Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

GUÍA TURÍSTICA DE LA CIUDAD DE CUENCA PARA DISPOSITIVOS MÓVILES ANDROID

Trabajo de graduación previo a la obtención del título de:

"Ingeniero en Sistemas y Telemática"

Autores:

Jorge Luis Calle Castro

Milton Fabián Largo Anguisaca

Director:

MsC. Diego Pacheco Prado

Cuenca – Ecuador

2017

DEDICATORIA

Este trabajo va dedicado en primer lugar a Dios quien hace posible que pueda mencionar a mis padres "***Miguel***" y "***Blanca***" quienes me han apoyado incondicionalmente a lo largo de mi vida en mis estudios, gracias por haberme enseñado el sacrificio y esfuerzo para obtener cada logro y haber hecho de mí un hombre de bien basado en los valores que nos han enseñado. A mi familia "***Mary***", "***Noelia***" y mi pequeña "***Karlita***", quienes son el pilar fundamental de mi vida y es por ellas por quien lucho día a día. A mis hermanos "***Adrián***" y "***Cristina***" por su apoyo y por siempre estar a mi lado.

A mi abuelita "***Digna***" por haber cuidado de mí y darme todo su amor y a alguien muy especial que desde el cielo estará muy orgulloso de este logro mi abuelito "***Manuel***".

A toda mi familia y amigos gracias por su apoyo y por estar pendientes que logre cumplir este objetivo.

Jorge Luis Calle Castro

A Dios.

Por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A mi madre ***Mayela***.

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien.

A mi padre ***Milton***.

Por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y que nunca me dé por vencido ante algún problema.

A mis hermanos ***Rommel y Alex.***

Por sus palabras de aliento que me ayudaron a seguir adelante, tomando las decisiones correctas en mi periodo estudiantil.

Milton Fabián Largo Anguisaca

AGRADECIMIENTOS

Al MsC. Diego Pacheco Prado Director de nuestra tesis, quien nos ha guiado durante el desarrollo del presente proyecto de graduación.

A mi gran amigo Fabián Largo quien me ha ayudado con su conocimientos y hemos podido salir adelante con este proyecto y cumplir con los objetivos planteados.

A todos los profesores de la Universidad del Azuay, quienes durante los años de estudio nos han impartido sus conocimientos.

Jorge Luis Calle Castro

Al Msc. Diego Pacheco por su tiempo y apoyo durante la etapa de desarrollo del proyecto, ya que con sus conocimientos y recomendaciones logramos cumplir el objetivo planteado.

A mi compañero Eduardo Gómez por compartir sus conocimientos de manera desinteresada y apoyarme durante el trascurso de desarrollo de este proyecto.

A Ornolis Vázquez que además de ser un buen amigo es un gran profesional que gracias a su experiencia supo recomendarme y aconsejarme las mejores prácticas para el desarrollo del presente proyecto

A Jorge Calle un gran amigo y compañero gracias por tu apoyo y tus conocimientos puestos en práctica en este proyecto

Milton Fabián Largo Anguisaca

INDICE DE CONTENIDOS

DEDICATORIA ii

AGRADECIEMIENTOS iv

INDICE DE CONTENIDOS ivi

INDICE DE FIGURAS ivxii

INDICE DE TABLAS xviv

RESUMEN ivii

ABSTRACT iviii

CAPÍTULO 1: ANÁLISIS DE LOS DISPOSITIVOS MÓVILES 1

Introducción 1

1.1 Versiones de sistemas operativos para móviles 1

1.1.1 Android 2

1.1.2 IOS 4

1.1.3 Windows Phone 5

1.2 Características de los dispositivos móviles 6

1.2.1 Portabilidad 7

1.2.2 Capacidad de procesamiento 7

1.2.3 Conexión a internet 8

1.2.4 Capacidad de almacenamiento 8

1.2.5 Capacidad de interacción 8

1.3 Tipos de dispositivos móviles 9

1.3.1 Dispositivos de gama baja 9

1.3.2 Dispositivos de gama alta 9

1.4 Redes de dispositivos móviles 10

1.5 Cantidad de dispositivos móviles por sistema operativo 10

1.6 Justificación del uso de la plataforma seleccionada 11

1.7 Geolocalización 12

1.7.1 Métodos de localización de un dispositivo Android 13

1.7.1.1 Alta precisión 13
1.7.1.2 Ahorro de Energía 13
1.7.1.3 Solo dispositivo 13
CAPÍTULO 2: ANÁLISIS DE HERRAMIENTAS Y ESTÁNDARES 14
Introducción 14
2.1 Mapas Offline 14
2.2 Mapas Online 15
2.3 Open StreetMap (OSM) 15
2.4 Google Maps 19
2.5 Realidad Aumentada 21
2.5.1 Métodos de activación de realidad Aumentada 22
2.5.2 Componentes de un sistema de realidad aumentada 23
2.5.3 Tipos de Realidad Aumentada 25
2.6 Algoritmos de ruteo 25
2.6.1 Representación de una red de caminos en forma de grafo 26
2.6.2 Algoritmos de enrutamiento 27
2.6.2.1 Algoritmos clásicos: 27
2.6.2.2 Algoritmos jerárquicos: 27
2.6.2.3 Algoritmos dirigidos al objetivo: 28
2.6.2.4. Algoritmos evolutivos: 28
CAPÍTULO 3 DESARROLLO DE SISTEMA 29
Introducción 29
3.1 Herramientas y tecnologías utilizadas 29
3.1.1 Android Studio 29
3.1.1.1 Estructura de un proyecto 29
3.1.1.2 Interfaz de Usuario 31
3.1.2 MySql Workbench 32
3.1.3 Base de datos SQLite 32
3.1.4 Método de autenticación OAUTH 36

3.1.5 Material Design 36
3.1.5.1 Objetivo de Material Design 37
3.1.5.2 Principios de Material Design 37
3.1.6 Retrofit - comunicación con el servidor 38
3.1.6.1 Estructura de un cliente REST 39
3.1.6.2 Construcción del cliente REST: 40
3.1.7 NinjaMock – Prototipo de App 41
3.1.8 NetBeans 41
3.1.9 PhayPhone 42
3.1.10 Implementación de Google Maps 45
3.1.11 Implementación de Google Directions API 56
3.1.12 Implementación de Street View API 58
3.1.13 Implementación de HockeyApp 60
3.1.14 Realidad Aumentada 62
3.1.14.1 Implementación de la Clase GeoObject 63
3.1.14.2 Creación de un marcador 64
3.1.14.3 Implementación de la función LocationChanged 65
3.1.14.4 Creación de multiples marcadores 65
3.2 Fase de Análisis de la aplicación 66
3.2.1 Requerimientos funcionales 66
3.2.2 Requerimientos no funcionales 67
3.2.2.1 Seguridad 67
3.2.2.2 Disponibilidad 67
3.2.2.3 Mantenibilidad 68
3.2.2.4 Portabilidad 68
3.2.2.5 Operabilidad 68
3.2.3 Restricciones del aplicativo 68
3.2.4 Arquitectura de la aplicación 69
3.2.5 Fase de diseño 70

3.2.5.1 Jerarquía de Usuarios 70
3.2.5.2 Diagramas de Caso de Uso 71
3.2.5.2.1 Caso de Uso General de la aplicación 72
3.2.5.2.2 Especificación Caso de Uso General de la aplicación 72
3.2.5.2.3 Caso de Uso Turista 73
3.2.5.2.4 Especificación Caso de Uso Turista 74
3.2.5.2.5 Especificación Caso de Uso Login Turista 75
3.2.5.2.6 Caso de Uso Dueño de Local 76
3.2.5.2.7 Especificación Caso de Uso Proceso Pago 76
3.2.5.2.8 Caso de Uso Actividad página web 77
3.2.5.2.9 Especificación Caso de Uso actividad página web 78
3.2.5.3 Diagrama de Clases 78
3.2.5.4 Diagrama de Secuencias 80
3.2.5.5 Diagrama de Navegación 80
3.2.5.5.1 Diagrama de navegación de aplicación 82
3.2.5.5.2 Diagrama de navegación de página web 82
3.2.5.6 Diseño de los prototipos de las pantallas de la aplicación 82
3.2.5.6.1 Diseño de la pantalla de inicio de sesión 82
3.2.5.6.2 Diseño de la pantalla para el registro del usuario ... 83
3.2.5.6.3 Diseño de la pantalla principal de la aplicación 84
3.2.5.6.4 Diseño del menú de la aplicación 84
3.2.5.6.5 Diseño del menú para ingreso de local 85
3.2.5.6.6 Diseño del prototipo para la aplicación web 85
3.2.5.6.7 Diseño para la pantalla de inicio de sesión en la página web 86
3.2.5.6.8 Diseño para la administración de datos de los locales desde la web 87

CAPÍTULO 4 RESULTADOS 88

4.1 Resultados del desarrollo de la aplicación 89
4.1.1 Registro de usuario 89
4.1.1.1 Registro de imagen para perfil de usuario 89
4.1.1.2 Registro de información de usuario 89
4.1.2 Inicio de sesión 91
4.1.3 Pantalla principal de la aplicación: 91
4.1.4 Listado de locales: 92
4.1.4.1 Información: 923
4.1.4.1.1 Galeria Fotografica: 923
4.1.4.2 Dirección 94
4.1.4.2.1 Visualización de Punto en Realidad Aumentada: 94
4.1.4.2.2 Google Directions: 95
4.1.4.2.3 Google Street: 96
4.1.4.3 Contactos: 96
4.1.5 Mapa: 97
4.1.6 Categorías: 97
4.1.7 Configuraciones: 98
4.1.8 Realidad Aumentada: 99
4.1.9 Historial: 101
4.1.10 Registro de local: 101
4.1.10.1 Selección de un plan de pagos: 101
4.1.10.2 Registro de imágenes de local: 103
4.1.10.3 Registro de información de local: 104
4.1.11 Cerrar Sesión: 105
4.2. Resultados de la página web 105
4.2.1 Inicio de sesión: 105
4.2.2 Listado de Locales: 106
4.2.3 Edición de Locales: 107
4.2.4 Ingreso de Locales 107

CAPÍTULO 5 DISCUSIONES Y CONCLUSIONES 108

RECOMENDACIONES 111

BIBLIOGRAFÍA 112

ANEXOS 120

Anexo 1 120

Resultados de encuestas 120

Pregunta 1: Por favor indicar que tipo de dispositivo móvil tiene 122

Pregunta 2: Qué modelo de dispositivo posee 122

Pregunta 3: Posee conexión a internet 123

Pregunta 4: Conoce usted aplicaciones de turismo 124

Pregunta 5: Prefiere usted aplicaciones de paga o gratuitas 124

Pregunta 6: Pagaría usted por usar una aplicación para promocionar su local 125

Justificaciones: 126

Pregunta 7: Está en desarrollo una aplicación local de turismo, nos podría ayudar con su información o sugerencia sobre qué información se debería mostrar en la aplicación. 126

Anexo 2 127

2.1 Resultados de evaluación de la aplicación mediante una etnografía enfocada 127

2.2 Resultados Obtenidos 128

Usuarios Locales 128

Usuarios Extranjeros 129

Anexo 3 130

3.1 Instalación de Android Studio 130

3.2 Configuración de Android SDK: Android SDK (Software Development Kit) contiene emuladores y depuradores para el desarrollo de aplicaciones en Android. Incorpora librerías gratuitas para ser implementadas en los proyectos. (Creative Commons, 2015) 135

Anexo 4 136

4.1 Resultados de reportes de HockeyApp: 136

INDICE DE FIGURAS

FIGURA 1.1: SISTEMAS OPERATIVOS PARA MÓVILES 2
FIGURA 1.2: ARQUITECTURA DE SISTEMA OPERATIVO ANDROID 3
FIGURA 1.3: ARQUITECTURA DE UN SISTEMA OPERATIVO IOS 5
FIGURA 1.4: ARQUITECTURA DE UN SISTEMA OPERATIVO WINDOWS PHONE 6
FIGURA 1.5: ESQUEMA DE PROCESADOR DE UN DISPOSITIVO MÓVIL ARQUITECTURA DE OMAP 5 8
FIGURA 1.6: NUMERO DE DISPOSITIVOS VENDIDOS EN EL MUNDO (UNIDADES) 11
FIGURA 1.7: CANTIDAD DE SISTEMAS OPERATIVOS EN USO 11
FIGURA 1.8: USUARIOS MÓVILES EN ECUADOR POR SISTEMA OPERATIVO 12
FIGURA 2.1: CAPTURA DE PANTALLA TOMADA DE LA WEB DE OPENSTREETMAP 16
FIGURA 2.2: SELECCIÓN DE PUNTO DE INTERÉS GOOGLE MAPS 20
FIGURA 2.3: SELECCIÓN OPCIÓN MAPA OFFLINE 20
FIGURA 2.4: SELECCIÓN DE ZONA PARA DESCARGA 20
FIGURA 2.5: DESCARGA DE MAPA 21
FIGURA 2.6: EJEMPLO DE REALIDAD AUMENTADA CREADO CON AURASMA 22
FIGURA 2.7: REALIDAD AUMENTADA VS REALIDAD VIRTUAL ARTOOLKIT 22
FIGURA 2.8: COMPONENTES DE UN SISTEMA CON REALIDAD AUMENTADA 24
FIGURA 2.9: ESQUEMA DEL PROCESO DE ARTOOLKIT, PARA EL DIBUJO DE UN PATRÓN SOBRE UN OBJETO. FUENTE: (ARTOOLKIT) 24
FIGURA 2.10: EVOLUCIÓN DE LA REALIDAD AUMENTADA 25
FIGURA 2.11: MODELO BASADO EN GRAFOS DE SECTOR VIAL 27
FIGURA 3.1: FRAGMENTO DE CÓDIGO DEL ARCHIVO ANDROIDMANIFEST 30
FIGURA 3.2: INTERFAZ DE ANDROID STUDIO 31
FIGURA 3.3: INTERFAZ DE MYSQL WORKBENCH 32
FIGURA 3.4: CLASE SQLITEOPENHELPER 33
FIGURA 3.5: MÉTODO INSERT() DE LA CLASE SQLITEOPENHELPER 33
FIGURA 3.6: MÉTODO UPDATE() Y DELETE() DE LA CLASE SQLITEOPENHELPER 34
FIGURA 3.7: MÉTODO EXECSQL 34
FIGURA 3.8: MÉTODO RAWQUERY 35
FIGURA 3.9: MÉTODO QUERY 35
FIGURA 3.10: MÉTODOS DE LA CLASECURSOR 35
FIGURA 3.11: PRINCIPIOS DE MATERIAL DESIGN (MATERIAL) 37
FIGURA 3.12: PRINCIPIOS DE MATERIAL DESIGN (LLAMATIVO) 38
FIGURA 3.13: PRINCIPIOS DE MATERIAL DESIGN (MOVIMIENTO) 38
FIGURA 3.14: RETROFIT - COMUNICACIÓN CON EL SERVIDOR 39

FIGURA 3.15: INICIALIZACIÓN DE LA CONEXIÓN A NUESTROS SERVICIOS 40
FIGURA 3.16: CONSTRUCCIÓN DEL CLIENTE REST 40
FIGURA 3.17: INTERFAZ NINJAMOCK 41
FIGURA 3.18: TECNOLOGÍAS SOPORTADAS POR NETBEANS 41
FIGURA 3.19: DESCARGA DEL JDK DE JAVA 42
FIGURA 3.20: INDEX DEL WEBSERVICE 42
FIGURA 3.21: PROCESO DE PAGO PAYPHONE 43
FIGURA 3.22: MÉTODO SETTRANSACTION DE PAYPHONE EN ANDROID 44
FIGURA 3.23: MÉTODO GETTRANSACTION DE PAYPHONE EN ANDROID 45
FIGURA 3.24: CONFIGURACIÓN DE GOOGLE PLAY SERVICES 46
FIGURA 3.25: ACTIVACIÓN DE GOOGLE PLAY SERVICES 46
FIGURA 3.26: ACEPTACIÓN DE LICENCIAS 47
FIGURA 3.27: DEPENDENCIA BUILD.GRADLE 48
FIGURA 3.28: FINGERPRINT SHA-1 49
FIGURA 3.29: CREACIÓN DE CLAVE DE API 49
FIGURA 3.30: CREDENCIALES DEL PROYECTO 50
FIGURA 3.31: CLAVE API PARA APLICACIÓN 50
FIGURA 3.32: CÓDIGO PARA INTEGRAR GOOGLE MAPS A LA APLICACIÓN 51
FIGURA 3.33: CÓDIGO PARA LA IMPLEMENTACIÓN DE FRAGMENT MANAGER 51
FIGURA 3.34: CÓDIGO PARA LA IMPLEMENTACIÓN DE MÉTODOS 52
FIGURA 3.35: CÓDIGO PARA LA IMPLEMENTACIÓN DE MÉTODOS 53
FIGURA 3.36: CONTROLES DE INTERFAZ DE USUARIO 54
FIGURA 3.37: MÉTODOS Y CONTROLADORAS DE UISETTINGS 54
FIGURA 3.38: CÓDIGO PARA IMPLEMENTACIÓN DE BOTÓN DE UBICACIÓN 55
FIGURA 3.39: MÉTODO ADDMARKER DE UN MARCADOR EN GOOGLE 55
FIGURA 3.40: MÉTODO ONMAPREADY DE UN MARCADOR EN GOOGLE 56
FIGURA 3.41: SOLICITUD DE URL ENVIADA A LA API 56
FIGURA 3.42: VALORES DE LATITUD-LONGITUD 57
FIGURA 3.43: EJEMPLO DE CONSTRUCCIÓN DE UNA RUTA 58
FIGURA 3.44: FRAGMENTO DE ARCHIVO DE DISEÑO 59
FIGURA 3.45: IMPLEMENTACIÓN DE LA CLASE ONSTREETVIEWPANORAMAREADYCALLBACK 59
FIGURA 3.46: CREACIÓN DE LA VISTA DE CONTENIDO 59
FIGURA 3.47: ENVIÓ DE DATOS PARA LA CREACIÓN DEL PANORAMA 59
FIGURA 3.48: AGREGAR EL SDK HOCKEYAPP 60
FIGURA 3.49: INTEGRACIÓN DE HOCKEYAPP 60
FIGURA 3.50: INTEGRACIÓN DE HOCKEYAPP 61

FIGURA 3.51: CREACIÓN DE INFORMES DE HOCKEYAPP 61
FIGURA 3.52: INVITACIÓN A BETATESTER DE LA APLICACIÓN 62
FIGURA 3.53: EJEMPLO DE IMPLEMENTACIÓN DE LA CLASE GEOOBJECT 64
FIGURA 3.54: SELECCIÓN DE LA IMAGEN PARA UN MARCADOR 65
FIGURA 3.55: IMPLEMENTACIÓN DE LA FUNCIÓN LOCATIONCHANGED 65
FIGURA 3.56: IMPLEMENTACIÓN DE ARRAY DE MARCADORES 66
FIGURA 3.57: EJEMPLO MODELO VISTA CONTROLADOR (MVC) 70
FIGURA 3.58: JERARQUÍA DE USUARIOS 71
FIGURA 3.59: CASO DE USO GENERAL DE LA APLICACIÓN 72
FIGURA 3.60: CASO DE USO TURISTA 74
FIGURA 3.61: CASO DE USO DUEÑO DE LOCAL 76
FIGURA 3.62: CASO DE USO ACTIVIDAD PAGINA WEB 76
FIGURA 3.63: DIAGRAMA DE CLASES 79
FIGURA 3.64: DIAGRAMA DE SECUENCIA REGISTRÓ TURISTA 80
FIGURA 3.65: DIAGRAMA DE NAVEGACIÓN DE LA APLICACIÓN 81
FIGURA 3.66: DIAGRAMA DE NAVEGACIÓN DE PÁGINA WEB 82
FIGURA 3.67: DISEÑO DE LA PANTALLA DE INICIO DE SESIÓN 83
FIGURA 3.68: DISEÑO DE LA PANTALLA PARA REGISTRO DE USUARIO 83
FIGURA 3.69: DISEÑO DE LA PANTALLA PRINCIPAL DE LA APLICACIÓN 84
FIGURA 3.70: DISEÑO DE LA PANTALLA DEL MENÚ DE LA APLICACIÓN 84
FIGURA 3.71: DISEÑO DE LA PANTALLA DEL MENÚ DE LA APLICACIÓN 85
FIGURA 3.72: ESTRUCTURA DE ARCHIVOS DE BOOTSTRAP 86
FIGURA 3.73: DISEÑO DE INICIO DE SESIÓN PÁGINA WEB 86
FIGURA 3.74: ADMINISTRACIÓN DE DATOS DE LOCALES 87
FIGURA 4.1: REGISTRO DE IMAGEN DE USUARIO 89
FIGURA 4.2: PROCESO DE REGISTRO DE USUARIO 90
FIGURA 4.3: SELECCIÓN DE CUIDAD 90
FIGURA 4.4: PANTALLA INICIO DE SESIÓN 91
FIGURA 4.5: PANTALLA PRINCIPAL DE APLICACIÓN 91
FIGURA 4.6: PANTALLA DE LISTADO DE LOCALES 92
FIGURA 4.7: PANTALLA DE INFORMACIÓN DE LOCAL 93
FIGURA 4.8: GALERÍA FOTOGRÁFICA 93
FIGURA 4.9: PANTALLA DE DIRECCIÓN DE LOCAL 94
FIGURA 4.10: VISUALIZACIÓN MEDIANTE REALIDAD AUMENTADA 94
FIGURA 4.11: PANTALLA QUE MUESTRA EL TRAZADO DE LA RUTA 95
FIGURA 4.12: NAVEGACIÓN ASISTIDA POR GPS 95
FIGURA 4.13: PANTALLA DE DIRECCIÓN DE LOCAL 96

FIGURA 4.14: INFORMACIÓN DE CONTACTO DE UN LOCAL.....96
FIGURA 4.15: PANTALLA DE MARCADORES DE LOS LOCALES EN EL MAPA.....97
FIGURA 4.16: PANTALLA DE CATEGORÍAS DE LOS LOCALES.....98
FIGURA 4.17: PANTALLA PARA CONFIGURACIONES DE LA APLICACIÓN 99
FIGURA 4.18: PUNTOS DE INTERÉS MEDIANTE REALIDAD AUMENTADA 100
FIGURA 4.19: DESCRIPCIÓN DE PUNTOS DE INTERÉS MEDIANTE REALIDAD AUMENTADA 100
FIGURA 4.20: PANTALLA QUE REGISTRA EL HISTORIAL DE VISITAS DEL USUARIO 101
FIGURA 4.21: PANTALLA QUE PERMITE LA SELECCIÓN DE UN PLAN DE PAGO.....102
FIGURA 4.22: INGRESO DE NUMERO PARA PROCESO DE PAGO 102
FIGURA 4.23: PROCESO DE VALIDACIÓN Y APROBACIÓN DE PAGO POR PARTE DE PROMOTOR.....103
FIGURA 4.24: PANTALLAS PARA EL REGISTRO DE IMÁGENES DE UN LOCAL.....104
FIGURA 4.25: PANTALLAS PARA EL REGISTRO DE INFORMACIÓN DE UN LOCAL 105
FIGURA 4.26: PANTALLAS DE NOTIFICACIÓN PARA CERRAR SESIÓN 105
FIGURA 4.27: LOGIN DE USUARIO EN PÁGINA WEB 106
FIGURA 4.28: LISTADO DE LOCALES DE UN USUARIO.....106
FIGURA 4.29: EDICIÓN DE LOCALES 107
FIGURA 4.30: INGRESO DE LOCALES 107
FIGURA 114: SISTEMATIZACIÓN DE RESULTADOS DE SISTEMAS OPERATIVOS 120
FIGURA 115: SISTEMATIZACIÓN DE RESULTADOS DE TIPO DE DISPOSITIVOS.....122
FIGURA 116: SISTEMATIZACIÓN DE RESULTADOS DE MODELOS DE DISPOSITIVOS MÓVILES. 123
FIGURA 117: SISTEMATIZACIÓN DE RESULTADOS DISPONIBILIDAD DE CONEXIÓN A INTERNET.....123
FIGURA 118: GRAFICA DE RESULTADOS CONOCIMIENTO APLICACIONES DE TURISMO. .124
FIGURA 119: GRAFICA DE RESULTADOS DE PAGO POR UNA APLICACIÓN 125
FIGURA 120: SISTEMATIZACIÓN DE RESULTADOS PAGO PARA PROMOCIONAR LOCAL MEDIANTE LA APLICACIÓN. 125
FIGURA 121: EJECUCIÓN DE ANDROID STUDIO PARA SU INSTALACIÓN.....130
FIGURA 122: PROCESO DE INSTALACIÓN DE ANDROID STUDIO.....130
FIGURA 123: ASISTENTE DE INSTALACIÓN DE ANDROID STUDIO 130
FIGURA 124: OPCIONES DE INSTALACIÓN DE ANDROID STUDIO.....131
FIGURA 125: TÉRMINOS DE INSTALACIÓN DE ANDROID STUDIO.....131
FIGURA 126: RUTA DE INSTALACIÓN DE ANDROID STUDIO 132
FIGURA 127: PROCESO DE INSTALACIÓN DE ANDROID STUDIO.....132
FIGURA 129: INICIALIZACIÓN DE ANDROID STUDIO.....133

FIGURA 130: PARÁMETROS DE CONFIGURACIÓN DE ANDROID STUDIO. 134
FIGURA 131: ACTUALIZACIÓN DE LIBRERÍAS Y REPOSITORIOS DE ANDROID STUDIO. 134
FIGURA 132: INICIALIZACIÓN DE ANDROID STUDIO 135
FIGURA 133: CONFIGURACIÓN DEL SDK DE ANDROID 135
FIGURA 134: REPORTE HOCKEYAPP.....137

INDICE DE TABLAS

TABLA 1. USUARIOS DE LA APLICACIÓN 71
TABLA 2. ESPECIFICACIÓN CASO DE USO REGISTRO LOCAL 73
TABLA 3. ESPECIFICACIÓN CASO DE USO REGISTRO TURISTA 75
TABLA 4. ESPECIFICACIÓN CASO DE USO LOGIN TURISTA 75
TABLA 5. ESPECIFICACIÓN CASO DE USO PROCESO PAGO 77
TABLA 6. ESPECIFICACIÓN CASO DE USO ACTIVIDAD PÁGINA WEB.....78
TABLA 7. RESULTADOS DE SISTEMAS OPERATIVOS 120
TABLA 8. FORMULARIO PARA REALIZAR ENCUESTAS A USUARIOS 121
TABLA 9. RESULTADOS DE TIPO DE DISPOSITIVO MÓVIL.....122
TABLA 10. RESULTADOS DE MODELO DE DISPOSITIVO MÓVIL.....122
TABLA 11. RESULTADOS DE DISPONIBILIDAD DE CONEXIÓN A INTERNET.....123
TABLA 12. RESULTADOS DE CONOCIMIENTO APLICACIONES DE TURISMO.....124
TABLA 13. RESULTADOS DE PAGO POR UNA APLICACIÓN.....124
TABLA 14. RESULTADOS DE PAGO PARA PROMOCIONAR LOCAL MEDIANTE LA APLICACIÓN
.....125
TABLA 15. PESOS DE CALIFICACIONES PARA ETNOGRAFÍA 127
TABLA 16. RESULTADOS ETNOGRAFÍA USUARIOS LOCALES 128
TABLA 17. RESULTADOS ETNOGRAFÍA USUARIOS EXTRANJEROS 129

RESUMEN

Hoy en día vivimos en un mundo donde la tecnología se basa en dispositivos móviles, el presente proyecto tiene como finalidad investigar y desarrollar una aplicación para dispositivos móviles Android, haciendo uso de las tecnologías de geoposicionamiento y realidad aumentada, que nos dará como resultado una aplicación útil y novedosa para el turismo del centro histórico de la ciudad de Cuenca.

La implementación se desarrolló de tal manera que son los dueños de los locales quienes ingresan la información, que consumirá la aplicación, es por ello que se estableció un plan de pagos que permita identificar a los usuarios.

ABSTRACT

Today we live in a world where technology is based on mobile devices. The aim of the present project was to research and develop an application for Android mobile devices by making use of geo-location and augmented reality technologies. This will result in a useful and innovative application for tourism, taking the historical center of the city of Cuenca as the area for demonstration. The implementation was developed in such a way, that the owners of the premises are the ones who enter the information that will be displayed in the application. In addition, a payment plan was established in order to identify users.

Translated by,
Lic. Lourdes Crespo

CAPÍTULO 1: ANÁLISIS DE LOS DISPOSITIVOS MÓVILES

Introducción

En este capítulo se realizará un análisis de los dispositivos móvil, existentes en el mercado local, con el fin de realizar un estudio, de las versiones de sistemas operativos en funcionamiento, así como también las características de mayor relevancia de los dispositivos y con estos datos poder determinar la accesibilidad hacia un dispositivo móvil, dependiendo de sus características y sistema operativo.

Además, se realizara un análisis del sistema operativo de los dispositivos móviles con mayor uso dentro de nuestro medio, el cual servirá como base para justificar la plataforma para la cual se desarrollará la aplicación de la presente investigación.

1.1 Versiones de sistemas operativos para móviles

Hoy en día la tecnología avanza de una manera sorprendente, es la sociedad quien se ha beneficiado con estos avances, desde el inicio de los años 80 hasta la actualidad el uso de dispositivos móviles ha logrado tener un enorme crecimiento, así como de su utilidad, pues paso de ser un simple teléfono a una herramienta de trabajo "Smartphone", por lo tanto, requieren de un sistema operativo que permita cumplir las exigencias de los usuarios, en un mundo de tecnología cambiante, es por eso que los fabricantes se encuentran en busca de la vanguardia de obtener el hardware con la mejor tecnología y a este dispositivo dotarle de un sistema operativo con la mejores características técnicas.

El sistema operativo del dispositivo es decir el "software", es aquel que inicia al dispositivo móvil y gestiona los recursos tanto en software como en hardware, permitiendo realizar la interacción con el usuario.

En la presente investigación se realizará un estudio de los sistemas operativos, para dispositivos [1]móviles con mayor uso dentro de nuestro mercado local.

[1]**DISPOSITIVO MÓVIL:** Son pequeñas maquinas que procesan datos, similar a un computador, con conexión a internet, con gran capacidad de almacenamiento y procesamiento.

Figura 1.1: Sistemas Operativos para móviles

Fuente: (Foundation G. C., 2016)

1.1.1 Android

Se caracteriza por ser un sistema operativo abierto, cuyo núcleo está basado en el kernel de Linux[2], es decir cualquier fabricante de dispositivos móviles, puede modificarlo y hacer uso del sistema operativo cuyo propietario es "Google".

1.1.1.1 Arquitectura

Su arquitectura de funcionamiento está basada en cuatro capas que conforman el sistema operativo:

1. **Aplicaciones:** Son aquellas que dan funcionalidad al dispositivo, estas son desarrolladas en Java (clientes de correo, calendario, mapas, contactos) (Basterra , Bertea , Borello , Castillo , & Venturi, 2017)
2. **Marco de trabajo de aplicaciones:** Es la capa a la cual los desarrolladores tienen acceso y es en donde se encuentra el API [3]de los frameworks[4] para aplicaciones base (permite la reutilización de componentes). (Basterra , Bertea , Borello , Castillo , & Venturi, 2017)
3. **Librerías:** Es el conjunto de librerías de C/C++, estos son usados por los componentes del sistema para el desarrollo de aplicaciones.
    - **Runtime de Android:** Contiene las librerías requeridas, para el funcionamiento del sistema. Cada aplicación Android al ejecutarse corre su propio proceso con su propia instancia de la máquina virtual de Dalvik.[5] (Basterra , Bertea , Borello , Castillo , & Venturi, 2017)

---

[2]**KERNEL DE LINUX:** Es el encargado de administrar los procesos del sistema operativo, permitiendo que el hardware y software puedan trabajar juntos. (Zabaljauregui, 2015)

[3]**API:** Interfaz para la programación de aplicaciones, se refiere a librerías que permiten la implementación de cierta funcionalidad mediante una interfaz. (Zerbst, 2004).

[4]**FRAMEWORK**: Es una estructura de software compuesta de varios componentes personalizables, que pueden ser usados para el desarrollo de una aplicación. (Molina Rivera, Sandoval Cardona, & Toledo Franco, 2012)

[5]**MÁQUINA VIRTUAL DE DALVIK**: Android utiliza su propia máquina virtual, lo cual permite la ejecución de multitareas en el dispositivo. (Basterra , Bertea , Borello , Castillo , & Venturi, 2017)

4. **Núcleo de Linux:** Proporciona servicios de seguridad, gestión para la memoria y procesos, pila de red, permite la creación de la capa de abstracción entre el hardware y software del dispositivo. (Basterra , Bertea , Borello , Castillo , & Venturi, 2017)

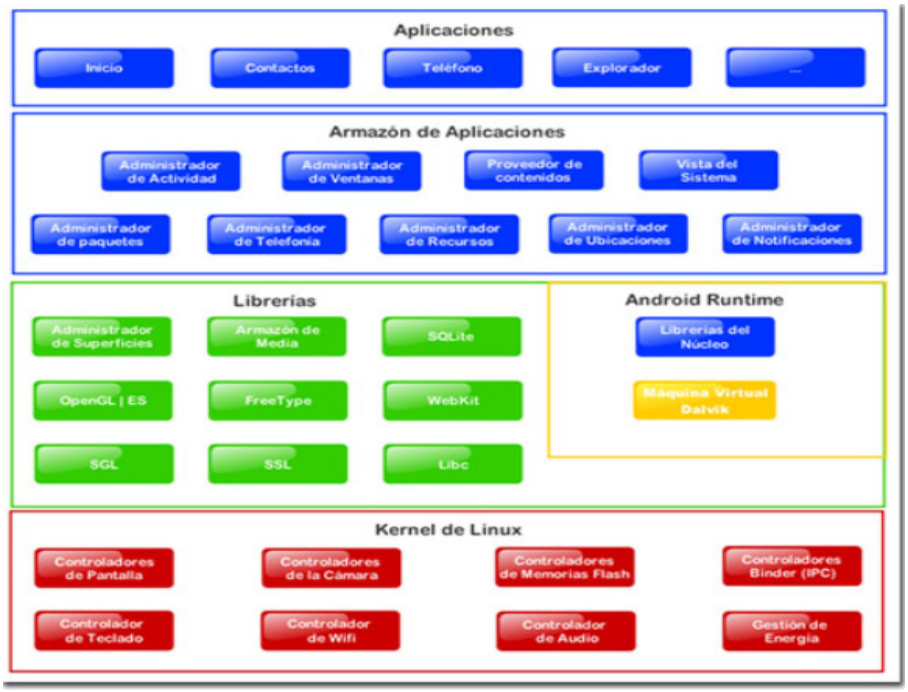


Figura 1.2: Arquitectura de sistema operativo Android

Fuente: (Basterra , Bertea , Borello , Castillo , & Venturi, 2017)

El hardware de Android está basado en una plataforma de arquitectura ARM[6], con una arquitectura de X86, lo cual permite la integración de cualquier hardware. Posee una estructura del sistema operativo que se compone de aplicaciones que se ejecutan en un *framework* java de aplicaciones, orientadas sobre el núcleo de las bibliotecas de java en la máquina virtual Dalvik, la cual se compila en tiempo de ejecución. (Basterra , Bertea , Borello , Castillo , & Venturi, 2017)

Las bibliotecas que se utilizan en las aplicaciones para dispositivos con sistema operativo Android, están desarrolladas en lenguaje C e incluyen:

a) Interfaz gráfica
b) Framework OpenCore
c) Base de datos SQLite
d) Interfaz de programación de API grafica OpenGL

[6]**ARQUITECTURA ARM:** Procesadores de tamaño muy reducido y alto rendimiento, con bajo consumo de energía, cuya arquitectura es x86. (Canel, 2009)

e) Motor de renderizado Webkit

f) Motor gráfico SGL

(Basterra , Bertea , Borello , Castillo , & Venturi, 2017)

1.1.2 IOS

Una de las principales características del sistema operativo de IOS, es que no permite modificaciones, más allá de sus ajustes, es por eso que se denomina un *sistema cerrado*, pero a la misma vez se caracteriza por ser un sistema estable y seguro, que puede ser usado solo en el hardware del fabricante "Apple", se limita a proporcionar su sistema operativo a terceros (Basterra , Bertea , Borello , Castillo , & Venturi, 2017), IOS está basado en Mac OS X, que es el sistema operativo de las computadoras Apple, esta desarrollo en Darwin BSD[7], del sistema operativo en Unix. (Rios, 2013)

1.1.2.1 Arquitectura

La arquitectura de funcionamiento de un sistema operativo IOS está basada en cuatro capas:

1. **Cocoa Touch:** Es la capa de mayor relevancia del sistema operativo IOS, ya que posee el conjunto de framework que proporciona el API de cocoa, el cual es usado para el desarrollo de aplicaciones.

   Esta capa está formada por dos frameworks fundamentales:

   1. **UIKit**: Es la capa que contiene las clases necesarias para el desarrollo de una interfaz de usuario.
   2. **Foundation Framework:** Permite definir las clases básicas, el acceso y manejo de objetos, así como los servicios para el manejo del sistema operativo.
2. **Media**: Es la capa que se encarga de proveer de los servicios de gráficos y multimedia a la capa superior.
3. **Core Services:** Es la encargada de proporcionar los servicios de los sistemas para el funcionamiento de las aplicaciones.
4. **Core OS:** Controla los ficheros del sistema, manejo de memoria, seguridad y drivers del dispositivo, son características que se denominan de bajo nivel.

[7]**Darwin BSD:** Es el núcleo del sistema operativo de Apple, Mac OS X, y se ejecuta en un núcleo de código abierto llamado XNU (X is Not Unix).

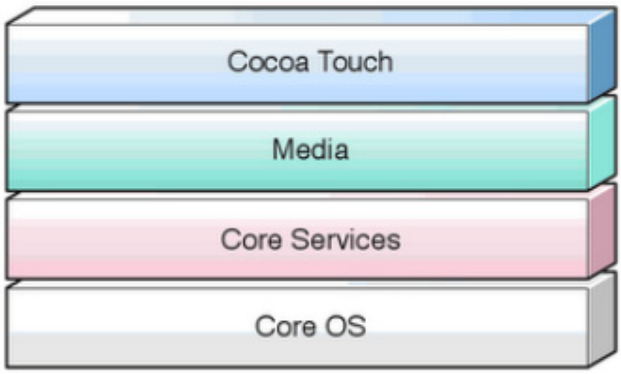


Figura 1.3: Arquitectura de un sistema operativo IOS

Fuente: (Rios, 2013)

1.1.3 Windows Phone

Sistema operativo que pertenece a Microsoft, logró alcanzar el 2,5% del mercado mundial, luego que su sistema tomo posesión de dispositivos Nokia. Ofrece un diseño diferente en su pantalla de inicio con lo cual logra dar una experiencia nueva al usuario, está basado en el núcleo del sistema operativo Windows CE.

1.1.3.1 Arquitectura

La arquitectura que utiliza el sistema operativo de Windows Phone, está compuesto por cuatro capas:

1. **Runtime de aplicaciones:** Es la capa que controla donde y como se ejecutan las aplicaciones. Permite determinar las limitaciones y los framework que se tienen para el desarrollo.
2. **Modelo de aplicación:** Las aplicaciones para Windows Phone se guardan en un archivo comprimido, en el cual podemos encontrar los ensamblados y recursos de la aplicación, estas de despliegan en un paquete denominado XAP[8]. A cada aplicación desarrollada y alojada en el Marketplace de Windows Phone se le asigna un ID único con un certificado de seguridad que es homologado por Windows.
3. **Modelo de UI:** Está compuesto por elementos, páginas y sesiones:
    - Los elementos son todos aquellos controles que se muestran al usuario.
    - Las páginas son las agrupaciones lógicas de los elementos.

[8]**XAP:** Es un paquete de aplicaciones Microsoft, que contiene archivos XAML y DLL, con directorios para la instalación de los aplicativos.

- Las sesiones son el conjunto de interacciones que el usuario realiza en la aplicación.

4. **Integración con la nube:** La versión de sistema operativo de Windows Phone, trae consigo aplicaciones como Exchange, Xbox, Skydrive, Facebook, etc. Las cuales permiten acceder a estos servicios directamente. (Rios, 2013)

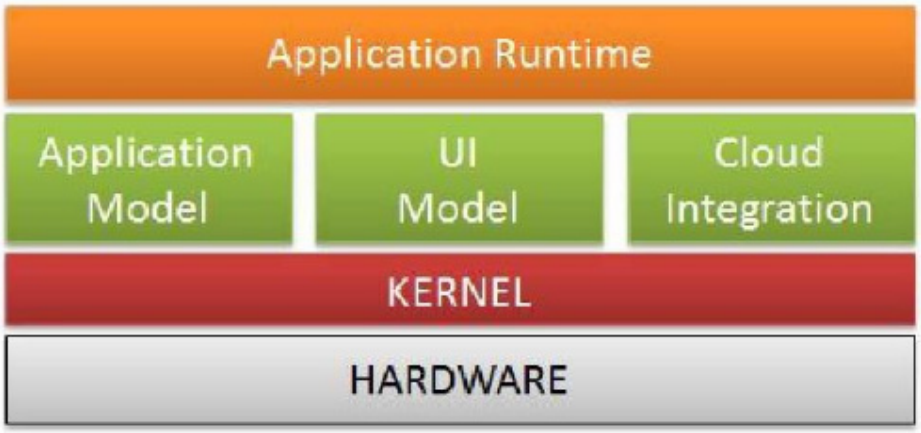


Figura 1.4: Arquitectura de un sistema operativo Windows Phone

Fuente: (Rios, 2013)

1.2 Características de los dispositivos móviles

Las Tecnologías de la Información (TICs), han permitido que se logre un avance importante a lo que hoy se denomina la "*Sociedad del Conocimiento*", ya que no existe un solo ámbito de la vida humana que no haya sido impactada por las TICs. Cada día existen nuevas tecnologías, entre los cuales se encuentran los dispositivos móviles (Mobility Computing) que se han convertido en una herramientas de trabajo, donde no se limita solo para el uso de la comunicación mediante voz, ya que ahora cuentan con un gran número de aplicaciones. (Clarke, Modet & Cº, España, 2014)

Los dispositivos móviles existentes en el mercado local cuentas con las siguientes características:

a) Portabilidad
b) Capacidad de procesamiento
c) Conexión a internet
d) Capacidad de almacenamiento
e) Capacidad de interacción

1.2.1 Portabilidad

Es la característica que permite al dispositivo ser transportado con gran facilidad, permitiendo hacer uso de sus características en cualquier lugar y ser utilizado mientras es transportado, para lo cual es dotado de baterías recargables, que permiten el uso de los dispositivos sin necesidad de requerir de una conexión a una fuente de energía, permitiendo tener acceso a los datos que se requieran desde cualquier ubicación. (Pozo, 2015)

1.2.2 Capacidad de procesamiento

Es la característica que permite dar la funcionalidad al dispositivo, es decir, donde se realizan todos los procesos y operaciones, todas estas funcionalidades vienen integradas en un solo chip SoC [9](System On a Chip), el cual contiene y permite controlar:

1. **Procesador de imagen:** Da la funcionalidad a la cámara y permite el manejo de imágenes.
2. **Sistema de codificación y decodificación de Vídeo:** Permite controlar los formatos de video que se pueden visualizar en el dispositivo móvil.
3. **Procesador de audio:** Permite la codificación y decodificación de los archivos que se pueden reproducir en el dispositivo móvil.
4. **Controlador de pantalla:** Muestra los gráficos en la pantalla del dispositivo móvil

[9]**SoC (System On a Chip):** Tecnología que permite integrar, varios componentes en un único chip. (Greaves, 2011)

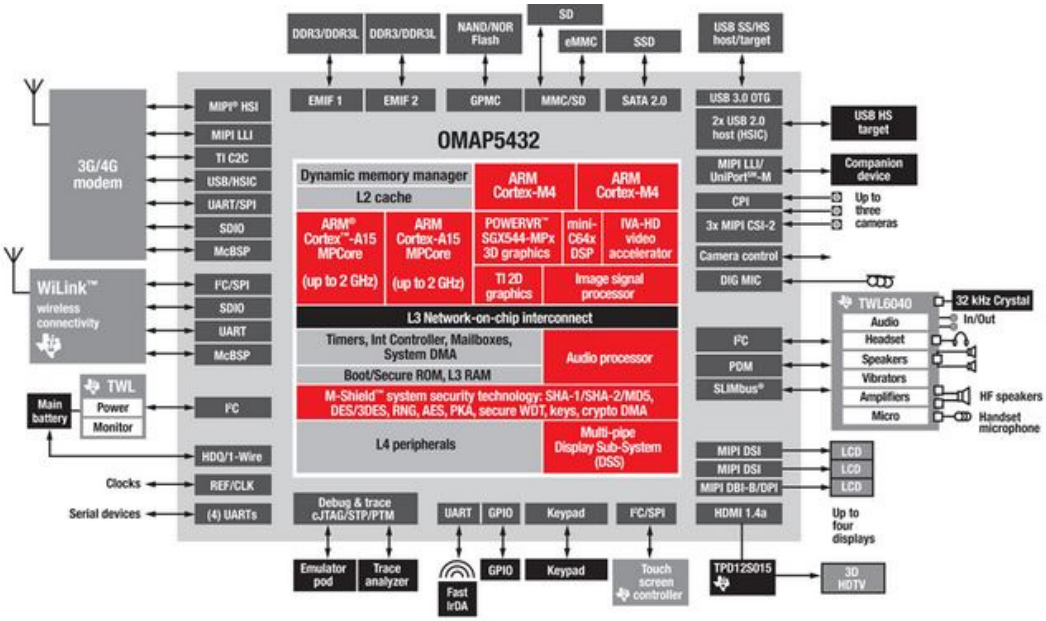


Figura 1.5: Esquema de procesador de un dispositivo móvil arquitectura de OMAP 5

Fuente: (Falcón, 2015)

1.2.3 Conexión a internet

Un dispositivo móvil puede enviar y recibir datos sin la necesidad de estar conectado a un enlace cableado, haciendo uso del dispositivo Wireless integrado, o su vez mediante el uso de datos móviles de operadoras telefónicas (Pozo, 2015).

1.2.4 Capacidad de almacenamiento

Todo dispositivo móvil para su correcta funcionalidad, ya sea de contenidos multimedia, mensajes de texto, correo, aplicaciones, etc., requieren un lugar de almacenamiento para lo cual cuenta con memoria:

- Memoria Flash (interna)
- MicroSD (externa)
- Memoria RAM

1.2.5 Capacidad de interacción

La capacidad de interacción de un dispositivo móvil, se define mediante la interfaz de usuario, es decir, la facilidad con la que se puede utilizar el dispositivo, para lo cual interviene la usabilidad y la ergonomía, estas definen los eventos que darán la funcionalidad a la pantalla tales como:

- Orden de los elementos.
- Tamaño de texto, imágenes.
- Presentación de menús y opciones.

Todas las características antes mencionadas varían dependiendo del tipo de dispositivo móvil ya sea por su capacidad y funcionalidad que tengan integradas. (Pozo, 2015)

1.3 Tipos de dispositivos móviles

El término de dispositivo móvil hoy en día ha sido reemplazado por el de Smartphone. Es un dispositivo al que se le puede agregar mayores funcionalidades mediante la instalación de programas, esto se lo realiza a través de una conexión de datos Wifi, el cual es denominado aprovisionamiento OTA [10](Over the Air), las aplicaciones pueden ser descargadas de acuerdo a la necesidad del usuario.

Según las características de un dispositivo móvil estos se clasifican en:

- Dispositivos de gama baja
- Dispositivos de gama alta

1.3.1 Dispositivos de gama baja

Son aquellos dispositivos, en donde el soporte para la instalación de aplicaciones es limitado, de igual manera su capacidad de procesamiento y almacenamiento es limitado. Cada vez son menos usados debido a que las aplicaciones y actualizaciones ya no se enfocan en estos dispositivos.

Estos dispositivos usan procesadores de la generación ARM9[11], que son chips de bajo coste, ya que poseen una arquitectura antigua (Falcón, 2015).

1.3.2 Dispositivos de gama alta

Son aquellos en donde se puede evidenciar los mayores avances tecnológicos, pues cuentan con los mejores procesadores existentes en el mercado, gran capacidad de almacenamiento y procesamiento, la mejor tecnología y resolución de pantalla y un diseño muy atractivo e innovador.

Estos dispositivos están orientados al funcionamiento de múltiples aplicaciones y contenido multimedia. Hoy en día se orientan hacia la realidad virtual, es por eso que

[10]**OTA:** Actualización del firmware o aplicaciones instaladas en el dispositivo móvil mediante una conexión a una red Wifi.
[11]**ARM9:** Son un grupo de procesadores de 32 bits, de un solo núcleo y de poca complejidad.

se pone énfasis en la resolución de la pantalla y de la cámara, para la obtención de imágenes con gran nitidez. Un dispositivo de gama alta debe tener una batería de larga duración y un sistema de carga rápida.

Estos dispositivos usan procesadores de tipo Cortex A8[12], estos tipos de procesador permiten el uso de aplicaciones de alto rendimiento (Falcón, 2015).

1.4 Redes de dispositivos móviles

Las redes telefónicas para los dispositivos móviles hoy en día se orientan hacia las comunicaciones sobre IP, estas permiten descargas de datos óptimas, como también aprovechan el espectro radioeléctrico[13], se clasifican en las siguientes:

- **Red de telefonía 3G:** Es una red de banda ancha, su velocidad va desde los 348 Kbps hasta los 2 Mbps, para su transmisión lo realiza mediante paquetes, la tecnología que utiliza se basa en UMTS[14], debido a que es de tercera generación. (Computación, 2014)
- **Red de telefonía 3.5G:** Es la optimización de la red 3G, mediante un scheduling[15] de UMTS R99, añade un nuevo canal downlink[16], permite el acceso al internet con mayor banda ancha y menor latencia hasta velocidades de 7.2 Mbps.
- **Red de telefonía 4G**: Utiliza el estándar LTE[17,] que está basado en su totalidad en protocolos IP, tiene una conexión de 326.5 Mbps en movimiento en velocidades de 15 Km/h, llegando a un máximo 326.5 Mbps pero cuando se encuentra el LTE (4G real), alcanza un máximo de 1 Gbps (Computación, 2014).

1.5 Cantidad de dispositivos móviles por sistema operativo

El número de dispositivos móviles a nivel mundial para el año 2015 alcanzó el 97% de penetración en el mercado, es decir que existen 7,9 mil millones de dispositivos móviles en el mundo con relación a los 7,3 millones que existían en el año 2014. (Ditrandia)

---

[12]**CORTEX A8:** Procesador que contiene una unidad NEON, con un motor SIMD de 128 bits.
[13]**ESPECTRO RADIOELÉCTRICO:** Es el medio por el cual se trasmiten las frecuencias de ondas de radio electromagnéticas, que permiten la comunicación.
[14]**UMTS: 3G, (Universal Mobile Telecommunications System**). Sistema universal de telecomunicaciones móviles (WORDp, 2013).
[15]**SCHEDULING:** Programación rápida, que permite adaptarse a las variaciones de las condiciones del radio y determinar una celda lo más rápido posible para él envió de datos.
[16]**DOWNLINK:** Canal que puede ser compartido por varios usuarios.
[17]**LTE:** (Long Term Evolution), evolución a largo plazo, estándar para comunicaciones inalámbricas de transmisión de datos de alta velocidad para teléfonos móviles y terminales de datos.

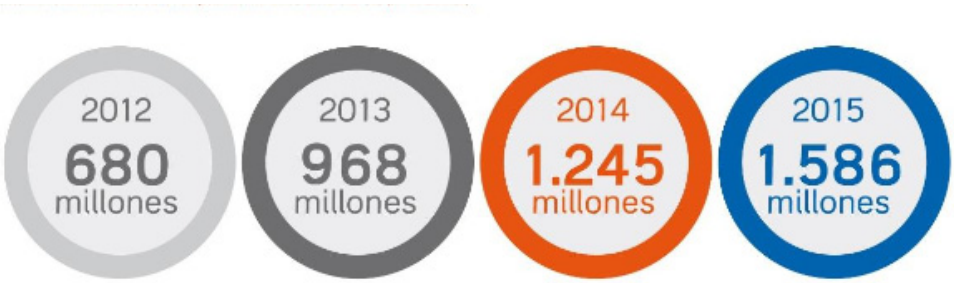


Figura 1.6: Numero de dispositivos vendidos en el mundo (unidades)

Fuente: (Ditrandia)

De los dispositivos que se han vendido en el mundo, el sistema operativo más utilizado sigue siendo *Android*, con un índice de penetración en el mercado del 84% en el año 2015, cifra que cada día sigue subiendo, le prosigue *IOS,* con cifras muy lejanas en cuanto a dispositivos, mientras que Windows Phone no se considera una cifra significante.

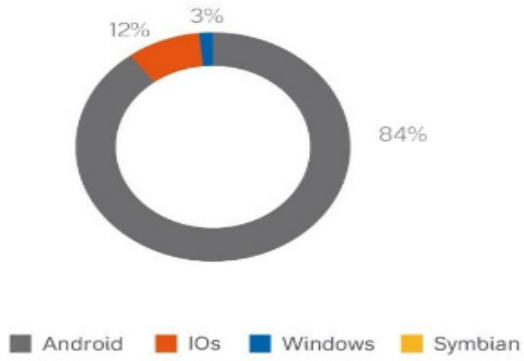


Figura 1.7: Cantidad de sistemas operativos en uso

Fuente: (Ditrandia)

1.6 Justificación del uso de la plataforma seleccionada

El uso de los 'Smartphone' no detiene su crecimiento, en el Ecuador, ya que para el año 2015 alrededor de 3'084.886 de ecuatorianos poseen un dispositivo móvil (INEC), esto significa que 53 de cada 100 ecuatorianos tienen a la mano un dispositivo móvil activado. (INEC, 2016)

Hoy en día el desarrollo de aplicaciones se encuentra en un crecimiento y evolución continua, cada vez encontramos un sin número de *apps* en el mercado, siendo Android la que concentra más del 50% del mercado (EcuadorInmediato, 2014),

adicional la compañía de investigación de marcado *Gartner,* publico cifras en el segundo trimestre (abril a junio) del 2016, en donde indica que Android lidera el mercado con un 86,2 por ciento, es por esta razón una de las justificaciones de la plataforma selecciona debido (Gartner, 2016).

> *Según el mapa de colonización Mobile 2014 de GuiaLocal.com, realiza un análisis de los dispositivos que utilizan los ecuatorianos para el acceso al internet, obteniendo los siguientes resultados en preferencia de sistemas operativos:*

- Android 58,87%
- IOS 26,47%
- BlackBerry 7,10%
- Windows Phone 3,09% (EcuadorInmediato, 2014)

Un reporte generado por la empresa Facebook determina que los usuarios que acceden a la red social, tienen una gran inclinación hacia el sistema operativo Android, seguido por el sistema operativo IOS.

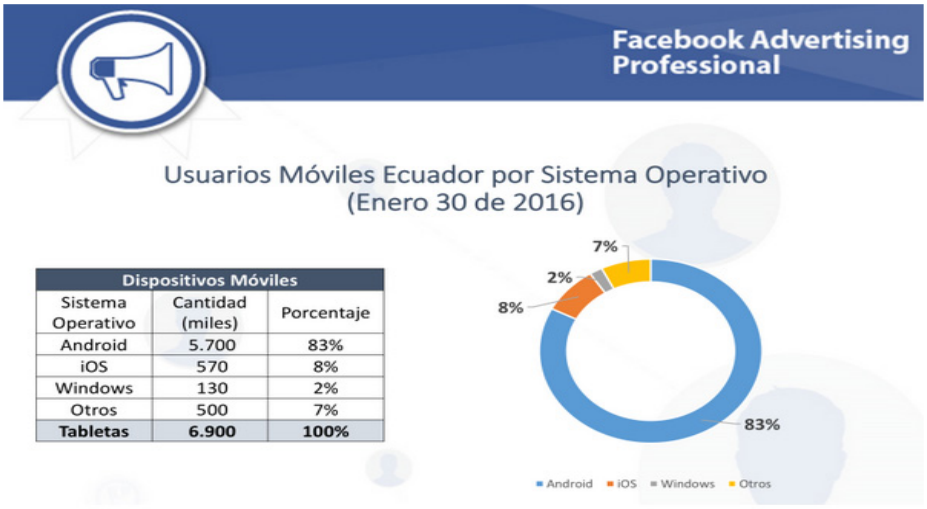


Figura 1.8: Usuarios móviles en Ecuador por sistema operativo

Fuente: (Ponce, 2016)

1.7 Geolocalización

La geolocalización nos permite conocer la ubicación geográfica exacta de cualquier objeto mediante el uso de un dispositivo móvil o computador con conexión a internet o una red de un proveedor (GPS), lo cual se logra con los siguientes datos:

- Latitud
- Longitud
- Altura
- Rumbo

Un ordenador o una aplicación pueden determinar la localización de un objeto o persona mediante tres parámetros:

- La aproximación a la torre de telefonía; por el tiempo que tarda la señal en ir de torre a torre.
- Por la fuerza de la señal recibida.
- La multilateralización (triangulación, combinación) de las señales de radio entre (varias) torres de radio de la red y el teléfono.

1.7.1 Métodos de localización de un dispositivo Android

El sistema operativo Android integra tres métodos para realizar la geolocalización que pueden ser utilizados desde un dispositivo móvil:

- Alta precisión
- Ahorro de Energía
- Solo dispositivo

1.7.1.1 Alta precisión: En este modo de alta precisión se permite al dispositivo determinar la posición a través de cualquier red disponible (WIFI, GPS) o red móvil, mediante los servicios de localización que provee Google en el sistema operativo Android (Araque, 2016).

1.7.1.2 Ahorro de Energía*:* El modo ahorro de energía, consume menos recursos del dispositivo móvil, el cual se limita a consumir datos solo de una red móvil o WIFI, elimina la posición mediante GPS, la exactitud de este modo no garantiza la proximidad a un punto (Araque, 2016).

1.7.1.3 Solo dispositivo: Este modo solo utiliza el GPS integrado en el dispositivo móvil para determinar la localización. (Araque, 2016)

CAPÍTULO 2: ANÁLISIS DE HERRAMIENTAS Y ESTÁNDARES

Introducción

El uso de mapas se ha convertido en parte esencial de nuestra vida, los mapas han sido usados desde hace siglos con el fin de representar el entorno para mostrar lugares, distancias, direcciones, navegación y búsquedas (Unidas, 2000) , con la ayuda de las tecnologías de geolocalización que incluyen los dispositivos móviles se ha permitido tener al alcance la cartografía digital (Seguros, 2016). Un mapa se genera mediante la representación de datos de objetos del mundo real los cuales se dividen en dos abstracciones:

- Objetos discretos (casas, calles)
- Objetos continuos (lluvias, elevaciones)

Para el almacenamiento de datos se utilizan dos métodos:

1. **Raster**: Es un tipo de imagen representada en mallas, divide al espacio en filas y columnas de celdas donde cada celda, representa un único valor. (Morales, 2016)
2. **Vectorial**: Los datos vectoriales se caracterizan por representar sus datos como puntos, líneas o polígonos. (Gatrell, 2012)

A la hora de utilizar mapas en un dispositivo móvil entre sus limitantes de uso, se requiere de una conexión a internet, ya sea mediante una red telefónica o WIFI, es por eso que nace la necesidad de identificar los tipos de mapas existentes:

- Mapas Offline
- Mapas Online

2.1 Mapas Offline

Son mapas que permiten hacer uso de sus características, sin la necesidad de una conexión a internet, es decir pueden ser descargados con anterioridad y almacenados en el dispositivo móvil, para lo cual utiliza el concepto de MbTiles[18,] permitiendo guardar los componentes que integran el mapa en una BD SQLite. Para realizar la descarga de un mapa en un dispositivo móvil se debe determinar el área que se quiere obtener del mapa, esto permitirá realizar búsquedas de lugares y obtener información de los lugares que se deseen visitar.

[18]**MBTILES:** Conjunto de millones de tiles, el cual almacena la combinación de ciertos lugares (calles, montañas, colores, etc.

2.2 Mapas Online

Un mapa online se caracteriza por permitir realizar localización y navegación, en tiempo real, mediante una conexión a internet, permitiendo la interacción ya que se puede colocar marcadores, líneas, figuras, texto que pueden ser animados con el fin de obtener un mapa interactivo.

Mediante el manejo del API de los mapas online se puede insertar un mapa completo en páginas web externas o a su vez en aplicaciones para dispositivos móviles, haciendo de sus herramientas.

2.3 Open StreetMap (OSM)

Es una plataforma que permite hacer uso de sus servicios geográficos de forma gratuita bajo los términos de licencia (ODbl)[19] Open Data License (Peris Martinez, 2014), los datos que conforman un mapa se los conoce como datos “vector” (Neis P, 2010). Los usuarios son quienes contribuyen a la actualización de la información, así como de la distribución. Para el proceso de actualización de los datos de los mapas se realizan los siguientes pasos:

1. Toma de datos
2. Subida de datos a los servidores
    - Edición gráfica de los datos
    - Edición alfanumérica de los datos
3. Renderizado de los mapas (Ferrer , Sanchez, & Tramoyeres)

[19]**ODBL:** Licencia libre que permite copiar, modificar y distribuir la base de datos de una aplicación de forma libre (Callejo, 2015).

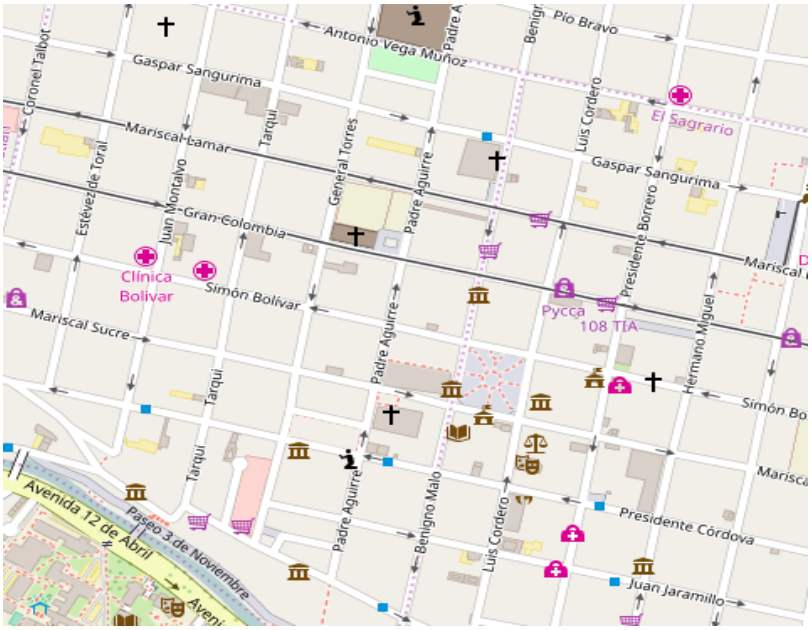


Figura 2.1: Captura de pantalla tomada de la web de OpenStreetMap

Fuente: (OpenStreetMap, 2017)

2.3.1 Toma de datos

Para la recolección de datos se lo puede realizar mediante observación directa, con el apoyo de un dispositivo GPS, el cual almacena los puntos en waypoint[21], recolección de imágenes y mapas de usuarios (Troya Carbajal, 2016).

2.3.2 Subida de datos a los servidores

Cuando la información es obtenida y consolidada, es integrada a la base de datos de OSM, para lo cual existen herramientas que el usuario puede utilizar, entre la cuales se tiene:

- Web Potlach2 (herramienta Online)
- JOSM (herramienta de escritorio)

Independientemente de la herramienta que se utilice, los datos obtenidos mediante el GPS, deben ser convertidos al formato estándar GPX[22] para poderlos subir al repositorio de trazas GPS de OSM.

[21]**WAYPOINT:** Los receptores GPS almacenan una posición, de un lugar determinado de este planeta, en su memoria (punto del camino) (Daneri, 2009).
[22]**GPX:** (Formato de Intercambio GPS) es un esquema XML pensado para transferir datos GPS entre aplicaciones. Se puede usar para describir puntos (waypoints), recorridos (tracks), y rutas (routes).

2.3.3 Edición gráfica de los datos

Para la edición de datos descargados de OSM, se puede utilizar las herramientas (Potlach2, JOSM o Merkaartor), OSM solo reconoce 2 tipos de datos gráficos:

- Nodos: Son aquellos elementos puntuales
- Vías: Conexiones entre nodos
    1. Abiertas: Aquellas que poseen entre 2 y 2000 nodos
    2. Cerradas: Vías que empiezan y terminan en el mismo nodo.
        - Áreas: Zonas que contienen las vías cerradas

2.3.4 Edición alfanumérica de los datos

Para la edición de OSM reconoce solo 2 tipos de datos alfanuméricos:

- Relación: Lista ordenada de nodos con un rol, como por ejemplo una restricción de giro.
- Etiqueta: Par de datos clave-valor que permite definir atributos.

En el modelo de datos alfanuméricos, se usa etiquetas “tags”, por parte de los usuarios para la identificación de ciertos elementos, esto con la finalidad de identificarlos fácilmente en el proceso de renderizado (Troya Carbajal, 2016).

2.3.5 Renderizado de los mapas

Se pueden obtener renderizados tanto en 2D como en 3D, el cual genera imágenes con información obtenida de la base de datos. La herramienta más usada para el renderizado es Mapnik[23] que toma los datos y los carga en una base de datos Postgres/PostGIS[24] para posteriormente renderizar tiles de 256x256. (Troya Carbajal, 2016)

2.3.6 Obtención de datos de OpenStreetMap

Para el manejo de la API de OSM, se debe conocer que está basada en la arquitectura RESTFul[25], el cual es un intercambio de XML y reconoce cuatro elementos:

---

[23]**MAPNIK:** Toolkit de código abierto para renderizar mapas.
[24]**POSTGIS:** Es un extensor de bases de datos espaciales, para la base de datos objeto-relacional de PostgreSQL.
[25]**REST:** Interfaz usada entre sistemas que usen HTTP para la obtención de datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. (BBVAOPEN4U, 2016)

Nodos: Son los puntos que contiene las coordenadas para localizar una determinada área en el mapa (Troya Carbajal, 2016).

Poseen información con las siguientes características:

- id: Identificador único
- lat y lon: Establece la posición geográfica en formato (WGS84[26])
- visible: Campo booleano que determina la visibilidad
- user: Información del usuario que creó la versión del nodo
- timestamp: Registro de marca de tiempo de creación
- version: Incremental para cada objeto.

Vías: Es la unión de dos nodos, representan calles, la unión de vías representa un área, que a su vez representa límites entre ciudades o países, en menos escala representan cuadras o parques (Troya Carbajal, 2016).

Son listas ordenadas de nodos con las siguientes características:

- id: Identificador único
- visible: Campo booleano que determina la visibilidad
- user: Información del usuario que creó la versión del nodo
- timestamp: Registro de marca de tiempo de creación
- version: Incremental para cada objeto.

Relaciones: Son listas ordenadas de objetos, nos permiten definir relaciones entre objetos, posee información con las siguientes características:

- id: Identificador único
- visible: Campo booleano que determina la visibilidad
- user: Información del usuario que creó la versión del nodo
- timestamp: Registro de marca de tiempo de creación

Etiquetas: Permite definir los atributos de un elemento, por ejemplo, los nombres de calles o sitios turísticos de una cuidad (Troya Carbajal, 2016).

[26]**WGS84:** Es un sistema de referencia terrestre y datum geodésico centrado y fijo en la Tierra. (REDERMAUS, 2015)

2.4 Google Maps

Es un servicio que ofrece Google, es una tecnología de mapeo con herramientas de gran potencial, con facilidad de uso, incluye una extensa información de ubicación de lugares, negocios, contactos, etc. Hoy en día, es una de las aplicaciones de mayor uso en el mundo, más de 800.000 sitios utilizan la API de Google Maps y existen alrededor de 250 millones de usuarios activos, por lo general su aplicación viene incluida en los dispositivos móviles. Google Maps permite consultar mapas que hayan sido descargados previamente sin necesidad de una conexión a internet. Cabe destacar que Google Maps no es de código abierto, permite hacer uso de sus funcionalidades y servicios bajo derechos de autor y términos de restricciones de uso.

Una búsqueda básica en Google Maps dará las siguientes opciones, ingresando como parámetro una dirección de destino:

- **Paper map:** La búsqueda que se realice convierte la ventana en un papel virtual “mapa”, lo cual permite la navegación, es decir, acercarse, alejarse en el área establecida.
- **Buscar ubicaciones:** Cuando se ha realizado la búsqueda inicial y se ha establecido el área, se puede realizar un zoom, lo cual nos permite una búsqueda de locales o ubicaciones con mayor detalle.
- **Direcciones:** En la búsqueda inicial se determinó una sola dirección, posteriormente podemos ingresar una segunda dirección y obtener rutas hacia los lugares.
- **Vista satélite:** Permite cambiar la vista a satelital o aérea en cualquier momento de la búsqueda. (Maps)

Google Maps permite guardar los mapas para posteriormente utilizarlos sin conexión, para lo cual se deberá seguir los siguientes pasos, con una conexión a internet desde el dispositivo móvil (Escobadia, 2016):

1. Abrir la aplicación de Google Maps con la zona o destino de interés.

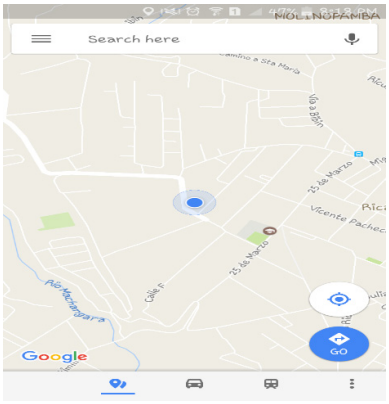


Figura 2.2: Selección de punto de interés Google Maps

Fuente: (Captura de pantalla Google Maps, Calle & Largo, 2017)

2. En el menú principal de la aplicación se selecciona la opción zonas sin conexión.

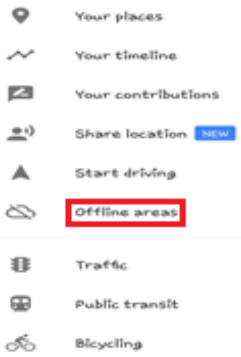


Figura 2.3: Selección opción Mapa Offline

Fuente: (Captura de pantalla Google Maps, Calle & Largo, 2017)

3. Permite descargar la zona que se haya seleccionado.

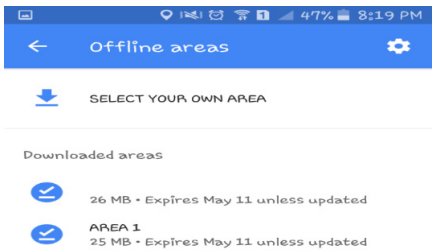


Figura 2.4: Selección de zona para descarga

Fuente: (Captura de pantalla Google Maps, Calle & Largo, 2017)

4. Indica el tamaño del mapa a descargar.

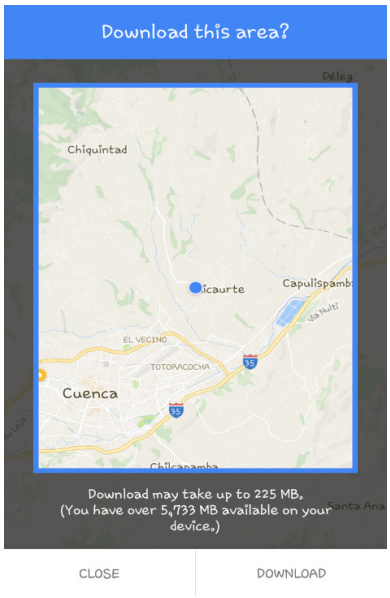


Figura 2.5: Descarga de mapa

Fuente: (Captura de pantalla Google Maps, Calle & Largo, 2017)

2.5 Realidad Aumentada

En los últimos años, la “realidad aumentada”, se ha convertido en una herramienta de gran utilidad en áreas como la medicina, educación, entretenimiento, turismo y robótica ya que, con su capacidad de insertar objetos virtuales en el mundo real, permite potencializar e innovar sus utilidades (Sanchez Escobedio, y otros, 2015).

La realidad aumentada es una tecnología que permite ver el mundo real, mediante una cámara (web o móvil), aumentando la información existente en nuestro entorno, mediante la implementación de elementos 2D, 3D, audiovisuales y multimedia (SL, 2016).

El uso de dispositivos móviles (Smartphone, tables) hoy en día se ha convertido en algo esencial siendo uno de los motivos con mayor importancia para el desarrollo de nuevas aplicaciones. Uno de ellos es la realidad aumentada que es una tecnología que permite una interacción de los elementos físicos con virtuales, dando como resultado al usuario una realidad virtual en tiempo real.

Un sistema de realidad aumentada, necesita de un dispositivo que permita obtener los datos en tiempo real, un equipo que permita procesar las imágenes reales y

convierta en virtual (software + procesador) y finalmente un medio para la proyección del resultado (pantalla).

Figura 2.6: Ejemplo de realidad aumentada creado con Aurasma

Fuente: (AumentaME, 2011)

Cabe destacar que existe una diferencia entre la realidad virtual y la realidad aumentada, la realidad virtual reemplaza el mundo real exterior con un mundo virtual, mientras que la realidad aumentada permite ver el mundo real y su alrededor con superposición de objetos 3D virtuales, lo cual logra una imagen de ilusión entre objetos del mundo real y virtual (Fernando, 2012).

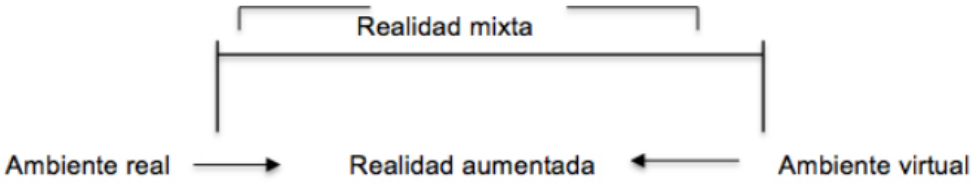


Figura 2.7: Realidad Aumentada vs Realidad Virtual ARToolKit

Fuente: (Fernando, 2012)

2.5.1 Métodos de activación de realidad Aumentada

Existen varios métodos para la activación de la realidad aumentada, las cuales pueden ser resumidas en:

1. **Dispositivo de reconocimiento del entorno:** Se necesita situar un objeto en el entorno que nos rodea, para lo cual se utiliza técnicas ya sea mediante

reconocimiento de imágenes captadas por una cámara de un dispositivo, por sensores o por geo posicionamiento.

2. **Cámara:** Capta la imagen que se está enfocando, para ser procesada y añadir la información digital.
3. **Basado en marcadores:** Utiliza imágenes del entorno, que son reconocidas por el dispositivo, las transforma en marcadores mediante la imagen recibida por la cámara, con lo cual se consigue la información de la posición del dispositivo, para colocar la imagen en realidad aumentada.
4. **Basado en posición y orientación del dispositivo:** Requiere de un sistema de geolocalización (GPS) y sistemas que reconozcan la orientación del dispositivo (brújula digital, acelerómetro), es por medio de estos dispositivos que se obtienen coordenadas, lo cual permite calcular la posición del objeto que debería visualizarme mediante la cámara del dispositivo. (Argelega, 2010)

2.5.2 Componentes de un sistema de realidad aumentada

Para la implementación de un sistema de realidad aumentada se requiere de componentes hardware y software, los cuales se detallan a continuación:

2.5.2.1 Componentes Hardware: Un sistema de realidad aumentada, al igual que un sistema informático, requiere de componentes hardware para llevar a cabo sus operaciones y procesos. (Pombo, 2010)

- **Cámara**: Toma la información del mundo real y lo envía hacia el software de realidad aumentada.
- **Pantalla para la visualización:** Dispositivo que muestra el resultado de lo real y lo virtual convertido en realidad aumentada.
- **Unidad de procesamiento:** Permite tomar los datos reales y convertirlos en realidad aumentada.
- **Componentes de almacenamiento:** Dispositivo que almacene la aplicación de realidad virtual.
- **Dispositivos de transmisión y recepción de datos**: Permiten la recepción y envió de información, mediante una conexión a internet. (Ordoñez, 2010)

Figura 2.8: Componentes de un sistema con realidad aumentada

Fuente: (Calle & Largo, 2017)

Hoy en día gracias a la tecnología, podemos tener todos los componentes de hardware integrados en un dispositivo móvil, el cual cumple con todos los requisitos para la implementación de un sistema de realidad aumentada.

2.5.2.2 Componentes Software: Una vez determinados los componentes hardware, se requiere de software para el funcionamiento del sistema de realidad aumentada, el cual puede ser mediante geo posicionamiento, imágenes o librerías de contenidos digitales. (AumentaME, 2011)

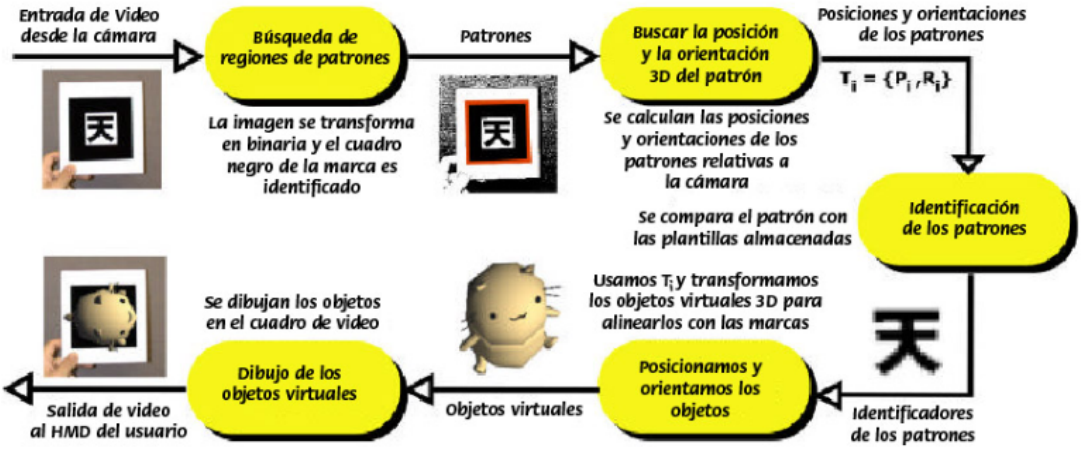


Figura 2.9: Esquema del proceso de ARToolKit, para el dibujo de un patrón sobre un objeto. Fuente: (ARToolkit)

2.5.3 Tipos de Realidad Aumentada

Se clasifican según sea el activador o marcador utilizado:

- Nivel 0: Código QR[27]
- Nivel 1: Patrón visual hecho ad-hoc B/N
- Nivel 2: Imagen 2d o una panorámica
- Nivel 3: Objeto 3d real o entorno 3d
- Nivel 4: Un punto del planeta (coordenadas GPS). (Garcia, 2015)

Figura 2.10: Evolución de la Realidad Aumentada

Fuente: (Izquierdo, 2010)

2.6 Algoritmos de ruteo

Un algoritmo de ruteo es aquel proceso que permite encontrar el camino óptimo dentro de una red, obteniendo el menor recorrido desde un punto de inicio hasta llegar a un punto de destino, específicamente el trayecto optimo entre dos puntos geográficos se lo conoce como enrutamiento geoespacial. (Ander, Borges, Yoseka, & Ainhoa, 2014)

Para realizar el cálculo de una ruta optima, el algoritmo de enrutamiento debe considerar múltiples factores:

- Tiempos de viaje
- Distancia a recorrer
- Costes
- Riesgos

[27]**CÓDIGO QR:** Es un código de barras bidimensional que puede almacenar los datos codificados (URL).

Para la representación de la realidad se lo realiza mediante grafos los cuales contienen información de los caminos a recorrer y cuáles son los costes de los recorridos que permitan llegar hacia el destino.

2.6.1 Representación de una red de caminos en forma de grafo

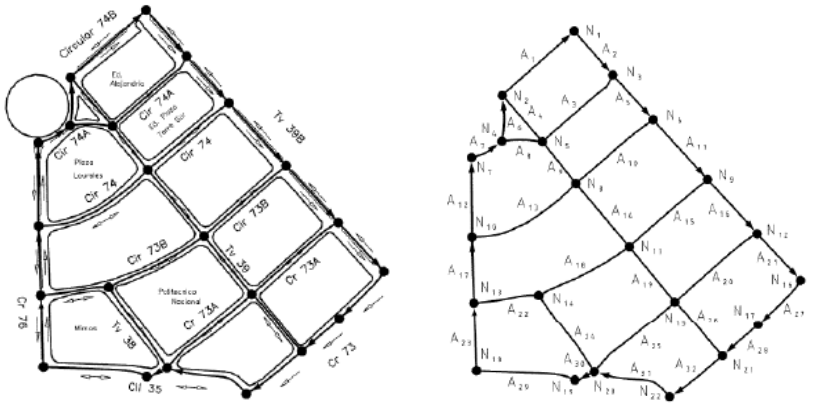
Un grafo esta compuestos esencialmente por nodos y aristas, los cuales permiten describir relaciones binarias entre los elementos que desee modelar. (Ander, Borges, Yoseka, & Ainhoa, 2014)

2.6.1.1 Nodos: Un nodo es aquel que permite la interconexión entre dos o más caminos en un punto geográfico, lo cual genera una ruta que puede dividir las direcciones conectadas, un nodo debe determinar restricciones tal como sucede en la vida real, por lo tanto un nodo representa la intersección entre caminos o puntos. (Ander, Borges, Yoseka, & Ainhoa, 2014)

2.6.1.2 Aristas: Representan los caminos que permiten recorrer de un nodo a otro, incluyen información sobre el coste del desplazamiento, el algoritmo de enrutamiento reconoce y determina cual es el nodo de inicio, el nodo destino y las aristas que debe recorrer para llegar a un destino, debido a que existen restricciones de dirección, en este proceso se realiza la suma de los costes de cada arista para poder optimizar una ruta, por lo tanto dos nodos pueden estar conectados por más de una arista. (Ander, Borges, Yoseka, & Ainhoa, 2014)

2.6.1.3 Costes: Se derivan de la longitud de las aristas, dependen de las siguientes variables:

- **Tiempo de viaje**: Combinan las longitudes entre aristas
- **Distancia recorrida:** Suma de la longitud de cada arista
- **Riesgos:** Determina la seguridad de las zonas a transitar.



a. b.

Figura 2.11: a. Modelo basado en grafos de sector vial

b. Grafo del modelo vial

Fuente: (Angel Restrepo & Marín Sepulveda, 2011)

2.6.2 Algoritmos de enrutamiento

Se debe determinar el tamaño del área geográfica y la complejidad que tiene el grafo para buscar un algoritmo de enrutamiento que permita obtener la mejor ruta sin requerir de procesos y tiempos elevados de búsquedas.

2.6.2.1 Algoritmos clásicos: Este tipo de algoritmo se expande de nodo a nodo, desde el punto inicial hasta todas las direcciones posibles hasta llegar al destino, realiza una búsqueda desde el nodo inicio al nodo final, dando una prioridad a las aristas con mayor proximidad al nodo destino con los menores pesos, un ejemplo de este algoritmo es el del Dijkstra.

El cual recorre todas las direcciones más cortas desde el nodo inicial, una vez que ha encontrado el camino más corto, mediante la suma de sus aristas este se detiene, debido a que hace un recorrido por todos los nodos en busca de la solución óptima, en grafos extensos el proceso se realiza con lentitud, pero obtiene una solución óptima frente a otras soluciones.

2.6.2.2 Algoritmos jerárquicos: Este tipo de algoritmo determina aristas o grafos con mayor prioridad que otros, una vez que se ha determinado esta jerarquía, el algoritmo solo usa las aristas de nivel bajo a las zonas cercanas al punto de inicio y destino para conectar con la arista de nivel superior.

2.6.2.3 Algoritmos dirigidos al objetivo: Este tipo de algoritmo tiene establecida la ruta, conoce el punto de partida y el destino, así como las aristas que tiene que recorrer, es por eso que es denominado algoritmo dirigido.

2.6.2.4 Algoritmos evolutivos: Este tipo de algoritmo se caracteriza por poseer un método heurístico, es decir permiten obtener valores para las variables de decisión que permiten satisfacer todas las restricciones, aunque estos valores no sean óptimos, pero cumplan con la función objetivo. (Ander, Borges, Yoseka, & Ainhoa, 2014)

CAPÍTULO 3 DESARROLLO DE SISTEMA

Introducción

Uno de los aspectos de mayor relevancia para el desarrollo de este proyecto son las herramientas a utilizar, ya que son las que permiten dar la funcionalidad y seguridad a la aplicación, así como también definir los diseños, que son como el usuario interactúa con la aplicación, establecer una arquitectura para la aplicación. Los requisitos funcionales y no funcionales que se tienen que cumplir para el desarrollo del aplicativo.

3.1 Herramientas y tecnologías utilizadas

3.1.1 Android Studio

Actualmente, es el entorno usado por los desarrolladores para aplicaciones Android, posee un potente editor de código, se basa en IntelliJ IDEA[28], ofrece múltiples funciones que ayudan a aumentar la productividad a la hora de la compilación de apps de Android entre ellas las siguientes:

- Sistema de compilación basado en Gradle flexible.
- Emulador de las aplicaciones con varias funciones.
- Entorno unificado, el cual permite desarrollar aplicaciones para todos los dispositivos Android.
- Permite realizar cambios en la aplicación mientras es ejecutada.
- Herramientas de Lint que permiten hacer seguimiento a problemas de rendimiento, usabilidad y compatibilidad.
- Soporte en línea con Google Cloud Plarform, para la integración con Cloud Messaging y App Engine (Creative Commons, 2015).

3.1.1.1 Estructura de un proyecto

Un proyecto desarrollado en Android Studio, contiene unos a varios módulos con archivos fuentes y de recursos:

- Módulos de Apps para Android
- Módulos de bibliotecas
- Módulos de Google App Engine

[28]**INTELLIJ IDEA:** Entorno de desarrollo Java, bajo concepto no instructivo e intuitivo, que permite escribir, depurar sin mayores complicaciones. (Arlandy Rodriguez, 2012)

Los archivos de un proyecto en Android Studio se organizan en módulos, con el fin de proporcionar un rápido acceso a cada uno de los archivos del proyecto, los archivos de compilación son visibles en el nivel superior de secuencias de comando gradle[29] y cada módulo de la aplicación contiene las siguientes carpetas:

- **Manifests**: Contiene el archivo AndroidManifest.xml.
    - El archivo AndroidManifest, nos permite realizar las configuraciones de la app, tales como:
        - Permisos de acceso de la aplicación al dispositivo (cámara, GPS, ubicación, acceso a internet).
    - Icono para la instalación de la aplicación en el dispositivo móvil.
    - Determina las bibliotecas, actividades, servicios, contenidos con las cuales debe tener relación la aplicación para su funcionamiento.
    - Determina la Actividad (pantalla) que será lanzada inicialmente. (Creative Commons, 2016)

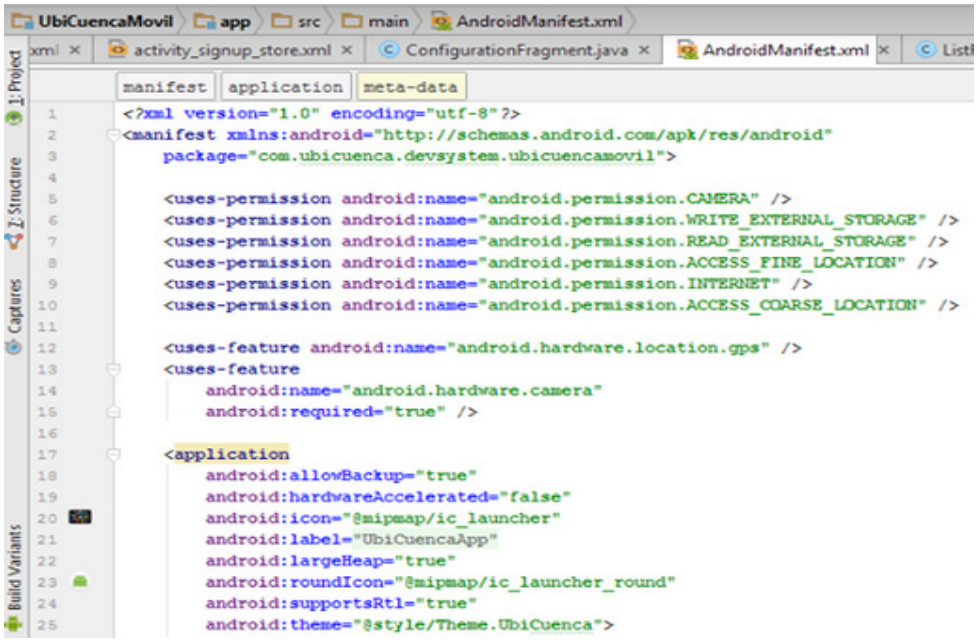


Figura 3.1: Fragmento de código del archivo AndroidManifest

Fuente: (Calle& Largo 2017)

- **Java**: Contiene los archivos de código fuente java.

[29]**GRADLE:** Herramienta de automatización para el proceso de desarrollo del proyecto, permite la compilación del código, el cual verifica si existen cambios. (Rodriguez, 2014)

- **Res**: Contiene todos los recursos requeridos por la aplicación tales como diseños XML, imágenes, IU, etc. (Creative Commons, 2015)

3.1.1.2 Interfaz de Usuario

La interfaz de usuario de un proyecto desarrollado en Android Studio contiene las siguientes áreas:

1. **Barra de herramientas**: Permite realizar el control de acciones, como la ejecución de la aplicación y guardar cambios.
2. **Barra de navegación:** Permite la exploración de los archivos del proyecto.
3. **Ventana de editor:** Área en donde se puede editar el código de la aplicación.
4. **Barra de la ventana de herramientas**: Se encuentra en la parte externa de la ventana del IDE[30] (Entornos Integrados de Desarrollo), contiene los botones que permiten expandir o contraer las ventanas de las herramientas del proyecto.
5. **Ventana de herramientas:** Permite la administración de los proyectos, así como también de las tareas y controles de versiones.
6. **Barra de estado:** Muestra el estado actual del proyecto indicando si existe algún error o advertencia mediante mensajes. (Creative Commons, 2015)

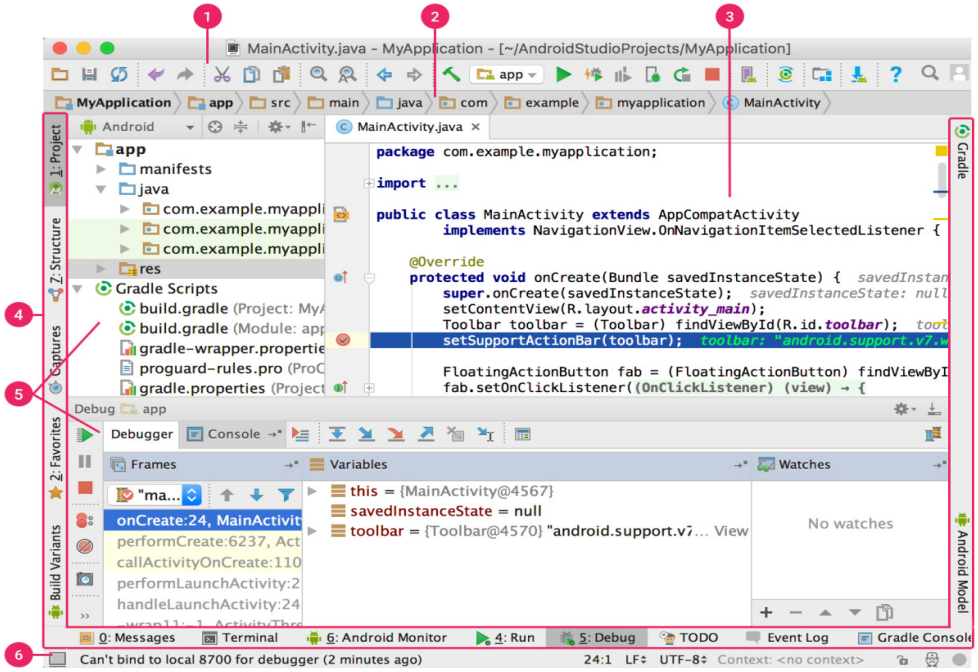


Figura 3.2: Interfaz de Android Studio

Fuente: (Creative Commons, 2015)

[30]**IDE:** Entornos Integrados de Desarrollo, software que integra herramientas para el desarrollo de aplicaciones, (Academia Android, 2014)

3.1.2 MySql Workbench

Es una herramienta para el diseño y administración de bases de datos para el sistema MySql, el cual permite:

- Configuraciones y conexiones de la base de datos.
- Editor de consultas SQL (Structured Query Language o lenguaje de consulta estructurada).
- Modelado de base de datos.
- Administración de la base de datos. (Oracle Corporation, 2017)

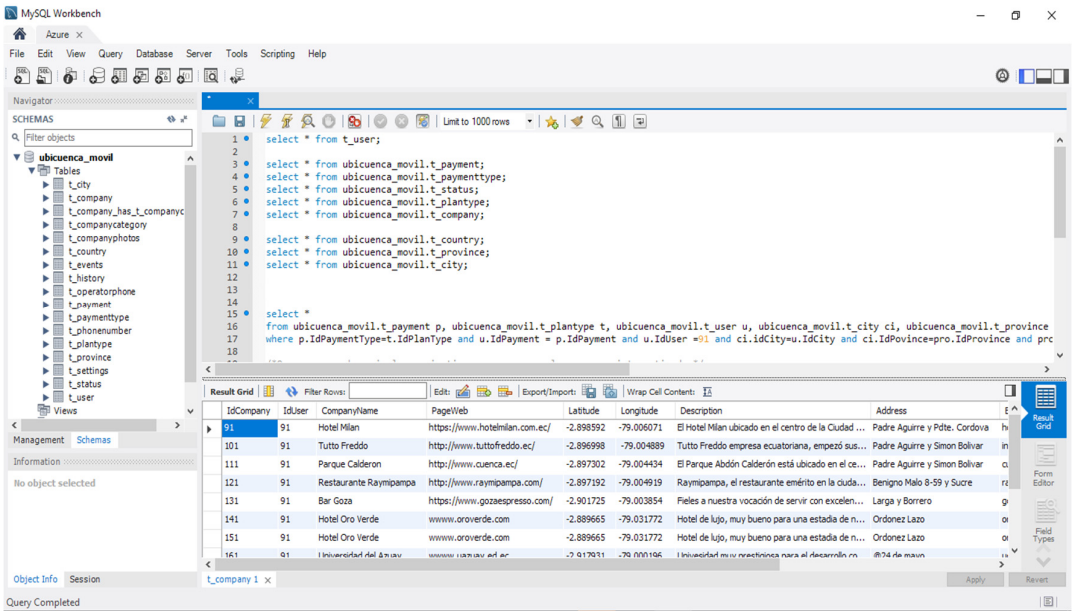


Figura 3.3: Interfaz de MySql Workbench

Fuente: (Oracle Corporation, 2017)

3.1.3 Base de datos SQLite

Es un gestor de bases de datos de código abierto que se caracteriza por ser extremadamente ligero. Almacena la base de datos en un único fichero que no requiere instalación alguna ya que Android incluye soporte a SQLite. No requiere de un servidor pues funciona localmente. Android incluye una API [31] (Aplication Programming Interface o Interfaces de programación de aplicaciones) completa para la gestión de bases de datos SQLite. (Lopez, 2015)

[31]**API: A**plication **P**rogramming **I**nterface, contiene un conjunto de protocolos y herramientas que pueden ser usados para el desarrollo de aplicaciones, el cual le permite dar funcionalidad.

3.1.3.1 Clase SQLiteOpenHelper: Permite la administración de la base de datos, es una clase auxiliar, la cual posee un solo constructor y dos métodos:

- OnCreate()

  Método que es llamado cuando la base de datos no existe, conecta a la base y crea según los parámetros enviados.
- OnUpgrade()

  Método que es llamado cuando la base de datos existe, pero la versión es anterior a la enviada, es por eso que envía los parámetros y convierte a la versión actual de la base de datos. (Lopez, 2015)

```
public class UsuariosSQLiteHelper extends SQLiteOpenHelper {

    //Sentencia SQL para crear la tabla de Usuarios
    String sqlCreate = "CREATE TABLE Usuarios (codigo INTEGER, nombre TEXT)";

    public UsuariosSQLiteHelper(Context contexto, String nombre,
                                Cursor Factoryfactory, int version) {
        super(contexto, nombre, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabasedb) {
        //Se ejecuta la sentencia SQL de creación de la tabla
        db.execSQL(sqlCreate);
    }

    @Override
    public void onUpgrade(SQLiteDatabasedb, intversionAnterior, intversionNueva) {
        //NOTA: Por simplicidad del ejemplo aquí utilizamos directamente la opción de
        //      eliminar la tabla anterior y crearla de nuevo vacía con el nuevo formato.
        //      Sin embargo lo normal será que haya que migrar datos de la tabla antigua
        //      a la nueva, por lo que este método debería ser más elaborado.

        //Se elimina la versión anterior de la tabla
        db.execSQL("DROP TABLE IF EXISTS Usuarios");

        //Se crea la nueva versión de la tabla
        db.execSQL(sqlCreate);
    }
}
```

Figura 3.4: Clase SQLiteOpenHelper

Fuente: (Lopez, 2015)

La clase SQLiteOpenHelper nos permite utilizar métodos insert(), update() y delete(), los cuales separan tablas, valores y condiciones en parámetros independientes, según el método que sea enviando. El método recibe tres parámetros:

1. Nombre de la tabla.
2. Se envían en casos puntuales, ej. Insertar registros vacíos.
3. Valores de registro que se vayan a enviar. (Lopez, 2015)

```
//Creamos el registro a insertar como objeto ContentValues
ContentValues nuevoRegistro = new ContentValues();
nuevoRegistro.put("codigo", "6");
nuevoRegistro.put("nombre","usuariopru");

//Insertamos el registro en la base de datos
db.insert("Usuarios", null, nuevoRegistro);
```

Figura 3.5: Método insert() de la clase SQLiteOpenHelper

Fuente: (Lopez, 2015)

Para los métodos update () y delete () los parámetros son similares, pero incluye un parámetro adicional la condición WHERE.

```
//Establecemos los campos-valores a actualizar
ContentValues valores = new ContentValues();
valores.put("nombre","usunuevo");

//Actualizamos el registro en la base de datos
db.update("Usuarios", valores, "codigo=6", null);
```

```
//Eliminamos el registro del usuario '6'
db.delete("Usuarios", "codigo=6", null);
```

Figura 3.6: Método update() y delete() de la clase SQLiteOpenHelper

Fuente: (Lopez, 2015)

3.1.3.2 Método execSQL: La API de Android de la base de datos nos permite la manipulación de los datos almacenados (inserción, actualización y eliminación), mediante el método execSQL() de la clase SQLiteDatabase, el cual permite ejecutar sentencias SQL. (Lopez, 2015)

```
//Insertar un registro
db.execSQL("INSERT INTO Usuarios (codigo,nombre) VALUES (6,'usuariopru') ");

//Eliminar un registro
db.execSQL("DELETE FROM Usuarios WHERE codigo=6 ");

//Actualizar un registro
db.execSQL("UPDATE Usuarios SET nombre='usunuevo' WHERE codigo=6 ");
```

Figura 3.7: Método execSQL

Fuente: (Lopez, 2015)

Las bases de datos SQLite que son creadas por las aplicaciones Android se almacenan en la memoria del dispositivo móvil y pueden ser accedidas solo por la aplicación propietaria (Lopez, 2015), contienen la siguiente ruta:

/data/data/paquete.java.de.la.aplicacion/databases/nombre_base_datos

3.1.3.3 Método rawQuery: Los métodos explicados anteriormente ofrecen funcionalidades para insertar, actualizar y eliminar datos, pero una de las tareas de mayor importancia es la recuperación y selección de datos de un registro de una base de datos SQLite. El método rawQuery de la clase SQLiteDatabase, recibe como parámetro un comando SQL, en donde se envían los campos que se desean obtener según los criterios de selección, el resultado obtenido es almacenado en un vector, el cual puede ser recorrido para obtener el registro que se requiera. (Lopez, 2015)

```
Cursor c = db.rawQuery(" SELECT codigo,nombre FROM Usuarios WHERE nombre='usu1' ", null);
```

Figura 3.8: Método rawQuery

Fuente: (Lopez, 2015)

3.1.3.4 Método Query: Perteneciente a la clase SQLiteDatabase, este método recibe varios parámetros: el nombre de la tabla, un arreglo con los campos que se desean obtener, la cláusula WHERE, este método permite hacer uso de las siguientes clausulas: (Lopez, 2015)

- GROUP BY
- HAVING
- ORDER BY

```
String[] campos = newString[] {"codigo", "nombre"};
String[] args = newString[] {"usu1"};

Cursor c = db.query("Usuarios", campos, "usuario=?", args, null, null, null);
```

Figura 3.9: Método Query

Fuente: (Lopez, 2015)

Los resultados de un objeto Cursor, pueden ser procesados y recorridos para obtener su información mediante métodos de la claseCursor, los cuales son:

- moveToFirst(): mueve el puntero del cursor al primer registro devuelto.
- moveToNext(): mueve el puntero del cursor al siguiente registro devuelto. (Lopez, 2015)

```
String[] campos = new String[] {"codigo", "nombre"};
String[] args = new String[] {"usu1"};

Cursor c = db.query("Usuarios", campos, "nombre=?", args, null, null, null);

//Nos aseguramos de que existe al menos un registro
if(c.moveToFirst()) {
     //Recorremos el cursor hasta que no haya más registros
     do{
          String codigo= c.getString(0);
          String nombre = c.getString(1);
     } while(c.moveToNext());
}
```

Figura 3.10: Métodos de la claseCursor

Fuente: (Lopez, 2015)

3.1.4 Método de autenticación OAUTH

Es un método de autorización para que un servicio pueda identificar al consumidor que accede a sus servicios, el objetivo de este método es añadir una capa de seguridad a los servicios, así como también proteger los datos de los usuarios cuando se concede acceso a aplicaciones de terceros. OAuth protege los servicios y datos mediante el uso de un código (token de acceso). (Garcia Delgado, 2014)

Para un correcto proceso de autorización, se basa en el siguiente protocolo:

- **Usuario**: Es quien hace uso del recurso y concede acceso a los recursos protegidos.
- **Cliente**: Es la aplicación que realiza solicitudes de recursos protegidos en nombre del propietario.
- **Servidor de autorización**: Es el sistema con el que ha de comunicarse el cliente, para que el usuario autorice el acceso a los recursos, después de autentificar correctamente, el proceso es representado por un token de acceso.
- **Token de acceso**: Respuesta por parte del servidor de autorización como respuesta a una solicitud correcta del cliente, permite al servidor identificar quien está intentando acceder al recurso y en nombre de quien, el token tiene un tiempo de vida específico.
- **Servidor del recurso**: Es el sistema que posee los recursos protegidos, es el que acepta y responde a las peticiones si estas son correctas mediante el uso del token. (Hardt, 2012)

3.1.5 Material Design

Está basado en objetos materiales, que son colocados en un espacio (lugar) y un tiempo (movimiento) (Perez, 2014), permite añadir diseños visuales con innovación para los usuarios, otorgando una experiencia de alta tecnología, sin importar la plataforma y tamaño de los dispositivos, una de sus limitantes principales es que se requiere que la versión de Android sea 5.0 (nivel de API 21) o superior. (Creative Commons, 2016)

Según (Perez, 2014) se basa en un diseño donde la profundidad, las superficies, los bordes, las sombras y los colores juegan un papel muy importante.

3.1.5.1 Objetivo de Material Design

Para el cumplimiento de los estilos y mejorar la experiencia del usuario, material design, se basa en los siguientes objetivos:

- Crear un lenguaje visual que cumpla con principios de un buen diseño y que estas se acoplen a las nuevas tecnologías.
- Crear un sistema que permita que el diseño se adapte en diferentes plataformas y dispositivos con la misma experiencia de uniformidad. (Aula formativa Online, 2016)

3.1.5.2 Principios de Material Design

Lo material como una metáfora: Principio que determina que lo material está unido a la realidad, es decir, que lo material se fusiona con el espacio y movimiento lo cual permite la integración con la tecnología y al mismo tiempo diseños con creatividad e imaginación. Como resultado obtenemos interfaces con superficies y bordes basadas en la realidad, con uso de atributos que permiten entender cada elemento de la interfaz, cabe destacar que el principio de la luz, superficie y movimiento son claves ya que trasmiten como los objetos interactúan el uno con el otro.

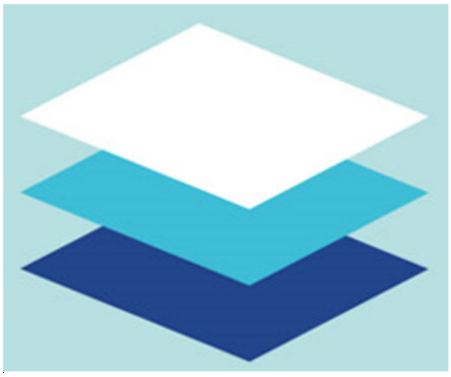


Figura 3.11: Principios de Material Design (material)

Fuente: (Lopez, 2015)

Llamativo e intencional: Principio que determina los elementos visuales de una aplicación haciendo uso de la teoría del diseño. Maneja y controla el uso de los colores, los espacios en blanco, sistemas de grillas, imágenes de calidad, tipografías de tamaño correcto. Todo este conjunto ayuda a determinar la jerarquía y lectura visual desde el lado del usuario.

Figura 3.12: Principios de Material Design (llamativo)

Fuente: (Lopez, 2015)

El movimiento otorga significado: Principio que determina que las animaciones deben tener la capacidad de captar la atención de los usuarios, pero sin interrumpir la experiencia de entender la interfaz, es por eso que debe existir una correcta relación entre las animaciones y la esencial de la aplicación.

Figura 3.13: Principios de Material Design (movimiento)

Fuente: (Lopez, 2015)

3.1.6 Retrofit - comunicación con el servidor

Está constituido como un cliente REST (Representational State Transfer)[32] para Android y Java, desarrollada por Square[33], permite hacer peticiones HTTP a un servidor:

[32]**REST:** Protocolo Cliente-Servidor que lleva información para ser procesada bajo las operaciones GET, POST, PUT, etc. (Gonzales Diez, 2016)
[33] **SQUARE**: Organización de desarrollo de bibliotecas de código abierto. (Square, Inc, 2016)

- GET: Petición al servidor en la cual se espera una respuesta con datos.
- POST: Petición al servidor en la cual se espera un OK/NOK.
- PUT: Su funcionamiento es similar al método POST, con la diferencia que el método PUT, integra mayor funcionalidad.
- DELETE: Petición al servidor de borrado, en la cual se espera un OK/NOK.
- HEAD: Su funcionamiento es similar al método GET, con la diferencia que su respuesta no contendrá un body. (Gonzales Diez, 2016)

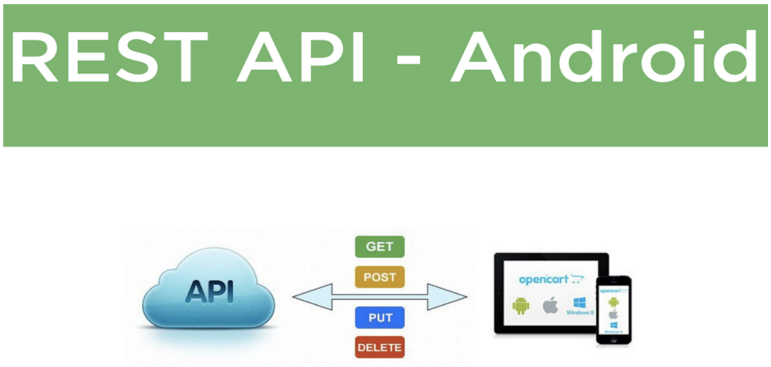


Figura 3.14: Retrofit - comunicación con el servidor

Fuente: (Garibay Cadenas, 2016)

3.1.6.1 Estructura de un cliente REST: Para su implementación y funcionamiento involucra tres conceptos:

- POJO: (Plain Old Java Object) modelo o clases, que contiene los objetos JSON [34] que devuelve el servidor.
- Interface: permite la creación de una clase que es la encargada de controlar las rutas que serán llamadas mediante los métodos GET y POST, se denomina una clase "servicio".
- Adaptador REST: Clase que tendrá la función del cliente REST. (Delgado Meraz, 2015)

[34] **JSON: (JavaScript Object Notation)** es un formato para el intercambio de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos. (Esquiva Rodriguez, 2014)

Retrofit es una librería de terceros por la tanto es necesario modificar la sección de *dependencies* del archivo *build.gradle*, de la carpeta *app* del proyecto, se debe tener una conexión activa al internet para realizar la descarga de los paquetes.

```
dependencies {
  compile 'com.google.code.gson:gson:2.1.0'
  compile 'com.squareup.retrofit2:retrofit:2.1.0'
  compile 'com.squareup.okhttp:okhttp-urlconnection:2.1.0'
}
```

3.1.6.2 Construcción del cliente REST: Una vez que el proyecto ha descargado las librerías y permisos requeridos, se debe crear una clase la cual realizará la función de adaptador REST, es la encargada de conectar las rutas del API con los métodos de nuestra interface.

```
public static Retrofit getRetrofit() {

    HttpLoggingInterceptor interceptor = new HttpLoggingInterceptor();
    interceptor.setLevel(HttpLoggingInterceptor.Level.BODY);
    OkHttpClient client = new OkHttpClient.Builder().addInterceptor(interceptor).build();

    return new Retrofit.Builder()
            .baseUrl(AppConfig.BASE_URL)
            .client(client)
            .addConverterFactory(GsonConverterFactory.create())
            .build();
}
```

Figura 3.15: Inicialización de la conexión a nuestros servicios

Fuente: (Calle & Largo, 2017)

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

    datos = OperacionesBaseDatos
            .obtenerInstancia(getApplicationContext());

    // Crear conexión al servicio REST
    getResponse = AppConfig.getRetrofit().create(FormatRequestApi.class);
```

Figura 3.16: Construcción del cliente REST

Fuente: (Calle & Largo, 2017)

3.1.7 NinjaMock – Prototipo de App

Permite realizar el diseño de la aplicación, se basa en la herramienta *drag&drop*, con lo cual se puede arrastrar los elementos y moverlos, para conseguir un diseño para un proyecto Android, es una herramienta Online, se debe crear una cuenta para poder hacer uso de sus herramientas.

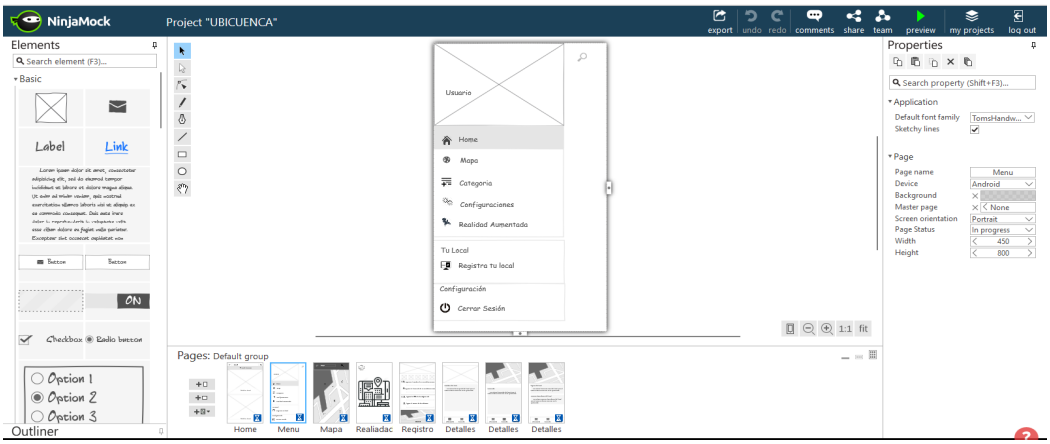


Figura 3.17: Interfaz NinjaMock

Fuente: (NinjaMock Online)

3.1.8 NetBeans

Es un entorno de desarrollo integrado (IDE) modular basado en estándares. Utiliza JAVA como lenguaje de programación, la cual es de código abierto, en la versión 8.2 que es la utilizada para el desarrollo de nuestros servicios web, soporta las siguientes tecnologías:

- Java EE 7, Java EE 6, and Java EE 5
- JavaFX 2.2.x and 8
- Java ME SDK 8.0
- Java Card 3 SDK
- Struts 1.3.10
- Spring 4.0.1, 3.2.7, 2.5
- Hibernate 4.2.6, 3.6.10
- Issue Tracking
  - Bugzilla 4.4 and earlier
  - Jira 3.13+ and 5.0+
- Node.js 4.0+
- PHP 7, 5.6, 5.5, 5.4, 5.3, 5.2, 5.1
- Groovy 2.1
- Grails 2.3, 2.2
- Apache Ant 1.9.7
- Apache Maven 3.0.5 and earlier
- C/C++/Fortran
- VCS
  - Subversion: 1.8, 1.7, 1.6
  - Mercurial: 2.8.x and later
  - Git 1.8.x and later

Figura 3.18: Tecnologías Soportadas por NetBeans

Fuente: (Oracle Corporation, 2016)

Uno de los requisitos para poder instalar NetBeans, es que nuestro sistema debe tener instalado Java SE Development Kit (JDK), el cual puede ser descargada de la página de java.

Figura 3.19: Descarga del JDK de java

Fuente: (Java SE Development Kit)

Para el desarrollo de los Servicios Web se utilizara PHP 5.6 desde la plataforma NetBeans.

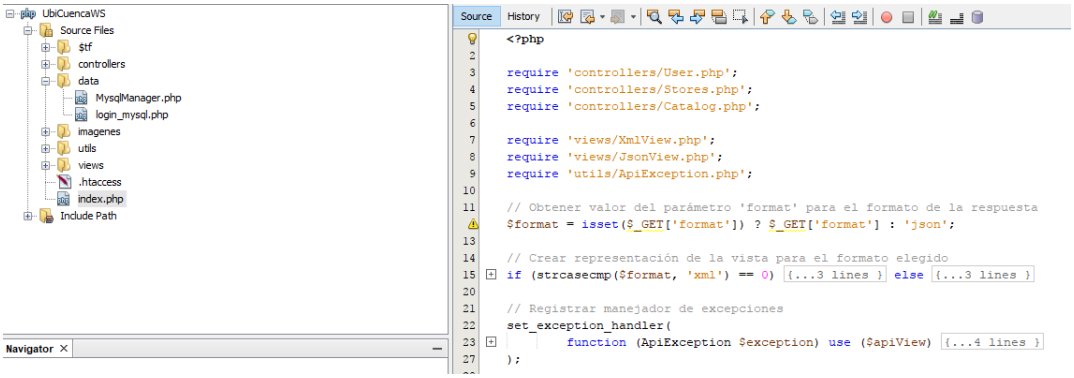


Figura 3.20: Index del WebService

Fuente: (Calle & Largo, 2017)

3.1.9 PhayPhone

Es un método de pago que puede ser implementado en las aplicaciones móviles o páginas web de comercio, permite realizar el registro de tarjetas de crédito en la aplicación, para posteriormente con el número de celular realizar los pagos, la aplicación genera una clave personal que autoriza la forma de pago, de esta forma se elimina que el usuario entregue sus tarjetas de crédito o débito al momento del sus pagos, es por ello que se garantiza que la transacción sea segura. Para obtener acceso a la información de desarrolladores debemos realizar un proceso de registro, en la página web de (PayPhone, 2017)

3.1.9.1 Pasos para un proceso de pago

Para llevar a cabo el proceso de pago, el API de PayPhone sigue los siguientes procedimientos:

1. Solicitud de regiones(GetRegions)
2. Solicitud de moneda(GetCurrency)
3. Verificación de datos para realizar una venta
    a. Verificación de datos previos al envío de la transacción
        i. Verificación de datos (SetTransaction)
        ii. Inicio de transacción (DoTransaction)
    b. Envío directo sin la verificación de datos
        i. Envío de transacción (SetAndDoTransaction)
4. Comprobación de estado de la transacción.
    a. Consulta por Id de transacción. (GetTransactionStatusById)
    b. Consultar por id de transacción de cliente (GetTransactionStatusByClientId)

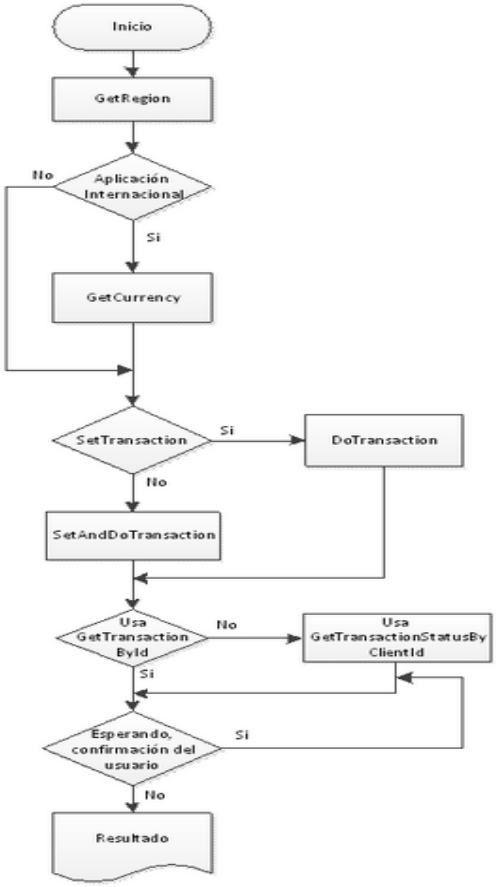


Figura 3.21: Proceso de pago PayPhone

Fuente: (PayPhone, 2017)

3.1.9.2 Métodos utilizados

Para la implementación de *PayPhone* dentro de la aplicación móvil se han utilizado los siguientes métodos del API:

- **SetAndDoTransaction:** Es la unión del método *SetTransaction* y *DoTransaction*, los cuales envían los datos de las transacción a PayPhone y devuelve los detalles del usuario que va a realizar el pago verificando su identidad, para posteriormente notificar al cliente del cobro que se realizó.

```
private class  SetAndDoTransaction extends AsyncTask<String, Integer, Boolean> {
  boolean resul = false;
  NumberVerifyResponseModel verifyNumber = new NumberVerifyResponseModel();
  protected Boolean doInBackground(String... params) {
     try {
        verifyNumber = instance.SetAndDoTransaction(data, phoneNumber, countryCode, typeCharge, nickname);
        transactionId = verifyNumber.getTransactionId();
        resul = true;
     } catch (PayPhoneWebException ex) {
        error = ex.getError();
      }
     return resul;
  }
  protected void onPostExecute(Boolean result) {
     pDialog.dismiss();
     if (result) {
     }
  }
}
```

Figura 3.22: Método SetTransaction de PayPhone en Android

Fuente: (PayPhone, 2017)

- **GetTransactionStatusById y GetTransactionStatusByClientId**: Métodos utilizados para obtener el estado de la transacción por su identificador, el cual puede ser por el id de la transacción o por *Clientid* y como resultado devuelve un estado indicando la autorización y datos de la transacción.

```
private class GetTransactionByClientId extends AsyncTask<String, Integer, Boolean> {
  boolean resul = false;
  BaseStatusResponseModel tx = new BaseStatusResponseModel();
  protected Boolean doInBackground(String... params) {
    try {
      tx = instance.GetTransactionById(transactionId);
      resul = true;
    } catch (PayPhoneWebException ex) {
      error = ex.getError();
      Log.e("ServicioRest", "Error!", ex);
    }
    return resul;
  }
  protected void onPostExecute(Boolean result) {
    int pending =TransactionStatus.Pending.getStatus();
    int canceled =TransactionStatus.Canceled.getStatus();
    int approved =TransactionStatus.Approved.getStatus();
    if (result) {
      if (tx.getStatus() == pending) {
        TareaWSStatusTransaction tarea = new TareaWSStatusTransaction();
        tarea.execute();
      } else {
        ResultResponseModel resultStatus = (ResultResponseModel) tx;
      }
    }
}
```

Figura 3.23: Método GetTransaction de PayPhone en Android

Fuente: (PayPhone, 2017)

3.1.10 Implementación de Google Maps

Permite hacer uso de sus servicios y características, para lo cual se puede realizar la implementación en las aplicaciones Android, se debe seguir las siguientes configuraciones y procedimientos:

3.1.10.1 Descargar y configurar Google Play Services: Es un conjunto de librerías de Google para poder hacer uso de sus funcionalidades, en este caso se realiza las configuraciones para Google Maps.

1. Se procede a la instalación del SDK, el cual permitirá agregar la dependencia requerida.

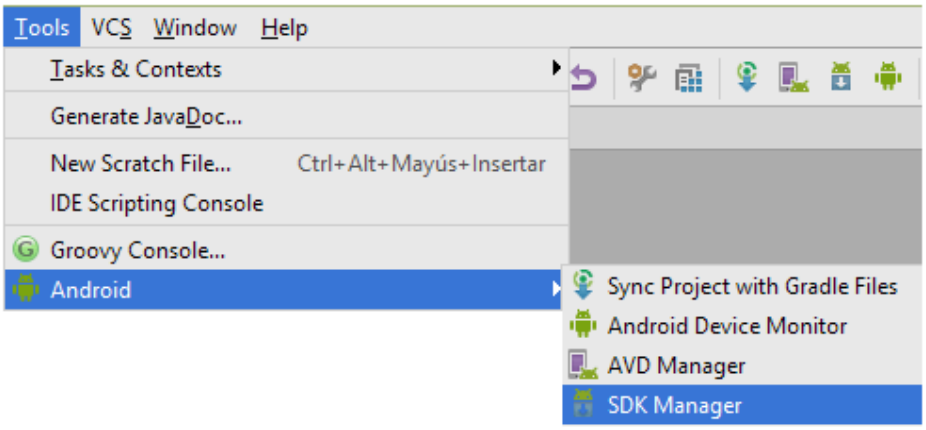


Figura 3.24: Configuración de Google Play Services

Fuente: (Revelo, 2016)

2. En las opciones del SDK Tools, se debe activar el paquete de Google Play Service y se confirma la instalación del servicio.

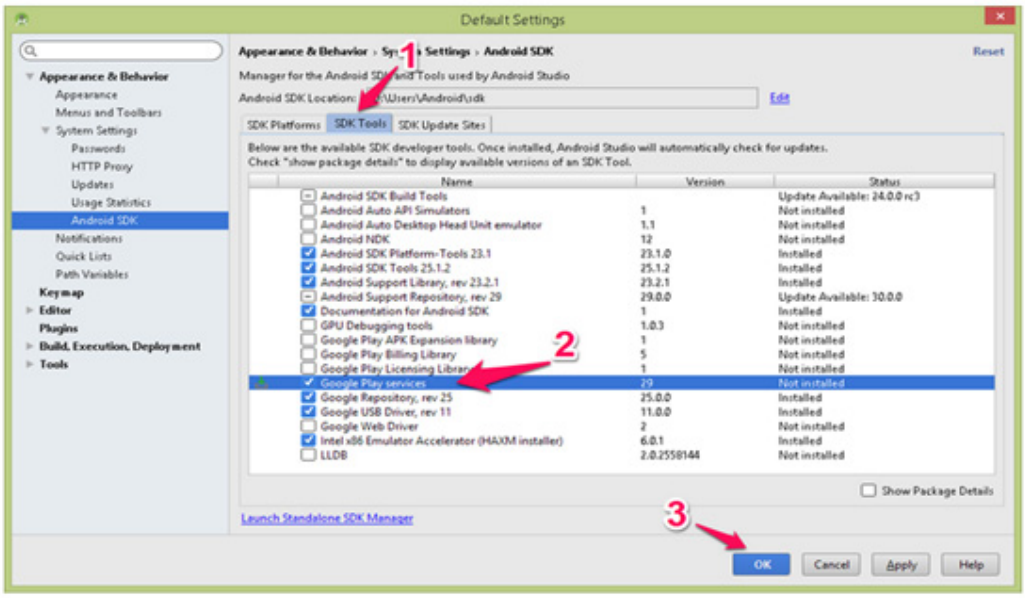


Figura 3.25: Activación de Google Play Services

Fuente: (Revelo, 2016)

3. Se acepta los términos de la licencia y se procede a la instalación de los servicios.

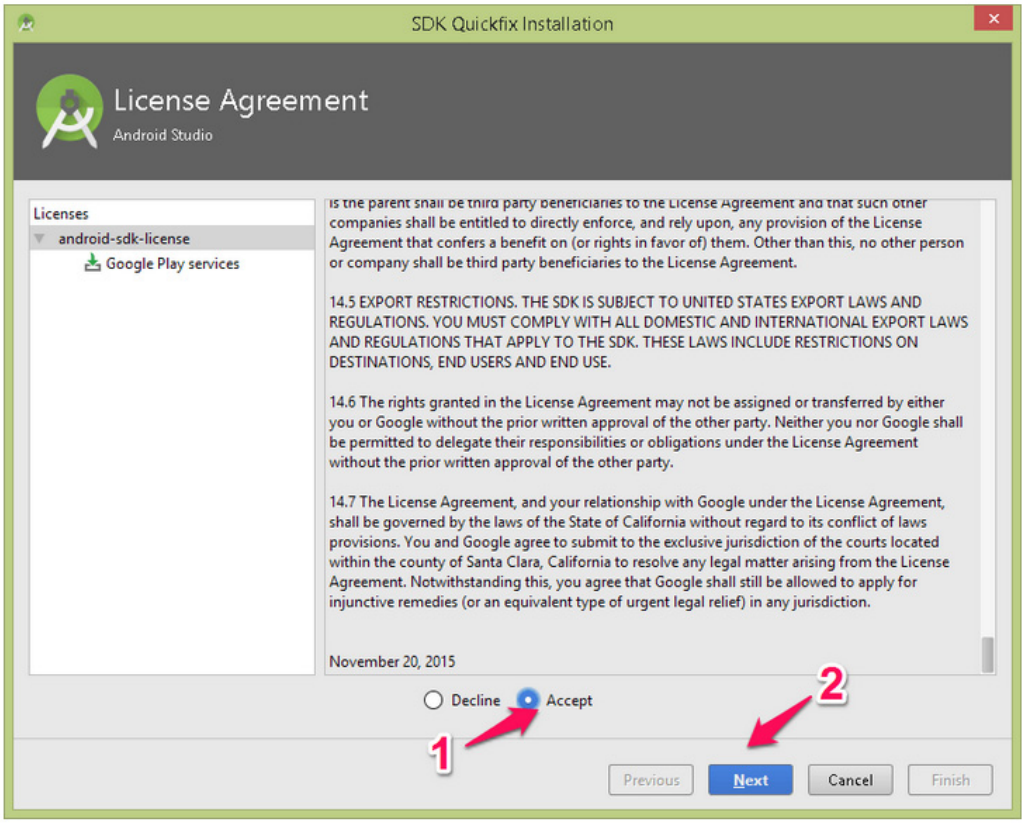


Figura 3.26: Aceptación de licencias

Fuente: (Revelo, 2016)

3.1.10.2 Añadir dependencia a build.gradle

Configuración que se debe realizar de la siguiente manera:

1. Seleccionar la ruta del siguiente directorio: File > Project Structure
2. Seleccionar el modulo app, luego la pestaña *Dependencies* y se procede a agregar una *Library Dependency.*

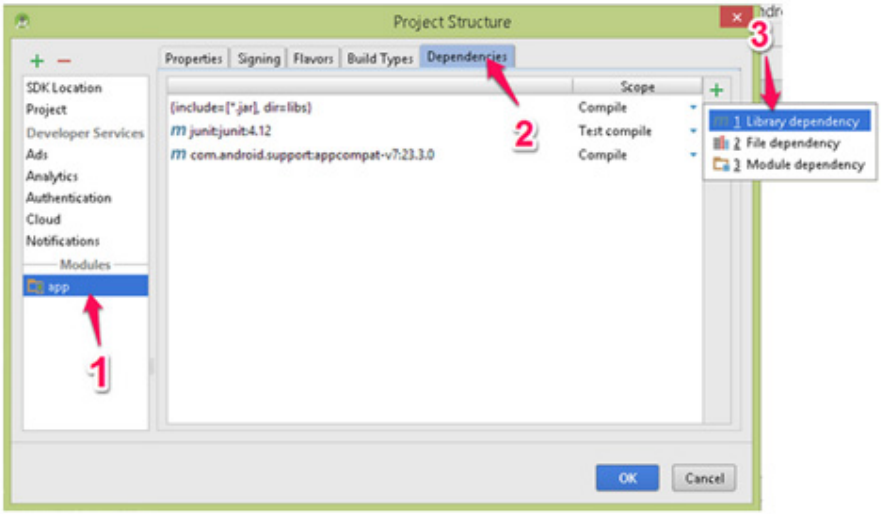


Figura 3.27: Dependencia build.gradle

Fuente: (Revelo, 2016)

3. Se agrega la librería “com.google.android.gms” que contenga el complemento “play-services.maps:v”

3.1.10.3 Obtención del Fingerprint SHA-1

La obtención de esta clave sirve para la creación de un entorno de desarrollo de Google y a su vez habilita los servicios en los servidores de Google*, el cual permite* hacer uso del *API* de Google Maps*,* mediante el siguiente proceso:

1. Se debe localizar el archivo debug.keystore el cual se encuentra en el siguiente directorio:
   - C:\Users\nombre_usuario\.android\
2. Mediante la consola nos ubicamos en la dirección del archivo y con el siguiente comando obtenemos la clave que contiene el archivo:
   - keytool -list -v -keystore "debug.keystore" -alias androiddebugkey -storepass android -keypass android

   El proceso anterior devolverá como resultado una clave de tipo SHA1 del siguiente tipo:

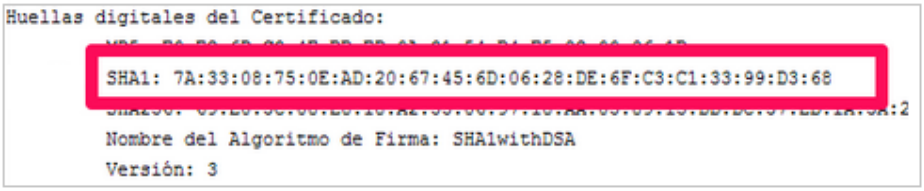


Figura 3.28: Fingerprint SHA-1

Fuente: (Revelo, 2016)

La clave obtenida servirá para la creación de proyectos en *Google Console Developers*, esta nos da acceso al API de Google Maps, la clave del API generada, deberá ser colocada en las credenciales del proyecto. (Revelo, 2016)

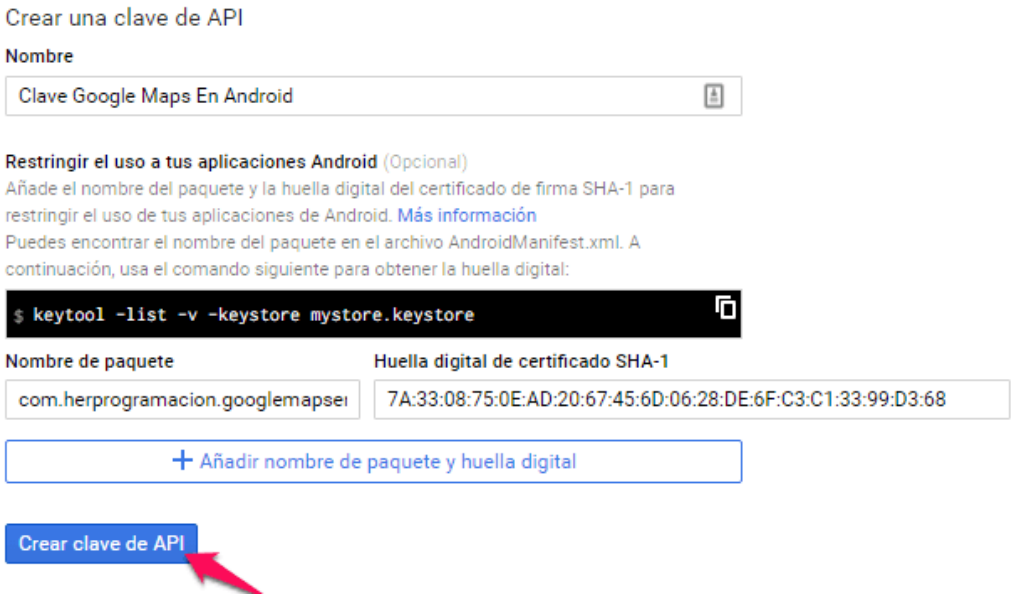


Figura 3.29: Creación de clave de API

Fuente: (Revelo, 2016)

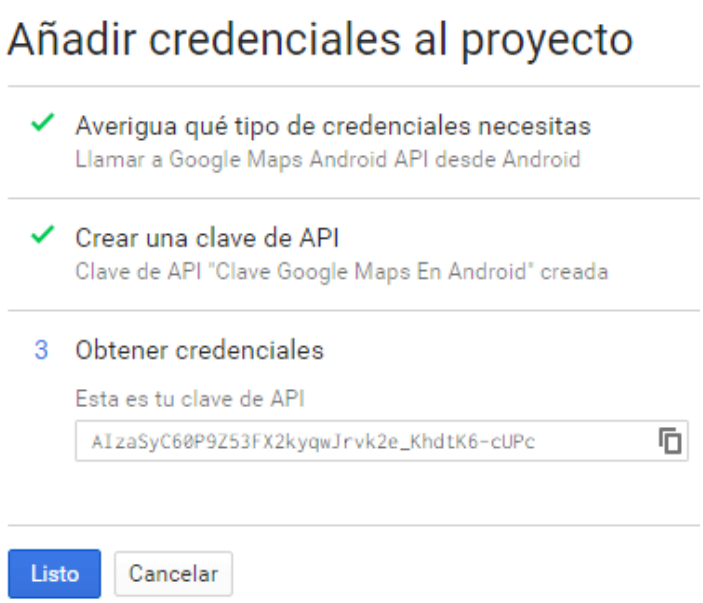


Figura 3.30: Credenciales del proyecto

Fuente: (Revelo, 2016)

3.1.10.4 Agregar clave de la API a la aplicación

En el archivo AndroidManifest.xml se agrega un <meta-data> al nodo <application> con la siguiente estructura, con la clave que se obtuvo en el apartado (3.1.10.3).

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyC60P9Z53FX2kyqwJrvk2e_KhdtK6-cUPc" />
```

Figura 3.31: Clave API para aplicación

Fuente: (Revelo, 2016)

3.1.10.5 Añadir un mapa a la aplicación

Para añadir un mapa de Google Maps y sus características a una aplicación se debe seguir los siguientes pasos y estructura para su correcto funcionamiento:

Añadir un fragmento de mapas dentro de una etiqueta <*fragment*> cuya referencia sea *MapFragment* o *SupportMapFragment, dentro de la clase activity_first_map.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".FirstMapActivity" />
```

Figura 3.32: Código para integrar Google Maps a la aplicación

Fuente: (Revelo, 2016)

3.1.10.6 Añadir un fragmento dinámicamente con el Fragment Manager

Se obtiene la instancia relacionada al contexto con getSupportFragmentManager() y se realiza una transacción add() en onCreate() de la actividad.

```
public class FirstMapActivity extends AppCompatActivity {

    private FirstMapFragment mFirstMapFragment;

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

        mFirstMapFragment = FirstMapFragment.newInstance();
        getSupportFragmentManager()
                .beginTransaction()
                .add(R.id.map_container, mFirstMapFragment)
                .commit();
    }
}
```

Figura 3.33: Código para la implementación de Fragment Manager

Fuente: (Revelo, 2016)

Posteriormente se realiza la implementación en el FirstMapActivity de la interfaz OnMapReadyCallback y sobrescribe su controlador onMapReady(), con lo cual el código debería quedar de la siguiente manera:

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;

public class FirstMapActivity extends AppCompatActivity
        implements OnMapReadyCallback {

    private FirstMapFragment mFirstMapFragment;

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

        mFirstMapFragment = FirstMapFragment.newInstance();
        getSupportFragmentManager()
                .beginTransaction()
                .add(R.id.map_container, mFirstMapFragment)
                .commit();

    }

    @Override
    public void onMapReady(GoogleMap googleMap) {

    }
}
```

Figura 3.34: Código para la implementación de métodos

Fuente: (Revelo, 2016)

Para que el mapa sea cargado se utiliza el método de asociación *getMapAsync()* el cual registra la actividad con *onCreate()*. Con esta acción *onMapReady()* proporciona la instancia del mapa cuando el servicio web de Google Maps solicite el contenido. (GoogleMap, 2017)

Todos los controladores reciben parámetros de la API.

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

    mFirstMapFragment = FirstMapFragment.newInstance();
    getSupportFragmentManager()
            .beginTransaction()
            .add(R.id.map_container, mFirstMapFragment)
            .commit();

    // Registrar escucha onMapReadyCallback
    mFirstMapFragment.getMapAsync(this);
}
```

Figura 3.35: Código para la implementación de métodos

Fuente: (Revelo, 2016)

3.1.10.7 Controles de interfaz de usuario sobre el mapa

Son aquellos controles que permiten manipular las funciones de acceso rápido, entre los cuales se tiene:

- **Control de zoom:** Se encuentran el parte inferior derecha sirven para acercar y alejar las imágenes del mapa.
- **Barra de tareas del mapa:** Botones que se encuentran la parte inferior derecha, su propósito es lanzar a la app de Google Maps.
- **Botón "Mi ubicación":** Botón que se encuentra ubicado en la parte superior derecha, el cual es activado con la capa "My Location", cuando es presionado da la ubicación actual con un punto azul en el mapa, indicando la posición.

Figura 3.36: Controles de interfaz de usuario

Fuente: (Revelo, 2016)

Métodos y controladores usados para las instancias *UiSettings* que llaman a cada uno de los métodos.

Controlador	Método
Zoom	`setZoomControlsEnabled(boolen)`
Botón "Mi ubicación"	`setMyLocationEnabled(boolean)`, `setMyLocationButtonEnabled(boolean)`
Barra de herramientas del mapa	`setMapToolbarEnabled(boolean)`

Figura 3.37: Métodos y controladoras de UiSettings

Fuente: (Revelo, 2016)

3.1.10.8 Añadir botón de ubicación

Para añadir la funcionalidad de mi ubicación a la aplicación se requieren permisos de la app, los cuales deben ser añadidos en la clase *AndroidManifest.xml.*

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

Figura 3.38: Código para implementación de botón de ubicación

Fuente: (Revelo, 2016)

3.1.10.9 Implementación de marcadores en Google Maps

Un marcador es un objeto interactivo el cual indica una ubicación dentro de la superficie del mapa, es utilizado para la representación de lugares, objetos y establecimientos. (Revelo, 2016).

Para añadir un marcador se usa el método *addMarker().*

```
public final Marker addMarker (MarkerOptions options)
```

Figura 3.39: Método addMarker de un marcador en Google

Fuente: (Revelo, 2016)

El método para la implementación de un marcador recibe un objeto *MarkerOptions*. El cual representa los marcadores según las propiedades que se le hayan sido especificadas, un marcador para su representación requiere de los siguientes parámetros:

- position()
- title()
- snippet()

Posteriormente se definen los marcadores con los métodos *addMarker()* en *onMapReady().*

```
@Override
public void onMapReady(GoogleMap googleMap) {
    // Coordenadas Japón
    LatLng latLng = new LatLng(36.2048, 138.2529);

    MarkerOptions markerOptions =
            new MarkerOptions()
                    .position(latLng)
                    .title("Japón")
                    .snippet("Primer ministro: Shinzō Abe");

    Marker marker = googleMap.addMarker(markerOptions);
}
```

Figura 3.40: Método onMapReady de un marcador en Google

Fuente: (Revelo, 2016)

3.1.11 Implementación de Google Directions API

Servicio de google que permite realizar el cálculo entre ubicaciones mediante solicitudes HTTP, la cual genera una solicitud string URL, con texto de coordenadas de latitud y longitud, junto con la clave API. (Google Developers, 2017)

Una solicitud enviada hacia la API de Google Directions debe contener la siguiente estructura:

```
https://maps.googleapis.com/maps/api/directions/outputFormat?parameters
```

Figura 3.41: Solicitud de URL enviada a la API

Fuente: (Google Developers, 2017)

En donde *outputFormat* puede contener los siguientes valores:

- Json
- Xml

3.1.11.1 Parámetros de solicitud

Para el correcto envió de las direcciones URL, los parámetros deberán ser enviados con el carácter (&), entre los parámetros que son obligatorios se tienen los siguientes:

- **Origin**: Contiene datos de la dirección, el valor de la latitud-longitud del sitio desde donde se quiere realizar los cálculos.

```
origin=41.43206,-81.38992
```

Figura 3.42: Valores de latitud-longitud

Fuente: (Google Developers, 2017)

- **Destination**: Contiene datos de la dirección, el valor de la latitud-longitud del sitio hacia donde se quiere realizar los cálculos.
- **Key**: Es la clave API de la aplicación que debe ser incluida en los parámetros enviados mediante la URL. (Google Developers, 2017)

3.1.11.2 Parámetros de solicitud opcionales

- **Mode**: Especifica el medio de transporte para proceder con el cálculo, para el desarrollo del proyecto se utilizó el modo
    - Walking: El cual se basa en traslados a pie por sendas peatonales y veredas.
- **Waypoints**: Especifica una matriz de puntos, los cuales especifican coordenadas de latitud y longitud, los waypoints construyen un trayecto permitiendo que se recorran las ubicaciones seleccionadas. (Google Developers, 2017)

3.1.11.3 Rutas

La API de Google Maps Directions, devuelve los resultados que son almacenados en una matriz (JSON) *routes*. La cual contiene el resultado del origen y del destino y forma una *legs* con los *waypoints* que se hayan especificado, para el trazado de una ruta se requiere de los siguientes datos:

- **Legs []**: matriz que contiene información entre dos ubicaciones, la cual se distingue por cada etapa de la ruta *steps*.
- **waypoint_order**: contiene la matriz con el orden de los *waypoitns* de la ruta calculada, para una ruta optimizada se los debe reordenar *optimize:true.*

```
"routes" : [
   {
      "bounds" : {
         "northeast" : {
            "lat" : 37.8079996,
            "lng" : -122.4074334
         },
         "southwest" : {
            "lat" : 37.7881005,
            "lng" : -122.4203553
         }
      },
      "copyrights" : "Map data ©2015 Google",
      "fare" : {
         "currency" : "USD",
         "value" : 6
         "text" : "$6.00"
      },
      ...
   }]
```


Figura 3.43: Ejemplo de construcción de una ruta

Fuente: (Google Developers, 2017)

3.1.12 Implementación de Street View API

La API de Google Street View permite obtener vistas panorámicas de 360 grados, mediante un conjunto de imágenes con proyección equirrectangular.

3.1.12.1 Agregar un panorama de Street View

Para la implementación de Street View en la aplicación se seguirá la siguiente estructura:

- Agregar un objeto *Fragment* a la clase *Activity* para la implementación del panorama Street View.
- Implementar la interfaz *OnStreetViewPanoramaReadyCallback* y usar el método de *callback* *onStreetViewPanoramaReady*(StreetViewPanorama), el cual permite administrar el objeto *StreetViewPanorama*.
- Llamar al método *getStreetViewPanoramaAsync*() en el fragmento para registrar el *callback*.

```
<fragment
    android:name="com.google.android.gms.maps.StreetViewPanoramaFragment"
    android:id="@+id/streetviewpanorama"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

Figura 3.44: Fragmento de archivo de diseño

Fuente: (Google Developers, 2016)

- Implementar la interfaz *OnStreetViewPanoramaReadyCallback* y configurar una instancia del *callback* en un objeto *StreetViewPanoramaFragment* o *StreetViewPanoramaView*.

```
public class MainActivity extends FragmentActivity
    implements OnStreetViewPanoramaReadyCallback {
...
}
```

Figura 3.45: Implementación de la clase OnStreetViewPanoramaReadyCallback

Fuente: (Google Developers, 2016)

- En el método *onCreate()* del *Activity* se debe realizar la vista del contenido.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    ...
}
```

Figura 3.46: Creación de la vista de contenido

Fuente: (Google Developers, 2016)

3.1.12.2 Ubicación del panorama

Se lo realiza mediante la clase *StreetViewPanorama.setPosition(),* la cual pasa objetos de tipo LatLng o radius dependiendo el área que se requiere.

```
private static final LatLng SAN_FRAN = new LatLng(37.765927, -122.449972);
mSvp.setPosition(SAN_FRAN);
```

Figura 3.47: Envió de datos para la creación del panorama

Fuente: (Google Developers, 2016)

3.1.13 Implementación de HockeyApp

Es una herramienta que nos permite distribuir la aplicación con la finalidad de que se realicen pruebas y de esta maneta obtener informes de fallos que se han producido durante su funcionamiento, es instalada en dispositivos de betatester quienes son los que evalúan la aplicación, HocheyApp envía datos de reportes de errores y caídas de la aplicación, cuando los desarrolladores suben actualizaciones el SDK reporta automáticamente que existen nuevas actualizaciones para la aplicación, si se ha realizado una correcta integración de la herramienta se puede obtener reportes detallados como los siguientes:

- Analíticas de uso: Porcentaje de uso de cada tester, versión del sistema operativo.
- Informes de bloqueo: Si la aplicación produce un bloqueo, falla en el almacenamiento del dispositivo. (GitHub , 2016)

El SDK de HockeyApp puede ser ejecutado en versiones del Sistema Operativo Android desde la versión 2.3 o posterior.

Para la integración de HockeyApp con nuestra aplicación se debe seguir el siguiente procedimiento:

1. **Obtener el SKD**: Agregar el SDK a las dependencias del módulo de aplicación en Android agregando la siguiente línea.

```
compile 'net.hockeyapp.android:HockeySDK:4.1.4'
```

Figura 3.48: Agregar el SDK HockeyApp

Fuente: (GitHub , 2016)

2. **Integración de HockeyApp**: Para lo cual en al archivo *build.gradle* se debe añadir la siguiente configuración.

```
manifestPlaceholders = [HOCKEYAPP_APP_ID: "$APP_ID"]
```

Figura 3.49: Integración de HockeyApp

Fuente: (GitHub , 2016)

En donde el parámetro $APP_ID debe ser reemplazado por el identificador de la aplicación HockeyApp, el cual se obtiene de la página de la aplicación, para la integración se debe guardar los cambios del archivo *build.gradle y* activar

la sincronización *Gradle*, dentro del archivo *AndroidManifest.xml* se debe agregar los metadatos para el *HockeySDK*.

```
<application>
        //your activity declarations and other stuff
        <meta-data android:name="net.hockeyapp.android.appIdentifier" android:value="${HOCKEYAPP_APP_ID}" />
</application>
```

Figura 3.50: Integración de HockeyApp

Fuente: (GitHub , 2016)

3. **Creación de informes de errores:** Agrega funciones para envios de informes de fallos, para lo cual se debe agregar el siguiente código en el *AndroidManifest.xml*

```
import net.hockeyapp.android.CrashManager;

public class YourActivity extends Activity {
  @Override
  public void onResume() {
    super.onResume();
    // ... your own onResume implementation
    checkForCrashes();
  }

  private void checkForCrashes() {
    CrashManager.register(this);
  }

}
```

Figura 3.51: Creación de informes de HockeyApp

Fuente: (GitHub , 2016)

4. **Envió de invitación**: Proceso que se realiza mediante envió de un correo electrónico, para la descarga de la aplicación en los dispositivos *betatester*.

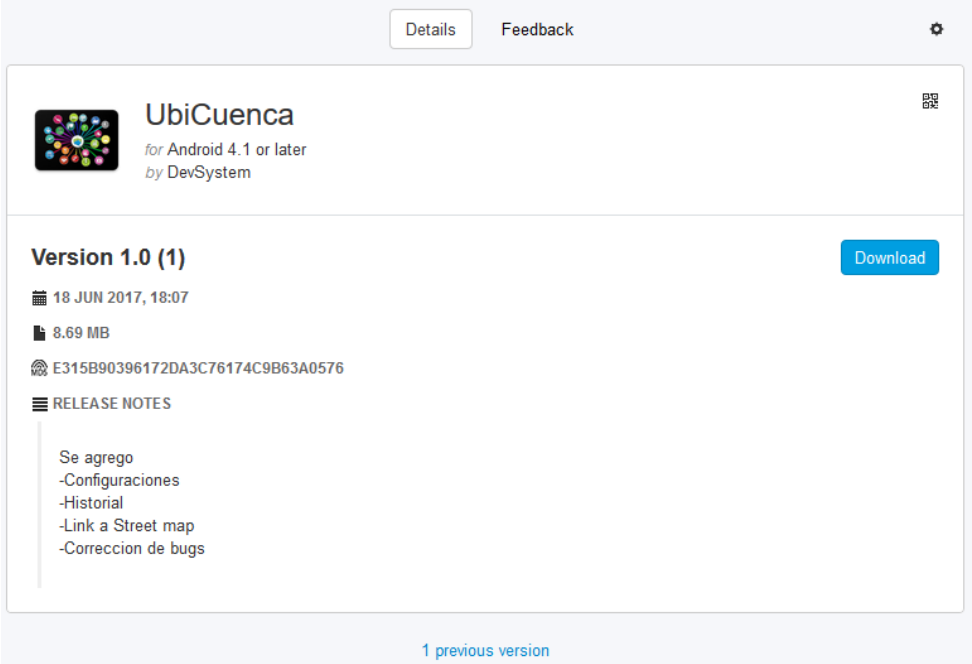


Figura 3.52: Invitación a betatester de la aplicación

Fuente: (Hockeyapp, 2017)

3.1.14 Realidad Aumentada

Para el desarrollo de la realidad aumentada se utilizó Wikitude SDK- JavaScript API, el cual es de pago, pero para el desarrollo ofrece una versión de prueba, con todas las funcionalidades, pero muestra en la pantalla marcas de agua con la palabra TRIAl, Wikitude, para su funcionamiento hace uso de la cámara, receptor GPS y la brújula del dispositivo móvil y requiere obtener los siguientes datos:

- Obtener la ubicación del dispositivo.
- Obtener la ubicación del punto de destino.
- Obtención de medidas en grados desde la ubicación actual, hacia el punto de destino, es por ello que se requiere de una correcta calibración de los sensores al iniciar la aplicación de realidad aumentada.

Para la obtención de la ubicación actual se lo realiza mediante el evento *onLocationChanged(),* posterior a la calibración de los sensores, se procede a la implementación de la clase *CameraViewActivity*, la cual llama a *SurfaceHolder.Callback*, para lanzar una imagen mediante la cámara en la pantalla del dispositivo, la clase *SurfaceHolder.Callback* implementa tres métodos:

- *surfaceChanged()*
- *surfaceCreated()*
- *surfaceDestroyed()*

Implementación de puntos de interés (POI), es el método en el cual se muestran los locales mediante un punto interpuesto en la pantalla del dispositivo, para la obtención de este resultado se lo debe implementar con las clases que a continuación se describen. (Wikitude GmbH, 2016)

3.1.14.1 Implementación de la Clase GeoObject

Esta clase permite representar una localización o punto de coordenadas mediante los siguientes parámetros:

- Latitud
- Longitud
- Altitud

La clase *GeoObject* maneja dos parámetros *ya sea en ARObject.enabled=true o ARObject.enabled=false* y es la encargada de realizar los cálculos si se encuentra habilitada para mostrar los elementos en la pantalla del dispositivo, mediante la cámara y es controlado de la siguiente manera:

```
//GeoObject se mantiene deshabilitado por defecto
var geoObject1 = new AR.GeoObject(geoLocation1, {
  enabled : false
});
//Se habilita el objeto geoObject1 para mostrar los elemnentos en la pantalla
geoObject1.enabled = true;
// Cuando es habilitada, la altitud se incrementa en 1 metro
// Cuando es deshabilitada, la altitud disminuye en 1 metro.
var geoObject2 = new AR.GeoObject(geoLocation1);
geoObject2.onEnterFieldOfVision = function() {
  geoObject2.locations[0].altitude++;
};
geoObject2.onExitFieldOfVision = function() {
  geoObject2.locations[0].altitude--;
};

//Se envian las dos ubicaciones
var geoObject3 = new AR.GeoObject( [ geoLocation1, geoLocation2 ], null);

// Se crea un nuevo objeto que enviara los eventos cuando en el estado true
var geoObject4 = new AR.GeoObject(locations, {
  //funcion que se ejecuta cuando se habilta la camara
  onEnterFieldOfVision : function(){ ... },
  //funcion que se ejecuta cuando se deshabilta la camara
  onExitFieldOfVision : function(){ ... },
  drawables : { cam :
      [drawable1, drawable2] //valores son enviados a clase GeoObject
  }
});

//Creacion del objeto
new AR.GeoObject(locations,
  { ...
    drawables : { cam : drawable1}
  }
);
```

Figura 3.53: Ejemplo de implementación de la clase GeoObject

Fuente: (Wikitude GmbH, 2016)

3.1.14.2 Creación de un marcador

Para la creación de un marcador se toman los datos que envía la clase *GeoObject* a la clase *GeoLocations*.

Para la representación de un marcador es necesario obtener la ubicación actual mediante la clase *context.onLocationChanged*, una vez establecida la ubicación se la dibujan los marcadores, que se encuentren dentro de rango establecido, se lo realiza mediante la clase *ImageDrawable*

La clase *ImageDrawable* toma la imagen que será establecida como marcador para mostrar en la pantalla, lo cual se la debe realizar en *ImageResource*, colocando el link en donde se encuentra almacenada la imagen.

```
markerDrawable_idle = new AR.ImageResource("assets/marker_idle.png");
```

Figura 3.54: Selección de la imagen para un marcador

Fuente: (Wikitude GmbH, 2016)

3.1.14.3 Implementación de la función LocationChanged

La función *onLocationChanged* comprueba si *initiallyLoadedData* ha sido llamada, en caso de ser correcta, se envían los datos de la posición a la función *loadPoisFromJsonData*, para que la API de Wikitude interprete los datos y dibuje en la pantalla los marcadores.

```
locationChanged: function locationChangedFn(lat, lon, alt, acc) {
    // request data if not already present
    if (!World.initiallyLoadedData) {
        var poiData = {
            "id": 1,
            "longitude": (lon + (Math.random() / 5 - 0.1)),
            "latitude": (lat + (Math.random() / 5 - 0.1)),
            "altitude": 100.0
        };
        World.loadPoisFromJsonData(poiData);
        World.initiallyLoadedData = true;
    }
}
```

Figura 3.55: Implementación de la función LocationChanged

Fuente: (Wikitude GmbH, 2016)

3.1.14.4 Creación de multiples marcadores

Para la implementación de varios marcadores se requiere que los datos sean almacenados en una arreglo de tipo POI, estos son enviados a la función *loadPoisFromJsonData*, la cual se encarga de crear los objetos dentro de un bucle, para cada objeto que se encuentra en la matriz se generan objetos de tipo *singlePo*i, para la creación de varios objetos se deben definir objetos de tipo *poiData*, estos finalmente se los deben almacenar en un array *markerList.*

```
// called to inject new POI data
loadPoisFromJsonData: function loadPoisFromJsonDataFn(poiData) {

    // empty list of visible markers
    World.markerList = [];

    // start loading marker assets
    World.markerDrawable_idle = new AR.ImageResource("assets/marker_idle.png");

    // loop through POI-information and create an AR.GeoObject (=Marker) per POI
    for (var currentPlaceNr = 0; currentPlaceNr < poiData.length; currentPlaceNr++) {
        var singlePoi = {
            "id": poiData[currentPlaceNr].id,
            "latitude": parseFloat(poiData[currentPlaceNr].latitude),
            "longitude": parseFloat(poiData[currentPlaceNr].longitude),
            "altitude": parseFloat(poiData[currentPlaceNr].altitude),
            "title": poiData[currentPlaceNr].name,
            "description": poiData[currentPlaceNr].description
        };

        World.markerList.push(new Marker(singlePoi));
    }

    World.updateStatusMessage(currentPlaceNr + ' places loaded');
}
```

Figura 3.56: Implementación de array de marcadores

Fuente: (Wikitude GmbH, 2016)

3.2 Fase de Análisis de la aplicación

Se definirán los requisitos y la arquitectura para la funcionalidad de la aplicación, es por ello que se requiere de un catálogo de requisitos tomando en cuenta los aspectos que se desea cubrir con la aplicación que se implementara.

3.2.1 Requerimientos funcionales

El proyecto denominado “Guía turística de la ciudad de Cuenca para dispositivos móviles Android”, permitirá al turista recorrer la cuidad conociendo sus atractivos turísticos, luego de haber hecho un análisis de aplicaciones de uso similar se ha determinado que la aplicación contenga los siguientes requisitos:

- Registro de Usuario
- Detalles de tiendas
    - Información
    - Dirección
    - Contactos
- Mapa
- Categorías
    - Parques
    - Hoteles
    - Restaurantes
    - Heladerías
    - Artesanías
    - Bares
    - Discotecas
    - Universidades
    - Iglesias
- Configuraciones
- Realidad Aumentada
- Registro de local
- Acerca de
- Ayuda

3.2.2 Requerimientos no funcionales

Son aquellos requisitos que complementan al proyecto, es decir, son atributos de calidad y definen el comportamiento de la aplicación.

3.2.2.1 Seguridad

Para el acceso a la aplicación se verifica que el usuario se haya registrado, ya que al momento de su registro se genera un token de seguridad[35] único, el cual garantiza la seguridad para el inicio se sesión.

3.2.2.2 Disponibilidad

En la primera instalación de la aplicación se necesitará de una conexión a internet para la descarga de datos necesarios. Los mismos serán almacenados en el dispositivo móvil para garantizar el funcionamiento de la aplicación permitiendo asi

[35]**TOKEN DE SEGURIDAD:** Contraseña de seguridad única, que se genera al momento del registro, la cual puede ser descifrada solo por la aplicación que la creo. (Calle & Largo, 2017)

usarla en modo offline, dependiendo de la funcionalidad a requerir por el usuario. La aplicación se encontrará disponible para su descarga e instalación en Google Play Store.

3.2.2.3 Mantenibilidad

La aplicación por ser un proyecto de tesis se la establecerá como un prototipo es por ello que solo cubrirá los requerimientos planteados como objetivos, en caso de ser de interés público luego de haberla publicada en Google Play Store, será necesario realizar actualizaciones y cambios que cubran todas las necesidades de los usuarios.

3.2.2.4 Portabilidad

Al ser una aplicación móvil, esta será compatible con dispositivos con sistema operativo Android, que tengan instalado como mínimo la versión 4.4, con el SDK 19, debido a herramientas que se utilizan para el desarrollo y funcionamiento (Realidad Aumentada, Material Desing).

La parte web del proyecto será compatible con todos los navegadores, se requiere que los plugins se encuentren actualizados para un mejor funcionamiento de la web.

3.2.2.5 Operabilidad

Serán los usuarios dueños de locales turísticos quienes actualicen información completa de los locales, una vez que se hayan registrado en la aplicación.

3.2.3 Restricciones del aplicativo

Las pruebas de funcionamiento de la aplicación se lo realizó en un dispositivo móvil HTC One S, el cual posee una pantalla Super AMOLED qHD de 4.3 pulgadas, con un procesador dual-core a 1.5GHz, con la versión de Android 4.4, es por ello que se han establecido las siguientes restricciones para la correcta funcionalidad de la aplicación y garantizar un óptimo procesamiento de datos, imágenes, mapas y tecnologías usadas para el desarrollo.

- Procesador: dual-core a 1.5GHz.
- Almacenamiento: 4Gb.
- Tamaño de pantalla: 540 x 960 pixels, 4.3 pulgadas.
- Navegación GPS.
- Wifi o red de datos.
- Cámara (Realidad Aumentada).
- Acelerómetro.
- Giroscopio.

Para el funcionamiento de la aplicación en modo offline los datos serán almacenados localmente en la base de datos propia del dispositivo (Sqlite), para ser enviados cuando el dispositivo establezca una conexión a internet.

3.2.4 Arquitectura de la aplicación

La aplicación móvil para su funcionamiento necesita de un dispositivo Android con la App instalada, con conexión a internet para la descarga de datos necesarios para su funcionamiento (en caso de ser instalada por primera vez). La aplicación realiza la conexión mediante un WebService para el intercambio de datos, en donde envía los parámetros requeridos al servidor, ya sea para solicitar o enviar información a la base de datos.

Adicionalmente el proyecto consta de una parte web en donde los usuarios pueden registrar datos de sus locales, este puede ser accedido desde un navegador web. El módulo web de igual manera realiza la conexión mediante un WebService para el procesamiento de datos.

3.2.4.1 Modelo Vista Controlador (MVC)

Arquitectura usada para aplicaciones software, la cual permite separar la lógica del negocio, de la interfaz del usuario, es por ello que el modelo MVC, se divide en tres modelos de abstracción:

- **Modelo:** Representa la lógica del negocio y es el que se encarga de forma directa de la integración de los datos y con las diversas clases (realizado por el desarrollador).
- **Vista:** Se encarga de mostrar información al usuario (conjunto de objetos de clases que conforman la interfaz gráfica).
- **Controlador:** Es el intermediario entre la vista y el modelo. Es el que controla las interacciones del usuario, ya que solicita datos al modelo y los entrega a la vista. (Pavon Mestras, 2010)

3.2.4.2 Flujo de control de MVC

1. El usuario realiza una acción en la interfaz.
2. El controlador procesa el evento de entrada.
3. El controlador notifica al modelo, la acción realizada por el usuario.
4. El proceso genera una vista nueva con los datos del modelo.
5. La interfaz de usuario pasa a un estado de espera de una nueva interacción. (Pavon Mestras, 2010)

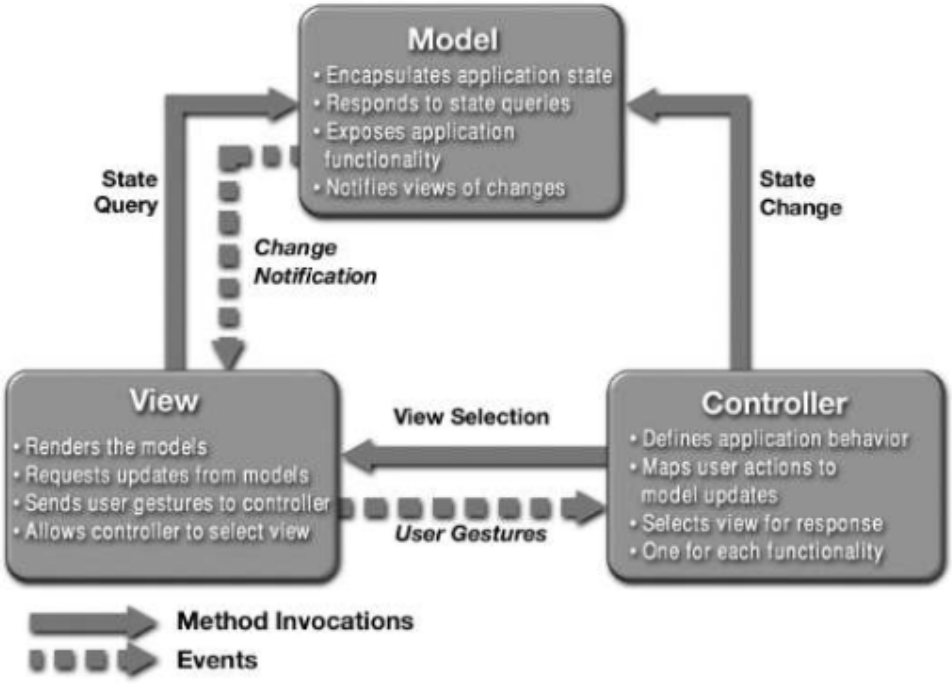


Figura 3.57: Ejemplo Modelo Vista Controlador (MVC)

Fuente: (Pavon Mestras, 2010)

3.2.5 Fase de diseño

Previo a la fase de implementación de la aplicación móvil se elaboran diseños, los cuales serán de uso intuitivo para el usuario, para poder determinar las características funcionales de la aplicación se utilizara el modelado UML[36] (Lenguaje Unificado de Modelado).

Entre los diagramas que se realizaran para el desarrollo del proyecto se tendrán los siguientes:

- Jerarquía de Usuarios.
- Diagrama de Caso de Uso.
- Diagrama de Secuencias.
- Diagrama de Navegación.

3.2.5.1 Jerarquía de Usuarios

Se detallan los usuarios finales que interactuaran con la aplicación

[36]**UML:** Conjunto de diagramas y estándares que sirven para el modelamiento del sistemas que son orientados a objetos, los cuales sirven para determinar la funcionalidad de la aplicación. (Booch, Jim , & Ivar , 2015)

Figura 3.58: Jerarquía de Usuarios

Fuente: (Calle & Largo, 2017)

3.2.5.2 Diagramas de Caso de Uso

Identificación de los usuarios

Usuarios de la aplicación	
Dueño de local	Usuario que sube información de los locales
Turista	Quien hace uso de la aplicación y consume los datos ingresados.

Tabla 1. Usuarios de la aplicación

3.2.5.2.1 Caso de Uso General de la aplicación

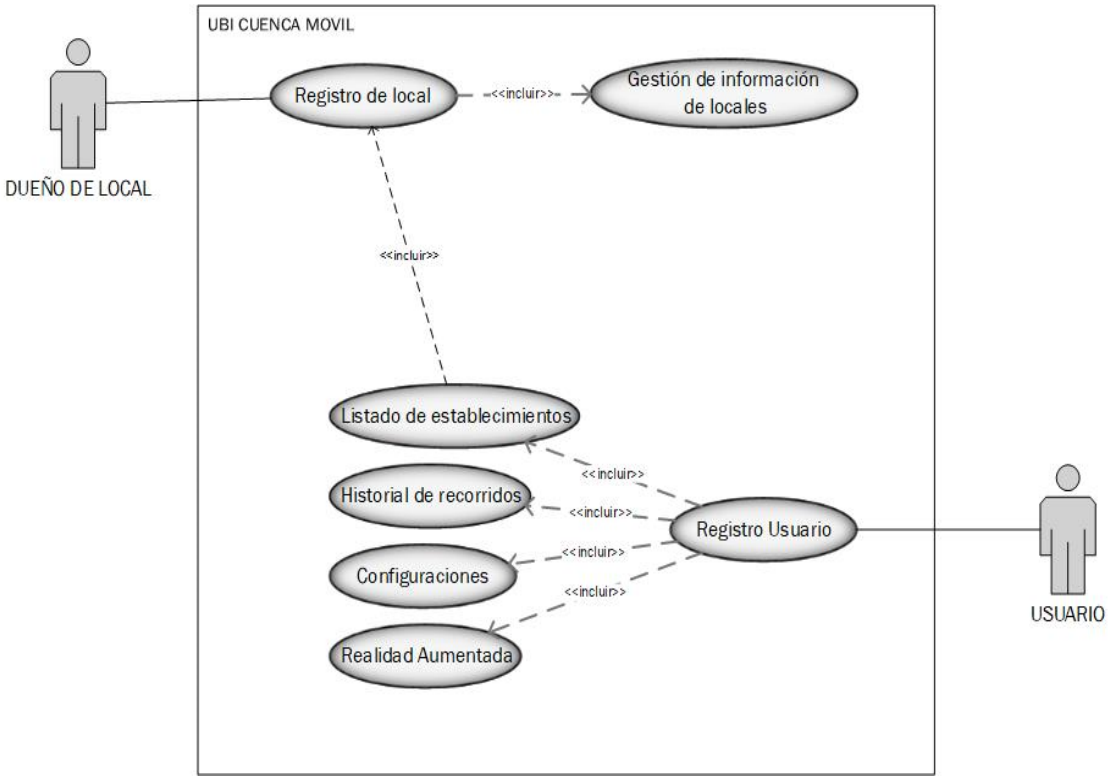


Figura 3.59: Caso de Uso General de la aplicación

Fuente: (Calle & Largo, 2017)

3.2.5.2.2 Especificación Caso de Uso General de la aplicación

CASO DE USO GENERAL DE LA APLICACION		
Nombre	Registro de Local	
Descripción	Registro de Local por parte de dueño	
Actores	Dueño de local	
Precondiciones	Aplicación ejecutada correctamente Conexión a internet Ser registrado como usuario	
Flujo del sistema	Paso	Acción
	1	Se requiere que se realice el proceso de registro de usuario
	2	Proceso de pago para poder ingresar datos de local
	3	Gestión de datos de local por parte de dueño
Post Condición	Se requiere del registro como usuario para poder acceder al registro de local	

Tabla 2. Especificación Caso de Uso Registro Local

3.2.5.2.3 Caso de Uso Turista

Este caso de uso detalla como el turista (usuario) interactúa con la aplicación en cada una de las opciones disponibles.

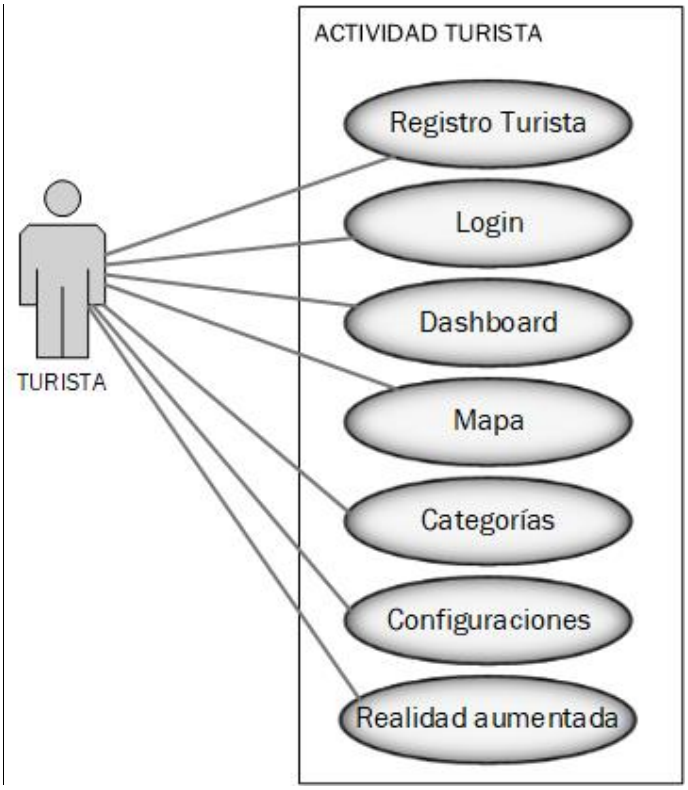


Figura 3.60: Caso de Uso Turista

Fuente: (Calle & Largo, 2017)

3.2.5.2.4 Especificación Caso de Uso Turista

CASO DE USO TURISTA		
Nombre	Registro Turista	
Descripción	Registro de Usuario a la aplicación	
Actores	Turista	
Precondiciones	Aplicación ejecutada correctamente Conexión a internet	
Flujo del sistema	Paso	Acción
	1 2	La aplicación solicita datos para realizar el proceso de registro. Aplicación valida si los datos ingresados son correctos y si el usuario no existe solicita creación de usuario
Post Condición	Si los datos no son correctamente validados no permite el registro del usuario.	

Tabla 3. Especificación Caso de Uso Registro Turista

3.2.5.2.5 Especificación Caso de Uso Login Turista

CASO DE USO LOGIN TURISTA		
Nombre	Login	
Descripción	Inicio de sesión de usuario en la aplicación	
Actores	Turista	
Precondiciones	Conexión a internet Usuario registrado Aplicación ejecutada correctamente	
Flujo del sistema	Paso	Acción
	1 2	La aplicación solicita datos para realizar proceso de registro Aplicación valida los datos ingresados por el usuario, sin son correctos permite iniciar sesión
Post Condición	Usuario debe estar registrado en la aplicación	

Tabla 4. Especificación Caso de Uso Login Turista

3.2.5.2.6 Caso de Uso Dueño de Local

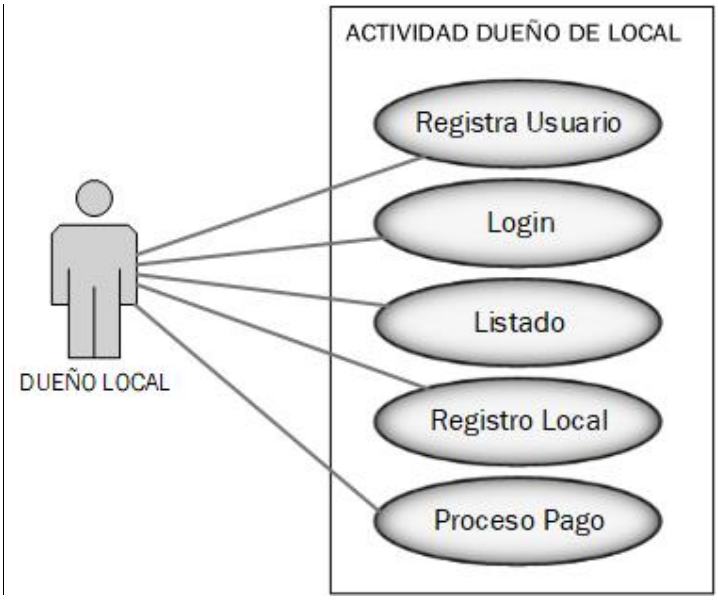


Figura 3.61: Caso de uso dueño de local

Fuente: (Calle & Largo, 2017)

3.2.5.2.7 Especificación Caso de Uso Proceso Pago

CASO DE USO PROCESO PAGO		
Nombre	Proceso pago	
Descripción	Realizar pago para que usuario pueda habilitar la opción de registrar locales en la aplicación, como rol dueño de local	
Actores	Dueño de local	
Precondiciones	Aplicación ejecutada correctamente Conexión a internet Registro como usuario	
Flujo del sistema	Paso	Acción
Post Condición	1	La aplicación valida que se encuentre registrado como como usuario
	2	Se procede con el proceso de pago
	3	El servicio Payphone valida los datos enviados y confirma el pago mediante una notificación.
	4	Se permite realizar el ingreso de locales

Tabla 5. Especificación Caso de Uso Proceso pago

3.2.5.2.8 Caso de Uso Actividad página web

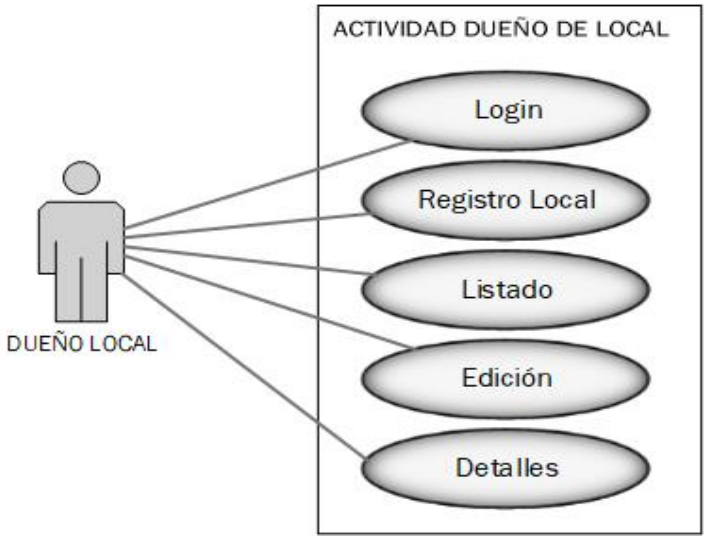


Figura 3.62: Caso de uso actividad página web

Fuente: (Calle & Largo, 2017)

3.2.5.2.9 Especificación Caso de Uso actividad página web

CASO DE USO ACTIVIDAD PAGINA WEB		
Nombre	Actividad página web	
Descripción	Permitir iniciar sesión en la página web para que los usuarios puedan registrar o editar información de sus locales.	
Actores	Dueño de local	
Precondiciones	Conexión a internet Registro como usuario dueño de local en la aplicación móvil Proceso de login correcto	
Flujo del sistema	Paso	Acción
	1 2 3	Se realiza validación de datos para determinar si son de usuario registrado como dueño de local Proceso de login exitoso Registro o edición de datos de un local
Post Condición	Datos validados para controlar que se encuentre registrado como usuario.	

Tabla 6. Especificación Caso de Uso actividad página web

3.2.5.3 Diagrama de Clases

El diseño se lo realizará de manera unificada es decir, se utilizara la misma base de datos para la parte web como para el aplicativo.

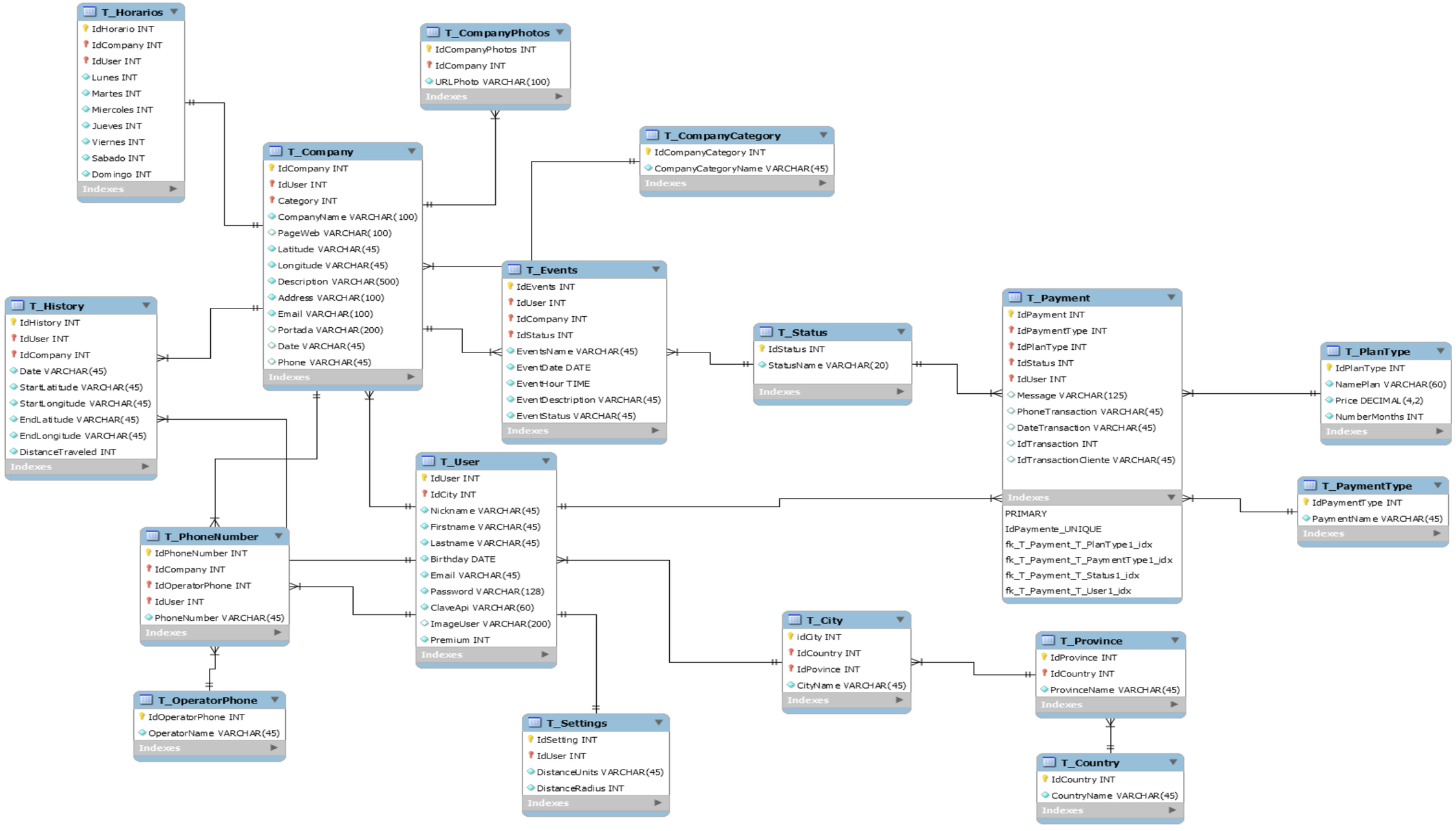


Figura 3.63: Diagrama de clases

Fuente: (Calle & Largo, 2017)

3.2.5.4 Diagrama de Secuencias

3.2.5.4.1 Diagrama de secuencia registró turista

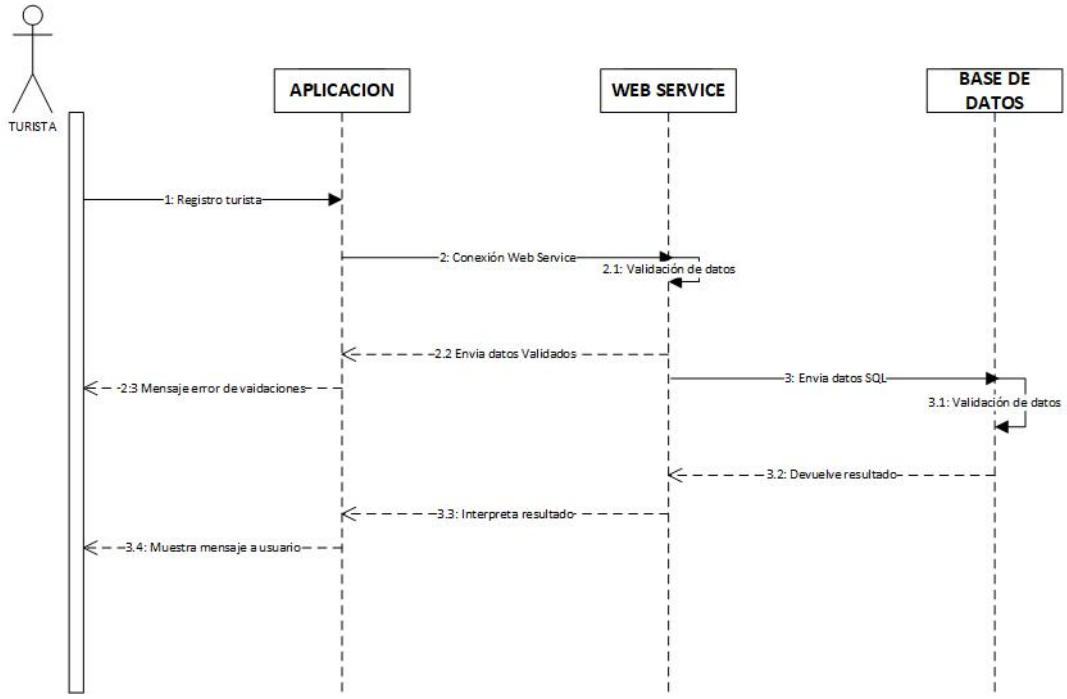


Figura 3.64: Diagrama de secuencia registró turista

Fuente: (Calle & Largo, 2017)

3.2.5.5 Diagrama de Navegación

Este diagrama nos indica el orden de interacción que el usuario tendrá con cada una de las pantallas de la aplicación.

3.2.5.5.1 Diagrama de navegación de la aplicación

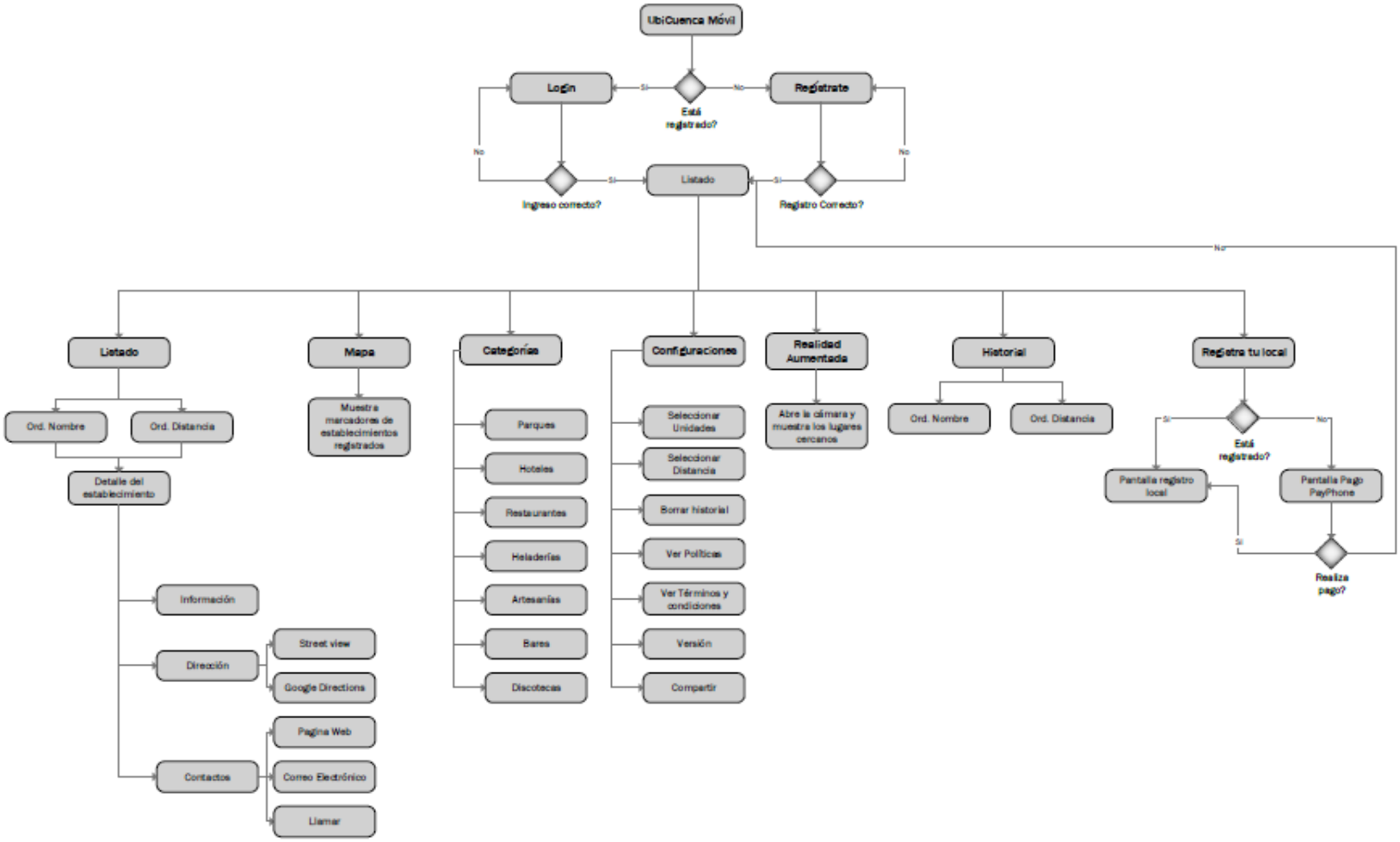


Figura 3.65: Diagrama de navegación de la aplicación

Fuente: (Calle & Largo, 2017*)*

3.2.5.5.2 Diagrama de navegación de página web

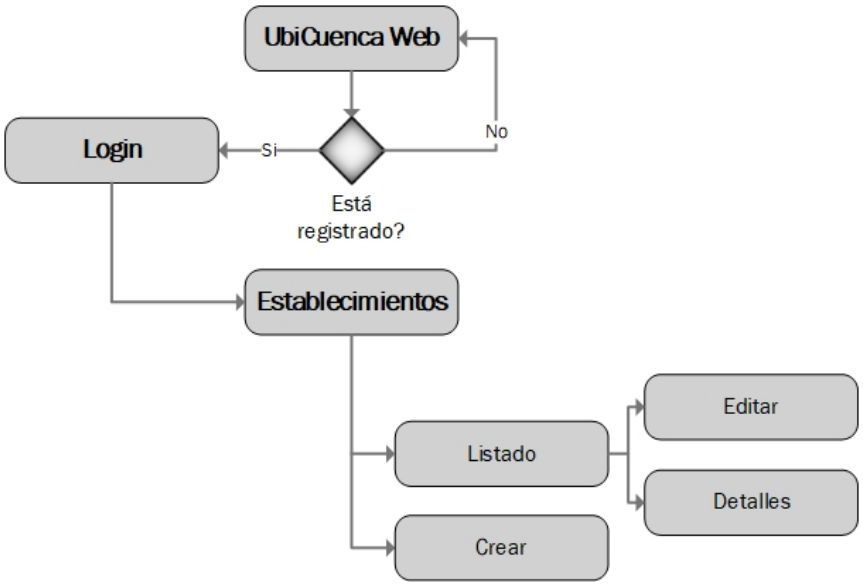


Figura 3.66: Diagrama de navegación de página web

Fuente: (Calle & Largo, 2017)

3.2.5.6 Diseño de los prototipos de las pantallas de la aplicación

Se establecerá un diseño preliminar de la aplicación, con la herramienta Ninja Mock, la cual ya se explicó con anterioridad.

3.2.5.6.1 Diseño de la pantalla de inicio de sesión

Es la primera pantalla que visualizará el usuario, en donde se tiene la opción de ingresar los datos en caso de ser un usuario registrado o a su vez existe un link la cual abrirá la pantalla para realizar el proceso de registro del usuario.

Figura 3.67: Diseño de la pantalla de inicio de sesión

Fuente: (Calle & Largo, 2017)

3.2.5.6.2 Diseño de la pantalla para el registro del usuario

Pantalla que permitirá al usuario registrarse en la aplicación, esta solicitara los datos requeridos para almacenarlos en la base de datos.

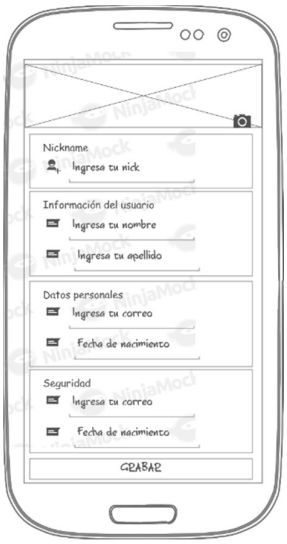


Figura 3.68: Diseño de la pantalla para registro de usuario

Fuente: (Calle & Largo, 2017)

3.2.5.6.3 Diseño de la pantalla principal de la aplicación

Pantalla que será la principal luego del proceso de inicio de sesión. Mostrará un listado con la información de interés del usuario.

Figura 3.69: Diseño de la pantalla principal de la aplicación

Fuente: (Calle & Largo, 2017)

3.2.5.6.4 Diseño del menú de la aplicación

Diseño que permitirá al usuario interactuar con la aplicación ya que tiene todas las opciones disponibles.

Figura 3.70: Diseño de la pantalla del menú de la aplicación

Fuente: (Calle & Largo, 2017)

3.2.5.6.5 Diseño del menú para ingreso de local

Pantalla que permitirá al usuario realizar el registro de los locales.

Figura 3.71: Diseño de la pantalla del menú de la aplicación

Fuente: (Calle & Largo, 2017)

3.2.5.6.6 Diseño del prototipo para la aplicación web

Se establecerá un diseño preliminar de la aplicación web, la cual será usada para el registro de locales por parte de los dueños de los establecimientos.

Para el desarrollo del aplicativo web se utilizaran las siguientes herramientas, que permitirán dar la funcionalidad requerida:

- Programación: Desarrollado en C#.
- Diseño: Se desarrolló basado en un framework, el cual es Bootstrap, el mismo que está basado en HTML y CSS, el uso de este framework facilita la creación de la interfaz de la página web de manera que sea una web responsive.

  Para el correcto funcionamiento de los plugins JavaScript de Bootstrap se requieren de las librerías de JQuery, por lo cual se debe incluir en las plantillas. (Robledano, 2015)

```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
│   ├── bootstrap-theme.min.css
│   └── bootstrap-theme.min.css.map
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
└── fonts/
    ├── glyphicons-halflings-regular.eot
    ├── glyphicons-halflings-regular.svg
    ├── glyphicons-halflings-regular.ttf
    ├── glyphicons-halflings-regular.woff
    └── glyphicons-halflings-regular.woff2
```

Figura 3.72: Estructura de archivos de Bootstrap

Fuente: (Robledano, 2015)

3.2.5.6.7 Diseño para la pantalla de inicio de sesión en la página web

Pantalla que permite a un usuario ingresar sus datos de inicio de sesión, previos a un registro y suscrito a un plan de pagos.

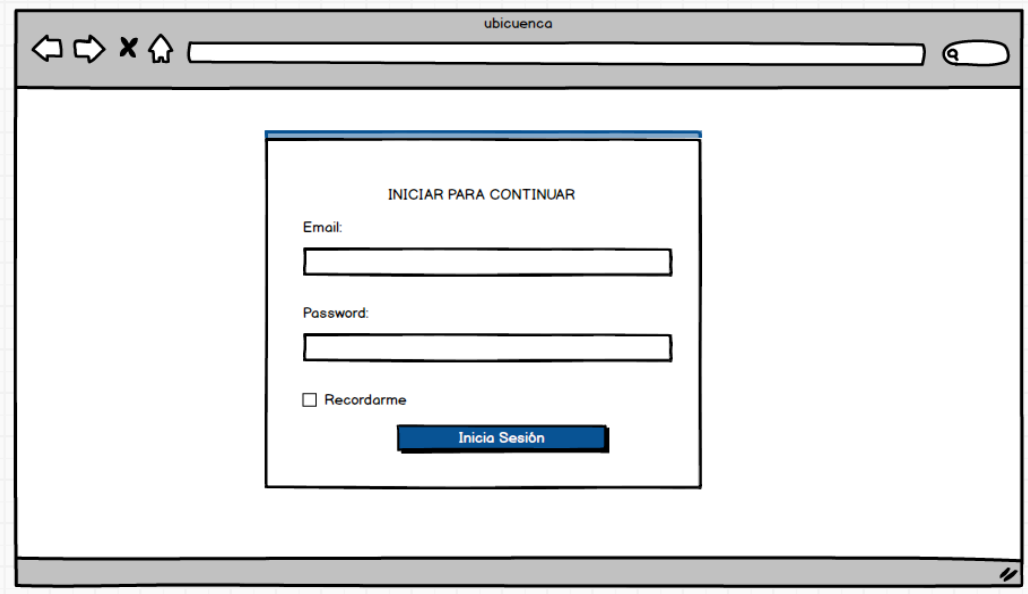


Figura 3.73: Diseño de Inicio de Sesión página web

Fuente: (Calle & Largo, 2017)

3.2.5.6.8 Diseño para la administración de datos de los locales desde la web

Pantalla que permite al usuario realizar un listado de la información que tiene almacenada, así como también la opción de crear nuevos locales.

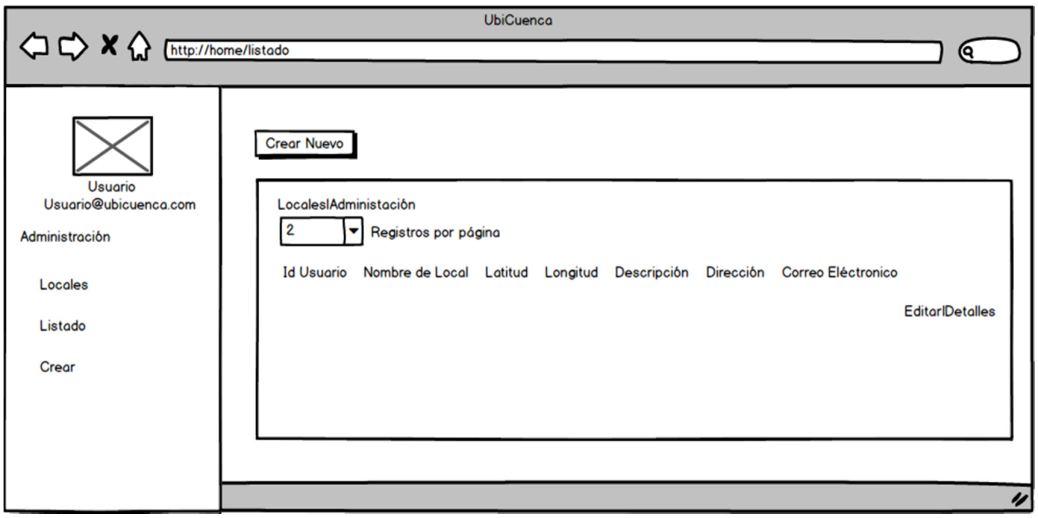


Figura 3.74: Administración de datos de locales

Fuente: (Calle & Largo, 2017)

CAPÍTULO 4 RESULTADOS

El desarrollo de la aplicación propuesta en el presente proyecto permitirá que los turistas hagan uso de la aplicación desde un dispositivo móvil con sistema operativo Android, debido a que es el sistema con mayor tendencia en nuestro medio, en un estudio realizado por nuestra autoría a 400 personas en la ciudad de Cuenca en el centro histórico se obtuvo que 272 personas usan Android, seguidas por 96 personas con el sistema operativo IOS y 32 con el sistema operativo Windows Phone. Es por ello que el aplicativo se desarrolló únicamente para dispositivos con plataforma Android (ver Anexo 1).

Se realizó un estudio sobre la implementación de un plan de pagos para que los dueños de los locales puedan realizar los registros, obteniendo como resultado la siguiente información:

Los dueños de los locales que si estarían dispuestos a pagar, supieron manifestar que no lo harían por razones tales como:

- Se da a conocer los servicios de local.
- Mayor acogida de los turistas.

Los dueños de los locales que no estarían dispuestos a pagar, supieron manifestar que no lo harían por razones tales como:

- No se conoce el tipo de aplicación, primero se tendría que realizar una prueba del aplicativo.
- Se maneja publicidad mediante otros medios.

Para determinar los requisitos de la aplicación, de igual manera se realizó un estudio centrado a dueños de establecimientos y usuarios en general dentro del centro histórico, teniendo como resultados (ver Anexo 1, Pregunta 7) información relevante sobre aspectos que debería tener la aplicación tales como:

- Que se manejen por categorías los locales comerciales
- Ubicación exacta del local en el mapa
- Que el turista tenga la opción de poder llegar al destino mediante una ruta que es generada por la aplicación.
- Que sea un aplicación innovadora en la ciudad
- Mostrar detalles del local
- Poder incluir varias fotos del local

- Mostrar horarios de atención, información, servicios y contactos
- Que la aplicación sea gratuita para el turista
- De fácil uso para el turista
- Que el turista pueda determinar un rango de búsqueda de locales comerciales
- El turista pueda ver varias fotos del local

4.1 Resultados del desarrollo de la aplicación

Se mostrara los resultados obtenidos luego de la etapa de desarrollo, indicando cada una de las pantallas y procesos que dan la funcionalidad a la aplicación para que pueda ser utilizada por el usuario final.

4.1.1 Registro de usuario: Pantalla que permite a un usuario nuevo realizar el registro de sus datos y poder iniciar sesión en el aplicativo, para lo cual se debe seguir el siguiente proceso:

4.1.1.1 Registro de imagen para perfil de usuario: Permite al usuario, tomar una foto o elegir una de la galería, esta imagen será la que se visualice en el perfil del usuario.

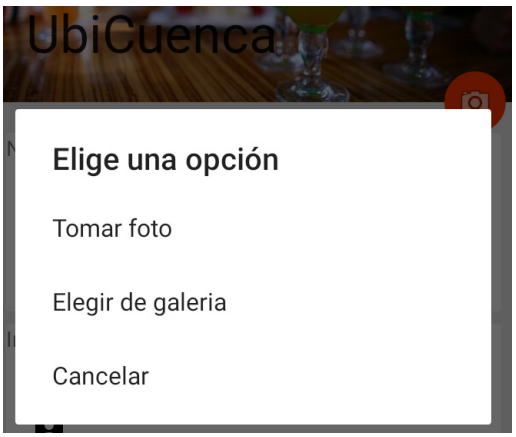


Figura 4.1: Registro de imagen de usuario

Fuente: (Calle & Largo, 2017)

4.1.1.2 Registro de información de usuario: Pantalla que permite al usuario ingresar la información requerida por la aplicación para su almacenamiento y registro como usuario y posterior inicio de sesión en el aplicativo, para lo cual se debe seguir el siguiente proceso:

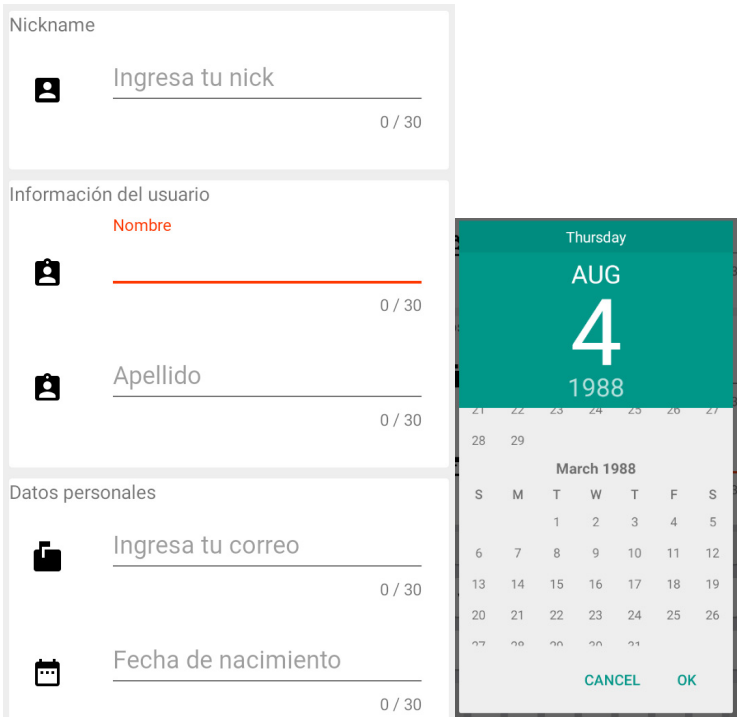


Figura 4.2: Proceso de registro de usuario

Fuente: (Calle & Largo, 2017)

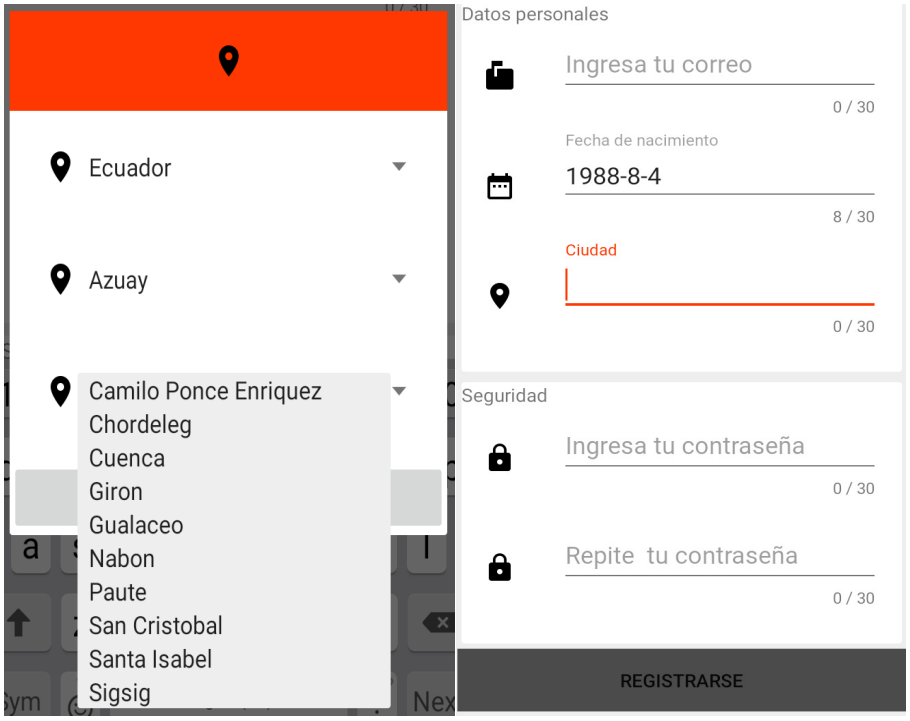


Figura 4.3: Selección de Cuidad

Fuente: (Calle & Largo, 2017)

4.1.2 Inicio de sesión: Pantalla que solicita los datos del usuario, para iniciar sesión en la aplicación.

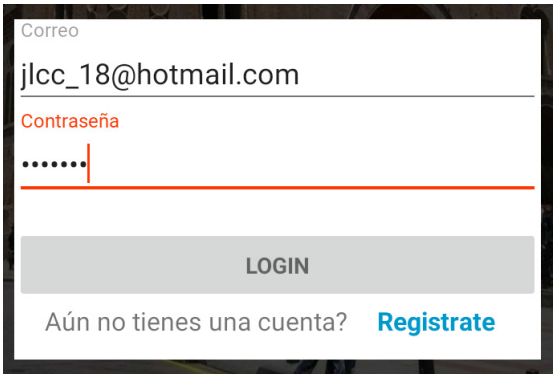


Figura 4.4: Pantalla inicio de sesión

Fuente: (Calle & Largo, 2017)

4.1.3 Pantalla principal de la aplicación: Contiene todas las opciones disponibles de la aplicación para la interacción con el usuario.

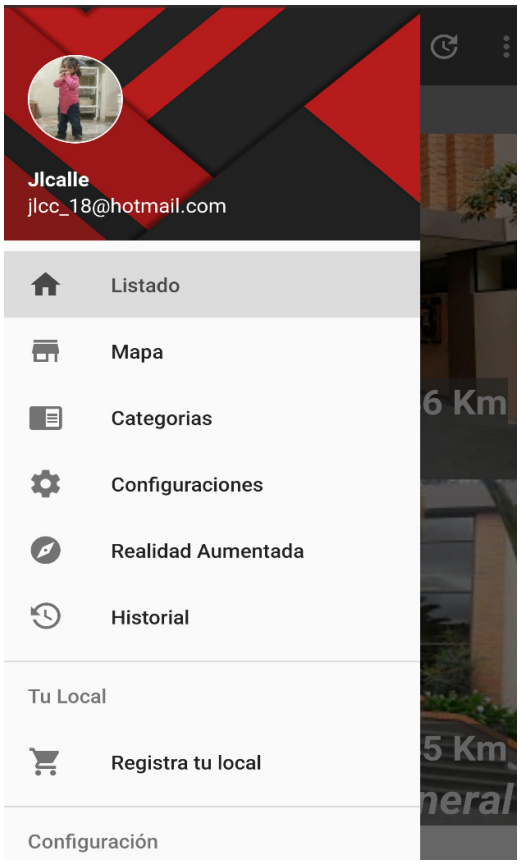


Figura 4.5: Pantalla principal de aplicación

Fuente: (Calle & Largo, 2017)

4.1.4 Listado de locales: Muestra los locales de preferencia, permite al usuario, ordenar el listado ya sea por nombre o distancia.

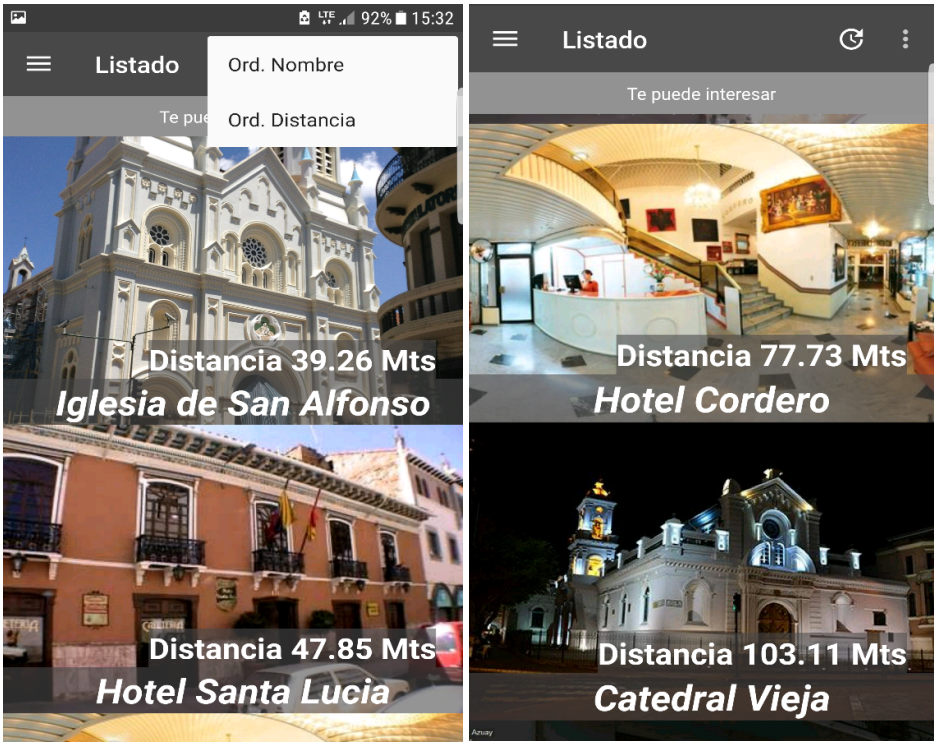


Figura 4.6: Pantalla de listado de locales

Fuente: (Calle & Largo, 2017)

La pantalla listado de locales permite ver información detallada de cada local la cual se refiere a:

4.1.4.1 Información: Pantalla que muestra descripción detallada del local.

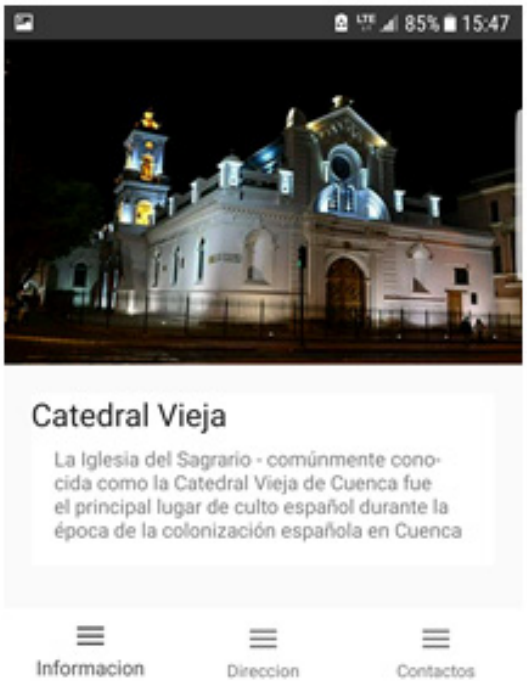


Figura 4.7: Pantalla de información de local

Fuente: (Calle & Largo, 2017)

4.1.4.1.1 Galería Fotográfica: Permite ver las imágenes almacenadas del local seleccionado del listado.

Figura 4.8: Galería Fotográfica

Fuente: (Calle & Largo, 2017)

4.1.4.2 Dirección: Pantalla que muestra la ubicación del local mediante un marcador en el mapa.

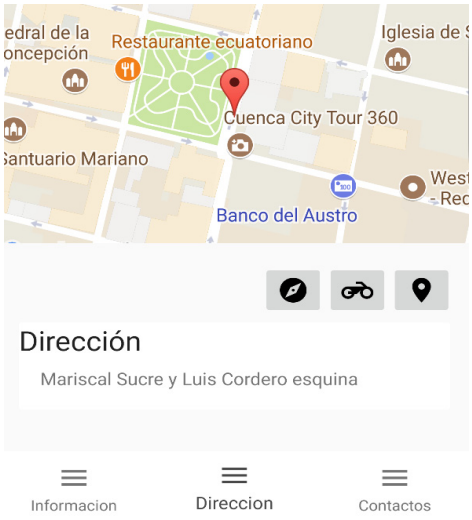


Figura 4.9: Pantalla de dirección de local

Fuente: (Calle & Largo, 2017)

4.1.4.2.1 Visualización de Punto en Realidad Aumentada: Pantalla que se habilita desde el icono , el cual permite visualizar únicamente el local seleccionado del listado mediante la realidad virtual.

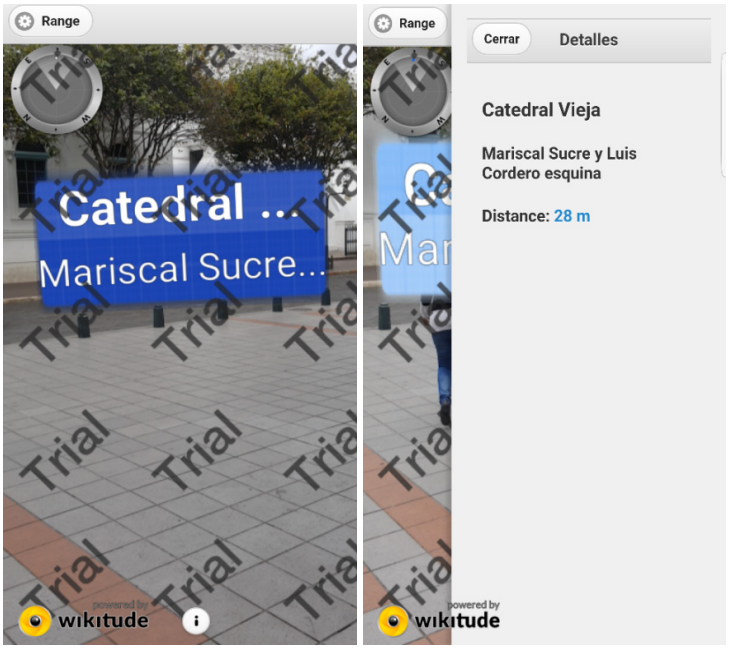


Figura 4.10: Visualización mediante realidad aumentada

Fuente: (Calle & Largo, 2017)

4.1.4.2.2 Google Directions: Pantalla que se habilita desde el icono el cual traza la ruta que el usuario debe tomar para llegar hacia el local de destino.

Figura 4.11: Pantalla que muestra el trazado de la ruta

Fuente: (Calle & Largo, 2017)

4.1.4.2.2.1 Navegación: Botón sobre al mapa de la ruta, el cual nos permite realizar la navegación asistida por el GPS.

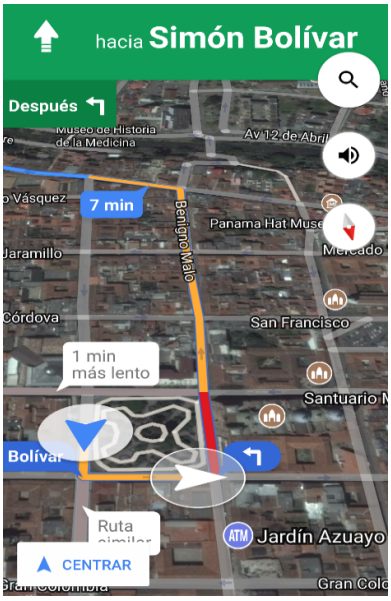


Figura 4.12: Navegación asistida por GPS

Fuente: (Calle & Largo, 2017)

4.1.4.2.3 Google Street: Pantalla que se habilita desde el icono la cual permite visualizar el local mediante Google Street.

Figura 4.13: Pantalla de dirección de local

Fuente: (Calle & Largo, 2017)

4.1.4.3 Contactos*:* Pantalla que muestra la información de contacto del local seleccionado del listado.

Figura 4.14: Información de contacto de un local

Fuente: (Calle & Largo, 2017)

4.1.5 Mapa: Pantalla que muestra mediante marcadores los locales registrados en la aplicación, según los parámetros de configuración de la distancia y radio establecidos.

Figura 4.15: Pantalla de marcadores de los locales en el mapa

Fuente: (Calle & Largo, 2017)

4.1.6 Categorías: Pantalla que muestra los locales registrados en la aplicación, dentro de cada una de las categorías establecidas:

- Parques
- Hoteles
- Restaurantes
- Heladerías
- Artesanías
- Bares
- Discotecas
- Universidades
- Iglesias

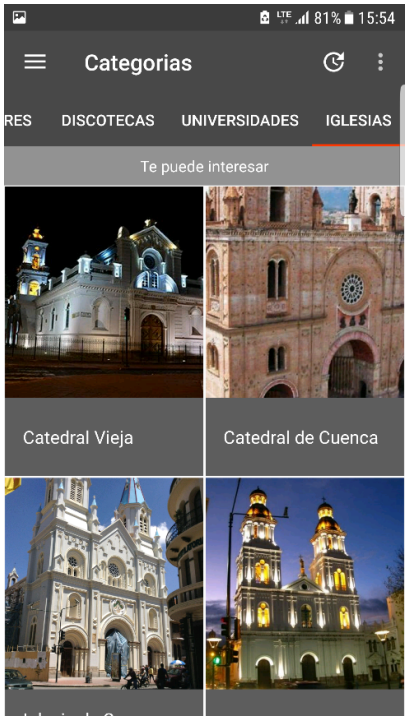


Figura 4.16: Pantalla de categorías de los locales

Fuente: (Calle & Largo, 2017)

4.1.7 Configuraciones: Pantalla que permite determinar al usuario unidades de medida y distancia para la búsqueda de locales, así como también borrar el historial de navegación de la aplicación, conocer los términos y condiciones de la aplicación, su versión y permite compartir el app a otros usuarios.

Figura 4.17: Pantalla para configuraciones de la aplicación

Fuente: (Calle & Largo, 2017)

4.1.8 Realidad Aumentada: Pantalla que mediante la cámara del dispositivo muestra los locales registrados, el usuario deberá dar los permisos para el acceso a la ubicación del dispositivo.

En las imágenes se puede ver los puntos que se encuentran registrados, y según la posición del usuario, se visualizan los marcadores, se puede seleccionar el marcador y este mostrara una descripción del punto de interés.

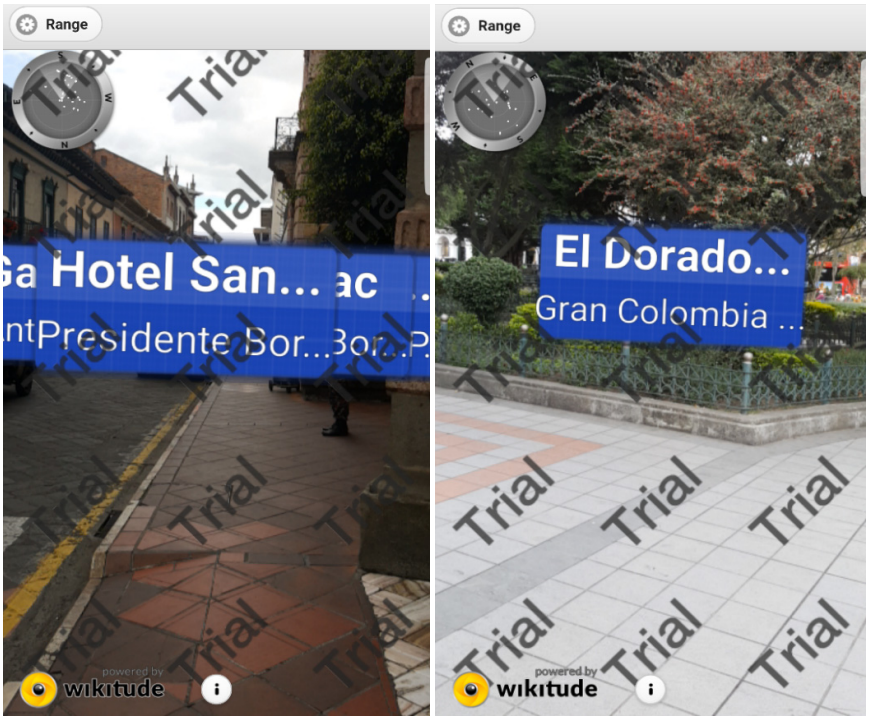


Figura 4.18: Puntos de interés mediante realidad aumentada

Fuente: (Calle & Largo, 2017)

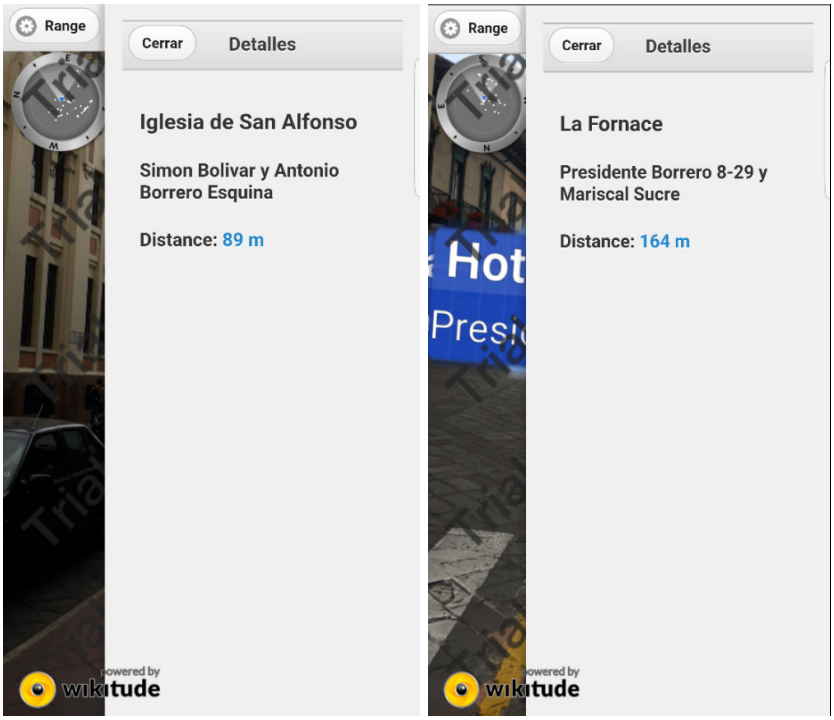


Figura 4.19: Descripción de puntos de interés mediante realidad aumentada

Fuente: (Calle & Largo, 2017)

4.1.9 Historial: Pantalla que muestra los lugares o locales que el usuario ha visitado.

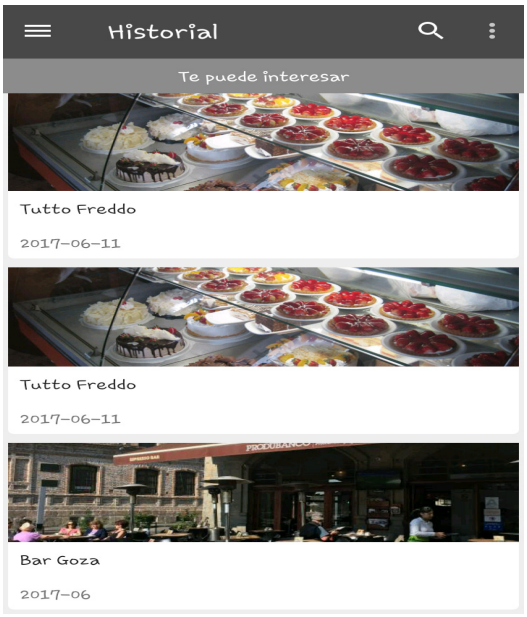


Figura 4.20: Pantalla que registra el historial de visitas del usuario

Fuente: (Calle & Largo, 2017)

4.1.10 Registro de local: Pantallas que permiten al usuario realizar el registro de la información de un local en el aplicativo.

Para el proceso de registro de la información de un local se debe seguir el siguiente orden para su correcto funcionamiento:

4.1.10.1 Selección de un plan de pagos: Se tiene que realizar un registro como un usuario dueño de local, esta opción se habilita luego de que el pago haya sido validado correctamente, para lo cual se deben seguir los siguientes pasos:

1. Selección de un plan de pagos, de las opciones disponibles

   - Plan de un año

   - Plan de 6 meses

   - Plan de un mes

Figura 4.21: Pantalla que permite la selección de un plan de pago

Fuente: (Calle & Largo, 2017)

2. Ingresamos el número de teléfono con el que se realizó la validación de la tarjeta en el sistema de payphone para que el pago pueda ser procesado.

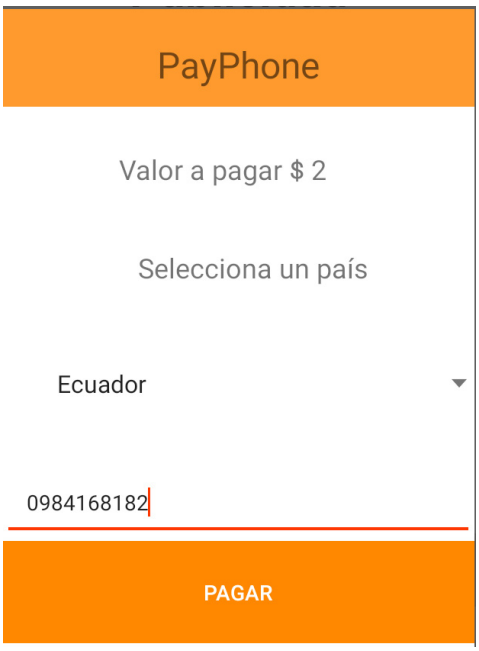


Figura 4.22: Ingreso de numero para proceso de pago

Fuente: (Calle & Largo, 2017)

3. El promotor deberá validar los datos y aceptar el pago

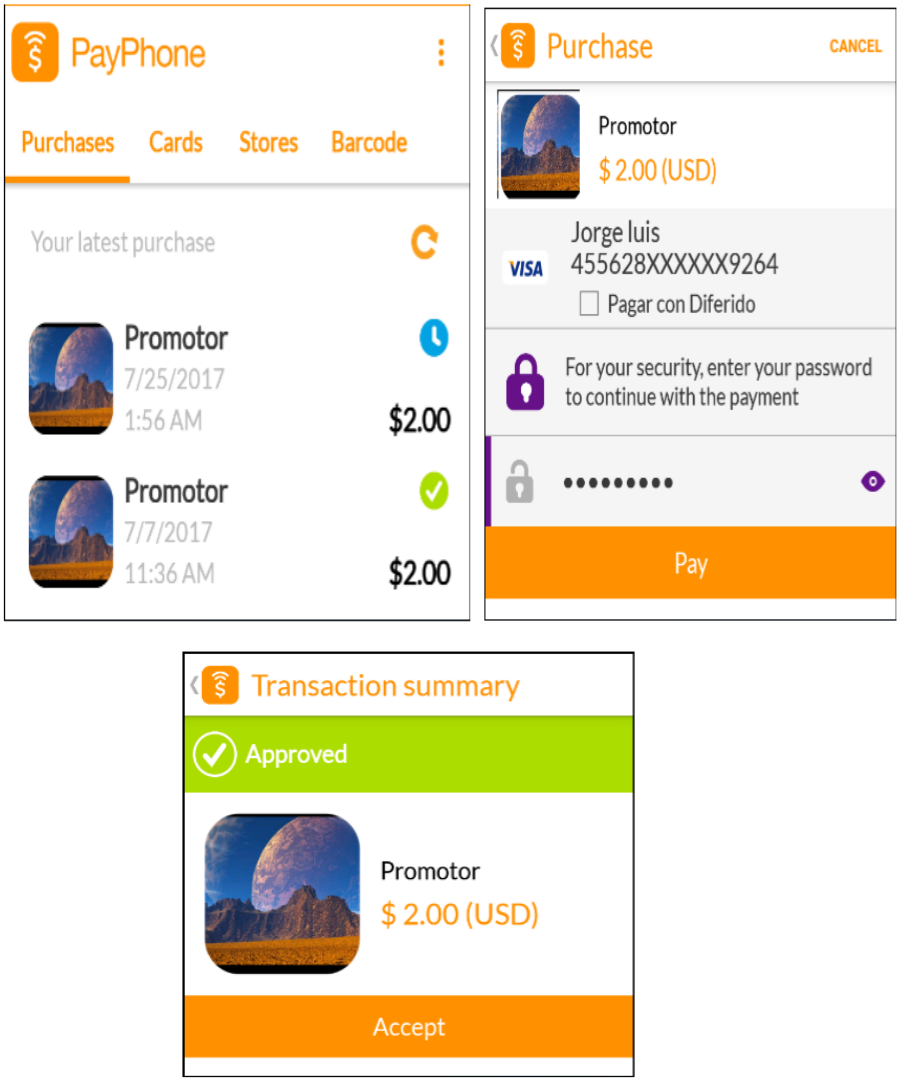


Figura 4.23: Proceso de validación y aprobación de pago por parte de promotor

Fuente: (Calle & Largo, 2017)

Posterior a la validación y aprobación del pago se podrá realizar el registro de los locales por parte de los dueños.

4.1.10.2 Registro de imágenes de local: El usuario tiene la opción de almacenar cinco fotos por cada establecimiento registrado, estas pueden ser tomadas directamente desde la cámara o a su vez seleccionadas desde el almacenamiento del dispositivo, para este proceso se requirió haber dado los permisos correspondientes a la aplicación.

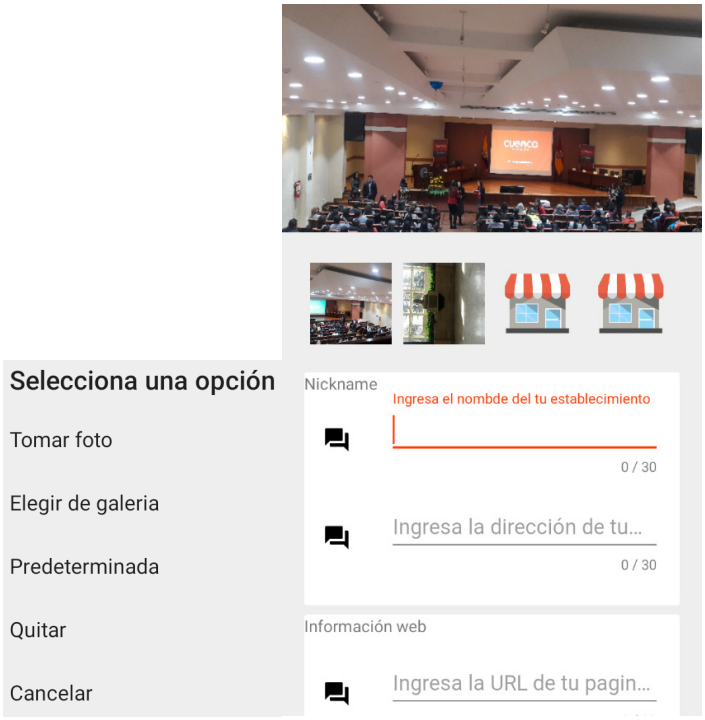


Figura 4.24: Pantallas para el registro de imágenes de un local

Fuente: (Calle & Largo, 2017)

4.1.10.3 Registro de información de local: Permite al usuario ingresar la información requerida por la aplicación, para su almacenamiento y registró como local.

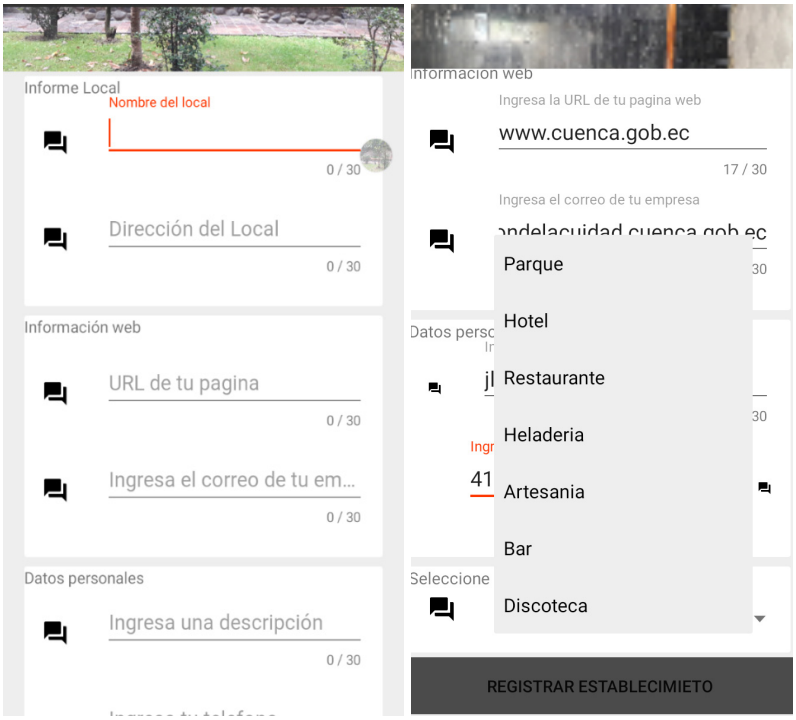


Figura 4.25: Pantallas para el registro de información de un local

Fuente: (Calle & Largo, 2017)

4.1.11 Cerrar Sesión: Cierra la sesión activa del usuario en la aplicación.

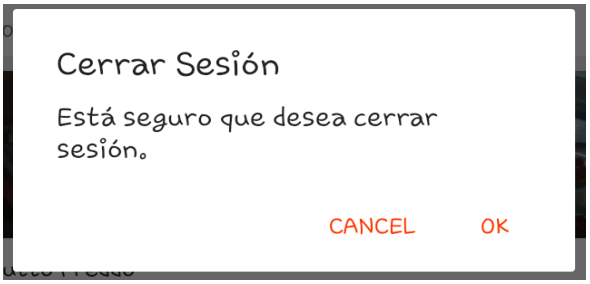


Figura 4.26: Pantallas de notificación para cerrar sesión

Fuente: (Calle & Largo, 2017)

4.2. Resultados de la página web

4.2.1 Inicio de sesión: Página que permite el inicio de sesión de un usuario, que ha realizado el proceso de registro desde la aplicación.

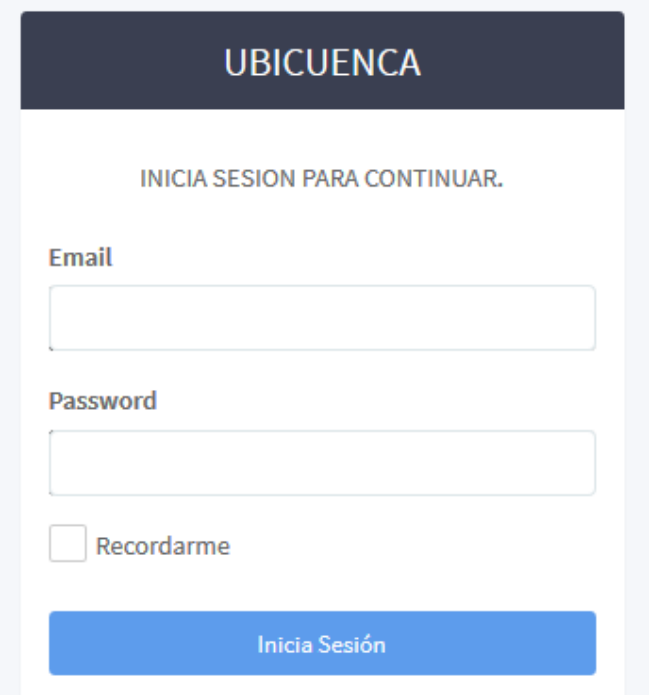


Figura 4.27: Login de usuario en página web

Fuente: (Calle & Largo, 2017)

4.2.2 Listado de Locales: Pantalla que permite ver mediante un listado los locales que ha registrado un usuario.

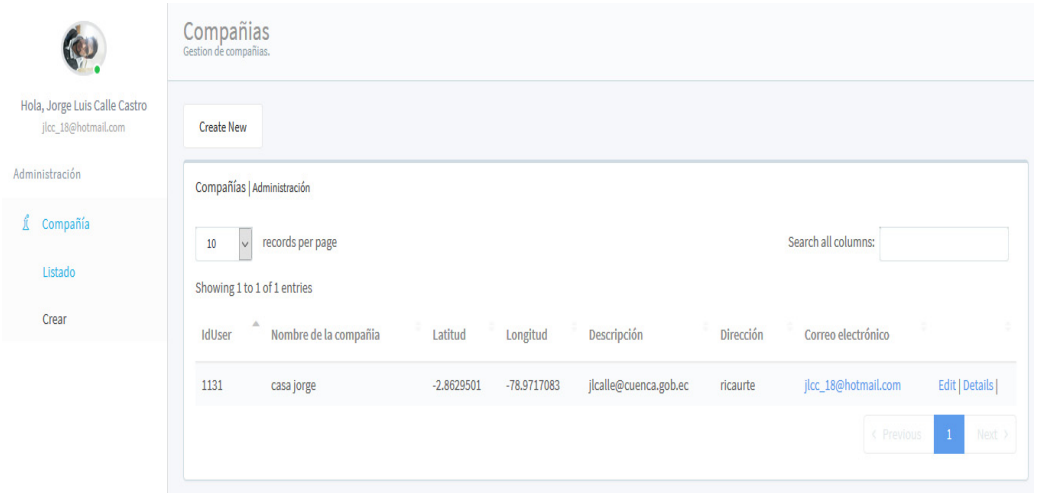


Figura 4.28: Listado de locales de un usuario

Fuente: (Calle & Largo, 2017)

4.2.3 Edición de Locales: Pantalla que permite a un usuario realizar modificaciones de la información ingresada de un local.

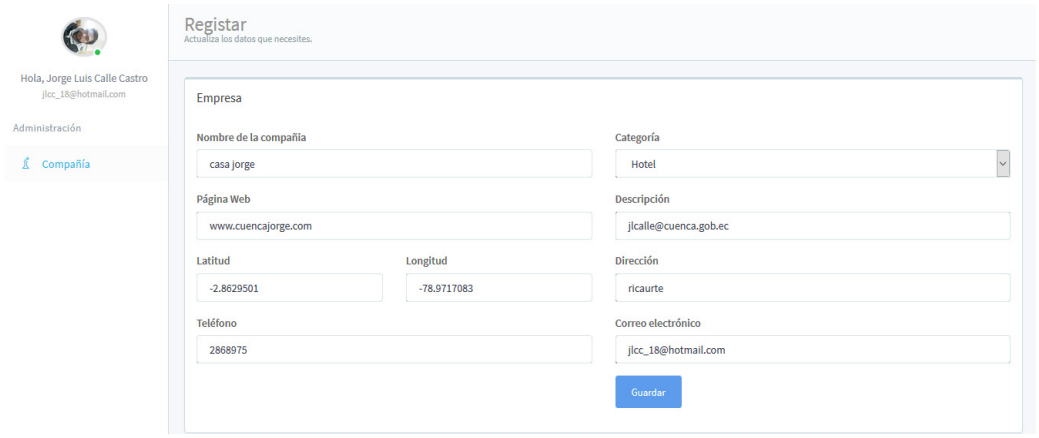


Figura 4.29: Edición de locales

Fuente: (Calle & Largo, 2017)

4.2.4 Ingreso de Locales: Pantalla que permite a un usuario realizar el ingreso de un nuevo local, para lo cual debe ingresar todos los datos requeridos.

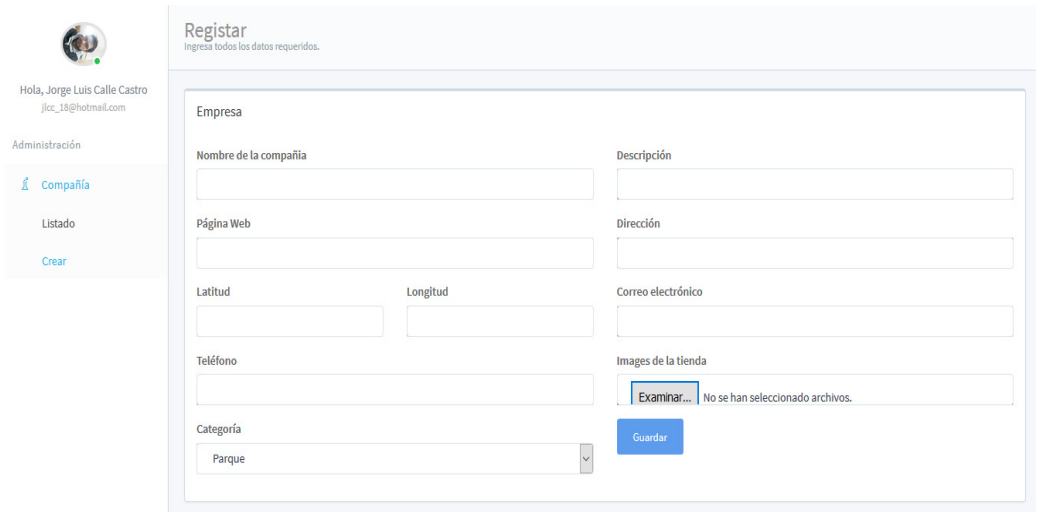


Figura 4.30: Ingreso de locales

Fuente: (Calle & Largo, 2017)

CAPÍTULO 5 DISCUSIONES Y CONCLUSIONES

Durante el desarrollo del presente trabajo de titulación denominado “guía turística de la ciudad de Cuenca para dispositivos móviles Android”, se utilizaron varias herramientas, las cuales van desde el análisis de la información, recopilación de requisitos hasta la obtención de una aplicación funcional. Durante el desarrollo del proyecto nos encontramos con algunas dificultades en las herramientas y tecnologías que se utilizaron:

No se pudo realizar la implementación de Google Maps de modo offline en la aplicación debido a que para el trazado de rutas a pie, se requiere de una conexión a internet, el modo offline funciona cuando se realiza la descarga del mapa y sirve para el trazado de rutas en vehículo.

Para el funcionamiento de la realidad aumentada el dispositivo móvil deberá tener como mínimo la versión 4.4 de sistema Android. Se requieren de permisos especiales que el usuario debe dar a la aplicación, tales como, ubicación, cámara y acceso al almacenamiento de imágenes y contenidos multimedia en el dispositivo.

El uso de un API para la realidad aumentada fue uno de los temas de mayor dificultad, debido a que para cumplir el objetivo de este proyecto, no se pudo encontrar un API de bajo coste, que nos permita realizar lo requerido, es decir la implementación de puntos de interés mediante marcadores, por tal motivo se realizó la implementación mediante el SDK de Wikitude para Android. Por lo cual en los resultados finales de la realidad aumentada, estos se verán con marcas de agua, debido a la limitación de la licencia de prueba del SDK.

Otro de los inconvenientes que se tuvo fue las actualizaciones del sistema operativo Android en su versión 7 en adelante, ya que se requiere de la reestructuración de la aplicación para su correcto funcionamiento. Para esta actualización se requiere de un manejo distinto de los procesos y métodos que controlan el almacenamiento de imágenes y contenidos multimedia, razón por la cual no está contemplada para estas versiones, esto es debido a que las actualizaciones fueron lanzadas cuando el proyecto ya estuvo en fases finales.

Se realizó la implementación de tal manera que desde la aplicación se puede realizar el registro de los locales y no solo desde la página web, como se planteó en un inicio, adicional se incluyó un método de pagos mediante el servicio de PayPhone.

El uso de Hockey app, ayuda a determinar errores que se producen en la aplicación durante la etapa de pruebas, con el objetivo de establecer una solución rápida y optima, esto se logra mediante los informes que son enviados por los betatester.

Por el momento la aplicación al ser un prototipo, no se está realizando la validación de datos de los establecimientos registrados, ya que para este proceso se requiere de personal especializado o de un sistema automatizado, el cual puede ser con una conexión o servicio al SRI, para la validación de los datos y constatación que estos sean verídicos

En el caso de Material Design sus efectos de funcionamiento y animación, varían de acuerdo a la versión del sistema operativo Android instalado en el dispositivo móvil.

La precisión de los sensores (GPS, Acelerómetro), no es exacta debido a que cuando se sale de la aplicación, el GPS es desactivado para ahorrar la batería del dispositivo, es por ello que cuando se ingresa nuevamente el número de satélites con los cuales se realiza la conexión ya no es la misma, por lo cual se requieren de aplicaciones de terceros que mantengan al GPS activo, pero consumen demasiada batería. Existen ocasiones en que el GPS, permanece conectado a la señal de ciertos satélites a pesar de que estos ya no se encuentren a su alcance, lo que hace que ya no busque señal de nuevos satélites y no se obtenga la ubicación exacta, otros de los factores que podría dar problemas es las carcasas de los dispositivos móviles, ya que existen de tipos metálicos los cuales puede interferir en la señal de precisión del GPS.

Se desarrolló la aplicación para la proforma Android debido a que luego de las investigaciones exhaustivas y el sondeo que se realizó en la ciudad de Cuenca se determina que es el sistema operativo con mayor número de usuario, por lo tanto es la razón principal para el prototipo sea considerado solo para dispositivos con el sistema operativo Android.

Como resultados del estudio mediante una etnografía enfocada, en lo que se refiere a usuarios locales, es decir de la cuidad y del país, se obtuvo calificaciones con un promedio general de 3,46/4, lo cual equivale a un 86,5% de aceptación de la aplicación, tomando en cuenta algunos aspectos de diseño y rendimiento por mejorar.

En lo que se refiere a usuarios de 36 a años en adelante, se complica el uso, debido a que se tiene que utilizar los sistemas de geoposicionamiento del dispositivo, también no se encuentran familiarizados con los sistemas de navegación, se requiere de ayuda para usar el aplicativo.

Como resultados del estudio mediante una etnografía enfocada, en lo que se refiere a usuarios extranjeros, se obtuvo calificaciones con un promedio general de 3,50/4, lo cual equivale a un 87,5% de aceptación de la aplicación, es decir se tiene mayor interés sobre la aplicación en este grupo de usuarios, teniendo como un inconveniente la traducción al inglés, ya que por motivos de ser un prototipo no se delimito ese alcance.

Al culminar el proyecto se realizaron pruebas de procesos de flujo, validaciones y comportamiento de las interfaces en varios dispositivos Android, comprobando el óptimo funcionamiento de cada una de las herramientas que se utilizaron durante el desarrollo, esto nos permite determinar y garantizar el cumplimiento de los objetivos planteados al inicio del proyecto, la cual fue obtener una aplicación que sirva de guía turística para dispositivos móviles con sistema operativo Android, mediante la implementación de la realidad aumentada como una innovación.

RECOMENDACIONES

Para una correcta implementación y corrección de errores, se recomienda hacer uso de la herramienta de Google Firebase, la cual realiza un análisis sobre el uso del aplicativo y la interacción con los usuarios, creando un informe con estadísticas cada vez que la aplicación tenga un fallo o deje de funcionar, estos datos son controlados y enviados en tiempo real.

Permite obtener un respaldo del contenido generado por la aplicación ya que se basa en la tecnología de Google Cloud Storage, además *Google Firebase* realiza un *test lab* de la aplicación, mediante dispositivos alojados en un centro de datos de Google.

Todas estas herramientas pueden ser monitorizadas desde una sola consola la cual puede ser web o desde un dispositivo móvil para lo cual se debe instalar y configurar el SDK de *Google Firebase.*

BIBLIOGRAFÍA

GitHub . (07 de 2016). *HockeySDK-Android*. Obtenido de https://github.com/bitstadium/HockeySDK-Android

Oracle Corporation. (2017). *MySql.*

Square, Inc. (2016). *Square Open Source*. Obtenido de http://square.github.io/

Wikitude GmbH. (03 de 05 de 2016). *Geo AR (Points of Interest)*. Obtenido de http://www.wikitude.com/external/doc/documentation/latest/android/poi.html#geo-ar-points-of-interest

Wikitude GmbH. (03 de 05 de 2016). *Wikitude SDK Android*. Obtenido de http://www.wikitude.com/external/doc/documentation/latest/android/gettingstartedandroid.html#getting-started

Academia Android. (16 de 05 de 2014). *IDE: Entornos Integrados de Desarrollo para Android*. Obtenido de https://academiaandroid.com/ide-entornos-integrados-de-desarrollo-para-android/

Ander, P., Borges, C., Yoseka, P., & Ainhoa, A. V. (2014). *Uso de algoritmos de enrutamiento para el cálculo de indicadores de sostenibilidad.* Universidad de Girona. Bilboa: Deusto Institute of Technology - DeustoTech Energ. Recuperado el 05 de 15 de 2017, de http://dugi-doc.udg.edu/bitstream/handle/10256/9041/UsoAlgoritmos-Article.pdf?sequence=2

Angel Restrepo, P. L., & Marín Sepulveda, L. (05 de 05 de 2011). *A COMPUTATIONAL METHOD TO OBTAINT OPTIMAL PATHS IN ROAD NETWORKS.* Obtenido de http://revistas.unal.edu.co/index.php/dyna/article/view/25773/39408

Araque, J. (25 de 06 de 2016). *Estas son las diferencias entre los tres modos de ubicación de Android*. (L. v. Galicia, Editor) Recuperado el 28 de 04 de 2016, de https://andro4all.com/2016/06/diferencias-tres-modos-ubicacion

Argelega, A. B. (20 de 01 de 2010). APLICACIÓN DE GESTIÓN DE INFORMACIÓN GEOLOCALIZADA EN ANDROID. Barcelona, Catalunya, España. Recuperado el 09 de 04 de 2017

Arlandy Rodriguez, M. (26 de 03 de 2012). *Trabajando con Intellij IDEA 11.* Obtenido de https://www.adictosaltrabajo.com/tutoriales/trabajando-intellijidea/

Aula formativa Online. (27 de 02 de 2016). *Que es material design.* Obtenido de http://blog.aulaformativa.com/que-es-el-material-design/

AumentaME. (19 de 08 de 2011). *Tipos de Realidad Aumentada*. (Creative Commons de reconocimiento-NoComercial-CompartirIgual 3.0 España (CC BY-NC-SA 3.0) .) Recuperado el 5 de 04 de 2017, de http://aumenta.me/node/36

Basterra , Bertea , Borello , Castillo , & Venturi. (11 de 03 de 2017). Android OS Documentation. Recuperado el 25 de 03 de 2017, de https://media.readthedocs.org/pdf/androidos/latest/androidos.pdf

BBVAOPEN4U. (23 de 03 de 2016). *API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos*. Obtenido de https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos

Booch, G., Jim , R., & Ivar , J. (08 de 03 de 2015). El Lenguaje Unificado de Modelado. Recuperado el 5 de 06 de 2017, de http://tecnocomputo.co/almacen/Root/Informatica%20Pdf/Informatica_Grady%20Booch%20Jim%20Rumbaug%20e%20Ivan%20Jacopson-El%20lenguaje%20Unificado%20de%20modelado%20UML.pdf

Callejo, M. S. (2015). Curso OpenStreetMap. Zaragoza, España. Recuperado el 05 de 04 de 2017, de https://osluz.unizar.es/files/documento_curso_osm_2015_1.pdf

Canel, S. M. (2009). Microcontroladores ARM. Argentina: TD II. Obtenido de http://www.electron.frba.utn.edu.ar/upload/Materias/95-0429/archivos/Cap10_2009_ARM7_apunte.pdf

Clarke, Modet & Cº, España. (2014). Tendencias en las redes tecnologicas moviles. Madrid, España: Creative Commons . Recuperado el 20 de 03 de 2016, de https://www.b2match.eu/system/murciatic2014/files/01_Informe_tendencias_en_las_tecnolog%C3%ADas_m%C3%B3viles.pdf?1409128108

Computación, I. D. (2014). Servicios Multimedia para Dispositivos Moviles. Obtenido de https://moodle2013-

14.ua.es/moodle/pluginfile.php/79922/mod_resource/content/3/smdm-01-plataformas.pdf

Creative Commons. (24 de 05 de 2015). *Conoce Android Studio.* Obtenido de https://developer.android.com/studio/intro/index.html?hl=es-419

Creative Commons. (2016). *Manifiesto de la app*. Obtenido de https://developer.android.com/guide/topics/manifest/manifest-intro.html?hl=es

Creative Commons. (2016). *Material Desing para Android.* Recuperado el 12 de 05 de 2017, de https://developer.android.com/design/material/index.html

Daneri, P. A. (2009). *Qué es un waypoin*. (Agrimensor) Recuperado el 08 de 04 de 2017, de http://www.alvarezdaneri.com.ar/gps/waypoints_track_rutas.htm

Delgado Meraz, J. J. (14 de 05 de 2015). *Retrofit en Android.* Recuperado el 28 de 05 de 2017, de https://j2deme.github.io/2015/05/14/Retrofit-en-Android.html

Ditrandia. (s.f.). Informe Mobile en España y el mundo 2016. España. Recuperado el 27 de 03 de 2017, de http://www.amic.media/media/files/file_352_1050.pdf

EcuadorInmediato. (05 de 01 de 2014). *EcuadorInmediato*. (EcuaMEX) Recuperado el 27 de 03 de 2017, de http://ecuadorinmediato.com/index.php?module=Noticias&func=news_user_view&id=2818754043&umt=ecuador_usuarios_prefieren_android

Escobadia, S. (21 de 04 de 2016). *Cómo consultar tus propios mapas personalizados desde donde quieras y sin conexión a Internet*. Recuperado el 13 de 04 de 2017, de http://turisdiy.com/consulta-tus-propios-mapas-personalizados-offline/

Esquiva Rodriguez, A. (12 de 06 de 2014). *JSON I Qué es y para qué sirve JSON*. Obtenido de https://geekytheory.com/json-i-que-es-y-para-que-sirve-json/

Falcón, C. D. (2015). Microprocesadores en Smartphones. Obtenido de http://www.iuma.ulpgc.es/~nunez/clases-micros-para-com/mpc1011-trabajos/mpc1011-Cassandra-Microprocesadores%20en%20SMARTPHONES.pdf

Fernando, R. (2012). De la realidad virtual a la. (R. Fernando, Ed.) Palermo, España. Recuperado el 10 de 04 de 2017, de http://fido.palermo.edu/servicios_dyc/opendc/archivos/4674_open.pdf

Ferrer , P. J., Sanchez, I., & Tramoyeres, S. (s.f.). Qué son OSM y JOSM. Recuperado el 05 de 04 de 2017, de http://taller-de-josm-imposm-tilemill.readthedocs.io/es/latest/osm/osm_intro.html

Foundation, G. C. (2016). *Sistemas Operativos para dispositivos moviles.* Recuperado el 10 de 03 de 2017, de https://www.gcfaprendelibre.org/tecnologia/curso/informatica_basica/sistemas_operativos/5.do

Foundation, T. A. (2016). *Material design.*

Garcia Delgado, A. (20 de 05 de 2014). Implementación de un proveedor de autorizaciones OAuth 2.0 con Scala. Madrid, España. Recuperado el 15 de 05 de 2017, de https://repositorio.uam.es/bitstream/handle/10486/661016/garcia_delgado_alvaro_tfg.pdf?sequence=1

Garcia, F. (15 de 12 de 2015). *Realidad Aumentada. Concepto,tipos,fundamentos,posibilidades.* Recuperado el 14 de 04 de 2017, de https://prezi.com/cm3xcq5sjwka/realidad-aumentada-conceptotiposfundamentosposibilidades/

Garibay Cadenas, V. (06 de 06 de 2016). *Consumiendo una API con Retrofit 2 en Android.* Recuperado el 13 de 05 de 2017, de https://stories.devacademy.la/mi-primer-app-con-retrofit-y-android-ac61a8954a2c

Gartner. (2016). Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016. Stamford, CT 06902 US: Gartner, Inc.

Gatrell, A. (28 de 03 de 2012). Concepts of space and geographical data. Recuperado el 02 de 03 de 2017, de http://www.um.es/geograf/sigmur/sigpdf/temario_3.pdf

Gonzales Diez, M. (30 de 06 de 2016). Clean Arquitecture y RXJAVA en Android. Barceloma, España. Recuperado el 05 de 13 de 2017, de http://diposit.ub.edu/dspace/bitstream/2445/102790/2/memoria.pdf

Google Developers. (05 de 10 de 2016). *Google Maps Street View API*. Obtenido de https://developers.google.com/maps/documentation/android-api/streetview?hl=es-419

Google Developers. (05 de 02 de 2017). *Google Maps Directions API*. Obtenido de https://developers.google.com/maps/documentation/directions/intro?hl=es-419

GoogleMap. (2017). *GoogleMap*. Obtenido de https://developers.google.com/android/reference/com/google/android/gms/maps/GoogleMap#nested-class-summary

Greaves, D. J. (2011). System on Chip Design and Modelling. (Parte II). (U. o. Cambridge, Ed.) Estados Unidos: DJG. Recuperado el 24 de 03 de 2016, de http://www.cl.cam.ac.uk/teaching/1011/SysOnChip/socdam-notes1011.pdf

Hardt, E. (09 de 2012). The OAuth 2.0 Authorization Framework. New York, USA. Recuperado el 15 de 05 de 2017, de https://tools.ietf.org/pdf/rfc6749.pdf

INEC. (2016). Tecnologias de las información y comunicación (TIC)-2016. Quito. Recuperado el 25 de 03 de 2017, de http://www.ecuadorencifras.gob.ec//tecnologias-de-la-informacion-y-comunicacion-tic/

Izquierdo, C. A. (2010). Desarrollo de un sistema de realidad aumentada en dispositivos moviles. Valencia, España. Recuperado el 15 de 04 de 2017, de https://riunet.upv.es/bitstream/handle/10251/8597/PFC%20-%20Desarrollo%20de%20un%20sistema%20de%20Realidad%20Aumentada%20en%20dispositivos%20m%C3%B3viles.pdf

Lopez, f. (10 de 04 de 2015). *Uso de SQLite en Android.* Obtenido de http://www.academia.edu/7529715/PR%C3%81CTICA_1_USO_DE_SQLITE_EN_ANDROID

Maps, G. (s.f.). Google Maps. Estados Unidos. Recuperado el 11 de 04 de 2017, de https://static.googleusercontent.com/media/www.google.com/es//educators/activities/Maps_JCescalante.pdf

Molina Rivera, Y. J., Sandoval Cardona, J., & Toledo Franco, S. A. (12 de 03 de 2012). SISTEMA OPERATIVO ANDROID: CARACTERÍSTICAS Y FUNCIONALIDAD. Pereira, Colombia. Recuperado el 25 de 04 de 2016, de

http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/2687/0053M722.pdf;jsessionid=9A541FAE87A5104DC2174FAE8969BA4F?sequence=1

Morales, A. (13 de 01 de 2016). *MappingGIS*. Recuperado el 11 de 04 de 2017, de https://mappinggis.com/2013/11/los-formatos-gis-vectoriales-mas-populares/

Neis P, Z. D. (2010). *OpenStreet Map.* Austria.

OpenStreetMap. (2017). *OpenStreetMap*. (Fundación OpenStreetMap) Recuperado el 10 de 04 de 2017, de https://www.openstreetmap.org/#map=17/-2.89730/-79.00149

Oracle Corporation. (23 de 08 de 2016). *NetBeans IDE 8.2 Release Notes.* Recuperado el 18 de 05 de 2017, de https://netbeans.org/community/releases/82/relnotes.html#tech

Ordoñez, R. P. (10 de 05 de 2010). Aplicación de realidad Aumentada. España. Recuperado el 20 de 04 de 2017, de http://fit.um.edu.mx/CI3/publicaciones/TechnicalReportCOMP-030-2010.pdf

Pavon Mestras, J. (15 de 09 de 2010). El patrón Modelo-Vista-Controlador (MVC). Madrid, España. Recuperado el 19 de 05 de 2017, de https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf

PayPhone. (2017). *PayPhone; Developer*. Obtenido de https://appdeveloper.payphonetodoesposible.com/

Perez, E. (09 de 11 de 2014). Que es material design. España. Recuperado el 12 de 05 de 2017, de https://elandroidelibre.elespanol.com/2014/11/que-es-material-design.html

Peris Martinez, D. (2014). Desarrollo de navegador para coches. València, España. Recuperado el 03 de 04 de 2017, de https://riunet.upv.es/bitstream/handle/10251/48178/PERIS%20-%20Desarrollo%20de%20navegador%20para%20coches%20basado%20en%20Android%20y%20OpenStreetMap.pdf?sequence=2

Pombo, H. L. (2010). Análisis y Desarrollo de Sistema de Realidad Aumentada. Madrid. Recuperado el 10 de 04 de 2017, de http://eprints.ucm.es/11425/1/memoria_final_03_09_10.pdf

Ponce, J. P. (2016). Estadisticas facebook ecuador. Quito: Formación Gerencial Business Training Center. Recuperado el 27 de 03 de 2017, de http://blog.formaciongerencial.com/2016/02/01/estadisticasfacebookecuador/

Pozo, J. D. (2015). Introducción a los dispositivos móvlies. Catalunya, España: Creative Commons. Recuperado el 21 de 03 de 2017, de https://desarrolloappandroid.files.wordpress.com/2013/06/tecnologia-desarrollo-dispositivos-moviles.pdf

REDERMAUS. (17 de 04 de 2015). *World Geodetic System 1984 (WGS84)*. Obtenido de https://confluence.qps.nl/pages/viewpage.action?pageId=42315395#app-switcher

Revelo, J. (26 de 05 de 2016). *Google Maps Android Api v2: Guía De Mapas*. Recuperado el 25 de 05 de 2017, de http://www.hermosaprogramacion.com/2016/05/google-maps-android-api-v2/#descargar-gps

Rios, M. C. (2013). COMPOSICIÓN DEL SISTEMA OPERATIVO MÓVIL IOS DE APPLE Y EL. Pereira, Colombia. Obtenido de http://ribuc.ucp.edu.co:8080/jspui/bitstream/handle/10785/1984/CDMIST81.pdf?sequence=1

Robledano, Á. (27 de 05 de 2015). *Tutorial Bootstrap 3: Introducción e instalación*. Recuperado el 28 de 05 de 2017, de https://openwebinars.net/blog/tutorial-bootstrap-3-introduccion-e-instalacion/

Rodriguez, J. A. (13 de 04 de 2014). *Que es Gradle.* Obtenido de http://josearodriguez.me/post/82569801132/que-es-gradle

Sanchez Escobedio, R. E., Ramirez Sanchez, J., Parroquin Amaya, P. C., Flores Garcia, S., Chavez Pierce, J. E., González Quezada , M. D., & Carrillo Saucedo, V. M. (2015). *REALIDAD AUMENTADA: DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA DE CORTE CONSTRUCTIVISTA PARA EL APRENDIZAJE DECONCEPTOS DE FÍSICA* (1 ed., Vol. 19). Juarez, Chihuahua, Mexico: Universidad Autónoma de Ciudad Juárez. Recuperado el 08 de 04 de 2017, de http://www.uacj.mx/DGDCDC/SP/Documents/RTI/2015/IIT/Realidad.pdf

Seguros, M. (25 de 05 de 2016). *Tu Blog Tecnologico*. Recuperado el 30 de 03 de 2017, de http://tublogtecnologico.com/aplicaciones-mapas-o-navegadores/

SL, A. P. (26 de 05 de 2016). La Realidad Aumentada: un nuevo prisma para ver el mundo. (Adarve, Ed.) Valencia, Pais. Recuperado el 8 de 04 de 2017, de http://www.adarveproducciones.com/uploads/ficha/fichero/APLICACIONES%20DE%20LA%20REALIDAD%20AUMENTADA_124.pdf

Troya Carbajal, S. O. (2016). IMPLEMENTACIÓN DE UN SISTEMA DE GEOLOCALIZACIÓN DE SITIOS TURISTICOS MEDIANTE TECNOLOGIA GPS. Ambato, Ecuador. Recuperado el 10 de 04 de 2017, de http://dspace.uniandes.edu.ec/bitstream/123456789/5263/1/PIUASIS010-2016.pdf

Unidas, N. (2000). Manual de sistemas de información geográfica y cartografía digital. NuevaYork, Estados Unidos : PUBLICACIÓN DE LAS NACIONES UNIDAS. Recuperado el 30 de 03 de 2017, de https://unstats.un.org/unsd/publication/SeriesF/SeriesF_79S.pdf

WORDp. (2013). GPRS, EDGE, UMTS, 3G, HSDPA, 4G. Obtenido de http://wordp.relatividad.org/ciencia/gprs-edge-umts-3g-hsdpa/

Zabaljauregui, M. (20 de 03 de 2015). Introducción al kernel de linux. Argentina. Recuperado el 25 de 04 de 2017, de http://linux.linti.unlp.edu.ar/images/d/d7/Presentacion-introduccion-kernel.pdf

Zerbst, S. (2004). 3D Game Engine Programmin. Thomsom, USA. Recuperado el 25 de 04 de 2016, de http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/hermida_t_r/capitulo2.pdf

ANEXOS

Anexo 1

Resultados de encuestas

Para establecer con exactitud la plataforma de desarrollo, se plateo un estudio con una muestra de 400 personas de la ciudad de Cuenca, tanto a usuarios locales como a extranjeros, esto se realizó en la fase de recolección de requisitos, los datos obtenidos sirvieron para determinar el alcance y funcionalidad de la aplicación.

1.1: Resultados de encuestas a usuarios

Sistema Operativo	Personas
Android	272
IOS	96
Windows Phone	32
Total	**400**

Tabla 7. Resultados de Sistemas Operativos

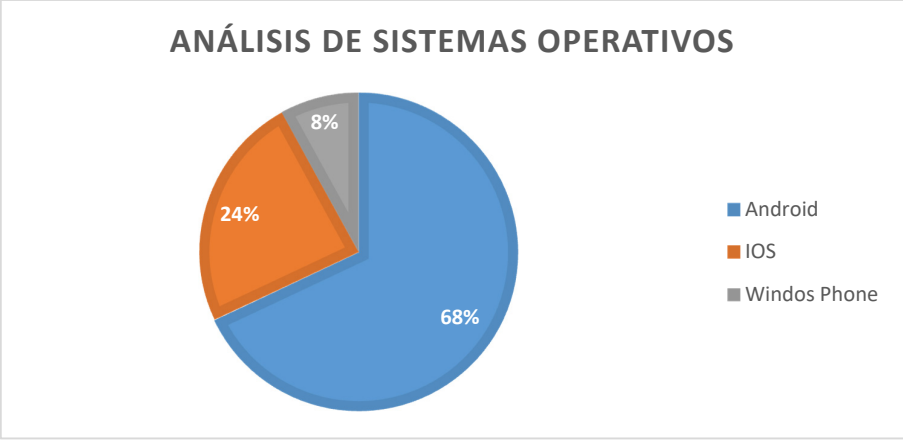


Figura 114: Sistematización de resultados de sistemas operativos

Fuente: (Calle & Largo, 2017)

1.2. Encuestas realizadas a dueños de locales

Para la obtención de estos datos se realizó una encuesta en donde se estableció una muestra de 20 locales comerciales, teniendo como base el siguiente formulario.

1.3 Formulario para realizar encuestas

ENCUESTA PARA DETERMINAR REQUERIMIENTOS			
1.Por favor indicar que tipo de dispositivo móvil tiene	Teléfono Básico ☐	Smartphone ☐	
2.Qué modelo de dispositivo posee	Android ☐	IPhone ☐	Windows Phone ☐
3.Posee conexión a internet	WIFI ☐	Red de datos ☐	
4.Conoce usted aplicaciones de turismo	Si ☐	No ☐	
5.Prefiere usted aplicaciones de paga o gratuitas	Paga ☐	Gratuitas ☐	
6.Pagaría usted por usar una aplicación para promocionar su local	Si ☐	NO ☐	
Justifique su respuesta:	---------- ---------- ----------		
7.Está en desarrollo una aplicación local de turismo, nos podría ayudar con su información o sugerencia sobre qué información se debería mostrar en la aplicación ---------- ---------- ----------			

Tabla 8. Formulario para realizar encuestas a usuarios

1.4 Resultado de encuestas de dueños de locales

Pregunta 1: Por favor indicar que tipo de dispositivo móvil tiene

Smartphone	24
Básico	6

Tabla 9. Resultados de tipo de dispositivo móvil

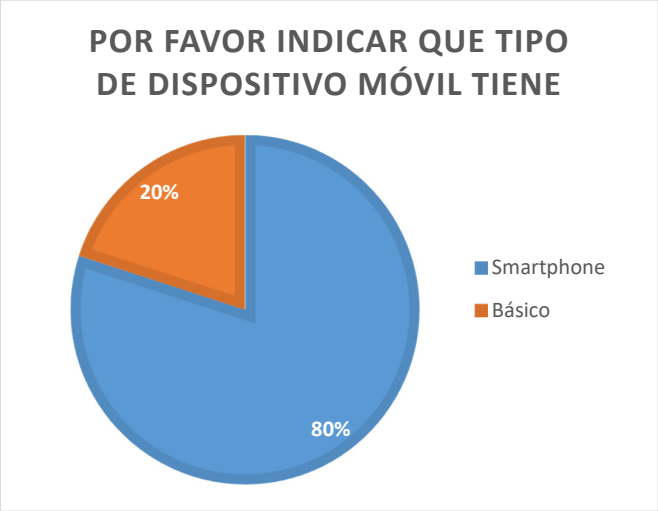


Figura 115: Sistematización de resultados de tipo de dispositivos

Fuente: (Calle & Largo, 2017)

Pregunta 2: Qué modelo de dispositivo posee

Android	21
IOS	7
Phone	2

Tabla 10. Resultados de modelo de dispositivo móvil

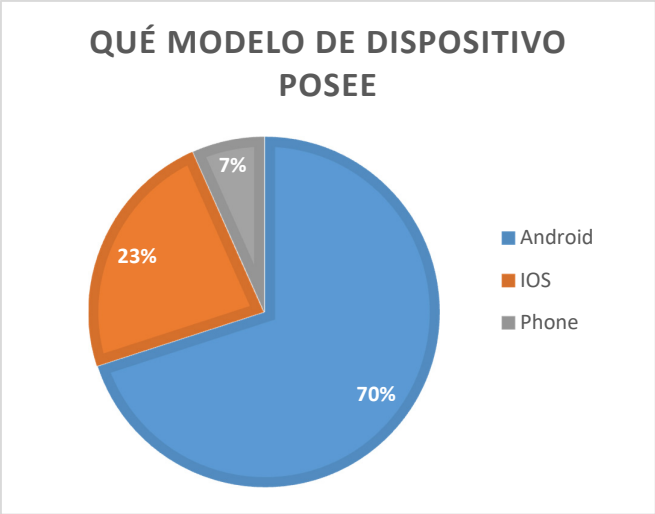


Figura 116: Sistematización de resultados de modelos de dispositivos móviles.

Fuente: (Calle & Largo, 2017)

Pregunta 3: Posee conexión a internet

WIFI	18
Datos	12

Tabla 11. Resultados de disponibilidad de conexión a internet

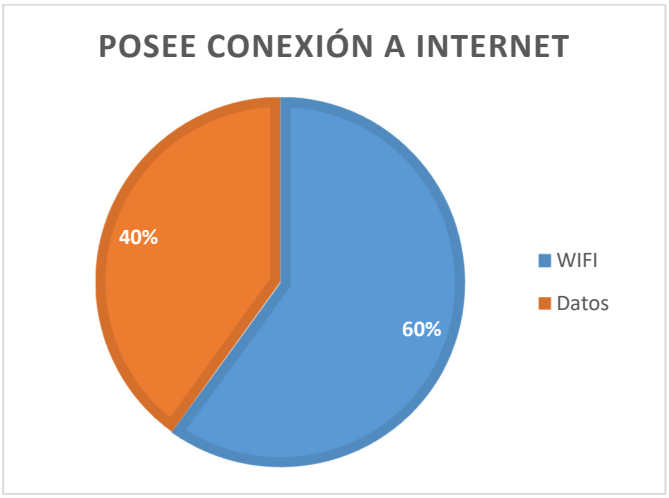


Figura 117: Sistematización de resultados disponibilidad de conexión a internet.

Fuente: (Calle & Largo, 2017)

Pregunta 4: Conoce usted aplicaciones de turismo

Si	22
No	8

Tabla 12. Resultados de conocimiento aplicaciones de turismo

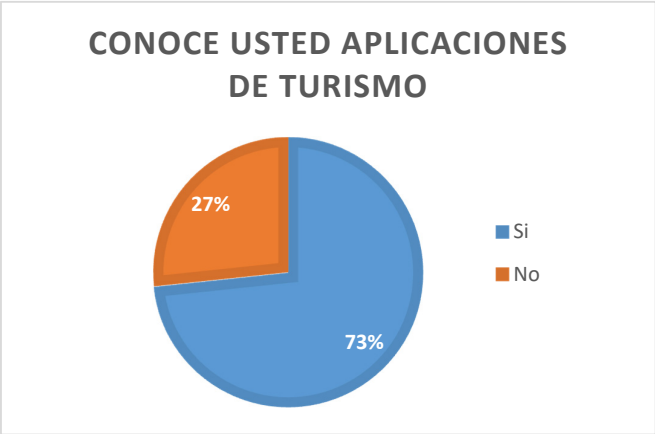


Figura 118: Grafica de resultados conocimiento aplicaciones de turismo.

Fuente: (Calle & Largo, 2017)

Pregunta 5: Prefiere usted aplicaciones de paga o gratuitas

Paga	6
Gratuitas	24

Tabla 13. Resultados de pago por una aplicación

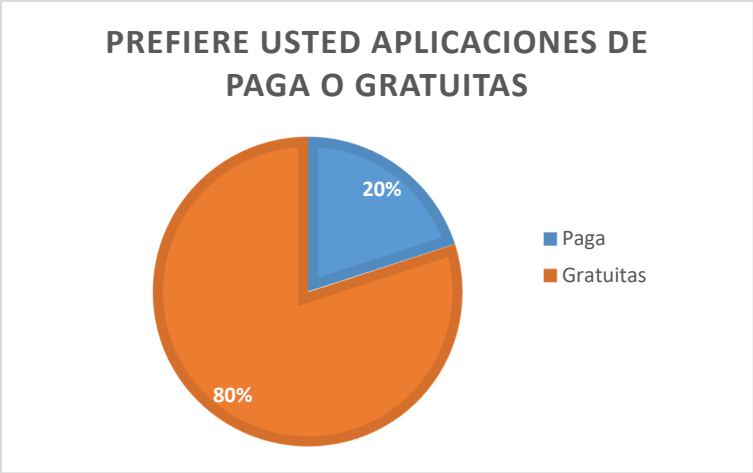


Figura 119: Grafica de resultados de pago por una aplicación

Fuente: (Calle & Largo, 2017)

Pregunta 6: Pagaría usted por usar una aplicación para promocionar su local

Si	12
No	18

Tabla 14. Resultados de pago para promocionar local mediante la aplicación

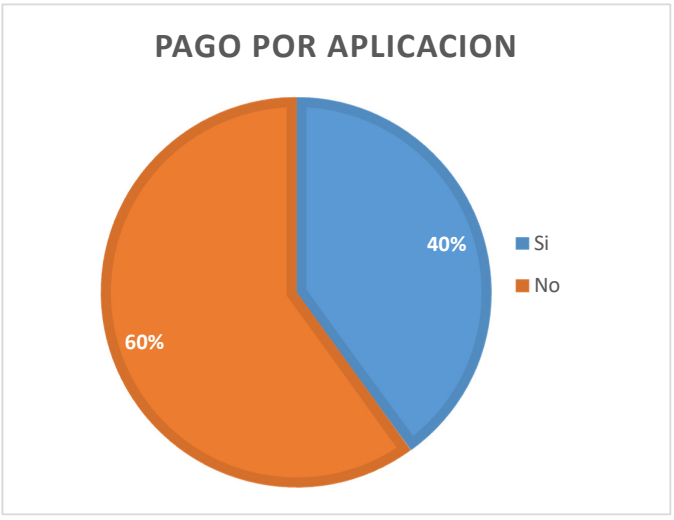


Figura 120: Sistematización de resultados pago para promocionar local mediante la aplicación.

Fuente: (Calle & Largo, 2017)

Justificaciones: Información que fue obtenida y colocada en la introducción del capítulo 4.

Pregunta 7: Está en desarrollo una aplicación local de turismo, nos podría ayudar con su información o sugerencia sobre qué información se debería mostrar en la aplicación.

Información que fue obtenida y colocada en la introducción del capítulo 4.

Anexo 2

2.1 Resultados de evaluación de la aplicación mediante una etnografía enfocada

Esta evaluación se llevó a acabo a usuarios finales es decir a los turistas que se encontraban dentro del centro histórico y son los que visualizan la información de los locales registrados en la aplicación, para lo cual se definió dos tipos de usuarios:

- Usuarios locales
- Usuarios extranjeros

Los usuarios seleccionados serán categorizados por edades, permitiendo así determinar si existe dificultad de uso en la aplicación, los rangos de edad que se delimitaron son los siguientes:

- 15 – 20 años
- 21 – 35 años
- 36 o más años

Para esta evaluación se pidió a los usuarios que utilicen la aplicación y que evalúen aspectos tales como:

- Diseño
- Operabilidad
- Facilidad de Navegación
- Eficiencia
- Rendimiento

Para lo cual se estableció un sistema de calificación con los siguientes pesos:

PESOS	CALIFICACION
0	Deficiente
1	Regular
2	Buena
3	Muy Buena
4	Excelente

Tabla 15. Pesos de Calificaciones para etnografía

2.2 Resultados Obtenidos

Usuarios Locales

Edad	Parámetro	Calificación											Promedio
15-20	diseño	3	3	4	3	3	4	3	3	4	3	4	3,36
	operabilidad	4	3	4	4	3	4	4	3	3	3	4	3,55
	facilidad de navegación	4	4	4	4	4	3	3	4	3	3	3	3,55
	eficiencia	3	3	3	4	4	3	4	3	4	4	4	3,55
	rendimiento	3	3	3	4	3	4	3	3	4	4	3	3,36
21 -35	diseño	3	4	3	4	4	4	3	2	3	4	4	3,45
	operabilidad	3	4	4	3	3	4	4	3	3	4	3	3,45
	facilidad de navegación	4	3	3	3	4	4	4	3	4	3	3	3,45
	eficiencia	4	4	3	4	3	3	3	3	4	4	4	3,55
	rendimiento	3	3	4	3	4	3	4	3	3	4	4	3,45
36 o mas	diseño	4	4	4	4	3	3	3	4	4	3	3	3,55
	operabilidad	3	3	3	3	4	4	3	3	3	4	4	3,36
	facilidad de navegación	4	3	4	2	3	4	3	4	3	3	4	3,36
	eficiencia	3	3	4	4	3	3	4	4	3	3	3	3,36
	rendimiento	3	4	3	4	4	4	3	3	3	4	4	3,55

Tabla 16. Resultados etnografía usuarios locales

Usuarios Extranjeros

Edad	Parámetro	Calificación											Promedio
15-20	diseño	4	3	4	3	3	4	3	4	3	4	3	3,45
	operabilidad	3	3	4	4	3	3	4	3	4	4	3	3,45
	facilidad de navegación	4	4	4	3	4	3	3	4	3	3	4	3,55
	eficiencia	3	3	4	4	3	3	3	3	4	3	4	3,36
	rendimiento	3	4	3	4	3	4	4	3	3	4	4	3,55
21 -35	diseño	4	3	4	4	4	4	4	3	3	4	3	3,64
	operabilidad	4	4	4	3	3	4	3	3	3	3	3	3,36
	facilidad de navegación	3	4	3	4	3	4	3	3	4	3	4	3,45
	eficiencia	4	4	4	4	4	4	3	3	3	4	4	3,73
	rendimiento	3	4	3	3	3	4	3	4	3	4	3	3,36
36 o mas	diseño	4	4	4	4	4	4	3	3	4	3	3	3,64
	operabilidad	4	3	4	3	3	3	4	4	3	4	3	3,45
	facilidad de navegación	4	4	4	3	3	3	3	3	3	4	4	3,45
	eficiencia	3	4	4	4	4	3	4	4	3	3	4	3,64
	rendimiento	4	3	3	4	3	4	4	3	3	4	3	3,45

Tabla 17. Resultados etnografía usuarios extranjeros

Anexo 3

3.1 Instalación de Android Studio

Para la correcta instalación de la herramienta de desarrollo Android Studio, se deberán seguir los siguientes pasos:

1. Ejecución del programa, en modo administrador, con el fin de otorgarle todos los permisos requeridos para la instalación.

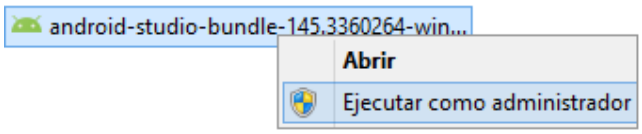


Figura 121: Ejecución de Android Studio para su instalación

Fuente: (Calle & Largo, 2017)

Se procede a descomprimir el paquete para la instalación.

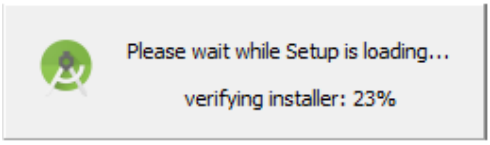


Figura 122: Proceso de instalación de Android Studio

Fuente: (Calle & Largo, 2017)

2. Asistente de instalación, en cual se deberá seleccionar la opción Next.

Figura 123: Asistente de instalación de Android Studio

Fuente: (Calle & Largo, 2017)

3. Se establece las opciones de instalación por defecto.

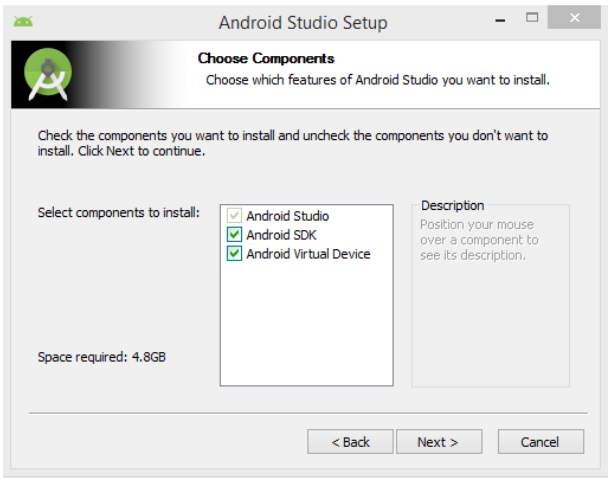


Figura 124: Opciones de instalación de Android Studio

Fuente: (Calle & Largo, 2017)

4. Aceptamos los términos de instalación

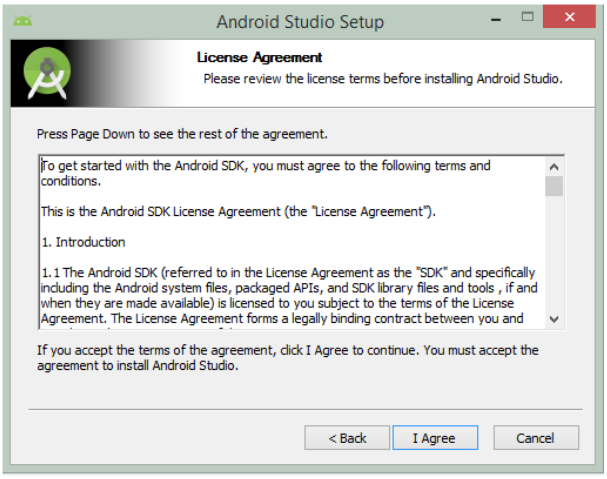


Figura 125: Términos de instalación de Android Studio

Fuente: (Calle & Largo, 2017)

5. Se especifican las direcciones de instalación para Android Studio y para el SDK, que requiere la instalación.

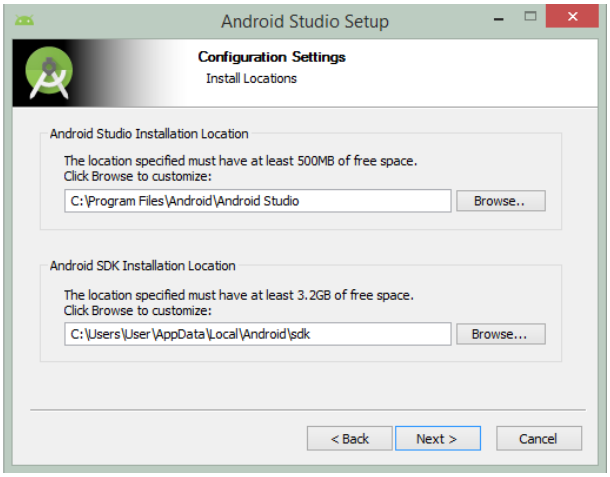


Figura 126: Ruta de instalación de Android Studio

Fuente: (Calle & Largo, 2017)

6. Se espera que el proceso de instalación concluya (10 minutos aproximadamente)

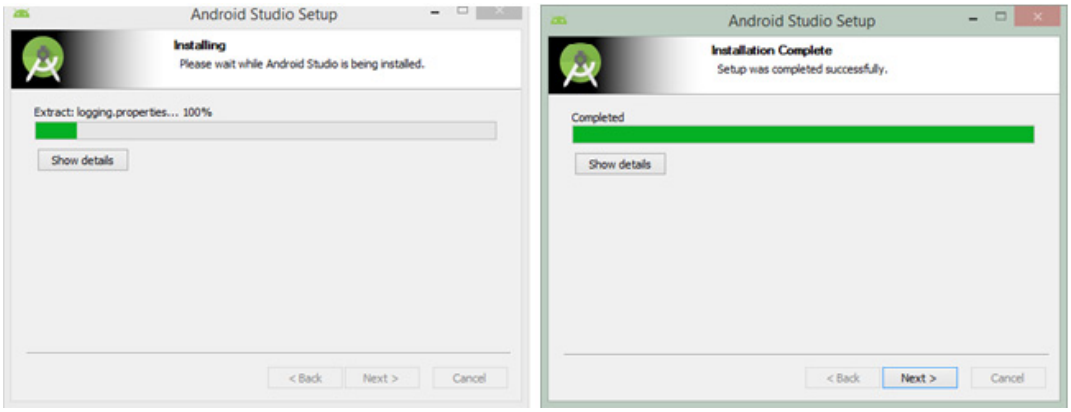


Figura 127: Proceso de instalación de Android Studio.

Fuente: (Calle & Largo, 2017)

7. Una vez completada la instalación, se debe determinar la ubicación en donde se desea almacenar los proyectos desarrollados o a su vez predeterminar la ubicación por defecto de Android Studio.

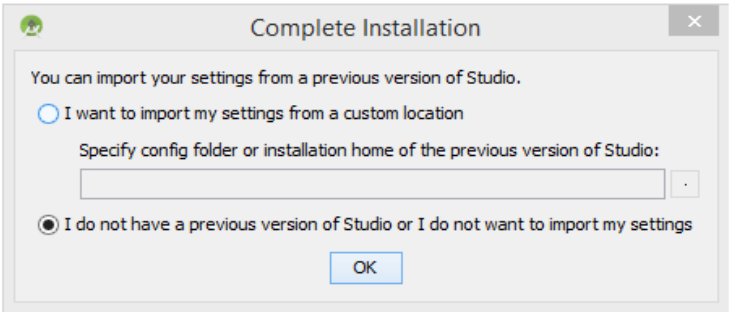


Figura 128: Rutas de almacenamiento de proyectos de Android Studio.

Fuente: (Calle & Largo, 2017)

8. Inicialización de Android Studio

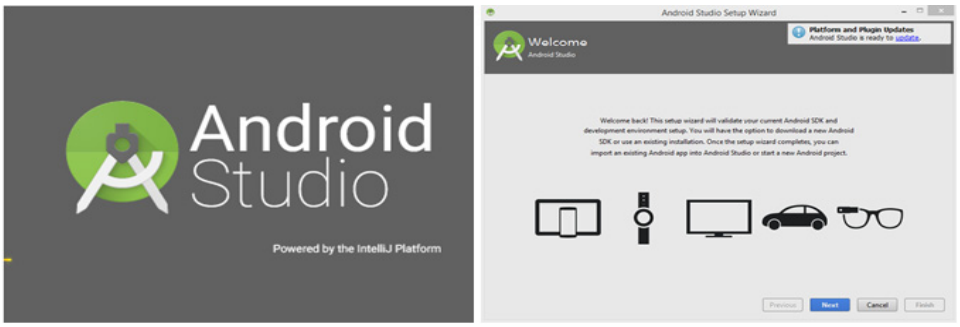


Figura 129: Inicialización de Android Studio.

Fuente: (Calle & Largo, 2017)

9. Seleccionamos el tipo de configuración que se desea para Android Studio.
    - Standard: Instala por defecto los componentes para el funcionamiento del software.
    - Custom: Nos permite seleccionar los componentes a instalar, así como también indicar la ruta del Android SDK Location.

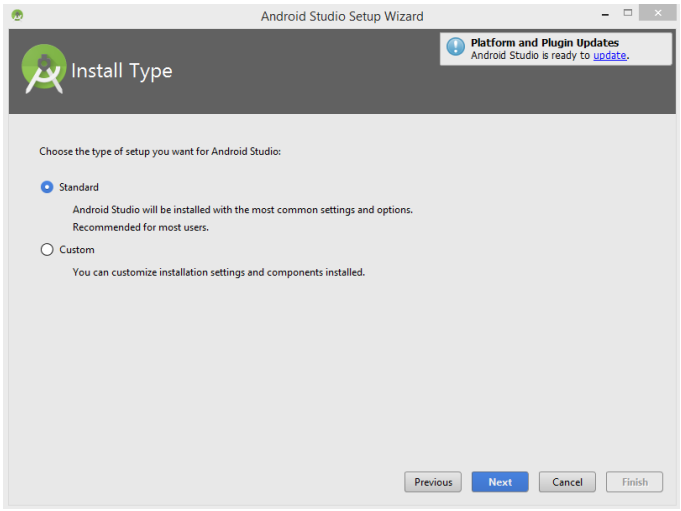


Figura 130: Parámetros de configuración de Android Studio.

Fuente: (Calle & Largo, 2017)

10. Luego de haber determinado los parámetros para el inicio de Android Studio, se debe realizar una actualización de las librerías y repositorios del sistema.

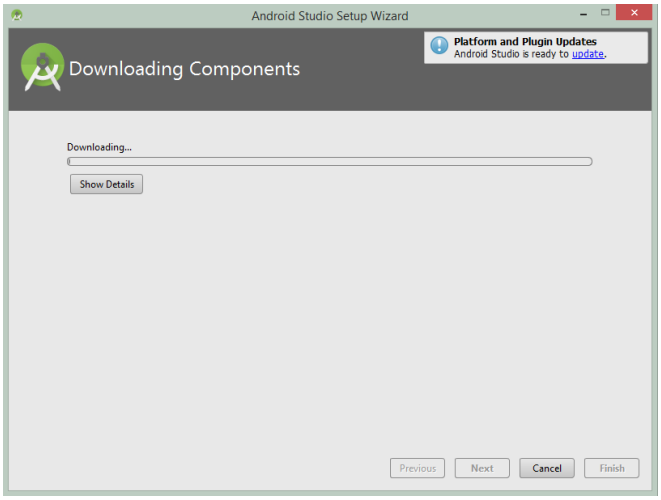


Figura 131: Actualización de Librerías y repositorios de Android Studio.

Fuente: (Calle & Largo, 2017)

11. Opciones disponibles luego de inicialización de Android Studio.

Figura 132: Inicialización de Android Studio

Fuente: (Calle & Largo, 2017)

3.2 Configuración de Android SDK: Android SDK (Software Development Kit) contiene emuladores y depuradores para el desarrollo de aplicaciones en Android. Incorpora librerías gratuitas para ser implementadas en los proyectos. (Creative Commons, 2015)

Al realizar la configuración del SDK establecemos la versión mínima con la cual funcionara nuestra aplicación, esto se debe establecer en la configuración inicial del proyecto.

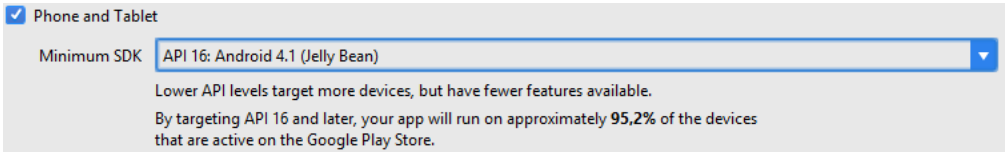


Figura 133: Configuración del SDK de Android

Fuente: (Calle & Largo, 2017)

Anexo 4

4.1 Resultados de reportes de Hockey App: Son datos que los usuarios betatester envían cuando la aplicación presenta errores.

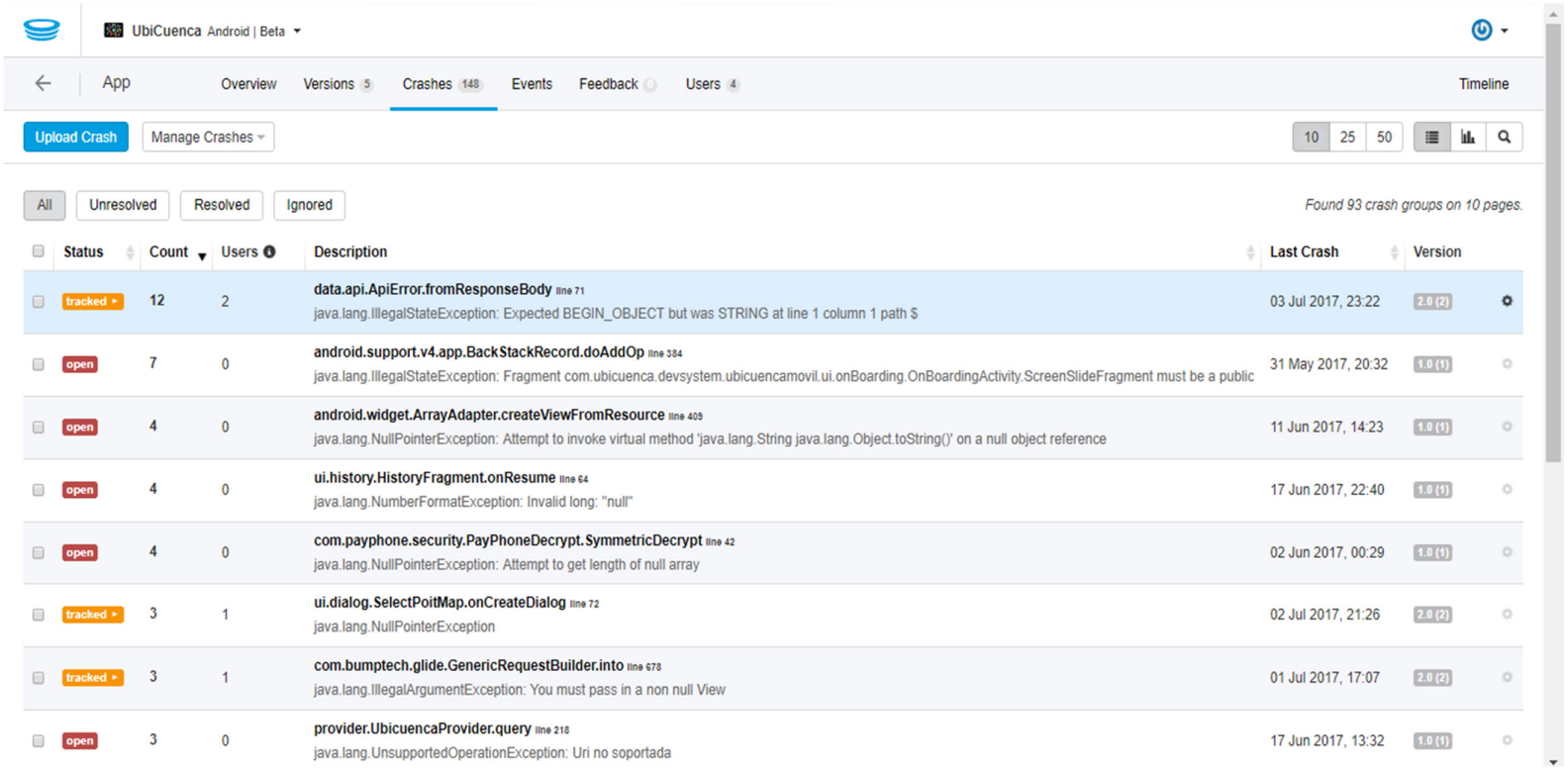


Figura 134: Reporte HockeyApp

Fuente: (Calle & Largo, 2017)

Doctora Jenny Ríos Coello, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay

C E R T I F I C A:

Que el Consejo de Facultad en sesión del 14 de febrero de 2017, conoció la petición de los señores **MILTON FABIAN LARGO ANGUISACA** con código 50469 y **JORGE LUIS CALLE CASTRO** , con código 50468, que presentan el diseño de su trabajo de titulación denominado: ***"GUIA TURISTICA DE LA CIUDAD DE CUENCA PARA DISPOSITIVOS MOVILES ANDROID",*** previa a la obtención del título de Ingeniero de Sistemas y Telemática.- El Consejo de Facultad acoge el informe de la Junta Académica y resuelve aprobar el diseño. Designa como **Director la ingniero Diego Pacheco Prado** y como miembros del Tribunal Examinador a los ingenieros Omar Delgado Inga y Andrés Patiño León; y,fija como plazo para la entrega del trabajo de titulación, seis meses contados desde la fecha de su aprobación, esto es hasta el **14 DE AGOSTO DE 2017,** debiendo el Director presentar a la Junta Académica, dos informes bimensuales del desarrollo del trabajo de titulación.

Cuenca, 15 de febrero de 2017

Dra. Jenny Ríos Coello
Secretaria de la Facultad

UNIVERSIDAD DEL AZUAY
FACULTAD DE ADMINISTRACION
SECRETARIA

CONVOCATORIA

Por disposición de la Junta Académica de la escuela de Ingeniería de Sistemas y Telemática, se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación: **"GUIA TURISTICA DE LA CIUDAD DE CUENCA PARA DISPOSITIVOS MOVILES ANDROID"**, presentado por los estudiantes Milton Fabian Largo Anguisaca y Jorge Luis Calle Castro, con códigos 50469 y 50468 respectivamente, previa a la obtención del grado de Ingeniero de Sistemas y Telemática, para el Martes, 24 de enero de 2017 a las 17h00.

Cuenca, 16 de enero de 2017

Dra. Jenny Ríos Coello
Secretaria de la Facultad

Ing. Diego Pacheco Prado .....

Ing. Omar Delgado Inga .....

Ing. Andrés Patiño León .....

Comunicado 16-01-2017 OK.

FECHA: 11-01-2017

ESCUELA DE SISTEMAS Y TELEMATICA

ESTUDIANTE: MILTON FABIAN LARGO ANGUISACA Y JORGE LUIS CALLE CASTRO

Sí procede Trab. Tit.

UNIVERSIDAD DEL AZUAY

ACTA

SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

1.5 **Nombre del estudiante:** Milton Fabian Largo Anguisaca y Jorge Luis Calle Castro
1.6 **Códigos:** 50469 y 50468 respectivamente
1.7 **Director sugerido:** Ing. Diego Pacheco Prado
1.8 **Codirector (opcional):** _____
1.4 **Tribunal:** Ing. Omar Delgado Inga
1.5 **Título propuesto:** ***"GUIA TURISTCA DE LA CIUDAD DE CUENCA PARA DISPOSITIVOS MOVILES ANDROID"***
1.6 **Resolución:**

1.6.1 Aceptado sin modificaciones _____

1.6.2 Aceptado con las siguientes modificaciones:

Delimitar claramente el alcance sobre el uso de algoritmos de ruteo

1.6.3 No aceptado
- Justificación:

Tribunal

Ing. Diego Pacheco Prado

Ing. Omar Delgado Inga

Ing. Andrés Patiño León

Sr. Milton Fabian Largo Anguisaca

Sr. Jorge Luis Calle Castro

Dra. Jenny Ríos Coello
Secretaria de la Facultad

RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN

1.1 **Nombre del estudiante:** Milton Fabian Largo Anguisaca y Jorge Luis Calle Castro
1.2 **Códigos:** 50469 y 50468 respectivamente
1.3 **Director sugerido:** Ing. Diego Pacheco Prado
1.4 **Codirector (opcional):**
1.1 **Título propuesto:** ***"GUIA TURISTICA DE LA CIUDAD DE CUENCA PARA DISPOSITIVOS MOVILES ANDROID"***
1.2 **Revisores (tribunal):** Ing. Omar Delgado Inga y Ing. Andrés Patiño León
1.5 **Recomendaciones generales de la revisión:**

	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
Línea de investigación				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?	✓			
Título Propuesto				
2. ¿Es informativo?	✓			
3. ¿Es conciso?	✓			
Estado del arte				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?	✓			
5. ¿Describe la teoría en la que se enmarca el trabajo		✓		
6. ¿Describe los trabajos relacionados más relevantes?	✓			
7. ¿Utiliza citas bibliográficas?	✓			
Problemática y/o pregunta de investigación				
8. ¿Presenta una descripción precisa y clara?	✓			
9. ¿Tiene relevancia profesional y social?	✓			
Hipótesis (opcional)				
10.¿Se expresa de forma clara?				
11.¿Es factible de verificación?				
Objetivo general				
12.¿Concuerda con el problema formulado?	✓			
13.¿Se encuentra redactado en tiempo verbal infinitivo?	✓			
14.¿Se encuentra redactado en tiempo verbal infinitivo?	✓			
Objetivos específicos				

15.¿Concuerdan con el objetivo general?	✓			
16.¿Son comprobables cualitativa o cuantitativamente?	✓			
Metodología				
17.¿Se encuentran disponibles los datos y materiales mencionados?	✓			
18.¿Las actividades se presentan siguiendo una secuencia lógica?	✓			
19.¿Las actividades permitirán la consecución de los objetivos específicos planteados?	✓			
20.¿Los datos, materiales y actividades mencionadas son adecuados para resolver el problema formulado?	✓			
Resultados esperados				
21.¿Son relevantes para resolver o contribuir con el problema formulado?	✓			
22.¿Concuerdan 23.con los objetivos específicos?	✓			
24.¿Se detalla la forma de presentación de los resultados?	✓			
25.¿Los resultados esperados son consecuencia, e 26.n todos los casos, de las actividades mencionadas?	✓			
Supuestos y riesgos				
27.¿Se mencionan los supuestos y riesgos más relevantes?	✓			
28.¿Es conveniente llevar a cabo el trabajo dado los supuestos y riesgos mencionados?	✓			
Presupuesto				
29.¿El presupuesto es razonable?	✓			
30.¿Se consideran los rubros más relevantes?	✓			
Cronograma				
31.¿Los plazos para las actividades son realistas?		✓		Revisar tiempo para levantamiento de información
Referencias				
32.¿Se siguen las recomendaciones de normas internacionales para citar?		✓		
Expresión escrita				
33.¿La redacción es clara y fácilmente comprensible?	✓			
34.¿El texto se encuentra libre de faltas ortográficas?	✓			

(*) Breve justificación, explicación o recomendación.

- Opcional cuando cumple totalmente,
- Obligatorio cuando cumple parcialmente y NO cumple.

- Delimitar claramente el alcance sobre el uso de algoritmos de ruteo

Ing. Diego Pacheco Prado

Ing. Omar Delgado Inga

Ing. Andrés Patiño León

UNIVERSIDAD DEL AZUAY

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

"Guía turística de la ciudad de Cuenca para dispositivos móviles Android"

Trabajo de graduación previo a la obtención del título de:

"Ingeniero en Sistemas y Telemática"

Autores:

Milton Fabian Largo Anguisaca

Jorge Luis Calle Castro

Fecha:

24/11/2016

Edición autorizada de 10.000 ejemplares
Del 778.501 al 788.500
Nº 0783848

UNIVERSIDAD DEL AZUAY

CONTENIDO

1.1 Datos Estudiante 3
1.2 Director Sugerido 3
1.3 Codirector Sugerido 4
1.4 Tribunal Designado 4
1.5 Aprobación 4
1.6 Línea de Investigación 4
1.6.1 Código UNESCO 4
1.6.2 Tipo de Trabajo 4
1.7 Área de Estudio 4
1.8 Título Propuesto 4
1.9 Subtitulo 4
1.10 Estado del Proyecto 5
2 Contenido 5
2.1 Motivación de la Investigación 5
2.2 Problemática 5
2.3 Resumen 6
2.4 Indagación exploratoria y base conceptual: 7
2.5 Objetivo General 9
2.6 Objetivos Específicos 9
2.7 Metodología 9
2.8 Alcances y Resultados esperados 11
2.9 Supuestos y Riegos 12
2.10 Presupuesto 13
2.11 Financiamiento 13
2.12 Esquema Tentativo 14
2.13 Cronograma 15
2.14 Referencias 20
2.15 Firma de Responsabilidad (Estudiantes) 22
2.16 Firma de Responsabilidad (Director Sugerido) 22
2.17 Fecha de Entrega 22

Edición autorizada de 10.000 ejemplares
Del 778.501 al 788.500
No 0783849

DATOS GENERALES

1.1 DATOS ESTUDIANTE

Jorge Luis Calle Castro

Código: 50468

Celular: 0984885230

Teléfono: 2868975

Correo Electrónico: jlcc_18@hotmail.com

Milton Fabián Largo Anguisaca

Código: 50469

Celular: 0984166182

Teléfono: 2863327

Correo Electrónico: fabianlargo1@hotmail.com

1.2 DIRECTOR SUGERIDO

Ingeniero Diego Pacheco

Celular: 0987861052

Correo Electrónico: dpacheco@uazuay.edu.ec

1.3 CODIRECTOR SUGERIDO

UNIVERSIDAD DEL AZUAY

1.4 TRIBUNAL DESIGNADO

1.5 APROBACIÓN

1.6 LÍNEA DE INVESTIGACIÓN

1.6.1 Código UNESCO

Línea: 1203.07 Informática

1.6.2 Tipo de Trabajo

Proyecto de investigación y desarrollo

1.7 ÁREA DE ESTUDIO

Las áreas que se aplicarán para el desarrollo de la tesis serán:

- Bases de Datos
- Programación Web
- Programación Móvil
- Diseño de Interfaz
- Análisis de Sistemas
- Calidad de Software

1.8 TÍTULO PROPUESTO

Guía turística de la ciudad de Cuenca para dispositivos móviles Android.

1.9 SUBTITULO

Digitalización de lugares de interés con realidad aumentada[1]

[1] Realidad Aumentada: "tecnología que mezcla entorno real (lo que nosotros/as podemos ver) y virtual (existente sólo aparentemente sin ser real). Es un modo de poder interactuar con la realidad física en tiempo real." (KZgunea, 2014)

Edición autorizada de 10.000 ejemplares
Del 778 501 al 788 500
Nº 0783850

1.10 Estado del Proyecto

Nuevo proyecto en la cual se desarrollara una aplicación android con realidad aumentada para la ciudad de Cuenca. Además de una página web en donde los locales (turísticos, comerciales, etc) ingresaran la información que deseen mostrar a los turistas.

Existen ya aplicaciones turísticas tales como TripAdvisor, Despegar, Booking, Google maps, Google Earth, etc. Pero están solo nos dan una visión general de la ciudad. Existe una página web llamada **ubicacuenca** (www.ubicacuenca.com), que muestra detalles de: promociones, eventos, hoteles etc. Aunque esta carece de una herramienta móvil, fuese interesante integrar la aplicación propuesta en esta plataforma web, ya que facilitaría el poder acceder a datos que necesita la aplicación.

2 Contenido

2.1 Motivación de la Investigación

Actualmente se observa una gran concurrencia de turistas que visitan la ciudad de Cuenca de los cuales la gran mayoría utilizan aplicaciones extranjeras, teniendo como resultado una información desactualizada, ya que los datos de los establecimientos no están en una frecuente actualización. Adicionalmente estas aplicaciones para su correcto funcionamiento, necesitan de una conexión a internet para utilizarlas, siendo esta una gran problemática, la cual tendría solución si se trabaja con mapas e información offline. Así, no se limitaría el uso de la aplicación. Siendo esta la razón y la justificación para el desarrollo de la aplicación para el trabajo de titulación.

2.2 Problemática

Aún se ve en la ciudad a turistas que consultan su mapa para visitar los puntos de interés, siendo esta una forma incómoda ya que tienen que consultar grandes papeles, información desactualizada y existen dificultades en su manejo.

Muchos países invierten en estos tipos de aplicaciones, con la finalidad de que el turista tenga a la mano herramientas tecnológicas de vanguardia, con las que se pueda conocer la ciudad, tales como es el caso de "**Chile Travel**" que fue lanzado hace aproximado 3 años atrás. (Tiempo, 2013).

UNIVERSIDAD DEL AZUAY

Otra aplicación interesante fue la que la Galicia-España, llamada **"Turismo Galicia"** en la que integran realidad aumentada, dando facilidades al turista de encontrar lugares interesantes que se encuentren a su alrededor. (PRESS, 2016).

Y la última novedad que nuestra ciudad presentó fue "Cuenca en tus manos" la aplicación es de una franquicia española cuyo nombre original es "Ciudad en tus manos" y que actualmente ya están en Guayaquil, Quito, Manabí, Santa Elena y Cuenca. Se la puede encontrar como una aplicación diferente para cada ciudad, siendo esto un problema, al ser una persona que viaje mucho necesitaría descargar la aplicación para cada ciudad a la que visita.

2.3 RESUMEN

Se va a desarrollar una aplicación para Android para el uso de los turistas en la ciudad de Cuenca, tomando como área de análisis el centro histórico la cual está limitada entre las calles Mariano Cueva, Gaspar Sangurima, Juan Montalvo y Presidente Córdova. Para complementar el proyecto se desarrollara una página web la cual será utilizada para que la aplicación consulte los datos. El registro de todos los locales que quieran colocar su información básica (Nombre del local, descripción, dirección, teléfono, fotos, hora de atención etc.). De esta manera el turista podrá conocer sobre el establecimiento de una manera global.

Se implementará una tecnología que no es nueva pero se está convirtiendo en tendencia para publicidad, juegos, educación etc., esta es la realidad aumentada (RA). Se utilizará con la finalidad de indicar al turista un detalle del establecimiento con la cámara de dispositivo móvil para lo cual debe estar a unos pocos metros del establecimiento. Se presentará simulando un cartel informativo en el que visualizará la información que fue ingresada desde la web de registro de los locales.

No siempre se tiene conexión a internet para lo cual se implementará el uso de mapas offline y online [2]de la ciudad de Cuenca.

[2] Mapas online: es una opción de usar mapas en tiempo real, para lo cual se necesita de una conexión a internet.

Edición autorizada de 10.000 ejemplares Del 778.501 al 788.500 Nº 0783851

2.4 INDAGACIÓN EXPLORATORIA Y BASE CONCEPTUAL:

En el mundo la tecnología avanza muy rápido, dando un área de trabajo con muchas opciones al éxito, una de la cuales son las aplicaciones móviles. En la actualidad son un campo de trabajo con amplia demanda las cuales cuentan con varias categorías que se puede explotar.

- Aplicaciones de Compras

Desde hace varios años las aplicaciones móviles han jugado un gran papel para el desarrollo y evolución de las compras, teniendo un gran nicho de mercado por la facilidad de acceso a lugares muy lejanos como también cercanos de una manera rápida y ágil (San-Martín, 2012)

Ejemplos:

OLX, MercadoLibre, Amazon, EBay, Patio Tuerca, AliExpress.

- Aplicaciones de Comunicación

Juega un rol muy importante ya que gracias a estas aplicaciones nos podemos comunicar con amigos, familia etc., en muchas de estas de forma gratuita siendo un gran ahorro económico y una gran facilidad de contacto con nuestros seres queridos.

Ejemplos

WhatsApp, Skype, Facebook Messenger, etc.

- Aplicaciones de deportivas

Muchas personas son fanáticas a los deportes, así que con este tipo de aplicaciones se les permite estar al día en todos los deportes de su interés.

Ejemplos

DIRECTV Sport, Liga de Campeones, Fútbol Ecuatoriano, Fox Sports, Etc.

- Aplicaciones de educación

En la actualidad la educación ha logrado grandes cambios debido al crecimiento de la tecnología, lo cual ha permitido mayor acceso a la información, teniendo una gran

UNIVERSIDAD DEL AZUAY

facilidad de búsquedas de la misma. Estas aplicaciones se caracterizan por el hecho que no se necesita estar en un lugar ni en un momento predeterminado para poder estudiar o aprender algo nuevo. (Valero, 2012)

Ejemplos

Doulingo, Edmorro, Lumosity, NeuroNation, etc.

- Aplicaciones de entretenimiento

En el mercado el área del entretenimiento es muy extensa, se tiene un sin número de aplicaciones para el ocio que pueden ser pagadas o gratis.

Ejemplos

NetFlix, Youtube, Spotify, etc.

- Aplicaciones de viajes y guías

En la época actual el internet móvil tiene una cobertura del 95% mundial, según informes de "La unión internacional de telecomunicaciones (ITU)" y el 64% de acceso a redes de tercera generación. (LUIS, 2015) Con estas estadísticas se puede tener grandes oportunidades en el área del turismo.

Este es el punto que se va a explorar para realizar una aplicación para la ciudad de Cuenca.

Ejemplos

Waze, Despegar, Maps, TripAdvisor, Guía Quito, etc.

Y muchas más aplicaciones que existen hoy en día en el mercado.

- Aplicaciones con realidad aumentada

Esta tecnología consiste en fusionar objetos (mundo real y mundo virtual) dando de esta manera la impresión de ver objetos que no existen en el mundo real a través de una aplicación (Lara, 2004). No es un tema nuevo, pero este año se lo explotó mucho en varias áreas siendo una de las más sonadas el juego de Pokemon Go. Pero existen muchas más aplicaciones que son aplicadas en distintas áreas.

Edición autorizada de 10.000 ejemplares
Del 778.501 al 788.500
Nº 0783852

Ejemplos

Layar (Publicidad), Aurasma (Entretenimiento), Wikitude (Turismo)

2.5 Objetivo General

Realizar una aplicación para móviles con sistema Android que sirva como guía turística para los visitantes de Cuenca.

2.6 Objetivos Específicos

1. Analizar información sobre aplicaciones turísticas en Android, con el fin de determinar las características esenciales de una app turística.
2. Recopilar información brindada por los dueños o responsables de los establecimientos para una guía turística de Cuenca.
3. Analizar la información brindada por los dueños o responsables de los establecimientos e información de los usuarios para la creación de la base de datos.
4. Estructurar la base de datos a usar, tanto para la parte web como para los dispositivos móviles.
5. Desarrollar una página web y una aplicación Android.
6. Implementar el módulo de realidad aumentada
7. Evaluar la aplicación con usuarios reales por medio de una etnografía enfocada.

2.7 Metodología

Para la elaboración de este trabajo se propone la ejecución de los siguientes pasos:

1. Análisis de la información investigada sobre aplicaciones turísticas en Android, con el fin de determinar las características esenciales de una app turística.
    a. Se presentará un prototipo con toda la funcionalidad que tendrá la aplicación (se tendrá una aplicación con datos de ejemplo)
2. Análisis de la información brindada por los dueños o responsables de los establecimientos para una guía turística de Cuenca.
    a. Todos los datos receptados serán tabulados, estudiados y graficados para una mejor comprensión de lo buscan los visitantes.

UNIVERSIDAD DEL AZUAY

3. Analizar la información brindada por los dueños o responsables de los establecimientos e información de los usuarios para la creación de la base de datos.
4. Estructurar y crear la base de datos (SQL). Se usará Workbench para el modelado y Microsoft SQL Server 2014 para la administración.
    a. Estructurar una base de datos para la parte web ya que esta se usará para alimentar al dispositivo móvil.
5. Estructurar y crear la base de datos (Sqlite), se usará SQL manager extensión para Firefox.
    a. Estructurar una base de datos interna en el móvil para guardar información básica del usuario. De esta manera no dependemos estrictamente de la conexión a internet y evitamos gastos de datos innecesarios.
6. Realizar la programación de la página web usando PHP, ccs3, JavaScript, Ajax. Todo se lo realizará con NetBeans Id 8.1
    a. Se analizará las ventajas y desventajas del uso de un framework [3] web para el desarrollo.
7. Realizar el desarrollo de la aplicación Android usando Android Studio, con base de datos local (Sqlite) y remota (MySql).
    a. Integración de API para manejo de mapas en la aplicación móvil
    b. Trazado de ruta mediante el uso de API.
8. Realizar el desarrollo de realidad aumentada usando el SDK que más se adapte a las necesidades de la aplicación
    a. Se analizará una serie de SDK que existen en el mercado y escogiendo la que mejor se adapte al proyecto a realizarse.
9. Evaluar la aplicación con usuarios reales por medio de una etnografía enfocada
    a. En este punto se visualizará los resultados con gráficos para una comprensión fácil. Esta evaluación es muy importante, ya que gracias a la información recolectada se podrá tener una visión amplia de la dificultad del uso de la aplicación.

[3] Framework.- Un Framework ofrece componentes como una librería, pero además provee de plantillas o esqueletos que definen el funcionamiento de las aplicaciones.

Edición autorizada de 10.000 ejemplares Del 778.501 al 788.500 Nº 0783853

10. Se depurará la aplicación en base a los resultados de la evaluación de campo.

2.8 Alcances y Resultados esperados

Se tomarán en cuenta las experiencias de los turistas que ya tienen un tiempo considerable en la ciudad para conocer sobre su trato y apego de los ciudadanos (que les agrada, que no les agrada).

De esta manera se podrá desarrollar la aplicación con las siguientes características:

1. Trazar una ruta óptima para guiar al usuario hacia su destino, para lo cual se utilizará una API que nos permita realizar este ruteo.
2. La aplicación será capaz de ayudar a descubrir que establecimientos puedo visitar mediante el uso de la realidad aumentada.
3. Mantener un historial de rutas visitadas por el turista.
4. El mapa tendrá la opción de usarse offline y online cuando encuentre conectividad.
5. Se podrá visualizar un listado e información de los establecimientos.
6. La información de los móviles se actualizará en tiempo real (dependiendo la conectividad).
7. La página será usada para ingresar información de los locales que deseen mostrar sus datos en la aplicación, para lo cual deberán realizar un registro en la página con los siguientes datos:
    a. ***Nombre del local.***- nombre comercial que se mostrará en la aplicación.
    b. ***Nombre del dueño.***- para contactos
    c. ***Dirección.***- podrá seleccionar la ubicación mediante un mapa que se integrara en el registro de establecimientos.
    d. ***Horas de atención.***- horas en las que el local atiende.
    e. ***Logo.***- logo del establecimiento.
    f. ***Teléfonos.***- números a los que se puede comunicar.
    g. ***Correo electrónico.***- se mostrar en la aplicación para cualquier duda o sugerencia.

2.9 Supuestos y Riegos

UNIVERSIDAD DEL AZUAY

Supuestos:

Supuestos	Alternativas de Solución
Contar con todo el material bibliográfico propuesto para el desarrollo del trabajo de investigación o el necesario.	Mantener una amplia bibliografía de los temas que se serán realizados antes de comenzar el proyecto.
Obtener artículos y estudios científicos previamente	Realizar búsquedas de información basándose en artículos científicos.
Contar con un plan de trabajos que nos permita cumplir con los objetivos planteados.	Establecer un plan de trabajos con tiempos que permitan mantener una holgura hasta la fecha de conclusión del proyecto.
Contar con la información de los locales comerciales del centro histórico.	Realizar una investigación con el fin de obtener la información requerida, para lo cual se puede acudir a la fundación turismo de cuenca.

Riegos:

Riesgos	Alternativas de Solución
El tiempo estimado para el desarrollo no sea el suficiente	Realizar un análisis minucioso en donde se determine los temas de investigación de cada capítulo, para poder determinar tiempos de avance para cada uno de ellos.
No encontrar el material bibliográfico necesario para el desarrollo	Asegurar mediante una revisión preliminar que exista información acerca de los temas que se requieren.
Perder Material Digital por causas de falla del computador de trabajo.	Mantener un plan de respaldo en la nube y en medios externos, los cuales se realizaran a diario.

Edición autorizada de 10.000 ejemplares
Del 778.501 al 788.500
Nº 0783854

2.10 PRESUPUESTO

Rubro-Denominación	Costo USD	Justificación
Útiles de Oficina	$200	- Hojas de Papel - Material de Escritorio - Copias - Empastado - Varios
Derechos Universitarios	$300	- Derecho de Tesis - Otros
Internet	$120	- Pago mensual de internet que se usara por la investigación
Imprevistos	$100	- Gastos adicionales que se presenten de última hora
Material Bibliográfico	$250	- Compra de libro, revistas tanto digitales como físicos para elaborar la investigación.
Viáticos	$200	- Combustible - Alimentación - Viajes
Trabajo humano	$3200	- Aporte intelectual
Total	**$3470**	

2.11 FINANCIAMIENTO

Financiamiento propio

2.12 Esquema Tentativo

UNIVERSIDAD DEL AZUAY

Portada

Dedicatoria

Agradecimientos

Abstract

Índice de Contenidos

Resumen

Introducción

Objetivo General

Objetivos Específicos

Capítulo 1 Análisis de los dispositivos móviles

Versiones de sistemas operativos para móviles

Android

IOS

Windows Phone

Características de los dispositivos móviles

Idioma

Costo

Uso

Cantidad de móviles

Justificación del uso de la plataforma seleccionada

Características de teléfonos

Capítulo 2 Análisis de herramientas y estándares

Mapas Offline

Mapas Online

Realidad Aumentada (AR)

Algoritmos de ruteo

Capítulo 3 Desarrollo de sistema

Crear diseño de la página web

Crear base de datos para la página web

Crear diseño para el dispositivo móvil

Edición autorizada de 10.000 ejemplares
Del 778.501 al 788.500
No 0783855

Crear base de todos para el dispositivo móvil

Programar página web

Programar dispositivo móvil

Integración de API para manejo de mapas

Trazado de rutas mediante API

Capítulo 4 Resultados

Capítulo 5 Discusión

Conclusiones

Recomendaciones

Referencias Bibliográficas

Glosario

Anexos

2.13 CRONOGRAMA

Objetivo Específico	Actividad	Resultado Esperado	Tiempo(Semanas)
Analizar la información sobre aplicaciones turísticas en Android, con el fin de determinar las características esenciales de una app turística.	Determinar el estado del arte. Revisión bibliográfica	Capítulos introductorios y códigos fuentes reutilizables de otras app.	1
Analizar información brindada por los dueños o responsables de los establecimientos para una guía turística de Cuenca.	Analizar información brindada por los dueños o responsables de los establecimientos para una guía turística de Cuenca.	Realizar un modelo con la que se va a distribuir la información proporcionado por los establecimientos	1
Analizar la información brindada por los dueños o responsables de los establecimientos e información de los usuarios para la creación de la base de datos.	Análisis para la creación de la base de datos a partir de los datos obtenidos en las encuestas	Tener un visión amplia de los campos, relaciones, tablas etc., a crear (Modelo entidad/relación)	1
Analizar información brindada por los dueños o responsables de los establecimientos e información de los usuarios para la creación de la base de datos para el dispositivo móvil.	Análisis para la creación de la base de datos a partir de los datos obtenidos en las encuestas.	Tener una visión amplia de los campos, relaciones, tablas etc., a crear.	1

UNIVERSIDAD DEL AZUAY

Especie valorada de 10000 pesetas
0783856

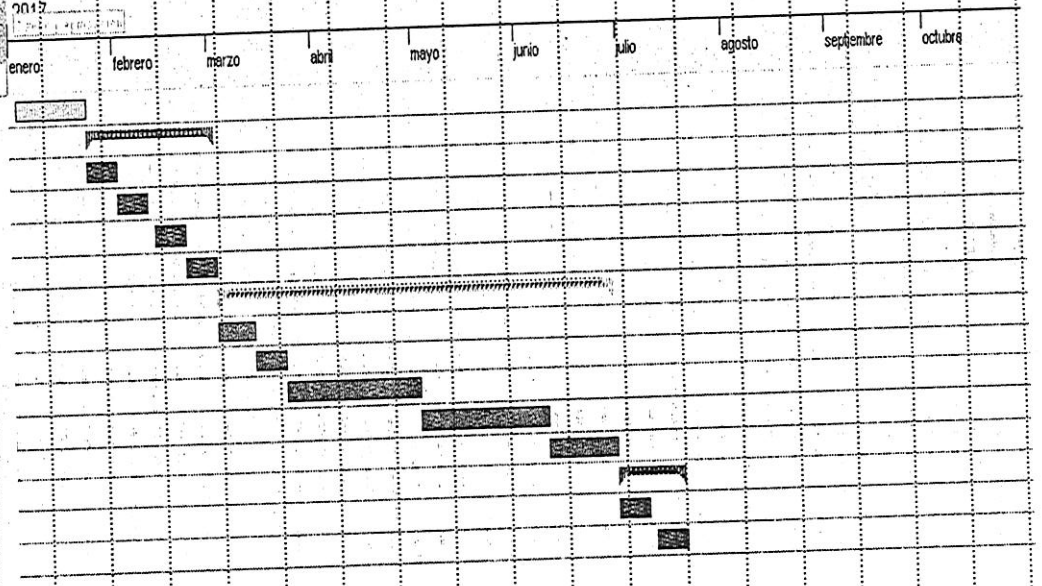
Estructurar y crear la base de datos (SQL), se usará Workbench para el modelado y Microsoft SQL Server 2014 para la administración.	Crear la base de datos en base al análisis previo.	Generar la base de datos con la que se va a trabajar para la creación de la página web y datos que se va a consumir desde el móvil.	1
Estructurar y crear la base de datos (Sqlite), se usará SQL manager extensión para Firefox.	Crear la base de datos en base al análisis previo	Generar la base de datos con la que se va a trabajar con el dispositivo móvil.	1
Realizar la programación de la página web usando php, HTML5, ccs3. Todo se lo realizará con NetBeans Id 8.1.	Desarrollar la página web para el registro de establecimientos y hospedar en un host.	Se tendrá la página web lista para que se pueda registrar los establecimientos y que pueda consumir la aplicación mediante un web service.	4
Realizar el desarrollo de la aplicación Android usando Android Studio, con base de datos local (Sqlite) y remota (MySql).	Desarrollar la estructura y codificación del sistema móvil para turistas. Esta tarea se realizará en conjunto con la página web.	Obtener la aplicación.	4
Realizar el desarrollo de realidad aumentada usando el SDK que más se	Se validará la mejor alternativa y se implementará en el proyecto Android	Se tendrá implementado y consumiendo los datos de la aplicación para mostrar la	2

adapte a las necesidades de la aplicación.		información necesaria en la aplicación.	
Evaluará la aplicación con usuarios reales por medio de una etnografía enfocada.	Se realizará una encuesta a turistas en la ciudad de Cuenca.	Se analizará los resultados para determinar las funcionalidades que tendrá la aplicación de acuerdo a las encuestas realizadas.	1
Se depurará la aplicación con base a los resultados de la evaluación en campo.	Se empieza el desarrollo de la aplicación en conjunto con los datos obtenidos de la información recolectada.	Aplicación culminada en base a las encuestas realizadas.	1
Total en semanas			18

UNIVERSIDAD DEL AZUAY

Ec. Guillermo [illegible] Fernández
No. 0783857

Nombre	Fecha de inicio	Fecha de fin
Diseño y aprobación	3/01/17	23/01/17
Analisis de información	24/01/17	2/03/17
Analizar la información sobre aplicaciones turísticas en Android	24/01/17	1/02/17
Analizar información brindada por los dueños o responsables de los establecimientos	2/02/17	10/02/17
Analizar información para bases de datos	13/02/17	21/02/17
Analizar información para creación de aplicación	22/02/17	2/03/17
Diseño y desarrollo de aplicación	3/03/17	28/06/17
Estructurar y crear la base de datos (SQL).	3/03/17	13/03/17
Estructurar y crear la base de datos (Sqlite).	14/03/17	22/03/17
Realizar la programación de la página web	23/03/17	1/05/17
Realizar el desarrollo de la aplicación	2/05/17	8/06/17
Realizar el desarrollo de realidad aumentada	9/06/17	28/06/17
Evaluación y depuración de resultados	29/06/17	18/07/17
Evaluará la aplicación con usuarios reales	29/06/17	7/07/17
Depuración de la aplicación	10/07/17	18/07/17



2.14 REFERENCIAS

ACISSI. (s.f.). *Seguridad Informática Ethical Hacking.* Barcelona: Epsilon.

Astudillo, K. (2013). *HACKING ÉTICO 101 – Cómo hackear profesionalmente en 21 días o menos!* Guayaquil.

Benítez, L. F. (12 de 2014). LA GESTIÓN DE RELACIÓN ENTRE LAS EMPRESAS Y TURISTAS A TRAVÉS DE LAS APLICACIONES MÓVILES COMO HERRAMIENTA DE MARKETING DE LOS DESTINOS TURÍSTICOS. Obtenido de http://www.eumed.net: http://www.eumed.net/rev/turydes/

Cabal, P. T. (2014). Las App como instrumento de información e información turística.

Cavia, J. F. (2013). Aplicaciones móviles de los destinos turísticos españoles, 4-14. Obtenido de itunews.

Cavia, J. F. (10 de 2013). *www.marcasturisticas.org.* Obtenido de www.marcasturisticas.org: www.marcasturisticas.org

EcuRed. (s.f.). *ecured.* Recuperado el 2013 de 06 de 25, de http://www.ecured.cu/index.php/Ataque_inform%C3%A1tico

Evron, R. V. (2006). *DNS Amplification Attacks.*

Evron, R. V. (17 de Marzo de 2006). *Isotf.* Recuperado el 2013 de 06 de 25, de http://www.isotf.org/news/DNS-Amplification-Attacks.pdf

GNU. (s.f.). *GNU Operation System.* Recuperado el 25 de 06 de 2013, de http://www.gnu.org/licenses/gpl-2.0.html

Hender, C. (2 de 2011). SISTEMAS DE INFORMACIÓN TURÍSTICOS.

James Michael Steward, G. K. (2011). *Global Knowledge.* Recuperado el 2013 de 06 de 25, de http://www.globalknowledge.nl/content/files/documents/224536/Security-White-Paper-10-Ways-Hackers-Breach-Security

KZgunea. (12 de 06 de 2014). *http://kzgunea.blog.euskadi.eus/.* Obtenido de http://kzgunea.blog.euskadi.eus/: http://kzgunea.blog.euskadi.eus/blog/2014/06/12/concepto-realidad-aumentada/

Lara, B. L. (2004). LA REALIDAD AUMENTADA: UNA TECNOLOGÍA EN ESPERA DE USUARIOS. *Revista Digital Universitaria*, 1-5.

Long, J. (2008). *Google Hacking.* Syngress Publishing.

Luis J Callarisa Fiol, J. S. (2012). LA IMPORTANCIA DE LAS COMUNIDADES VIRTUALES PARA EL ANALISIS DEL VALOR DE MARCA. EL CASO DE TRIPADVISOR EN HONG KONG Y PARIS*. Obtenido de www.papersdeturisme.gva.es: www.papersdeturisme.gva.es

LUIS, Á. J. (2 de 1 de 2015). *http://www.elmundo.es/.* Obtenido de http://www.elmundo.es/: http://www.elmundo.es/tecnologia/2015/12/01/565d834b46163fc6438b4623.html

María del Pilar Ramos, A. G. (2011). *Seguridad Informatica.* Madrid: Paraninfo.

Edición autorizada de 10.000 ejemplares Nº 0783858

Murillo, J. C. (212). *Administración avanzada de redes.*

PRESS, E. (8 de 1 de 2016). *La nueva aplicación móvil de realidad aumentada 'Turismo de Galicia', una guía para los visitantes.* Obtenido de http://www.20minutos.es/: http://www.20minutos.es/noticia/2644374/0/nueva-aplicacion-movil-realidad-aumentada-turismo-galicia-guia-para-visitantes/

RUBIO, J. E. (2014). APP PARA PROMOCIÓN DEL PATRIMONIO CULTURAL DE LA UTP SOBRE ANDROID APLICANDO REALIDAD AUMENTADA.

San-Martín, S. (2012). La venta por teléfono desde el punto de vista de las empresas españolas. *Universia BussinesReview*, 124-131.

Tiempo, R. E. (31 de 8 de 2013). *Aplicación para móviles con información turística.* Obtenido de http://www.eltiempo.com.ec/: http://www.eltiempo.com.ec/noticias/novedades/23/319224/aplicacion-para-moviles-con-informacion-turistica

Tony Bradley, C. I. (s.f.). Recuperado el 2013 de 06 de 25, de http://netsecurity.about.com/od/newsandeditorial1/a/aazeroday.htm

Tori, C. (2008). *Hacking Ético.* Rosario.

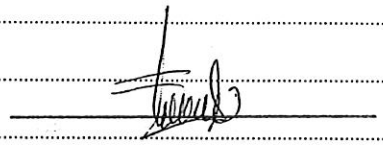
Valero, C. C. (2012). Tendencias actuales en el uso de dispositivos móviles en educación. *La Educ@ción Digital Magazine N 147*, 1-5.

W3C. (s.f.). *http://www.w3c.es/*. Obtenido de http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb

2.15 Firma de Responsabilidad (Estudiantes)

UNIVERSIDAD DEL AZUAY

Jorge Luis Calle Castro



Milton Fabián Largo Angüisaca

2.16 Firma de Responsabilidad (Director Sugerido)

Ing. Diego Pacheco

2.17 Fecha de Entrega

01 de febrero de 2017

Edición autorizada de 10.000 ejemplares
Del 778.501 al 788.500
Nº 0783861