarios

Para este requerimiento hemos simulados 300 usuarios simultáneos repitiendo el requerimiento 100 veces, por lo que nuestro número total de requerimientos será 3000. En la Figura 5.4 se muestran los valores arrojados por el sistema.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec
Peticion HTTP	3000	2307	2185	3178	3525	4086	952	6226	0,00%	40,4/sec	116,5
Total	3000	2307	2185	3178	3525	4086	952	6226	0,00%	40,4/sec	116,5

Figura 5.4 – Resultados de la petición de tipo Http.

Se indica en la Figura 5.4 que a pesar de realizarse 3000 peticiones al sistema de forma continua, no se obtuvo porcentaje de error, el rendimiento fue de 40.4 peticiones/segundo y mantuvo una media por requerimiento de 2.18 segundos. Lo que es aún aceptable para el usuario final ya que tendría que esperar 2 segundos para poder visualizar sus datos en la aplicación.

En cuanto al tiempo de respuesta promedio (línea azul), tal como se indica en la figura siguiente, se muestra como varía de acuerdo al tiempo, además podemos visualizar en la Figura 5.5 el rendimiento (línea negra) medido en requerimientos/segundo.

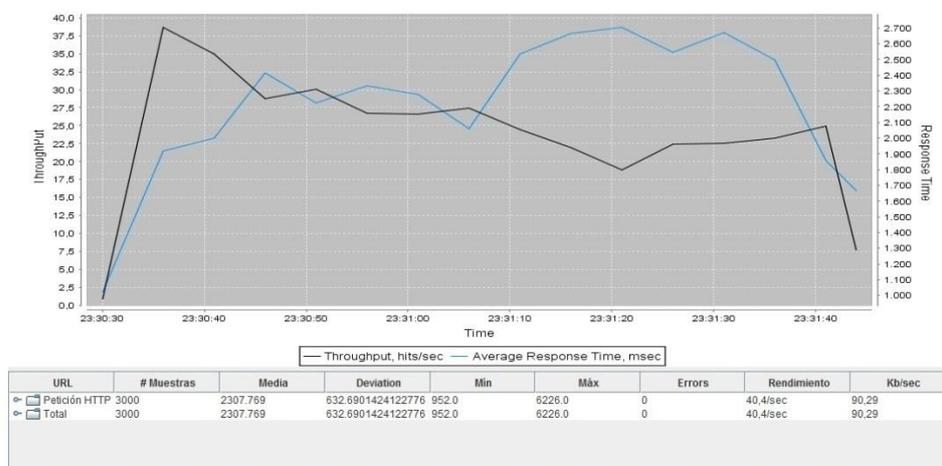


Figura 5.5 – Tiempo de respuesta y el tiempo medio según las pruebas.

Esto pone en evidencia el alto nivel de efectividad por parte de la aplicación, así como la disponibilidad de la aplicación web, que a

pesar de realizarle 3000 consultas de forma simultánea, el rendimiento es muy bueno, y el tiempo de respuesta aceptable.

Prueba Listado Bancos

El Servicio de listado bancos nos devuelve todos los bancos que se encuentren activos en el sistema web. Para posteriormente permitir su filtrado en la aplicación móvil. Probamos los siguientes escenarios:

Gran Cantidad de Usuarios

Para este requerimiento hemos simulados 200 usuarios simultáneos repitiendo el requerimiento 100 veces, por lo que nuestro número total de requerimientos será 2000. En la Figura 5.6 se muestran los valores arrojados por el sistema.

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec
Petición Banc.	2000	1325	1305	1624	1752	2297	435	2808	0.00%	34.1/sec	41.1
Petición Cajér.	2000	1501	1463	1823	2013	2417	519	2541	0.00%	33.6/sec	96.8
Total	4000	1413	1382	1738	1914	2409	435	2808	0.00%	66.6/sec	136.2

Figura 5.6 – Simulación con 200 usuarios simultáneos.

Cómo podemos observar en la Figura 5.7, a pesar de haber realizado 2000 peticiones al sistema de forma continua, no tuvo porcentaje de error y un buen tiempo de respuesta promedio de 1.32 segundos. Lo que es de aceptable para el usuario final ya que tendría que esperar 1 segundo para cargar sus datos.

De igual forma podemos observar su tiempo de respuesta promedio (línea azul) y su rendimiento (línea negra) medido en requerimientos/segundo.

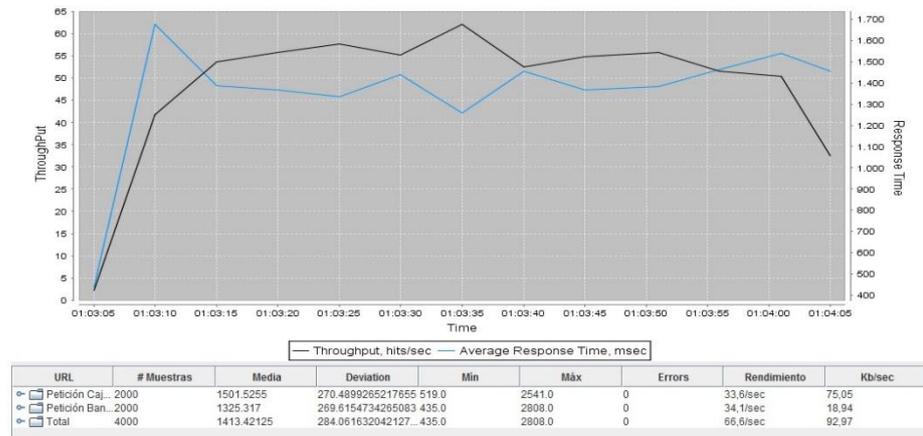


Figura 5.7 – Tiempo de respuesta y tiempo medido en requerimientos/segundo.

CONCLUSIONES Y RECOMENDACIONES

Al finalizar el desarrollo de la aplicación, y en base a la experiencia luego de la investigación necesaria para la realización de la misma, se puede concluir que:

1. Se logró almacenar toda la información referente a cajeros y Bancos, en una aplicación web, lo cual nos brinda mayor disponibilidad de dicha información.

2. Se hace uso de una interfaz sencilla, con la finalidad de tener legibilidad en cualquier dispositivo en el cual se visualice la aplicación.
3. La interacción que puede realizar el usuario con la aplicación, la navegación dentro del mapa de la ciudad, así como la distinción de los cajeros de acuerdo a la entidad bancaria, se hace de la forma más sencilla posible, con el fin de facilitar su uso.
4. Esta aplicación será de gran utilidad, especialmente para personas ajenas a la ubicación desde donde se realiza la consulta.
5. La aplicación, al ser desarrollada para dispositivos móviles que tengan como sistema operativo Android 4.0 o superior, brinda la posibilidad de ser utilizada en la mayoría de los dispositivos en uso a nivel mundial.
6. La aplicación hace uso del mapa proporcionado por Google Maps, lo cual permite que se pueda implementar en cualquier ubicación a nivel mundial.

Dependiendo el tipo de persona que desee hacer uso de la aplicación, pudiendo ser estas: usuarios o nuevas entidades bancarias, se recomienda:

1. Antes de realizar la consulta, verificar en su dispositivo móvil, que este encendido el sistema GPS del equipo.
2. Tomar en cuenta que al realizar la consulta, es probable que exista un consumo mínimo de sus datos móviles.
3. Al momento de ingresar un nuevo cajero al sistema, se debe tomar en cuenta que el banco al que será anexado, debe estar registrado previamente en el sistema.
4. Tomar en cuenta que la información de los cajeros, así como la ubicación de los mismos, es la proporcionada por la entidad bancaria o por la investigación realizada al momento del desarrollo de la aplicación (2015).
5. La configuración del API de Google Maps para Android requiere cierto nivel de conocimientos técnicos, por lo que se hace necesario consultar los pasos necesarios en portales confiables.

6. La aplicación podría ser mejorada de contenido, por ejemplo incluir información de todos los ATMS de las distintas ciudades del país.
7. La aplicación podría incorporar la función de que el usuario móvil pueda añadir un ATM no registrado en la base de datos.
8. La aplicación podría incluir publicidad de cada Institución Bancaria registrada, previo convenio con los Bancos, para de esta manera obtener beneficios económicos.
9. La aplicación podría ser subida a Google Play, para que los usuarios puedan descargarla de forma gratuita.

GLOSARIO

Aplicación: programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos.

Usuario: persona que utiliza algún tipo de objeto o que es destinataria de un servicio.

Algoritmo: conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos.

Framework: estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

Kernel: Parte esencial de un sistema operativo que provee los servicios más básicos del sistema.

Open Source: programas informáticos que permiten el acceso a su código de programación.

BIBLIOGRAFÍA

- [1]. «La verdadera Historia de Android-Llega Android 1.0 Apple Pie (2007),» Androidsis [En línea]. Disponible: <http://www.androidsis.com/la-verdadera-historia-de-android-llega-android-1-0-apple-pie-2007/>. [Último acceso: Febrero 2013].
- [2]. Imnmandroid, «Banana Bread (v1.1),» Versiones de Android [En línea]. Disponible: <https://Imnmandroid.wordpress.com/android-1-0/banana-bread-v1-1/>. [Último acceso: Julio 2013].
- [3]. Facilísimo, «Cupcake, Android 1.5,» Android [En línea]. Disponible: http://tecnologia.facilísimo.com/blogs/moviles/android/cupcake-android-1-5_629089.html. [Último acceso: Marzo 2011].
- [4]. Aplicaciones en android, «Características de Android Donut 1.6,» Android Donut 1.6: Especificaciones y características [En línea]. Disponible: <http://www.aplicacionesenandroid.com/android-donut-1-6-especificaciones-y-caracteristicas/>. [Último acceso: Octubre 2013].
- [5]. Aplicaciones en android, « Android Eclair 2.0 2.1: Especificaciones y características,» [En línea]. Disponible: <http://www.aplicacionesenandroid.com/android-eclair-2-0-2-1-especificaciones-y-caracteristicas/>. [Último acceso: 2013].
- [6]. Genbeta «Android 2.2 Froyo: Un repaso a las principales novedades,» [En línea]. Disponible: <http://www.genbeta.com/sistemas-operativos/android-22-froyo-un-repaso-a-las-principales-novedades> [Último acceso: Mayo 2010].

- [7]. AndroidZone, «Android 2.3 Gingerbread: Características y Novedades,» Novedades Android 2.3 Gingerbread [En línea]. Disponible: <http://androidzone.org/2010/12/android-2-3-gingerbread-novedades/>. [Último acceso: Diciembre 2010].
- [8]. Aplicaciones en android, «Características Android HoneyComb 3.0 3.1 3.2,» Android Honeycomb 3.0 3.1 3.2: Especificaciones [En línea]. Disponible: <http://www.aplicacionesenandroid.com/android-honeycomb-3-0-3-1-3-2-especificaciones/>. [Último acceso: Octubre 2013].
- [9]. «Android 4.0.x Ice Cream Sandwich,» Anexo:Historial de versiones de Android [En línea]. Disponible: http://es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android. [Último acceso: Octubre 2011].
- [10]. Muy Android, «Conoce las características que trae el Android 4.1 Jelly Bean,» [En línea]. Disponible: <http://muyandroid.net/conoce-las-caracteristicas-que-trae-el-android-4-1-jelly-bean/>. [Último acceso: Octubre 2013].
- [11]. Todoandroid.es, « Que es Jelly Bean 4.1: novedades y características de esta versión de android,» [En línea]. Disponible: <http://www.todoandroid.es/index.php/faq-de-android/65-versiones/790-que-es-jelly-bean-41-novedades-y-caracteristicas-de-esta-version-de-android.html>. [Último acceso: Febrero 2013].
- [12]. Andro 4 All, « Android 4.4 KitKat ya es oficial, ¡te contamos todas las novedades!,» [En línea]. Disponible: <http://andro4all.com/2013/10/android-4-4-kitkat>. [Último acceso: Abril 2014].

[13]. Todoandroid.es, «Qué es Android 5.0 Lollipop: novedades y características de esta versión de android,» [En línea]. Disponible: <http://www.todoandroid.es/index.php/faq-de-android/65-versiones/1672-que-es-android-5-0-lollipop-novedades-y-caracteristicas-de-esta-version-de-android.html>. [Último acceso: Marzo 2015].

[14]. Fandroides, « QUÉ ES Y PARA QUÉ SIRVE EL SDK,» [En línea]. Disponible: <http://www.fandroides.com/que-es-y-para-que-sirve-el-sdk/>. [Último acceso: Abril 2014].

[15]. Camino Geek, «APACHE JMETER; REALIZA PRUEBA DE COMPORTAMIENTO A TUS SERVIDORES,» [En línea]. Disponible: <http://www.caminogeek.com/apache-jmeter-realiza-prueba-de-comportamiento-a-sus-servidores/>. [Último acceso: Mayo 2013].

[16]. Xataka Android, «¿Qué es Android?,» [En línea]. Disponible: <http://www.xatakandroid.com/sistema-operativo/que-es-android/>. [Último acceso: Febrero 2011].

[17]. Cake PHP, «Entendiendo el Modelo - Vista - Controlador,» [En línea]. Disponible: <http://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>. [Último acceso: Mayo 2013].

[18]. Wikipedia, «Google App Engine,» [En línea]. Disponible: http://es.wikipedia.org/wiki/Google_App_Engine. [Último acceso: Abril 2015].

[19]. Masadelante.com, «¿Qué es una Base de datos? - Definición de Base de datos,» [En línea]. Disponible: <https://www.masadelante.com/faqs/base-de-datos>. [Último acceso: 2015].

ANEXOS

ANEXO 1

Pruebas de Funcionalidad del Sistema Web

Tabla 33 – Diseño de Casos de Pruebas, Caso de Uso: CU01 Ingresar Banco.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU01: Ingresar Banco								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	*Usuario con acceso al sistema	Validar ingreso de bancos de acuerdo a la variables Nombre y Estado	<ul style="list-style-type: none"> * Sistema solicita el ingreso del nombre para el Banco a ingresar. * Registra el nombre. * Sistema solicita seleccionar el estado para el Banco a ingresar. * Una vez 	<ul style="list-style-type: none"> * Sistema guarda el Banco ingresado según datos especificados. * Sistema muestra el nuevo Banco en la lista de Bancos registrados. * Sistema muestra mensaje de 	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

				registrados los datos, se da clic sobre el botón "Guardar"	registro exitoso.					
2	07-04-15	*Usuario con acceso al sistema	Validar que se ingrese el nombre del Banco a ingresar	<p>* Sistema solicita el ingreso del nombre para el Banco a ingresar.</p> <p>* No se ingresa un nombre.</p> <p>* Sistema solicita seleccionar el estado para el Banco a ingresar.</p> <p>* Una vez registrados los datos, se da clic sobre el botón "Guardar"</p>	<p>* Sistema no realiza el ingreso del Banco.</p> <p>* Sistema solicita de forma obligatoria un nombre.</p> <p>* Sistema muestra mensaje de registro no exitoso.</p>	Reglas de negocio.	Negativo	07-04-15	Aceptado	1

3	07-04-15	*Usuario con acceso al sistema	Validar el ingreso de caracteres especiales en el campo "Nombre"	<ul style="list-style-type: none"> * Sistema solicita el ingreso del nombre para el Banco a ingresar. * Se insertan caracteres especiales en el campo nombre * Sistema solicita seleccionar el estado para el Banco a ingresar. * Una vez registrados los datos, se da clic sobre el botón "Guardar" 	<ul style="list-style-type: none"> * Sistema no realiza el ingreso del Banco. * Sistema solicita que se vuelva a ingresar el nombre sin caracteres especiales. * Sistema muestra mensaje de registro no exitoso. 	Reglas de negocio.	Negativo	07-04-15	Aceptado	1
---	----------	--------------------------------	--	--	---	--------------------	----------	----------	----------	---

Tabla 34 – Diseño de Casos de Pruebas, Caso de Uso: CU02 Consultar Banco.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU02: Consultar Banco								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	*Existen Bancos ingresados al sistema	Mostrar los Bancos existentes en el sistema	* Escoger la opción Banco en la barra de menú del Sistema * Seleccionar la opción Consultar	* Se mostrará un listado con todos los bancos existentes en el sistema, con su respectivo estado, además de las opciones de editar y eliminar.	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

Tabla 35 – Diseño de Casos de Pruebas, Caso de Uso: CU03 Editar Banco.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU03: Editar Banco								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	*El Banco a editar existe	Editar uno o más campos del Banco seleccionado	*Mostrar todos los bancos a través de la página Consultar. *Escoger el Banco que se desea editar dando clic en el botón  *Modificar los campos que se deseen editar. *Dar clic sobre el botón Guardar.	* Sistema guardará los cambios especificados. *Sistema muestra el listado de todos los bancos, incluyendo el banco editado. *Sistema muestra un mensaje indicando modificación exitosa	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

2	07-04-15	*El Banco a editar existe	Validar que estén llenos los campos para modificar un Banco	<p>*Mostrar todos los bancos a través de la página Consultar.</p> <p>*Escoger el Banco que se desea editar dando clic en el botón </p> <p>* No se llena el campo Nombre.</p> <p>*Dar clic sobre el botón Guardar.</p>	<p>* Sistema no realiza la modificación del Banco.</p> <p>* Sistema solicita que se llene el campo Nombre de forma obligatoria.</p> <p>* Sistema muestra mensaje de modificación no exitosa.</p>	Reglas de negocio.	Negativo	07-04-15	Aceptado	1
---	----------	---------------------------	---	--	--	--------------------	----------	----------	----------	---

3	07-04-15	*El Banco a editar existe	Validar el nombre del Banco	<p>*Mostrar todos los bancos a través de la página Consultar. *Escoger el Banco que se desea editar dando clic en el botón </p> <p>* Se ingresa un caracter especial en el campo Nombre. *Dar clic sobre el botón Guardar.</p>	<p>* Sistema no realiza la modificación del Banco. * Sistema solicita el ingreso de los valores correctos, sin caracteres especiales. * Sistema muestra mensaje de modificación no exitosa.</p>	Reglas de negocio.	Negativo	07-04-15	Aceptado	1
---	----------	---------------------------	-----------------------------	---	---	--------------------	----------	----------	----------	---

Tabla 36 – Diseño de Casos de Pruebas, Caso de Uso: CU04 Eliminar Banco.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU04: Eliminar Banco								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	*El Banco a eliminar existe	Eliminar un Banco del sistema	*Mostrar todos los bancos a través de la página Consultar *Escoger el Banco que se desea eliminar dando clic en el botón . *Sistema solicita confirmación para eliminar el Banco *Dar clic sobre el botón Aceptar .	* Sistema elimina el Banco de su listado	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

Tabla 37 – Diseño de Casos de Pruebas, Caso de Uso: CU05 Ingresar Cajero.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU05: Ingresar Cajero								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	* El Banco al que pertenece el Cajero existe	Ingresar un nuevo Cajero	*Escoger la opción: Cajero . *Escoger la opción: Nuevo . *Seleccionar el Banco al que pertenece el cajero. *Llenar los campos con los datos para el nuevo cajero en las cajas de texto. * Seleccionar la o las redes a la que pertenece este cajero, dando clic sobre las casillas presentadas. *Clic sobre el botón Guardar .	* Sistema registra exitosamente el nuevo cajero. * Sistema redirecciona a la página Consultar . * Sistema muestra mensaje de ingreso exitoso del cajero.	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

2	07-04-15	* El Banco al que pertenece el Cajero existe	Validar el ingreso de todos los campos requeridos	<p>*Escoger la opción: Cajero.</p> <p>*Escoger la opción: Nuevo.</p> <p>*Seleccionar el Banco al que pertenece el cajero.</p> <p>*No llenar uno o más de los campos solicitados.</p> <p>* Seleccionar la o las redes a la que pertenece este cajero, dando clic sobre las casillas presentadas.</p> <p>*Clic sobre el botón Guardar.</p>	<p>* Sistema no registra el nuevo cajero.</p> <p>* Sistema muestra mensaje solicitando se llene el o los campos que faltan.</p> <p>* Sistema muestra mensaje de ingreso no exitoso del cajero.</p>	Reglas de negocio.	Negativo	07-04-15	Aceptado	1
---	----------	--	---	---	--	--------------------	----------	----------	----------	---

3	07-04-15	* El Banco al que pertenece el Cajero existe	Validar el ingreso de caracteres especiales en los campos requeridos	<p>*Escoger la opción: Cajero.</p> <p>*Escoger la opción: Nuevo.</p> <p>*Seleccionar el Banco al que pertenece el cajero.</p> <p>*Al ingresar los datos requeridos, se hace el ingreso de un caracter especial.</p> <p>* Seleccionar la o las redes a la que pertenece este cajero, dando clic sobre las casillas presentadas.</p> <p>*Clic sobre el botón Guardar.</p>	<p>* Sistema no registra el nuevo cajero.</p> <p>* Sistema muestra mensaje solicitando se llene el o los campos sin caracteres especiales.</p> <p>* Sistema muestra mensaje de ingreso no exitoso del cajero.</p>	Reglas de negocio.	Negativo	07-04-15	Aceptado	1
---	----------	--	--	--	---	--------------------	----------	----------	----------	---

Tabla 38 – Diseño de Casos de Pruebas, Caso de Uso: CU06 Consultar Cajero.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU06: Consultar Cajero								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	* El Cajero a consultar existe	Consultar los cajeros de un determinado Banco en el Sistema	*Escoger la opción: Cajero . *Escoger la opción: Consultar . *Seleccionar el Banco al que pertenece el cajero. *Clic en el botón Consultar .	* Sistema muestra el listado de cajeros de acuerdo al Banco seleccionado	Reglas de negocio	Positivo	07-04-15	Aceptado	1

2	07-04-15	* El Cajero a consultar existe	Consultar la información de un cajero en específico	<p>*Escoger la opción: Cajero.</p> <p>*Escoger la opción: Consultar.</p> <p>*Seleccionar el Banco al que pertenece el cajero.</p> <p>*Clic en el botón Consultar.</p> <p>*Clic sobre el nombre del Cajero que se desea consultar</p>	* Sistema muestra la información relacionada al cajero seleccionado	Reglas de negocio	Positivo	07-04-15	Aceptado	1
---	----------	--------------------------------	---	---	---	-------------------	----------	----------	----------	---

Tabla 39 – Diseño de Casos de Pruebas, Caso de Uso: CU07 Editar Cajero.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU07: Editar Cajero								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	*El Cajero a editar existe	Editar uno o más campos del Cajero seleccionado	<p>*Mostrar todos los Cajeros a través de la página Consultar, del menú Cajero.</p> <p>*Escoger el Cajero que se desea editar dando clic en el botón </p> <p>*Modificar los campos que se deseen editar</p> <p>*Dar clic sobre el botón Guardar</p>	<p>* Sistema guardará los cambios especificados</p> <p>. *Sistema muestra el listado de todos los cajeros, incluyendo el Cajero editado.</p> <p>*Sistema muestra un mensaje indicando modificación exitosa</p>	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

2	07-04-15	*El Cajero a editar existe	Validar que todos los campos del Cajero a editar contengan datos	<p>*Mostrar todos los Cajeros a través de la página Consultar, del menú Cajero.</p> <p>*Escoger el Cajero que se desea editar dando clic en el botón </p> <p>* Uno o más de un campo se quedan sin llenar.</p> <p>*Dar clic sobre el botón Guardar.</p>	<p>* Sistema no guardará los cambios especificados</p> <p>* Sistema solicita que se llene el o los campos faltantes.</p> <p>* Sistema muestra un mensaje indicando modificación no exitosa</p>	Reglas de negocio.	Negativo	07-04-15	Falló	1
---	----------	----------------------------	--	---	--	--------------------	----------	----------	-------	---

3	07-04-15	*El Cajero a editar existe	Validar el ingreso correcto de datos al modificar un Cajero	<p>*Mostrar todos los Cajeros a través de la página Consultar, del menú Cajero.</p> <p>*Escoger el Cajero que se desea editar dando clic en el botón </p> <p>*En uno o más campos se ingresan caracteres especiales.</p> <p>*Dar clic sobre el botón Guardar.</p>	<p>* Sistema no guardará los cambios especificados</p> <p>. *Sistema solicita que se llene de forma correcta el o los campos que están incorrectos.</p> <p>*Sistema muestra un mensaje indicando modificación no exitosa</p>	Reglas de negocio.	Negativo	07-04-15	Falló	1
---	----------	----------------------------	---	---	--	--------------------	----------	----------	-------	---

Tabla 40 – Diseño de Casos de Pruebas, Caso de Uso: CU08 Eliminar Cajero.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU08: Eliminar Cajero								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	*El Cajero a editar existe	Eliminar un Cajero del sistema	*Mostrar todos los cajeros a través de la página Consultar, del menú Cajero. *Escoger el cajero que se desea eliminar dando clic en el botón  *Sistema solicita confirmación para eliminar el Cajero. *Dar clic sobre el botón Aceptar .	* Sistema elimina el Cajero de su listado	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

Tabla 41 – Diseño de Casos de Pruebas, Caso de Uso: CU09 Subir Cajero.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU09: Subir Cajero								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	*El ID del Banco al que pertenecen los cajeros debe existir	Asignar una lista de cajeros, definida en un archivo externo, a un determinado Banco	*Escoger la opción: Cajero *Escoger la opción: Subir *Escoger la opción: Seleccionar archivo . * Seleccionar desde el directorio el archivo que contiene el listado de cajeros *Dar clic sobre el botón Abrir . *Dar clic sobre el botón Subir .	* Sistema asigna el listado de cajeros de la lista al banco indicado. *Sistema muestra un mensaje de confirmación.	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

Tabla 42 – Diseño de Casos de Pruebas, Caso de Uso: CU10 Ubicar Cajero.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU10: Ubicar Cajero								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	*El cajero al que se le va a asignar la posición, debe existir	Ubicar el marcador de cada Cajero en la posición deseada dentro del mapa de Google Maps.	<ul style="list-style-type: none"> *Escoger la opción: Mapa * Seleccionar el Banco al cual pertenece el cajero. * Dar clic en Consultar. * Dar clic en el botón  . * Seleccionar el Cajero deseado. * Mover el marcador a la posición deseada. * Dar clic en el botón Guardar. 	<ul style="list-style-type: none"> - * Sistema asigna la posición geográfica al cajero, mediante latitud y longitud. *Sistema muestra un mensaje de confirmación. 	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

ANEXO 2

Pruebas de Funcionalidad de la Aplicación Android

Tabla 43 – Diseño de Casos de Pruebas, Caso de Uso: CU11 Elegir Banco.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU11: Elegir Banco								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	Ninguna	Seleccionar el Banco del cual se desea conocer los cajeros más cercanos	* Iniciar la aplicación. * Escoger en el menú la opción Bancos.	* El sistema muestra un listado con los nombres de los bancos disponibles	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

Tabla 44 – Diseño de Casos de Pruebas, Caso de Uso: CU12 Elegir Redes asociadas.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU12: Elegir redes asociadas								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	Ninguna	Seleccionar las redes de las cuales se desea conocer los cajeros más cercanos	* Iniciar la aplicación. * Escoger en el menú la opción Redes.	* El sistema muestra un listado con los nombres de las redes disponibles	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

Tabla 45 – Diseño de Casos de Pruebas, Caso de Uso: CU13 Consultar Cajeros cercanos por Banco.

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU13: Consultar cajeros cercanos por Banco								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	Escoger un Banco	Consultar los cajeros más cercanos de acuerdo al Banco	<ul style="list-style-type: none"> * Iniciar la aplicación. * Escoger en el menú la opción Bancos. * Marcar la casilla correspondiente al Banco o los Bancos de los cuales se desea consultar. * Pulsar sobre el botón Ok. 	<ul style="list-style-type: none"> * El sistema muestra mediante un mapa proporcionado por Google Maps, la ubicación geográfica de los 10 cajeros más cercanos a su posición. * Se puede además navegar en el mapa, se mostrará de forma automática los 10 cajeros más cercanos a esa posición. 	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

Tabla 46 – Diseño de Casos de Pruebas, Caso de Uso: CU14 Consultar Cajeros cercanos por Red Bancaria

Diseño Casos de Pruebas							Clasificación	Ciclo 1 – Ejecución de pruebas		
Caso de Uso CU14: Consultar cajeros cercanos por Red Bancaria								Fecha	Resultado	N Incidente
ID	Fecha	Precondiciones	Descripción	Pasos	Resultado Esperado	Tipo Caso de Prueba				
1	07-04-15	Escoger Red Bancaria	Consultar los cajeros más cercanos de acuerdo la red bancaria a la que pertenece	<ul style="list-style-type: none"> * Iniciar la aplicación. * Escoger en el menú la opción Redes Bancarias. * Marcar la casilla correspondiente a la o las redes de las cuales se desea consultar. * Pulsar sobre el botón Ok. 	<ul style="list-style-type: none"> * El sistema muestra mediante un mapa proporcionado por Google Maps, la ubicación geográfica de los 10 cajeros más cercanos a su posición. * Se puede además navegar en el mapa, se mostrará de forma automática los 10 cajeros más cercanos a esa posición. 	Reglas de negocio.	Positivo	07-04-15	Aceptado	1

