



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
SISTEMAS E INFORMÁTICA

AUTOR: VERA YÁNEZ, DANIEL AUGUSTO

TEMA: DESARROLLO DE UNA APLICACIÓN MÓVIL PARA APOYAR AL  
TURISMO DEL CENTRO HISTÓRICO DE QUITO, UTILIZANDO  
REALIDAD AUMENTADA Y GEOLOCALIZACIÓN, PARA LA EMPRESA  
VLBS CIA.LTDA

DIRECTOR: ING. DÍAZ, OSWALDO  
CODIRECTOR: ING. MARCILLO, DIEGO

SANGOLQUÍ, FEBRERO 2014

**CERTIFICACIÓN**

Certificamos que el presente trabajo titulado “DESARROLLO DE UNA APLICACIÓN MÓVIL PARA APOYAR AL TURISMO DEL CENTRO HISTÓRICO DE QUITO, UTILIZANDO REALIDAD AUMENTADA Y GEOLOCALIZACIÓN, PARA LA EMPRESA VLBS CIA.LTDA” fue realizado en su totalidad por el Sr. Daniel Augusto Vera Yáñez, como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

FEBRERO DE 2014

---

**ING. OSWALDO DÍAZ**  
**DIRECTOR**

---

**ING. DIEGO MARCILLO**  
**CODIRECTOR**

**DECLARACIÓN**

Yo, Daniel Augusto Vera Yáñez, declaro que el presente trabajo es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación personal y que he consultado las referencias bibliográficas que se incluyen en el documento.

La Universidad de las Fuerzas Armadas - ESPE puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual por su reglamento y por la normativa institucional vigente.

Sangolquí, Febrero de 2014

---

**Daniel Augusto Vera Yáñez**

**AUTORIZACIÓN**

Yo, Daniel Augusto Vera Yáñez, autorizo a la Universidad de las Fuerzas Armadas - ESPE a que publique en el repositorio digital de la biblioteca Alejandro Segovia el presente proyecto de tesis, así como también los materiales y documentos relacionados a la misma.

Sangolquí, Febrero de 2014

---

**Daniel Augusto Vera Yáñez**



**DEDICATORIA**

Con todo mi cariño y mi amor para las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme con valores, principios, empeño, carácter, perseverancia y coraje para conseguir mis objetivos y darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban.

A mi familia por siempre mi corazón y mi agradecimiento.

**DANIEL AUGUSTO VERA YÁNEZ**

## **AGRADECIMIENTO**

Al finalizar un trabajo tan duro como el desarrollo de esta tesis, un análisis objetivo muestra inmediatamente que la magnitud de este aporte hubiese sido imposible sin la participación de personas e instituciones que han facilitado las cosas para que este trabajo llegue a un feliz término. Por ello, es para mí un verdadero placer agradecer de manera especial y sincera al Ing. Oswaldo Díaz por aceptarme para realizar esta tesis bajo su dirección. Su apoyo y confianza en mi trabajo y su capacidad para guiar mis ideas, siempre enmarcadas en su orientación y rigurosidad, han sido la clave del buen trabajo que se ha realizado juntos, el cual no se puede concebir sin su siempre oportuna participación.

Quiero expresar también mi más sincero agradecimiento al Ing. Diego Marcillo por su importante aporte y participación activa en el desarrollo de esta tesis, no cabe duda que su participación ha enriquecido el trabajo realizado.

A mis padres que participaron e hicieron posible este proyecto, muchas gracias por su apoyo y enseñanza.

**DANIEL AUGUSTO VERA YÁNEZ**

## TABLA DE CONTENIDO

RESUMEN .....	X
GLOSARIO DE TÉRMINOS .....	74
BIBLIOGRAFÍA .....	76
ANEXO A .....	78
ANEXO B .....	81
CAPÍTULO 1 INTRODUCCIÓN .....	1
1.1 TEMA.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA .....	1
1.3 ANTECEDENTES .....	2
1.4 JUSTIFICACIÓN E IMPORTANCIA .....	4
1.5 OBJETIVOS	
1.5.1 OBJETIVO GENERAL .....	5
1.5.2 OBJETIVOS ESPECÍFICOS.....	5
1.6 ALCANCE.....	6
1.7 METODOLOGÍA	
1.7.1 FASES OOHDM .....	7
1.7.2 ANÁLISIS DE REQUERIMIENTOS .....	7
1.7.3 REQUERIMIENTOS FUNCIONALES.....	7
1.7.4 REQUERIMIENTOS NO FUNCIONALES .....	8
1.7.5 DISEÑO CONCEPTUAL.....	8
1.7.6 DISEÑO NAVEGACIONAL.....	8
1.7.7 DISEÑO DE LA INTERFAZ ABSTRACTA.....	8
1.7.8 IMPLEMENTACIÓN.....	9
1.8 RECURSOS .....	10
1.9 FACTIBILIDAD	
1.9.1 FACTIBILIDAD TÉCNICA.....	10
1.9.2 FACTIBILIDAD TECNOLÓGICA .....	11
1.9.3 FACTIBILIDAD OPERATIVA .....	11
1.9.4 FACTIBILIDAD OPERACIONAL.....	11
1.9.5 FACTIBILIDAD ECONÓMICA .....	11
1.9.6 FACTIBILIDAD LEGAL .....	12
CAPÍTULO 2 MARCO TEÓRICO .....	14
2.1 Realidad Aumentada .....	14
2.1.1 ¿Cómo funciona la Realidad Aumentada? .....	14
2.1.2 Áreas de aplicación de la realidad aumentada .....	16
2.2 La Geolocalización, Coordenadas hacia el Éxito .....	17
2.3 Fundamentación	
2.3.1 Aspectos Generales	
2.3.1.1 Dispositivos Móviles (Hardware).....	18
2.3.1.2 Aplicaciones Móviles (Software) .....	20

2.3.1.2 iOS .....	21
2.3.1.3 Android .....	22
2.3.1.4 Windows Phone .....	22
2.3.2 Lenguaje de programación C#.....	23
2.3.3 Xcode (IDE) .....	25
2.3.4 iOS SDK.....	28
2.3.5 Unity (Game Engine) .....	30
2.3.6 Vuforia SDK .....	31
2.4 Metodología de Diseño de Hipermedia Orientada a Objetos (OOHDM)	
2.4.1 Historia .....	31
2.4.2 Definición .....	33
2.4.3 Características Fundamentales .....	33
2.4.4 Etapas ó Fases .....	34
2.4.5 ¿Por qué usar OOHDM en el desarrollo de una aplicación móvil?.....	35
2.5 Lenguaje Unificado de Modelado (UML) .....	36
2.6 Modelo Vista Controlador (MVC).....	36
2.6.1 Descripción del patrón .....	37
2.7 Adelantos Tecnológicos de la RA	
2.7.1 Google Glass .....	39
2.7.2 Aurasma.....	40
2.7.3 Par Works .....	41
<b>CAPÍTULO 3 ANÁLISIS Y DISEÑO DE LA APLICACIÓN MÓVIL .....</b>	<b>42</b>
3.1 Especificación de requerimientos	
3.1.1 Requerimientos funcionales.....	42
3.1.2 Requerimientos no funcionales.....	42
3.2 Introducción	
3.2.1 Propósito.....	43
3.2.2 Definiciones, acrónimos y abreviaturas .....	43
3.3 Identificación de Roles y Tareas .....	44
3.4 Especificación de Escenarios .....	45
3.5 Especificación de uso por Actor .....	45
3.6 Diseño Conceptual	
3.6.1 Diagramas de Secuencia .....	49
3.6.2 Diagrama de Clases .....	52
3.6.3 Diseño Navegacional .....	53
3.6.4 Diseño de Interfaz Abstracta.....	53
3.6.5 Arquitectura del Sistema .....	55
<b>CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBAS DE LA APLICACIÓN MÓVIL .....</b>	<b>56</b>
4.1 Unity Game Engine	
4.1.1 Instalación.....	57
4.1.2 Interfaz de usuario de Unity .....	59

	IX
4.1.3 Importar paquetes Unity a un proyecto .....	60
4.2 Vuforia SDK .....	61
4.2.1 Instalación de Vuforia .....	63
4.3 Construcción de la aplicación móvil en Unity .....	64
4.3.1 Consideraciones de la programación de RA utilizando Vuforia ....	64
4.3.2 Consideraciones especiales de la geolocalización	
4.3.2.2 Ruta para llegar a un punto geográfico específico .....	66
4.3.2 Consideraciones especiales de la geolocalización	
4.3.2.1 Distancia entre dos puntos geográficos .....	66
4.4 Pruebas de la Aplicación .....	67
4.5 Pruebas de Contenido .....	68
4.5.1 Pruebas de Usuario	
4.5.1.1 Pruebas de Interfaz .....	69
4.5.1.2 Pruebas de facilidad de uso .....	69
4.5.1.3 Pruebas de navegación .....	69
4.5.2 Pruebas de componentes	
4.5.2.1 Prueba de configuración .....	70
4.5.2.2 Prueba de seguridad .....	70
4.5.2.3 Prueba de desempeño .....	70
4.5.3 Deployment de la aplicación .....	70
CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES .....	72

### **Listado de Tablas**

Tabla 1. Fases de OODHM.....	35
Tabla 2. Ver ayuda aplicación.....	46
Tabla 3. Entrar configuración.....	46
Tabla 4. Iniciar aplicación.....	47
Tabla 5. Despliega información RA.....	48
Tabla 6. Despliega información Geolocalización.....	48
Tabla 7. Punto de interés cercano.....	49

## Índice de Figuras

Figura 1. Marcador para Realidad Aumentada. ....	15
Figura 2. Ejemplo de RA markerless. ....	15
Figura 3. iPhone 5s. ....	18
Figura 4. iPad Air. ....	19
Figura 5. Samsung Galaxy S4. ....	20
Figura 6. Samsung Galaxy Tab 3. ....	20
Figura 7. Verifica selección de proyectos en Xcode. ....	27
Figura 8. Árbol de un proyecto realizado en Xcode. ....	28
Figura 9. Ejemplo de Diagrama MVC. ....	37
Figura 10. Google Glass. ....	40
Figura 11. Aplicación Par Works. ....	41
Figura 12. Actores del sistema. ....	44
Figura 13. Casos de uso del Usuario. ....	45
Figura 14. Casos de uso para GUI. ....	47
Figura 15. Diagrama de secuencia – Ver ayuda aplicación. ....	49
Figura 16. Diagrama de secuencia – Entrar configuración. ....	50
Figura 17. Diagrama de secuencia – Iniciar aplicación. ....	50
Figura 18. Diagrama de secuencia – Despliega información RA. ....	51
Figura 19. Diagrama de secuencia – Despliega información geolocalización. ....	51
Figura 20. Diagrama de secuencia – Punto de interés cercano. ....	52
Figura 21. Diagrama de clases. ....	52
Figura 22. Diagrama de contexto. ....	53
Figura 23. Arquitectura del sistema. ....	55
Figura 24. Activación Unity. ....	57
Figura 25. Registrarse con la cuenta Unity. ....	58
Figura 26. Interfaz de Unity. ....	59
Figura 27. Asset Store. ....	60
Figura 28. Importar paquete Unity. ....	61
Figura 29. Image Target. ....	62
Figura 30. Frame Maker. ....	62
Figura 31. Página oficial de Vuforia. ....	63
Figura 32. La imagen representa un marcador de realidad aumentada. ....	64
Figura 33. Muestra las características del bloque central ESPE. ....	65
Figura 34. Muestra un objeto RA al ser reconocido su marcador. ....	65
Figura 35. Captura de la aplicación que muestra la distancia y el rumbo. ....	67
Figura 36. Proceso de prueba – diseño. ....	68
Figura 37. Proceso de prueba – pruebas. ....	68

## RESUMEN

En el presente artículo se muestra la aplicación de técnicas de realidad aumentada RA y la geolocalización para la creación de una aplicación móvil que ayude al turismo del Centro Histórico de Quito, llamado “QuitoRA”. Quito capital del Ecuador recibe anualmente 545.000 turistas, en donde surge la necesidad de una aplicación compatible con los dos sistemas operativos móviles más populares (iOS, Android) del mercado, la cual ayude al turista nacional y extranjero a identificar las principales iglesias y zonas históricas de la ciudad. “Quito RA” logra esto usando la cámara del dispositivo móvil y apuntando a una iglesia, el software reconoce la edificación y muestra información sobre el mismo, la aplicación utiliza la geolocalización para dirigir al usuario al siguiente punto reconocible. Como IDE de desarrollo se utilizó Unity Game Engine, el cual permitió desarrollar una aplicación que sea compatible con iOS y Android. Para la realidad aumentada se escogió Vuforia SDK, el cual permitió agregar las funcionalidades de la realidad aumentada en nuestra aplicación de una forma sencilla, al ser compatible con Unity. La metodología utilizada fue OOHDM (Object Oriented Hypermedia Design Methodology ) ya que es compatible con el desarrollo de aplicaciones móviles y da una buena base para el proceso.

**Palabras Clave:** Turismo, dispositivo móvil, Realidad Aumentada, Geolocalización, OOHDM.

**ABSTRACT**

This article demonstrates the use of augmented reality and geolocation for the creation of a mobile application that helps the tourists of Quito's Historic Center, called "QuitoRA". Ecuador's capital Quito receives 545.000 tourists annually where the need for an application that supports the two most popular mobile operating systems of the market (iOS, Android) which helps the domestic and foreign tourists to identify the main churches and historic areas of the city. Using the mobile device camera and point to a church, the software recognizes and displays information about the church, the application uses geolocation to direct the user to the next recognizable point. As development IDE Unity game engine which allowed us to develop an application that supports iOS and Android. Vuforia SDK added the augmented reality functionality in the application in a simple way as it is supported by Unity. The methodology used was as OOHDM that supports the development of mobile applications and provides a good basis for development.

**KeyWords:** Tourism, mobile device, Augmented Reality, Geolocalitation, OOHDM.



## **CAPÍTULO 1: GENERALIDADES**

### **1.1 TEMA**

Desarrollo de una Aplicación móvil para apoyar al Turismo del Centro Histórico de Quito, utilizando Realidad Aumentada y Geolocalización, para la Empresa VLBS CIA.LTDA.

### **1.2 PLANTEAMIENTO DEL PROBLEMA**

La ciudad de Quito tiene un gran número de visitantes extranjeros y nacionales a lo largo de todo el año con lo cual surge la necesidad de mostrar a los turistas los lugares importantes de la ciudad, como es el caso del Centro Histórico. Se puede ayudar con esta labor elaborando una aplicación móvil que informe sobre ciertos lugares representativos del centro histórico como son sus iglesias, con lo cual el visitante podrá aprender sobre la historia del lugar que está visitando.

En la actualidad existen varias aplicaciones turísticas que se apoyan en la realidad aumentada para mejorar la experiencia del usuario, una de estas aplicaciones móviles es Nearest Tube desarrollada por la Empresa Acrossair, la cual usa la cámara del dispositivo móvil y muestra unas etiquetas que indican al usuario los monumentos históricos, estaciones de metro, estaciones de bus, restaurantes más cercanos. Actualmente la aplicación es utilizada en las importantes ciudades el mundo New York, Paris, Londres, etc. (acrossair company, 2014)

### 1.3 ANTECEDENTES

La ciudad de San Francisco de Quito es la capital del Ecuador y la segunda ciudad más grande y más poblada del país. Se encuentra situada en las faldas del volcán activo Pichicha, aproximadamente en las coordenadas 0°13'23"S 78°30'45"O y su altitud promedio es de 2800 msnm. Quito fue la primera capital del mundo en ser declarada Patrimonio Cultural de la Humanidad, por parte de la UNESCO en 1978, reconocida como Capital Americana de Cultura 2011. (Peralta, 2003)

Quito tiene un gran número de turistas extranjeros y nacionales, los cuales llegan a ser 545.000 anualmente con una tasa de crecimiento anual de un 5%. Los meses con mayor afluencia de turistas extranjeros son junio, agosto, octubre y noviembre. El 32% de los visitantes (no residentes) son Norteamericanos, siendo el 27% de turistas de Estados Unidos. Las nacionalidades de la mayoría de turistas que recibe Quito son: estadounidense, colombiana, española, venezolana, canadiense, alemana, argentina y británica. Entre los lugares más visitados de la ciudad de Quito se encuentran el centro histórico con el 35.1% y el Teleférico con un 11.5%. El ingreso anual de divisas por gasto turístico en el 2012 fue de 289 millones de dólares (Empresa Pública Metropolitana Quito Turismo, 2013).

Sobre la base de las consideraciones anteriores, surge la necesidad de mostrar a los visitantes los lugares importantes de la ciudad, como es el caso del centro histórico. Se puede mejorar esta tarea desarrollando una aplicación móvil que informe sobre lugares representativos del centro histórico como son sus iglesias, con lo cual el turista podrá aprender sobre la historia del lugar que está visitando y en su propio idioma, principalmente en

inglés, español. La solución es muy viable ya que en los últimos años disponer de un dispositivo inteligente se ha convertido en una necesidad del día a día. Los sistemas operativos móviles iOS y Android, ocupan casi el 100% del mercado (Gottesman, 2013). Los aspectos antes mencionados facilitan que casi cualquier persona pueda acceder a la aplicación turística para el Centro Histórico.

La aplicación va a ser uso de realidad aumentada (RA). La RA es la combinación de una vista en tiempo real del mundo físico (real) con objetos generados por computador (sonido, gráfico, texto, video y información GPS), los cuales aumentan (suplementan) los elementos visuales de la vista. La tecnología RA da la oportunidad de generar contenido educativo basado en experiencias del mundo real, creando así una enseñanza más visual, facilitando el modo en el que las personas pueden aprender desde su dispositivo móvil (Haag, 2013).

Con el propósito de que el usuario de la aplicación tenga conocimiento de la localización geográfica de los puntos de interés de la aplicación se utilizó técnicas de geolocalización. La geolocalización se puede definir como un servicio que integra la localización o posición de un dispositivo móvil con otra información que da un valor añadido al usuario (Schiller & Voisard, 2004).

En un principio se pretendía que la aplicación sea exclusiva para la plataforma iOS, pero esto causaría que se pierdan todos los posibles usuarios Android, por lo cual se decidió utilizar el Game Engine Unity. Unity permite realizar una aplicación multiplataforma, lo que quiere decir que solo

se programa una vez y se puede compilar para una amplia variedad de plataformas entre ellas iOS y Android.

La aplicación utilizará la realidad aumentada para ayudar al usuario a reconocer las diferentes iglesias que existen en el Centro Histórico y conocer los principales detalles de estas. La geolocalización dará al turista una referencia de donde está el siguiente punto de interés, tomando en cuenta la posición en la que se encuentra en ese momento. Con esta aplicación el turista no requerirá de tener un guía de turismo para poder aprender sobre el lugar que está visitando.

En el presente documento se explica las herramientas utilizadas para desarrollar la aplicación así como las pruebas realizadas con la misma y los resultados obtenidos.

#### **1.4 JUSTIFICACIÓN E IMPORTANCIA**

El presente proyecto tiene como finalidad principal desarrollar una aplicación móvil que usando realidad aumentada se haga el reconocimiento de iglesias del Centro Histórico de Quito y por medio de la geolocalización identifique en donde se encuentra ubicado el usuario dentro del lugar y le presente información sobre el sitio turístico y la localización del usuario en la aplicación.

Al momento no existe ninguna aplicación similar en el mercado, la cual apoyada por el gran número de dispositivos móviles que existen actualmente, ayudará al turismo de Quito brindando información histórica a los usuarios de esta aplicación.

Según la Cámara de Turismo de Pichincha (CAPTUR), a Quito llega un 65% del turismo que ingresa al país y que permanece en promedio dos días,

para luego distribuirse a otros lugares. El plan estratégico para el 2020 busca duplicar el número de visitantes e incrementar entre un 20% y 30% el gasto del turista en la ciudad. Quito recibe en promedio 1,5 millones de turistas al año justificando el desarrollo de esta proyecto (Cámara Provincial Turismo Pichincha, 2013).

## **1.5 OBJETIVOS**

### **1.5.1 OBJETIVO GENERAL**

“Desarrollar una aplicación móvil interactiva de una guía turística del Centro Histórico de Quito utilizando realidad aumentada y geolocalización para difusión a los turistas”.

### **1.5.2 OBJETIVOS ESPECÍFICOS**

- ✓ Revisar los fundamentos de la realidad aumentada aplicada al reconocimiento de edificaciones para turismo en dispositivos móviles.
- ✓ Revisar las características básicas de la herramienta de desarrollo Unity, con el lenguaje C# y la librería Vuforia.
- ✓ Realizar el análisis de requerimientos de la aplicación con la norma IEEE STD 830 - 1998.
- ✓ Diseñar y desarrollar la aplicación móvil empleando la metodología OOHDM con UML y el patrón de diseño MVC.
- ✓ Implementar la aplicación móvil con realidad aumentada y geolocalización.

## 1.6 ALCANCE

El tema “Desarrollo de una Aplicación móvil para apoyar al Turismo del Centro Histórico de Quito, utilizando Realidad Aumentada y Geolocalización, para la Empresa VLBS CIA.LTDA” contemplará lo siguiente:

- ✓ La aplicación será desarrollada utilizando el IDE de programación Unity y las librerías Unity SDK y Vuforia SDK.
- ✓ La aplicación implementada será compatible con tecnología iPhone y con las versiones de IOS 4 y superiores.
- ✓ Permitirá reconocer 7 iglesias del Centro Histórico de Quito.
- ✓ Conocerá la ubicación de 7 lugares dentro del Centro Histórico de Quito.
- ✓ Presentará información histórica de forma escrita para los 14 lugares previamente mencionados (7 iglesias y 7 lugares geográficos).
- ✓ Presentará un menú simple donde se seleccionará ejecutar la cámara del dispositivo para la utilización de la realidad aumentada y también otra opción para que la aplicación muestre la ubicación del usuario de acuerdo a un punto geográfico conocido, donde la aplicación mostrará la información del mismo.
- ✓ Desarrollar los manuales técnicos y de usuario de la aplicación.

## 1.7 METODOLOGÍA

Para la elaboración del software se ha definido el uso de la metodología Object Oriented Hypermedia Design Methodology (OOHDM) junto con el lenguaje Unificado de Modelado (UML); con el objetivo de agilizar el diseño de la aplicación móvil. Esta metodología se creó pensando en las aplicaciones web, cuya principal característica es la navegabilidad,

característica fundamental de las aplicaciones móviles, la cual no es tomada en cuenta en otras metodologías, haciendo a OOHDM una gran herramienta para el desarrollo móvil.

### **1.7.1 Fases OOHDM**

La metodología OOHDM consta de las siguientes fases (Pressman, 2010):

- ✓ Análisis de Requerimientos.
- ✓ Diseño Conceptual.
- ✓ Diseño Navegacional.
- ✓ Diseño de la Interfaz Abstracta.
- ✓ Implementación.

### **1.7.2 Análisis de Requerimientos**

Se reúne la información necesaria para definir las funcionalidades, procesos y actividades que conformarán la creación de la aplicación móvil (IEEE Org, 2008).

### **1.7.3 Requerimientos funcionales**

- ✓ El sistema reconocerá 7 edificaciones situadas en el Centro Histórico de Quito, Ecuador.
- ✓ El sistema mostrará la información de los 7 lugares geográficos situados en el Centro Histórico de Quito, Ecuador.
- ✓ La aplicación permitirá tomar fotos usando la cámara del dispositivo móvil.

- ✓ El software mostrará a qué distancia se encuentra el usuario del punto de interés más cercano.

#### **1.7.4 Requerimientos no funcionales**

- ✓ La aplicación funcionará bajo los sistemas operativos iOS (Apple Inc) y Android (Google Inc).
- ✓ El software requiere que el dispositivo tenga conexión a internet, mientras la aplicación está en funcionamiento.
- ✓ Para que el software pueda reconocer las edificaciones se necesita que el objeto este iluminado, por lo cual se debe usar la aplicación durante el día.

#### **1.7.5 Diseño Conceptual**

Se elaboró un esquema conceptual representado por los objetos de dominio o clases y las relaciones entre dichos objetos. El diagrama de clases de la aplicación se encuentra en la Figura 17.

#### **1.7.6 Diseño Navegacional**

Se precisan las clases de navegación como enlaces, nodos y estructuras de acceso generalizadas del esquema conceptual. Se describe la estructura navegacional en términos de contextos navegacionales. El diagrama navegacional está detallado en la Figura 18.

#### **1.7.7 Diseño de la Interfaz Abstracta**

Se define la forma en la cual deben aparecer los contextos navegacionales. Se incluye el modo en que dichos objetos de interfaz



activarán la navegación y el resto de funcionalidades de la aplicación (Pressman, 2010).

### **1.7.8 Implementación**

Dedicada a la puesta en marcha de la aplicación, es donde los objetos de interfaz se enlazan con los objetos de implementación. Los modelos fueron elaborados sin tomar en cuenta la plataforma de implementación; en esta fase se tiene en cuenta los entornos en los cuales se va a correr la aplicación.

El **Modelo Vista Controlador (MVC)** es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (Modelo, Vista y Controlador). El Patrón MVC se ve frecuentemente en aplicaciones Web, donde la Vista es la página HTML y el código que provee de datos dinámicos a la página; el Modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; el Controlador es el responsable de recibir los eventos de entrada desde la Vista (Pressman, 2010).

## **Herramientas**

### **Análisis de Requerimientos**

- ✓ Diagramas de Casos de Uso

### **Modelo Conceptual**

- ✓ Diagrama de Entidad-Relación
- ✓ Diagrama de Clases
- ✓ Diagrama de Secuencia

## 1.8 RECURSOS

Las herramientas necesarias para el desarrollo del sistema son:

- ✓ Se utilizará Unity, el cual es un entorno de desarrollo C# y cumple perfectamente con las exigencias del sistema a desarrollar.
- ✓ Como framework base para el desarrollo se utilizará el iOS 5 de Apple, la librería gráfica Unity SDK, como la librería de realidad aumentada Vuforia SDK.
- ✓ El sistemas operativos en los cuales van a ser instaladas las herramientas de desarrollo serán Mac OSX Mountain Lion.

## 1.9 FACTIBILIDAD

### 1.9.1 FACTIBILIDAD TÉCNICA

Los conocimientos necesarios para completar el proyecto son:

- ✓ Programación orientada a objetos.
- ✓ Lenguaje de programación C#.
- ✓ IDE Unity.
- ✓ Programación para iOS.
- ✓ Framework iOS.
- ✓ Framework Unity SDK.
- ✓ Framework Vuforia SDK.

## 1.9.2 FACTIBILIDAD TECNOLÓGICA

El Hardware con que se cuenta para la realización del proyecto es:

- ✓ 1 Macbook, cuyas características principales son:
  - **Marca:** Apple MacBook.
  - **Procesador:** Intel Core 2 Duo.
  - **Velocidad:** 2.13Ghz
  - **Memoria RAM:** 4GB
  - **Disco:** 160 GB.
- ✓ 1 Iphone 4, en el que se realizarán las pruebas.
- ✓ 1 Tablet Android, en la que se realizarán las pruebas.

## 1.9.3 FACTIBILIDAD OPERATIVA

El sistema se encontrará instalado en el dispositivo móvil del cliente y su único requerimiento es que el dispositivo tenga acceso a internet (Wi-Fi o 3G). La aplicación se encontrará disponible las 24 horas del día los 365 días del año, la aplicación operará en el Centro Histórico de Quito.

## 1.9.4 FACTIBILIDAD OPERACIONAL

La aplicación demanda los siguientes requisitos por parte del usuario:

- ✓ Manipulación de su dispositivo (smartphone, tablet).
- ✓ Instalación de aplicaciones en el dispositivo móvil.

## 1.9.5 FACTIBILIDAD ECONÓMICA

- ✓ Para la realización de este sitio web se utilizará software libre debido a que las licencias son gratuitas, de tal manera se cumple la política de las entidades públicas de trabajar con sistemas open source, adicionalmente no se

demandarán mayores gastos por la adquisición de licencias en herramientas de desarrollo. Tanto el IDE de desarrollo como el framework que se van a utilizar tienen licencia, pero su uso es gratuito para el desarrollo.

- ✓ Tanto la Macbook, como los demás dispositivos para el desarrollo serán de propiedad del desarrollador (tesista).

### **1.9.6 FACTIBILIDAD LEGAL**

A continuación se hará un resumen de las leyes y regulaciones más relevantes con el estudio realizado.

#### **Reglamento para la prestación del servicio móvil avanzado**

En la Resolución No. 498-25-CONATEL-2002 indica lo siguiente: El espectro radioeléctrico para el Servicio Móvil Avanzado están divididos de la siguiente manera:

- ✓ 824 MHz a 849 MHz.
- ✓ 869 MHz a 894 MHz.
- ✓ 1710 MHz a 2025 MHz.
- ✓ 2110 MHz a 2200 MHz.

Las cuales han sido divididas entre las tres operadoras celulares que funcionan en el Ecuador como se muestra a continuación:

- ✓ CONECEL S.A., que funciona en la banda A, el grupo de frecuencias comprendido entre los siguientes rangos: 824 a 835 MHz, 845 a 846.5 MHz, 869 a 880 MHz, 890 a 891.5 MHz.

- ✓ OTECEL S.A., que funciona en la banda B, el grupo de frecuencias comprendido entre los siguientes rangos: 835 a 845 MHz, 846.5 a 849 MHz, 880 a 890 MHz, 891.5 a 894 MHz.
- ✓ TELECSA, que funciona en la banda C-C' de 1900 Mhz. Sus rangos de operación son de 1895 a 1910 MHz y de 1975 a 1990 MHz.

Los equipos celulares deben operar dentro de estos rangos de frecuencias para que sean funcionales en el territorio ecuatoriano (Noboa, 2001).

### **Ley Especial de Telecomunicaciones Reformada**

En la Ley N°184 de la Corporación Nacional de Telecomunicaciones estipula:

**Art. 25.- Derecho al servicio.-** Todas las personas naturales o jurídicas, ecuatorianas o extranjeras, tienen el derecho a utilizar los servicios públicos de telecomunicaciones condicionado a las normas establecidas en los reglamentos y al pago de las tasas y tarifas respectivas. Las empresas legalmente autorizadas establecerán los mecanismos necesarios para garantizar el ejercicio de los derechos de los usuarios (Registro Oficial N°. 996, 1992).

## CAPÍTULO 2: MARCO TEÓRICO

### 2 Antecedentes Investigativos

#### 2.1 Realidad Aumentada

La Realidad Aumentada (RA) es un campo de la investigación computacional, con la cual se trata de combinar datos generados por computadora, con el mundo real.

La Realidad Aumentada ha ido evolucionando conjuntamente con la Realidad Virtual, la cual trata de simular la realidad en un mundo simulado. Entre los primeros intentos para simular la realidad se encuentran:

- ✓ “Cine de Experiencia” Morton Heiling lo construyó en 1962, juntamente con 5 filmes a través del cual se motivan los sentidos del espectador (vista, tacto, oído y olfato).
- ✓ En los 80 la compañía de Jaron Lanier construyó los primeros guantes y anteojos de Realidad Virtual.

El investigador Tom Caudell fue contratado por la empresa Boeing en 1992, para encontrar una alternativa a los tediosos tableros de configuración de cables que utilizaban los trabajadores. Su solución fue tableros virtuales y gafas especiales, así “aumentó” la realidad y de ahí viene el término Realidad Aumentada (Haag, 2013).

##### 2.1.1 ¿Cómo funciona la Realidad Aumentada?

La Realidad Aumentada superpone gráficos, audio u otros elementos audiovisuales, a un ambiente real en tiempo real. La realidad aumentada se basa en un sistema de seguimientos de movimiento (Tracking System). Se apoya en marcadores, los cuales están dentro de la visión de las cámaras,

estos se utilizan para que la computadora tenga un punto de referencia sobre el cual superponer las imágenes. Estos marcadores son predefinidos por el usuario, cada marcador es exclusivo para que imagen a ser superpuesta. Los marcadores son imágenes sencillas como marcos de cuadros como se observa en la Figura 1.



Figura 1. Marcador para Realidad Aumentada.

En los últimos años se ha desarrollado RA “markerless”, la cual no tiene que trabajar con tarjetas o cosas extrañas al ambiente. Los sistemas son capaces de reconocer formas simples, como formas geométricas sencillas, mesas, frutas, como por ejemplo reconocer el cuerpo de una persona y poner ropa virtual, como se puede ver en la Figura 2.

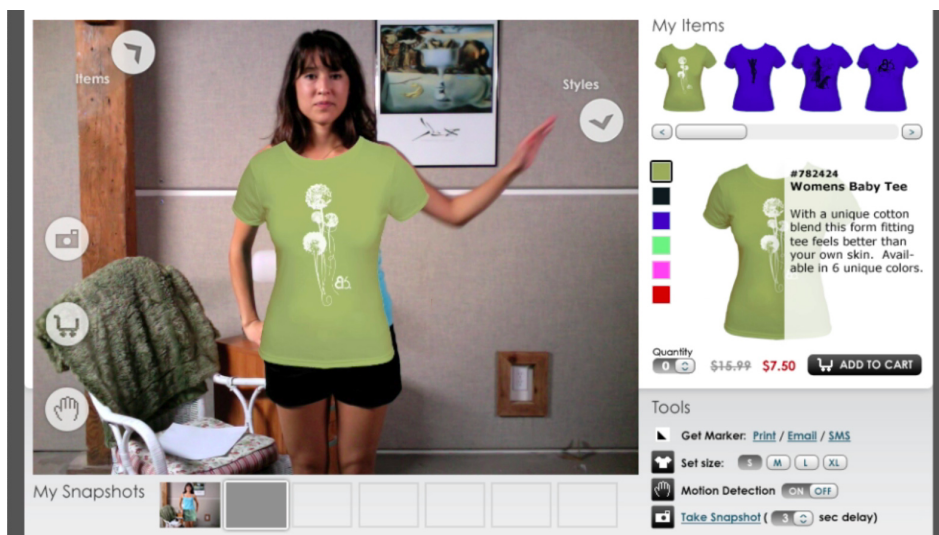


Figura 2. Ejemplo de RA markerless.

Con el auge de los smartphones, se han ido incrementando las aplicaciones que usen Realidad Aumentada. Hace algunos años la Realidad Aumentada solo se la encontraba en los laboratorios ya que para generar una aplicación RA se necesitaba un gran procesamiento de datos, esto se ha ido reduciendo al pasar el tiempo y cada vez se realiza una aplicación más compleja en un dispositivo cada vez más pequeño.

### **2.1.2 Áreas de aplicación de la realidad aumentada**

- ✓ **Turismo:** En la actualidad existe gran número de aplicaciones que apoyan a las personas que se encuentran en una ciudad desconocida, como es el caso de Nearest Tube de la empresa Acrossair, la cual se apoya de la RA para informar al usuario la dirección y la distancia de puntos turísticos de la ciudad, como también de paradas de autobús o metro más cercanas. (acrossair company, 2014)
- ✓ **Educación:** La aplicación AR Liver desarrollada por la empresa iSO-FORM, permite visualizar en tiempo real un modelo en 3D de un hígado humano anatómicamente correcto, que permite a los estudiantes de medicina tener una herramienta más para aprender Anatomía. (iSO-FORM, LLC, 2014)
- ✓ **Entretenimiento:** Una interesante aplicación de RA fue desarrollada por la compañía suiza de muebles IKEA, la cual contiene los muebles más recientes que se ofrecen y el usuario utilizando su smartphone o tablet puede observar en tiempo



real como se vería cierto mueble en la sala de su casa, ya que la aplicación utiliza la cámara del dispositivo y sitúa un modelo en 3D del mueble seleccionado. (IKEA Inc., 2014)

## **2.2 La Geolocalización, Coordenadas hacia el Éxito**

La geolocalización es la capacidad de obtener coordenadas geográficas mediante herramientas informáticas. Gracias a la tecnología de GPS que ahora está incorporada en smartphones y computadores personales, millones de personas tienen acceso a esta tecnología (Schiller & Voisard, 2004).

En estos últimos años la geolocalización se ha visto explotada por las redes sociales, que permiten dar a conocer los lugares en donde se ha estado y también conocer la localización de nuestros “amigos”. Esto ha llevado a redefinir las palabras intimidad, privacidad y relación. Entre las aplicaciones de redes sociales que utilizan la geolocalización se encuentran: Facebook, Twitter y Foursquare.

Las aplicaciones sociales con geolocalización ayudan a sobrepasar diferentes necesidades como:

- ✓ Funcionales: Encontrar un lugar al que gustaría visitar (restaurantes, tiendas, bares, etc.)
- ✓ Pertenencia: el ser humano es social por lo que disfruta formar parte de un grupo o sociedad y esto es lo que ofrecen las redes sociales formar parte de un grupo que es “actual” (a la moda).
- ✓ Estima: ser felicitados por nuestros logros, sin importar la magnitud que estos tengan.

- ✓ Autorrealización: Ayudar a otra persona, mostrándole la localización de cierto lugar que ha visitado con anterioridad y da una opinión sobre el lugar en cuestión, da la sensación de ser útiles.

## 2.3 Fundamentación

### 2.3.1 Aspectos Generales

#### 2.3.1.1 Dispositivos Móviles (Hardware)

La empresa Apple Inc. ha influido enormemente en los últimos años al desarrollo de dispositivos móviles creando una fuerte competencia. Sus dispositivos más representativos con el iPhone 5s y el iPad Air que serán descritos brevemente a continuación.

El iPhone 5s fue introducido el 10 de septiembre del 2013, es el primer teléfono que tiene un procesador de 64 bits de 1.3 GHz el Apple A7, cuenta con un sistema de identificación de huella digital, una memoria RAM de 1 GB y una cámara trasera con 8 megapíxeles. En la Figura 3 se muestra un iPhone 5s. (Apple Inc., 2014)

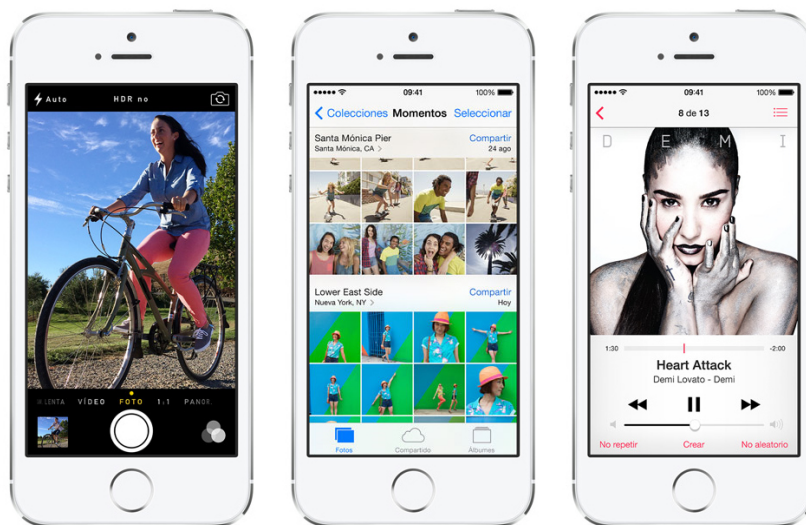


Figura 3. iPhone 5s.

El iPad Air fue introducido el 22 de octubre del 2013, también cuenta con el procesador de 64 bits Apple A7 convirtiéndola en la primera Tablet de 64 bits del mercado. Cuenta con una pantalla 9.7 pulgadas, una cámara frontal de 1,2 megapíxeles y una cámara posterior de 5 megapíxeles. En la Figura 4 se muestra un iPad Air. (Apple Inc., 2014)



Figura 4. iPad Air.

El Grupo Samsung es el principal rival de Apple Inc., compañía fundada en 1938 en Corea del Sur, siendo la empresa más grande de dicho país. Samsung ofrece una gran variedad de dispositivos móviles pero los más importantes son: Samsung Galaxy S4 y Samsung Galaxy Tab 3, los cuales serán descritos a continuación.

El Samsung Galaxy S4 introducido el 14 de marzo del 2013, esta compuesto por un procesador Qualcomm de 1.9 GHz, memoria RAM de 2GB, cámara delantera de 2 mpx y cámara trasera de 13 mpx. En la Figura 5 me muestra un Samsung Galaxy S4. (Samsung Group, 2014)



Figura 5. Samsung Galaxy S4.

La Samsung Galaxy Tab 3 introducida el 24 de junio del 2013 cuenta con un procesador Intel Atom de 1.6 GHz, memoria RAM de 1 GB, con cámara frontal de 1.3 mpx y cámara posterior de 3.15 mpx. Viene en tres diferentes tamaños de pantalla 7.0, 8.0 10.1 pulgadas. En la Figura 6 se muestra una Samsung Galaxy Tab 3. (Samsung Group, 2014)



Figura 6. Samsung Galaxy Tab 3.

### 2.3.1.2 Aplicaciones Móviles (Software)

Las aplicaciones móviles son programas que pueden ser descargados a un dispositivo móvil (celular, tablet, reproductor de mp3), que ofrecen distintas funcionalidades: comunicación, ofimática, entretenimiento.

Cada dispositivo móvil según su marca utilizan un sistema operativo (SO) diferente, entre los que más usuarios tienen se encuentran: iOS (Apple), Android (Google) y Windows Phone (Microsoft) (Gottesman, 2013).

### **2.3.1.3 iOS**

Es un sistema operativo móvil de la empresa Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV. Apple, Inc. no permite la instalación de iOS en hardware de terceros. En mayo de 2010 en los Estados Unidos, tenía el 59% de consumo de datos móviles (incluyendo el iPod Touch y el iPad).

La interfaz de usuario de iOS está basada en el concepto de manipulación directa, usando gestos multitáctiles. Los elementos de control consisten de deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee de una interfaz fluida. La interacción con el sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a sacudir el dispositivo (por ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical al apaisado u horizontal).

iOS se deriva de Mac OS X, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Unix. iOS cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de "Servicios Principales", la capa de "Medios" y la capa de "Cocoa Touch". La versión

actual del sistema operativo (iOS 6.1.3) ocupa más o menos 770 megabytes variando por modelo (Asokan, 2013).

#### **2.3.1.4 Android**

Es un sistema operativo basado en Linux y orientado a dispositivos móviles, como teléfono inteligente y tablets. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en el 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los Estados Unidos, en el segundo y tercer trimestres de 2010, con una cuota de mercado de 43,6% en el tercer trimestre. Android tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos.

A la fecha, existen cerca de 200.000 aplicaciones disponibles para Android. Android Market es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Los programas están escritos en el lenguaje de programación Java (Asokan, 2013).

#### **2.3.1.5 Windows Phone**

Windows Phone es un sistema operativo móvil desarrollado por Microsoft, como sucesor de la plataforma Windows Mobile. A diferencia de su predecesor, está enfocado en el mercado de consumo generalista en lugar del mercado empresarial. Con Windows Phone, Microsoft ofrece una

nueva interfaz de usuario que integra varios servicios propios como SkyDrive, Skype y Xbox Live en el sistema operativo.

El 29 de octubre de 2012 se lanzó al mercado la última versión del sistema operativo, Windows Phone 8. Esta versión incluye nuevas funciones que de acuerdo a Microsoft lo harán competitivo con SO como iOS de Apple o Android de Google.

Windows Phone 8 es una nueva versión del sistema operativo solo para nuevos dispositivos. Entre las nuevas características se incluyen (Microsoft, 2014):

- ✓ Multitarea para aplicaciones de terceros.
- ✓ Núcleo Windows NT para soporte de procesadores de varios núcleos.
- ✓ Tarjeta de memoria externa intercambiable.
- ✓ Transferencias de ficheros por Bluetooth.
- ✓ Cifrado de datos almacenados.
- ✓ Aplicaciones en código nativo.
- ✓ Actualizaciones directamente en el teléfono (sin usar el PC).
- ✓ Captura de pantalla.

### **2.3.2 Lenguaje de programación C#**

C# es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores generar diversas aplicaciones sólidas y seguras que se ejecutan en NET Framework. Puede utilizar este lenguaje para crear aplicaciones cliente para Windows tradicionales, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, y muchas tareas más.

La sintaxis de C# es muy expresiva, pero también es sencilla y fácil de aprender. La sintaxis de C# basada en signos de llave podrá ser reconocida inmediatamente por cualquier persona familiarizada con C, C++ o Java. Los desarrolladores que conocen cualquiera de estos lenguajes pueden empezar a trabajar de forma productiva en C# en un plazo muy breve. La sintaxis de C# simplifica muchas de las complejidades de C++ y proporciona características eficaces tales como tipos de valores que admiten valores NULL, enumeraciones, delegados y acceso directo a memoria, que no se encuentran en Java. C# admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos e iteradores, que permiten a los implementadores de clases de colección definir comportamientos de iteración personalizados que el código cliente puede utilizar fácilmente. En C# 3.0, las expresiones Language-Integrated Query (LINQ) convierten la consulta con establecimiento inflexible de tipos en una construcción de lenguaje de primera clase.

Como lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método Main que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. Una clase puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces. Los métodos que reemplazan a los métodos virtuales en una clase primaria requieren la palabra clave override como medio para evitar redefiniciones accidentales. En C#, una estructura es como una clase sencilla; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia.



Además de estos principios básicos orientados a objetos, C# facilita el desarrollo de componentes de software a través de varias construcciones de lenguaje innovadoras, entre las que se incluyen las siguientes:

- ✓ Firmas de métodos encapsulados denominadas delegados, que habilitan notificaciones de eventos con seguridad de tipos.
- ✓ Propiedades, que actúan como descriptores de acceso para variables miembro privadas.
- ✓ Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
- ✓ Comentarios en línea de documentación XML.
- ✓ Language-Integrated Query (LINQ) que proporciona funciones de consulta integradas en una gran variedad de orígenes de datos.

El proceso de generación de C# es simple en comparación con el de C y C++, y es más flexible que en Java. No hay archivos de encabezado independientes, ni se requiere que los métodos y los tipos se declaren en un orden determinado. Un archivo de código fuente de C# puede definir cualquier número de clases, estructuras, interfaces y eventos (Ferguson, Brian, Beres, Boutquin, & Gupta, 2003).

### **2.3.3 Xcode (IDE)**

Xcode es el entorno de desarrollo integrado (IDE) de Apple Inc. y se suministra gratuitamente junto con Mac OS X. Xcode trabaja conjuntamente con Interface Builder, una herencia de NeXT, una herramienta gráfica para la creación de interfaces de usuario.

Xcode incluye la colección de compiladores del proyecto GNU (GCC), y puede compilar código C, C++, Objective-C, Objective-

C++, Java y AppleScript mediante una amplia gama de modelos de programación, incluyendo, pero no limitado a Cocoa, Carbón y Java (Apple Inc.).

Tipos de proyecto para iOS de Xcode:

- ✓ **Master-Detail Application:** Presentan la información de forma jerárquica usando múltiples vistas (diseño de pantallas). También puede ser enfocadas para iPad que muestran más de una vista en la pantalla al mismo tiempo.
- ✓ **Open GL Game:** Creación de cualquier tipo de aplicación que represente imágenes, animaciones o gráficos 3D.
- ✓ **Page-Based Application:** Punto de inicio para crear una aplicación basada en páginas.
- ✓ **Single View Application :** Una vista simple para implementar la interfaz.
- ✓ **Tabbed Application :** Usado para crear aplicaciones que utilizan una barra de menú con diferentes opciones para el usuario.
- ✓ **Utility Application:** Aplicaciones que presentan una vista principal y permiten que el usuario acceda a otra vista para realizar personalizaciones básicas.
- ✓ **Empty Application:** Plantilla universal (cualquier dispositivo con iOS) que sirve de punto de partida para crear cualquier tipo de aplicación,

La Figura 7 ilustra los diferentes tipo de proyecto de Xcode.

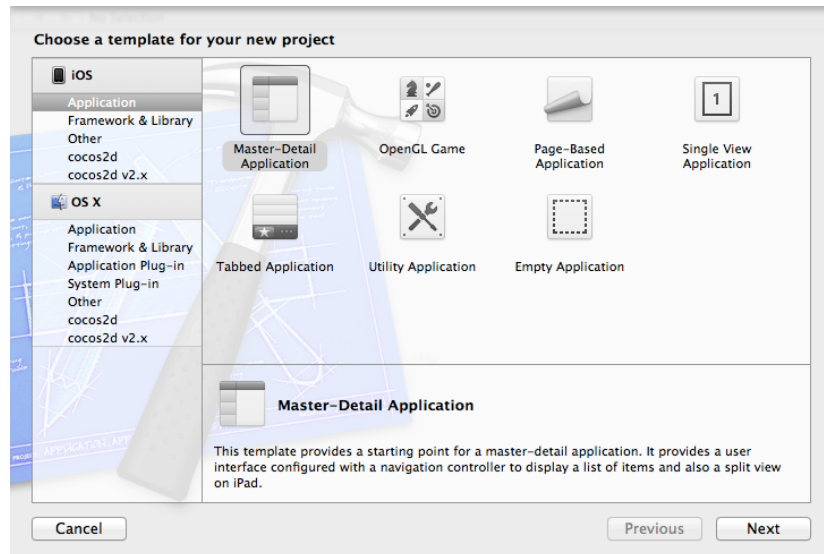


Figura 7. Verifica selección de proyectos en Xcode.

Tipos de ficheros de un proyecto:

- ✓ **Info.plist:** Como todos los plist, es un XML (introducido por NeXT) constituido de propiedades de configuración en UTF-8.
- ✓ **Ficheros .h:** Declaraciones de interfaces, clases y atributos.
- ✓ **Ficheros .m:** Implementación de clases y métodos definidos con extensión .h.
- ✓ **Ficheros .pch:** Cabeceras precompiladas que se añaden a las clases.
- ✓ **Ficheros de objetivo:** Resultado de la compilación de un conjunto de ficheros de código, puede ser una librería o una aplicación. Un conjunto de objetivos forman un producto.

La Figura 8 muestra los tipos de fichero que ofrece Xcode.

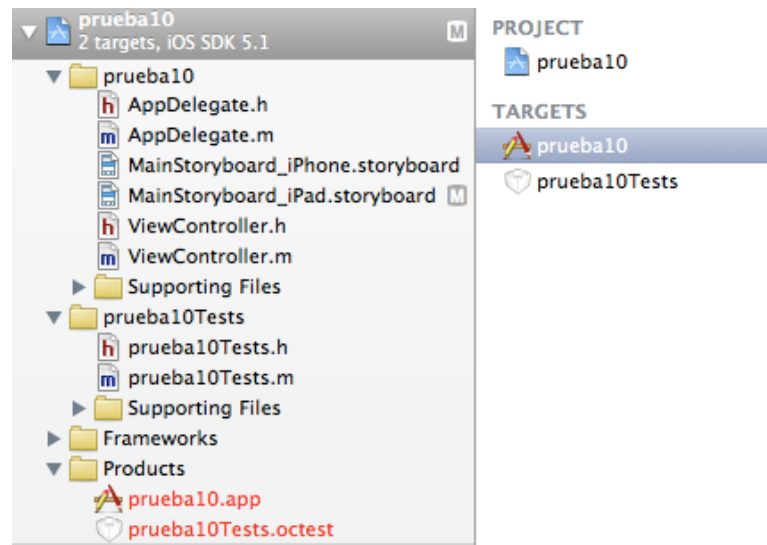


Figura 8. Árbol de un proyecto realizado en Xcode.

### 2.3.4 iOS SDK

Anteriormente conocido como iPhone SDK, es un entorno de desarrollo de software que permite a los desarrolladores, crear programas nativos para el iPhone, iPod y iPad, conocidas como “apps”, utilizando el lenguaje Objective-C y Xcode.

El SDK es un conjunto de librerías y cada una da una funcionalidad específica:

- ✓ Cocoa Touch
  - **Eventos y controles Multi-touch:** Controla los eventos relacionados con la pantalla táctil del dispositivo.
  - **Soporte acelerómetro:** Elemento que registra el movimiento del dispositivo.
  - **Localización (Regionalización):** Coordenada geográfica del dispositivo en el globo terráqueo.
  - **Soporte cámara:** Controla todas las funciones de la cámara.

✓ Media

- **OpenAL:** Librerías código abierto que manejan el sonido que emite o graba el dispositivo.
- **Mezcla de audio y grabación:** Eventos de grabación e edición de audio.
- **Reproducción de video**
- **Formato archivos de imágenes:** Tipos de imágenes soportados por el dispositivo.
- **Quartz:** motor gráfico 2D.
- **Core Animation:** motor gráfico 2D.
- **OpenGL ES:** motor gráfico 3D de código abierto.

✓ Core Services

- **Networking:** Intercomunicación entre dispositivos usan un canal común.
- **Base de datos SQLite integrada:** Motor de base de datos simplificado para el uso en dispositivos móviles.
- **Core Location:** Librería que maneja la geolocalización del dispositivo.
- **Threads (hilos):** Procesos computacionales que son procesados al mismo tiempo.
- **CoreMotion:** Librería que maneja los eventos del acelerómetro.

✓ Mac Os X Kernel

- **TCP/IP:** protocolo de red para la comunicación de distintos dispositivos.
- **Sockets:** Intercambio de información entre dos programas computacionales, siguiendo un orden establecido.
- **Manejo energía:** Consumo de la batería del dispositivo.
- **Sistema Archivos:** Forma en la cual están organizados los archivos dentro del disco duro.
- **Seguridad:** Método para asegurar que los usuarios no roben o alteren información.

En el iOS SDK se incluye el “iPhone Simulator” el cual simula un iPhone o iPad en el computador del desarrollador. Cabe mencionar que la simulación no es completa ya que el iPhone Simulator no da soporte al acelerómetro (Apple Inc., 2014).

### 2.3.5 Unity (Game Engine)

Unity es un motor de video juegos multiplataforma creado por Unity Technologies. Unity está disponible como plataforma de desarrollo para Windows y OS X, y permite crear juegos para Windows, OS X, Linux, Xbox 360, PlayStation 3, Wii, Wii U, iPad, iPhone y Android. Gracias al Plug-In Web de Unity, también se pueden desarrollar juegos de navegador, para Windows y Mac. Su última versión, la 4.2, fue lanzada en julio de 2013. Desde la página oficial se pueden descargar dos versiones: Unity y Unity Pro (Smith & C, 2013).

### **2.3.6 Vuforia SDK**

Una aplicación basada en Vuforia SDK, usa la pantalla del dispositivo móvil como un espejo mágico hacia una realidad aumentada donde el mundo real y el mundo virtual parecen coexistir. Los objetos 3D virtuales se superponen en la imagen de la cámara y parecen que forman parte del mundo real (Qualcomm Inc., 2014).

Una aplicación desarrollada con Vuforia tiene las siguientes características:

- ✓ Una rápida detección de marcadores (targets).
- ✓ Creación de marcadores en tiempo de ejecución.
- ✓ Detección y seguimiento de marcadores en superficies cilíndricas.
- ✓ Reconocimiento de texto.
- ✓ No se pierde el objeto, si el dispositivo se mueve.
- ✓ Detección simultánea de más de 5 marcadores.
- ✓ Mejores resultados en condiciones del mundo real (baja luz, marcador parcialmente cubierto).

## **2.4 Metodología de Diseño de Hipermedia Orientada a Objetos (OOHDM)**

### **2.4.1 Historia**

Con el crecimiento de Internet la creación de portales Web para ser publicados y obtener una verdadera acogida ha tomado nuevos enfoques tanto en el tamaño como en su complejidad; requiriendo de adecuados servicios y características de navegación como punto fundamental para el éxito de la aplicación final.

Por ello los investigadores, han dedicado su esfuerzo al estudio del análisis y diseño antes del desarrollo con el fin de crear metodologías prácticas y eficientes que ayuden a diseñar la hipermedia y admitan evolución y reusabilidad.

Una de las metodologías para aplicaciones en entorno Web que más se ha consolidado es OOHDM (*Object Oriented Hypermedia Design Methodology*) que fue propuesta y diseñada por D. Schwabe, G. Rossi, y S. D. J. Barbosa en el año 1996 (Universidad Nacional de la Plata) ya que se preocupa en todos sus diagramas de caracterizar el diseño de las aplicaciones hipermedia; enfocándose en un proceso de Ingeniería del Software.

Esta metodología es una extensión de **HDM** la cual fue creada por Franca Garzotto, Paolo Paolini y Daniel Schwabe en 1991, dentro del marco del proyecto HYTEA de la Comunidad Europea, cuyo objetivo era crear un modelo que fuera de utilidad para realizar el diseño de una aplicación de hipertexto.

El enfoque principal de HDM era crear un modelo consistente que posea casi en su totalidad las características conceptuales, visuales, navegacionales y abstractas de un sitio Web, es decir modelar la estructura del hipertexto-hipermedia antes de su desarrollo; lo cual fue adoptado también por OOHDM con inclusión de la orientación a objetos (Pressman, 2010).



### **2.4.2 Definición**

OOHDM es una metodología orientada a objetos y ampliamente aceptada para el desarrollo de aplicaciones hipermedia, y en particular de aplicaciones Web. Se basa en dos premisas fundamentales:

- ✓ La utilización de casos de uso para la fácil captura y definición de requisitos por parte de los usuarios y clientes no expertos; de tal manera que los analistas posteriormente puedan generar los user interaction diagrams (UIDs) o modelos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de la interfaz.
- ✓ Comenzar el diseño del sistema con el fin de que las necesidades de interacción del sitio Web sean las deseadas por los usuarios.

### **2.4.3 Características Fundamentales**

Apropiada para un comportamiento complejo: OOHDM provee las mejores herramientas para controlar el desarrollo de una aplicación que tiene un complejo comportamiento.

Separación del diseño con respecto al desarrollo: Permite que la complejidad del desarrollo de software sea menor ya que ésta ocurre a diferentes niveles: “dominios de aplicación sofisticados (financieros, médicos, geográficos, etc.); la necesidad de proveer acceso de navegación simple a grandes cantidades de datos, y por último la aparición de nuevos dispositivos para los cuales se deben construir interfaces “Web fáciles de usar”.

#### 2.4.4 Etapas ó Fases

OOHDM propone el desarrollo de aplicaciones Web a través de cinco etapas donde se combinan notaciones gráficas UML (Lenguaje de Modelado Unificado) con otras propias de la metodología y son:

- ✓ Definición u Obtención de Requerimientos.
- ✓ Diseño o Modelo Conceptual.
- ✓ Diseño Navegacional.
- ✓ Diseño de Interfaces Abstractas.
- ✓ Implementación.

En cada etapa se crean un conjunto de modelos orientados a objetos que describen un diseño particular. El éxito de esta metodología es la clara identificación de los tres diferentes niveles de diseño en forma independiente de la implementación, como se puede ver en la Tabla 1 (Rossi, 2000).

Tabla 1. Fases de OODHM.

<b>Etapas</b>	<b>Productos</b>	<b>Formalismos</b>	<b>Mecanismos</b>	<b>Descripción</b>
<b><i>Obtención de Requerimientos</i></b>	Casos de Uso (actores, escenarios)	Plantillas del formato del documento, Diagramas de Interacción de Usuario (UIDs)	Técnicas de Observación, Entrevistas	Se crea un documento que describe actividades y requerimientos de los usuarios
<b><i>Diseño Conceptual</i></b>	Clases, subsistemas, relaciones, atributos	Modelos Orientados a Objetos	Clasificación, agregación, generalización y especialización	Se modela la semántica del dominio de la aplicación
<b><i>Diseño Navegacional</i></b>	Nodos, enlaces, estructuras de acceso, contextos navegacionales, transformaciones de navegación	Vistas Orientadas a Objetos, Cartas de navegación orientadas a objetos, Clases de Contexto	Clasificación, agregación, generalización y especialización	Se tiene en cuenta el perfil del usuario y las tareas. Se enfatiza en los aspectos cognitivos. Se crea la estructura de navegación de la aplicación
<b><i>Diseño de Interfaz Abstracta</i></b>	Objetos de la interfaz abstracta, respuestas a eventos externos, transformaciones de la interfaz	Vistas Abstractas de Datos (ADV), Diagramas de Configuración, Cartas de navegación de los ADVs	Mapeado entre la navegación y los objetos visibles	Se modelizan los objetos visibles. Se describe la interfaz para los objetos de navegación. Se define el aspecto de los objetos de la interfaz
<b><i>Implementación</i></b>	Aplicación en funcionamiento	Los soportados por el entorno	Los que provea el entorno	Se realiza la puesta en producción del sistema

Fuente: (Silva, 2002)

#### 2.4.5 ¿Por qué usar OOHDM en el desarrollo de una aplicación móvil?

Una aplicación móvil es muy similar a un sitio Web ya que ambas se basan en interfaces y necesitan de un diseño navegacional para que el usuario pueda explorar todo el contenido que ofrece la aplicación. Estas características se ven contempladas en OOHDM lo cual la hace perfecta para el diseño de la aplicación móvil.

## **2.5 Lenguaje Unificado de Modelado (UML)**

UML (Unified Modeling Language) o Lenguaje Unificado de Modelado es un lenguaje gráfico para visualizar, especificar, construir y documentar los elementos de un sistema, establece un conjunto de notaciones, diagramas estándar para modelar sistemas orientados a objetos describiendo la semántica esencial de estos diagramas y los símbolos en ellos utilizados.

UML representa una recopilación de mejores técnicas de ingeniería que se han probado ser exitosas en el desarrollo de sistemas grandes y complejos, es por ello que UML puede usarse con una gran variedad de metodologías o procesos de desarrollo de software, siendo un lenguaje para especificar y no un método o un proceso (Pressman, 2010).

## **2.6 Modelo Vista Controlador (MVC)**

Como señala Juan Pavón Mestras, en su publicación referente a la Estructura de las Aplicaciones Orientadas a Objetos. El patrón Modelo-Vista-Controlador (MVC), el Modelo Vista Controlador (MVC) es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC, se usa frecuentemente en aplicaciones web, pero también se utiliza en aplicaciones móviles y de escritorio (desktop). El modelo es el Sistema de Gestión de Bases de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista, como se puede ver en la Figura 9. (Pressman, 2010)

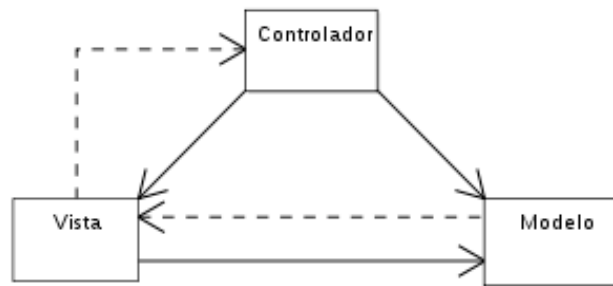


Figura 9. Ejemplo de Diagrama MVC.

### 2.6.1 Descripción del patrón

- ✓ **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la *vista* y su *controlador* facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado
- ✓ **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- ✓ **Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso

a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

- ✓ El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
- ✓ El controlador recibe (por parte de los objetos de la interfaz- vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- ✓ El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- ✓ El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier

cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones Web puesto que las clases de la vista están desconectadas del modelo y del controlador. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista. Por ejemplo en el MVC usado por Apple en su framework Cocoa. Suele citarse como Modelo-Interface-Control, una variación del MVC más puro.

- ✓ La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

## **2.7 Adelantos Tecnológicos de la RA**

A continuación se mostrarán algunos de los adelantos más recientes, relacionados con la Realidad Aumentada.

### **2.7.1 Google Glass**

Google Glass son unas gafas de realidad aumentada desarrolladas por Google, las cuales serán lanzadas al público en el 2014. Tendrán una pequeña pantalla que mostrará información disponible al usuario, similar a un teléfono inteligente, sin utilizar las manos, permitiendo el acceso al Internet mediante comandos de voz. Entre sus características se encuentran (Google Inc., 2014):

- ✓ Funciones por comandos de voz (fotos, videos, mensajes).
- ✓ Tomar fotos, captura de video (lo que uno está viendo).
- ✓ Compartir videos en vivo.
- ✓ Navegación (mapas).
- ✓ Hacer búsquedas en el internet (preguntas).
- ✓ Traducciones a otros idiomas.
- ✓ Provee información sin que se lo pregunte, dependiendo de la localización (tiempo, información de vuelos)

La Figura 10 muestra la forma de los Google Glass.



Figura 10. Google Glass.

### **2.7.2 Aurasma**

Aurasma es una aplicación de realidad aumentada de código abierto, compatible con las plataformas Android y iOS, la cual permite al usuario crear sus propios marcadores y objetos de realidad aumentada.

Esta aplicación ha sido exitosamente usada por maestros de escuelas primarias, los cuales agregan marcadores a las tareas de diferentes materias por ejemplo matemáticas y estos marcadores cargan videos cortos de la clase, donde los estudiantes pueden volver a repasar algo que estudiaron en clase en caso de no haber entendido bien o si quieren repasar. Esto a provocado que más tareas saquen la más alta calificación.



Las posibilidades que ofrece Aurasma solo se ven limitadas por nuestra imaginación (Aurasma Inc., 2014).

### 2.7.3 Par Works

Es una aplicación de realidad aumentada para dispositivos móviles, desarrollada por la empresa MARS, esta aplicación reconoce edificaciones (casa, edificios), las cuales están en alquiler o en venta, al apuntar la cámara del dispositivo móvil a la edificación, la aplicación la reconocerá y cargará información en la pantalla, por ejemplo marcará las diferentes zonas de la casa (comedor, dormitorios, etc). La aplicación también mostrará información que ayude al posible comprador a decidirse, si se permite mascotas, cargará videos de cómo es la casa por dentro.

Con esta aplicación el cliente no se ve obligado a tener asistencia por algún vendedor especializado o la necesidad de que los dueños actuales se encuentren en su casa, para poder obtener información sobre la edificación que esta interesando, como se pude ver en la Figura 11 (PAR Works Inc., 2014).

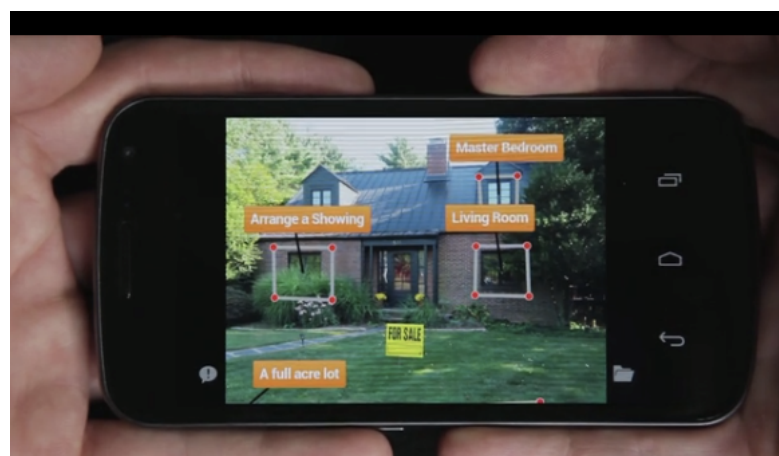


Figura 11. Aplicación Par Works.

## **CAPITULO 3: ANÁLISIS Y DISEÑO DE LA APLICACIÓN MÓVIL**

### **3 Especificación de requerimientos**

La presente especificación de requerimientos pertenece al desarrollo del proyecto “Desarrollo de una Aplicación Móvil para apoyar al Turismo del Centro Histórico de Quito, utilizando Realidad Aumentada y Geolocalización, para la empresa VLBS CIA. LTDA”, como Tesis para la obtención del título de Ingeniería en Sistemas e Informática y está desarrollada siguiendo las directrices de la metodología OOADM y del Lenguaje Unificado de Modelado (UML) junto con la Ingeniería de Software.

#### **3.1.1 Requerimientos funcionales:**

- ✓ El sistema reconocerá 7 edificaciones situadas en el Centro Histórico de Quito Ecuador.
- ✓ El sistema mostrará información de 7 lugares geográficos situados en el Centro Histórico de Quito Ecuador.
- ✓ La aplicación permitirá tomar fotos usando la cámara del dispositivo.
- ✓ El software mostrará a qué distancia se encuentra el usuario del punto de interés más cercano.

#### **3.1.2 Requerimientos no funcionales**

- ✓ La aplicación funcionará bajo los sistemas operativos iOS (Apple) y Android (Google).
- ✓ El software requiere que el dispositivo tenga conexión a internet, mientras la aplicación está en funcionamiento.

- ✓ Para que el software pueda reconocer las edificaciones se necesita que el objeto este iluminado, por lo cual se debe usar la aplicación en el día.

## 3.2 Introducción

### 3.2.1 Propósito

El propósito del presente apartado es definir los requerimientos que debe tener la aplicación móvil la cual se llamará Quito RA (Quito Realidad Aumentada) de aquí en adelante. Con la especificación de requerimientos se formalizarán las funcionalidades de la aplicación.

### 3.2.2 Definiciones, acrónimos y abreviaturas

- ✓ **3D:** Abreviatura utilizada para definir un ambiente de trabajo gráfico en tres dimensiones (three dimensions).
- ✓ **Target:** Corresponde a la abreviatura utilizada para definir el marcador que va a usar la aplicación para cargar los objetos virtuales al mundo real.
- ✓ **RA:** Abreviatura utilizada para definir el término Realidad Aumentada.
- ✓ **GUI:** Corresponde a la abreviatura del término Interface Gráfica de Usuario (Graphics User Interface) que se define como el medio por el cual el usuario va a interactuar con las aplicación.
- ✓ **ImageTarget:** Abreviatura utilizada para definir un grupo de uno o más targets (marcadores) reconocibles por la aplicación para cargar los objetos virtuales.

### 3.3 Identificación de Roles y Tareas

#### ✓ Roles

- Usuario
  - Es el actor que tiene acceso a la aplicación y que usa sus funcionalidades.
- GUI
  - El sistema que retorna información y alertas al usuario.

La Figura 12 muestra los dos actores del sistema.



Figura 12. Actores del sistema.

#### ✓ Tareas

- Usuario
  - Navegar en los menús de la aplicación.
  - Usar la cámara del dispositivo móvil dentro de la aplicación, para reconocimiento de los objetivos.
  - Tomar fotos usando la aplicación.
  - Acercarse a puntos de geolocalización para que la aplicación muestre información del lugar.
- GUI
  - Reconocer las edificaciones (iglesias).
  - Mostrar mensajes informativos al usuario.

### 3.4 Especificación de Escenarios

#### Rol Usuario

- ✓ Navegar por la aplicación: El usuario es capaz de usar todas las funcionalidades que ofrece la aplicación Quito RA.

### 3.5 Especificación de casos de uso por Actor

#### Actor: Usuario

Como se puede observar los casos de uso del actor Usuario en la Figura 13 y que a continuación se mencionan:

- ✓ USU-RA-01: Ver ayuda aplicación.
- ✓ USU-RA-02: Entrar configuración.
- ✓ USU-RA-03: Iniciar aplicación.

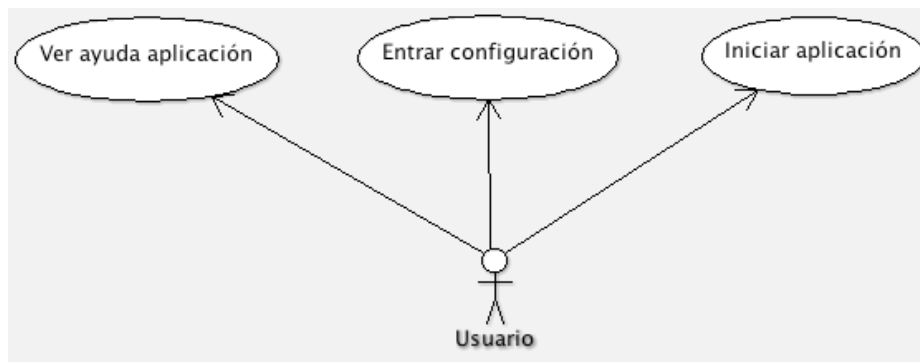


Figura 13. Casos de uso del Usuario.

## Casos de uso del Usuario

La Tabla 2 muestra la opción ayuda de la aplicación.

Tabla 2. Ver ayuda aplicación.

<b>USU-RA-01: Ver ayuda aplicación</b>	
<b>Resumen:</b>	Despliega la información necesaria para usar la aplicación.
<b>Prioridad:</b>	Media
<b>Actores Directos:</b>	Usuario
<b>Escenarios</b>	
<b>Tipo Escenario</b>	<b>Descripción</b>
<b>Principal:</b>	El sistema muestra la información básica para poder utilizar todas las funcionalidades de la aplicación.
<b>Secundario:</b>	Ninguna
<b>Pre-condiciones</b>	
El usuario debe ingresar a la opción Ayuda! en el menú principal.	
<b>Post-condiciones</b>	
Ninguna	

La Tabla 3 muestra la opción de entrar en la configuración.

Tabla 3. Entrar configuración.

<b>USU-RA-02: Entrar configuración</b>	
<b>Resumen:</b>	Despliega las diferentes opciones de configuración.
<b>Prioridad:</b>	Baja
<b>Actores Directos:</b>	Usuario
<b>Escenarios</b>	
<b>Tipo Escenario</b>	<b>Descripción</b>
<b>Principal:</b>	El sistema muestra todas las opciones de configuración que permite la aplicación.
<b>Secundario:</b>	Ninguna
<b>Pre-condiciones</b>	
El usuario debe ingresar a la opción Configuración en el menú principal.	
<b>Post-condiciones</b>	
Ninguna	

La Tabla 4 muestra la opción iniciar la aplicación.

Tabla 4. Iniciar aplicación.

USU-RA-03: Iniciar aplicación	
<b>Resumen:</b>	El actor indica que esta listo para iniciar la aplicación.
<b>Prioridad:</b>	Alta
<b>Actores Directos:</b>	Usuario
<b>Escenarios</b>	
<b>Tipo Escenario</b>	<b>Descripción</b>
<b>Principal:</b>	El usuario indica que esta listo para iniciar el sistema.
<b>Secundario:</b>	Ninguna
<b>Pre-condiciones</b>	
El usuario debe ingresar a la opción Empezar en el menú principal.	
<b>Post-condiciones</b>	
Ninguna	

### Actor: GUI

Se puede observar los casos de uso del actor Usuario en la Figura 14 y que a continuación se mencionan:

- ✓ GUI-RA-01: Despliega información RA.
- ✓ GUI-RA-02: Despliega información Geolocalización.
- ✓ GUI-RA-03: Punto de interés cercano.

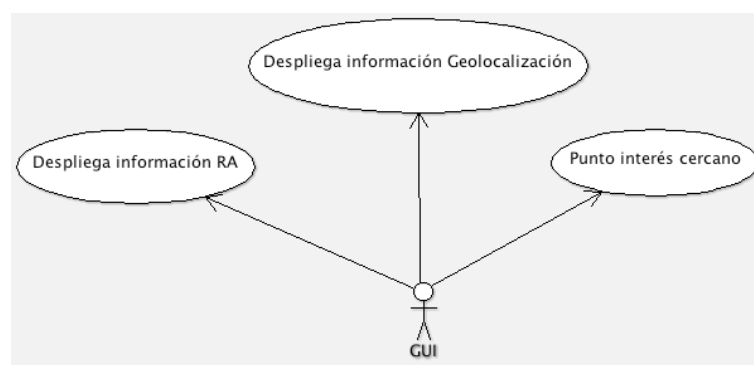


Figura 14. Casos de uso para GUI.

## Casos de uso de GUI

La Tabla 5 indica el caso de uso de desplegar la información RA.

Tabla 5. Despliega información RA.

GUI-RA-01: Despliega información RA									
<b>Resumen:</b>	Despliega la información acerca de una iglesia reconocida								
<b>Prioridad:</b>	Alta								
<b>Actores Directos:</b>	GUI								
<table border="1"> <thead> <tr> <th colspan="2">Escenarios</th> </tr> <tr> <th>Tipo Escenario</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><b>Principal:</b></td> <td>El sistema muestra la información acerca de la edificación que fue reconocida.</td> </tr> <tr> <td><b>Secundario:</b></td> <td>Ninguna</td> </tr> </tbody> </table>		Escenarios		Tipo Escenario	Descripción	<b>Principal:</b>	El sistema muestra la información acerca de la edificación que fue reconocida.	<b>Secundario:</b>	Ninguna
Escenarios									
Tipo Escenario	Descripción								
<b>Principal:</b>	El sistema muestra la información acerca de la edificación que fue reconocida.								
<b>Secundario:</b>	Ninguna								
<b>Pre-condiciones</b>									
<ul style="list-style-type: none"> <li>• La cámara del dispositivo debe estar activa.</li> <li>• Que el sistema reconozca e identifique la edificación.</li> </ul>									
<b>Post-condiciones</b>									
Ninguna									

La Tabla 6 muestra el caso de uso despliega información de geolocalización.

Tabla 6. Despliega información Geolocalización.

GUI-RA-02: Despliega información Geolocalización									
<b>Resumen:</b>	Despliega la información acerca de un punto geográfico.								
<b>Prioridad:</b>	Alta								
<b>Actores Directos:</b>	GUI								
<table border="1"> <thead> <tr> <th colspan="2">Escenarios</th> </tr> <tr> <th>Tipo Escenario</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><b>Principal:</b></td> <td>El sistema muestra la información acerca del punto geográfico, que su posición se encuentra almacenada.</td> </tr> <tr> <td><b>Secundario:</b></td> <td>Ninguna</td> </tr> </tbody> </table>		Escenarios		Tipo Escenario	Descripción	<b>Principal:</b>	El sistema muestra la información acerca del punto geográfico, que su posición se encuentra almacenada.	<b>Secundario:</b>	Ninguna
Escenarios									
Tipo Escenario	Descripción								
<b>Principal:</b>	El sistema muestra la información acerca del punto geográfico, que su posición se encuentra almacenada.								
<b>Secundario:</b>	Ninguna								
<b>Pre-condiciones</b>									
<ul style="list-style-type: none"> <li>• El usuario inició la aplicación.</li> <li>• Encontrarse en las coordenadas geográficas, del punto que reconocido.</li> </ul>									
<b>Post-condiciones</b>									
Ninguna									



La Tabla 7 indica el caso de uso punto de interés más cercano.

Tabla 7. Punto de interés cercano.

GUI-RA-03: Punto de interés cercano									
<b>Resumen:</b>	Despliega un mensaje con la distancia del punto de interés más cercano								
<b>Prioridad:</b>	Alta								
<b>Actores Directos:</b>	GUI								
<table border="1"> <thead> <tr> <th colspan="2">Escenarios</th> </tr> <tr> <th>Tipo Escenario</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><b>Principal:</b></td> <td>El sistema muestra un mensaje con la distancia, del punto de RA o geolocalización más cercano.</td> </tr> <tr> <td><b>Secundario:</b></td> <td>Ninguna</td> </tr> </tbody> </table>		Escenarios		Tipo Escenario	Descripción	<b>Principal:</b>	El sistema muestra un mensaje con la distancia, del punto de RA o geolocalización más cercano.	<b>Secundario:</b>	Ninguna
Escenarios									
Tipo Escenario	Descripción								
<b>Principal:</b>	El sistema muestra un mensaje con la distancia, del punto de RA o geolocalización más cercano.								
<b>Secundario:</b>	Ninguna								
<b>Pre-condiciones</b>									
<ul style="list-style-type: none"> <li>El usuario inició la aplicación.</li> </ul>									
<b>Post-condiciones</b>									
Ninguna									

### 3.6 Diseño Conceptual

#### 3.6.1 Diagramas de Secuencia

##### Actor: Usuario

Las Figuras 15, 16 y 17 ilustran los diagramas de secuencia del actor usuario.

- ✓ USU-RA-01: Ver ayuda aplicación.

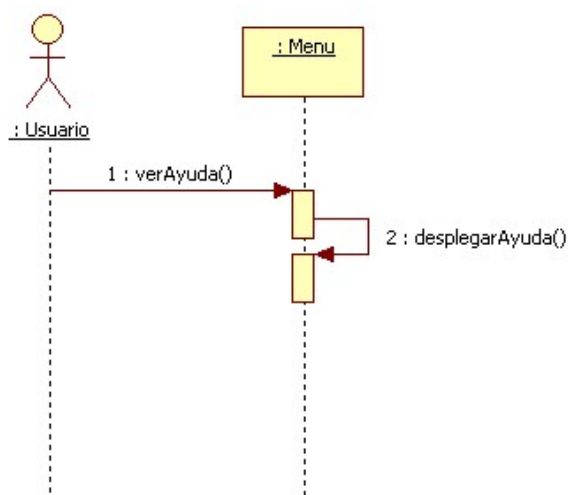


Figura 15. Diagrama de secuencia – Ver ayuda aplicación.

✓ USU-RA-02: Entrar configuración.

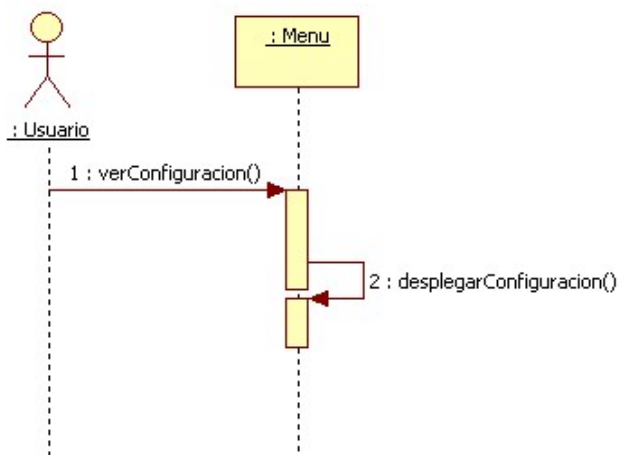


Figura 16. Diagrama de secuencia – Entrar configuración.

✓ USU-RA-03: Iniciar aplicación.

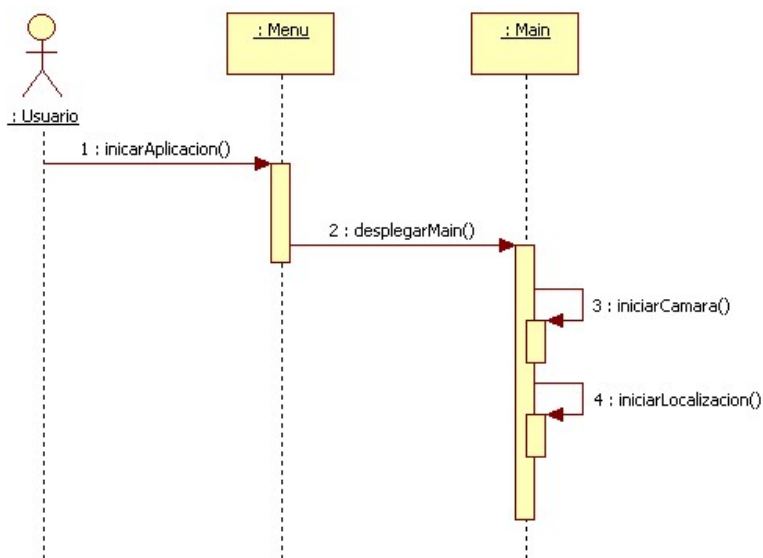


Figura 17. Diagrama de secuencia – Iniciar aplicación.

**Actor: GUI**

Las Figuras 18, 19 y 20 muestran los diagramas de secuencia del actor GUI.

- ✓ GUI-RA-01: Despliega información RA.

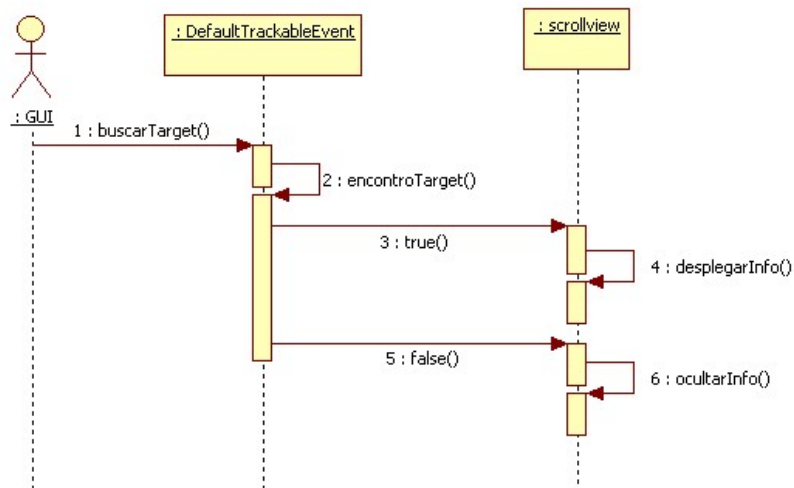


Figura 18. Diagrama de secuencia – Despliega información RA.

- ✓ GUI-RA-02: Despliega información Geolocalización.

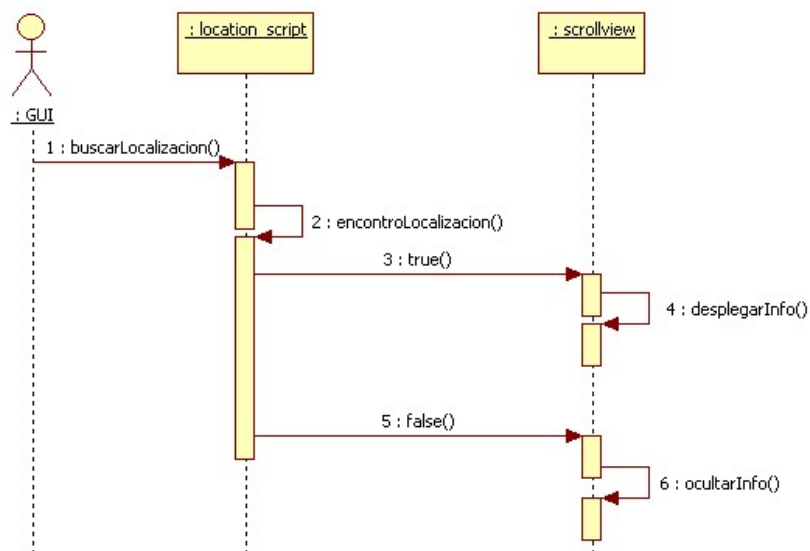


Figura 19. Diagrama de secuencia – Despliega información geolocalización.

- ✓ GUI-RA-03: Punto de interés cercano.

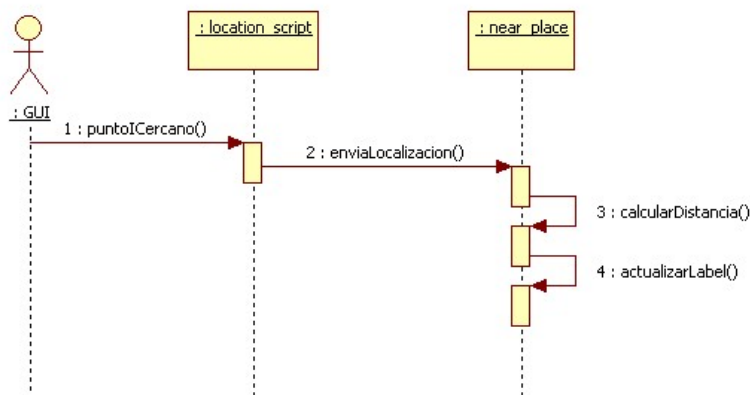


Figura 20. Diagrama de secuencia – Punto de interés cercano.

### 3.6.2 Diagrama de Clases

Las clases de la aplicación se muestran en la Figura 21 y son las siguientes:

- ✓ Location\_script.
- ✓ XmlReader.
- ✓ Shoot\_button.
- ✓ Scrollview.

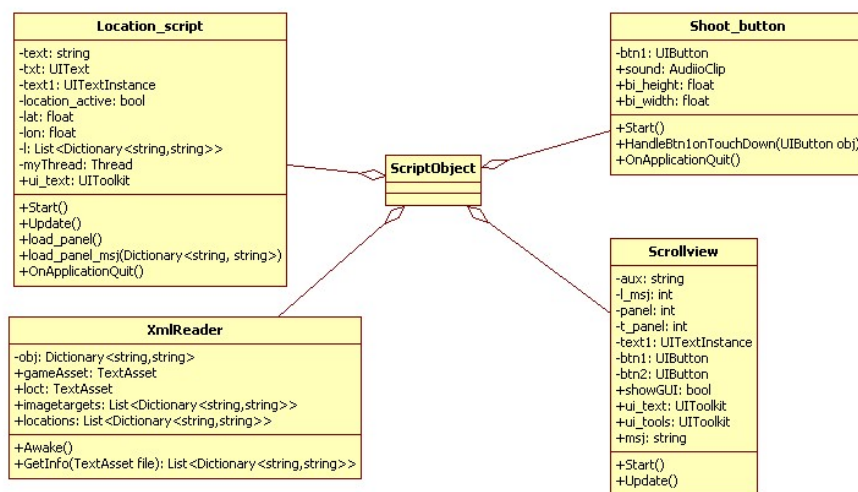


Figura 21. Diagrama de clases.

### 3.6.3 Diseño Navegacional

El sistema está basado en escenas, en la que el usuario puede ir navegando. Posee los siguientes objetos y contextos navegacionales (Figura 22):

#### Objetos Navegacionales:

- ✓ Menú principal.
- ✓ Escena de ayuda.
- ✓ Escena de configuración.
- ✓ Escena principal.

#### Contextos Navegacionales:

- ✓ Iniciar la aplicación.

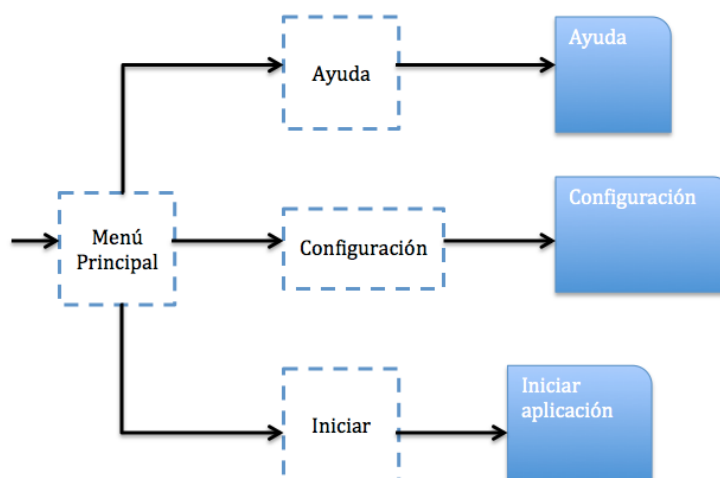


Figura 22. Diagrama de contexto.

### 3.6.4 Diseño de Interfaz Abstracta

A continuación se mostrarán las ADVs (Vista de Datos Abstracta), modelos abstractos que especifican la organización y el comportamiento de la interfaz.

## Vista de Datos Abstracta

- ✓ Nodo Menú Principal

<b>ADV Nodo Menú Principal</b>
<b>ADV Nodo Iniciar</b>
<b>ADV Nodo Ayuda</b>
<b>ADV Nodo Configuración</b>

- ✓ Nodo Ayuda

<b>ADV Nodo Ayuda</b>
<b>Texto Como usar la aplicación</b>
<b>&lt;Regresar Menú&gt;</b>

- ✓ Nodo Configuración

<b>ADV Nodo Configuración</b>
<b>Desactivar punto cercano</b>
<b>&lt;Regresar Menú&gt;</b>

- ✓ Nodo Aplicación

<b>ADV Nodo Aplicación</b>
<b>&lt;Regresar Menú&gt;</b>

### 3.6.5 Arquitectura del Sistema

En la Figura 23 se muestra la arquitectura de la aplicación móvil.

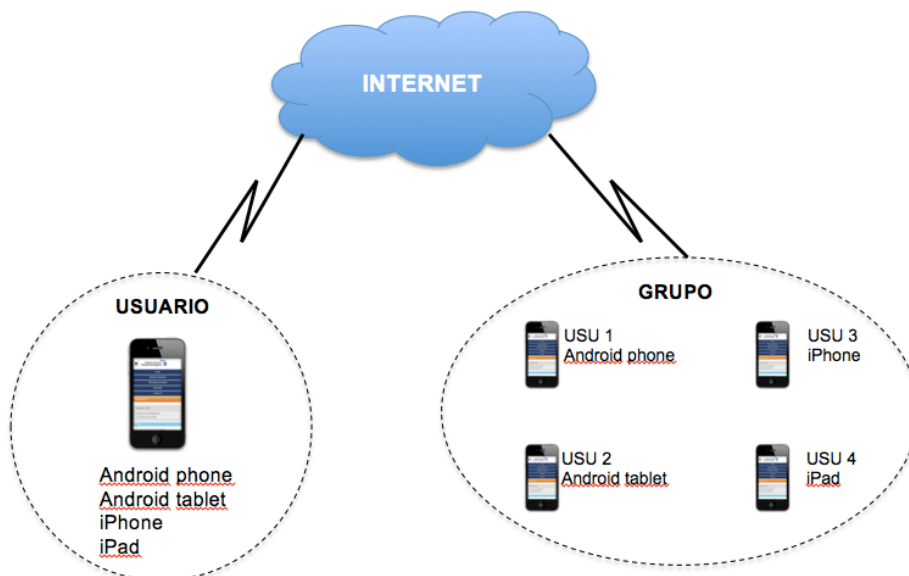


Figura 23. Arquitectura del sistema.

## CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS DE LA APLICACIÓN

### MÓVIL

#### 4 Lenguaje de programación C#

C# lenguaje de programación presentado en .Net Framework, procede de C++. Sin embargo, C# es un lenguaje orientado a objetos, moderno y seguro.

Microsoft diseñó C# de modo que retuviera casi toda la sintaxis de C y C++. Los programadores que estén familiarizados con esos lenguajes pueden escoger el código C# y empezar a programar de forma relativamente rápida. Sin embargo, la gran ventaja de C# consiste en que sus diseñadores decidieron no hacerlo compatible con los anteriores C y C#. Aunque esto puede parecer un mal asunto, en realidad es una buena noticia. C# elimina las cosas que hacían que fuese difícil trabajar con C y C++. Como todo el código C es también C++. C++ tenía que mantener todas las rarezas y deficiencias de C. C# parte de cero y sin ningún requisito de compatibilidad, así que puede mantener los puntos fuertes de sus predecesores y descartar las debilidades que complicaban las cosas a los programadores de C y C++.

Como lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método Main que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. Una clase puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces. Los métodos que reemplazan a los métodos virtuales en una clase primaria requieren la palabra clave `override` como medio para evitar redefiniciones accidentales. En C#, una estructura es como una clase



sencilla; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia (Ferguson, Brian, Beres, Boutquin, & Gupta, 2003).

## 4.1 Unity Game Engine

Unity es un poderoso motor de videojuegos, el cual está disponible en dos versiones Indie (gratis) y Pro (pago).

### 4.1.1 Instalación

- ✓ Ingresar en [www.unity3d.com](http://www.unity3d.com)
- ✓ En la página Descargar , dar click en la última versión de Unity. Esperar a que termine la descarga (existe versión para Windows o Mac OSX).
- ✓ Una vez que el software está instalado. Saldrá una ventana de activación donde pregunta qué tipo de licenciase desea utilizar: Unity Gratis o 30 días de Unity Pro, como se muestra en la Figura 24.

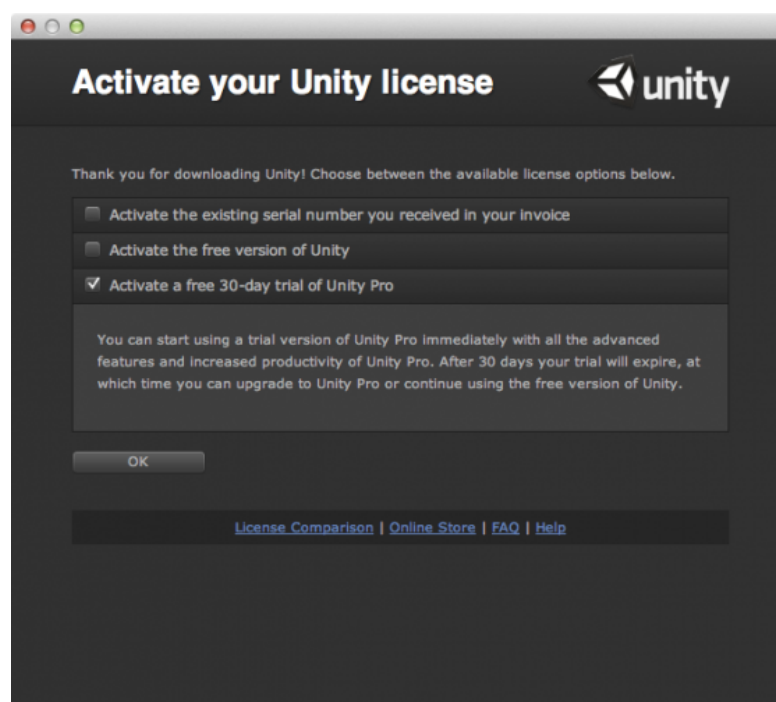
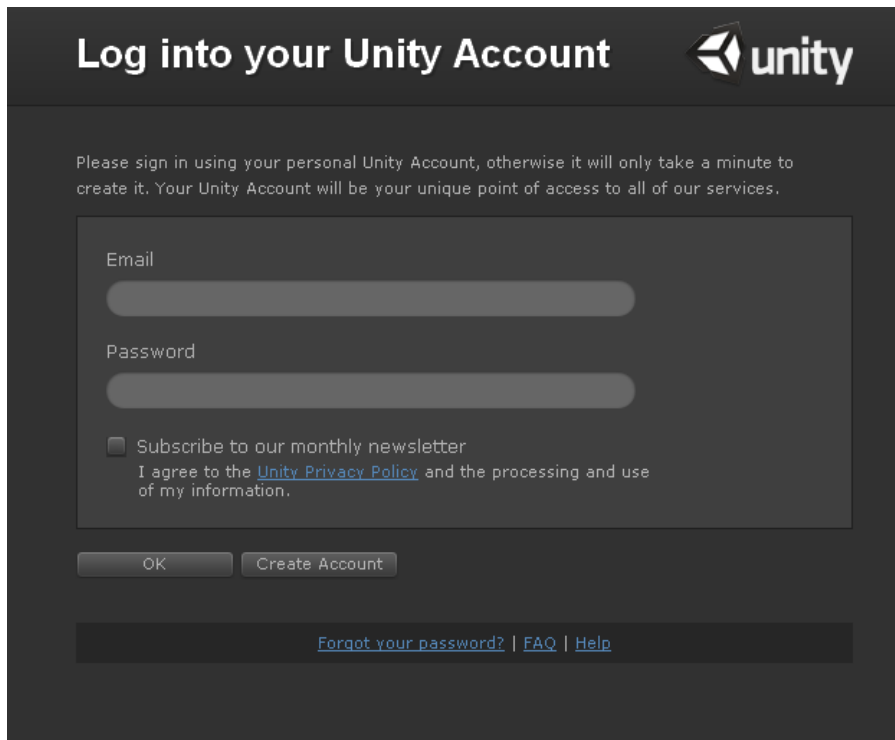


Figura 24. Activación Unity.

- ✓ Seleccionar el tipo de licencia deseado y la siguiente pantalla será para ingresar o crear una cuenta en Unity (Figura 25).



The image shows a dark-themed web interface for logging into a Unity account. At the top left, the text "Log into your Unity Account" is displayed in white. To the right is the Unity logo, which consists of a stylized white cube icon followed by the word "unity" in a lowercase, sans-serif font. Below the header, a paragraph of text reads: "Please sign in using your personal Unity Account, otherwise it will only take a minute to create it. Your Unity Account will be your unique point of access to all of our services." The main content area is a light gray box containing two input fields: "Email" and "Password", each with a rounded rectangular input area. Below these fields is a checkbox labeled "Subscribe to our monthly newsletter" and a line of text: "I agree to the [Unity Privacy Policy](#) and the processing and use of my information." At the bottom of the form are two buttons: "OK" and "Create Account". Below the form, there is a dark gray bar with three links: "[Forgot your password?](#) | [FAQ](#) | [Help](#)".

Figura 25. Registrarse con la cuenta Unity.

- ✓ Una vez registrado ya se puede usar Unity de inmediato.

## 4.1.2 Interfaz de usuario de Unity

La Figura 26 muestra la interfaz principal de Unity.

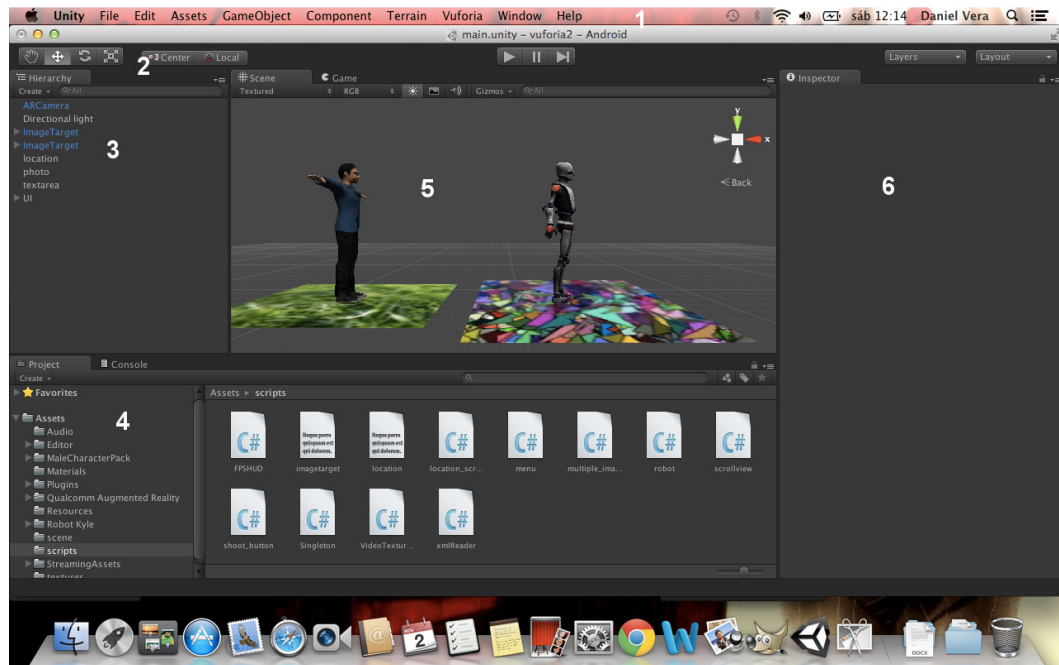


Figura 26. Interfaz de Unity.

- ✓ **Menú:** Da acceso a la lista de comandos: asset import/export, preferences setting, game object creation, componentes, terrain, layout y documentation.
- ✓ **Toolbar:** Incluye las herramientas de transformación (manipulación de objetos y navegación de la escena), herramientas de control (correr, pausar y detener el nivel), herramientas de listas (manejo de capas).
- ✓ **Hierarchy:** Lista con los objetos del juego (personajes, cámaras, luces).
- ✓ **Project:** En donde se organizan todos los archivos de nuestro proyecto (escenas, scripts, modelos).

- ✓ **Scene:** Escena en donde coloca, rota y selecciona los objetos que van a estar presentes en el nivel.
- ✓ **Inspector:** Ventana en donde se configura cualquier objeto de la escena, se selecciona desde la ventana Hierarchy.

### 4.1.3 Importar paquetes Unity a un proyecto

Los paquetes que provee Unity, pueden ahorrar un gran tiempo de desarrollo, usualmente contienen recursos (texturas, materiales) y nuevas características. Existen paquetes desarrollados por otras personas o empresas que ofrecen nuevas características o personajes, los cuales se los puede descargar desde la Asset Store (<https://www.assetstore.unity3d.com/>), como se en el ejemplo de la Figura 27.

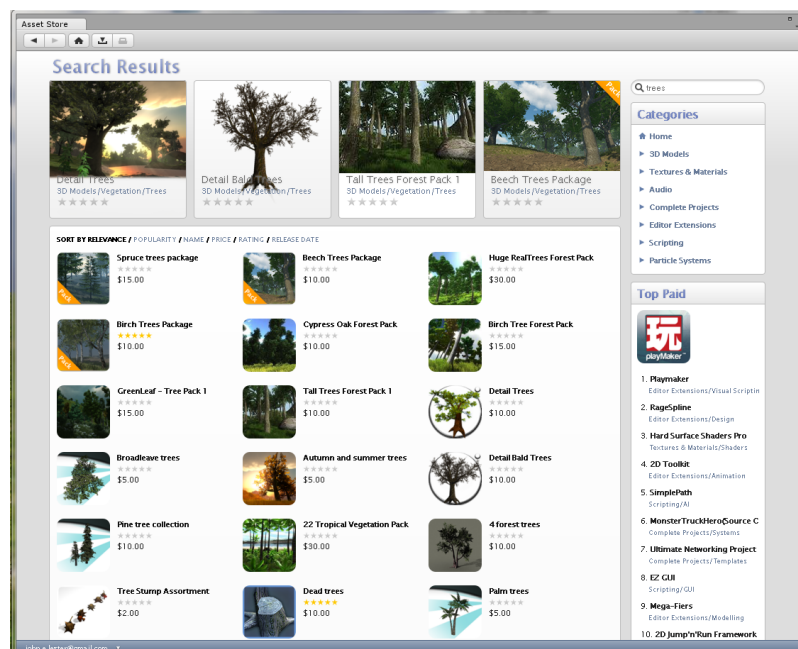


Figura 27. Asset Store.

Para importar un paquete a nuestro proyecto:

- ✓ Dentro del editor de Unity, se entra al menú Asset.
- ✓ Se selecciona Import Package. Da una lista de paquetes que ofrece Unity, pero si se quiere importar un paquete descargado de la Asset Store selecciona el Custom Package, como se visualiza en la Figura 28.

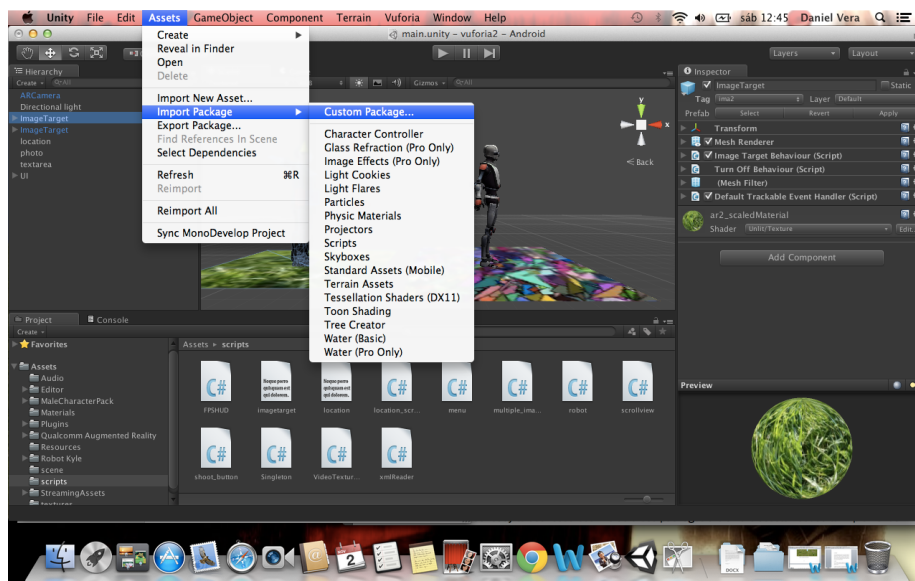


Figura 28. Importar paquete Unity.

- ✓ El contenido del paquete se importará a nuestro proyecto.

## 4.2 Vuforia SDK

Vuforia SDK es un grupo de librerías que permite utilizar realidad aumentada en el proyecto que se está desarrollando, esta SDK se encuentra disponible para Android, iOS y como paquete para Unity, todas son distribuidas de forma gratuita por la empresa Qualcomm Inc. Vuforia usa la técnica de rastreo de marcadores, la cual busca patrones en las imágenes para cargar objetos.

Las características que ofrece Vuforia son:

- ✓ **ARCamera:** Ofrece una cámara que detecta los marcadores en tiempo real.
- ✓ **Image Target:** Este es el componente más utilizado en Vuforia, usándolo la aplicación puede detectar cualquier imagen con la que fue entrenada y desplegar el contenido RA encima de la imagen (Figura 29).



Figura 29. Image Target.

- ✓ **Frame Maker:** Son marcadores cuadrados con un código dentro de sus márgenes. Existen 100 Frame Markers que ofrece Vuforia, que su aplicación puede usar para desplegar contenido RA. La ventaja de los Frame Markers es que requieren un menor procesamiento y se pueden detectar más en una sola escena sin afectar al desempeño de la aplicación (Figura 30).

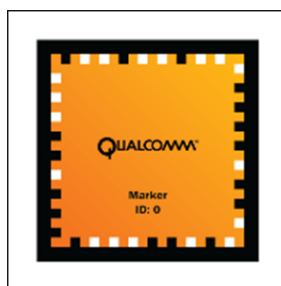


Figura 30. Frame Maker.

- ✓ **Multi-Targets:** Permiten detectar una caja física desde cualquier ángulo, la caja debe tener suficientes detalles para que sea fácilmente detectable. Con esto se puede hacer que el contenido RA pueda rotar alrededor de la caja.
- ✓ **Virtual Button:** Los botones virtuales permiten que el usuario interactúe con el marcador y cuando la aplicación detecta la interacción, se puede modificar el contenido RA.

#### 4.2.1 Instalación de Vuforia

- ✓ Ir a la dirección <https://developer.vuforia.com/resources/sdk/unity>
- ✓ Descargar la extensión para Unity de Vuforia, como se visualiza en la Figura 31.

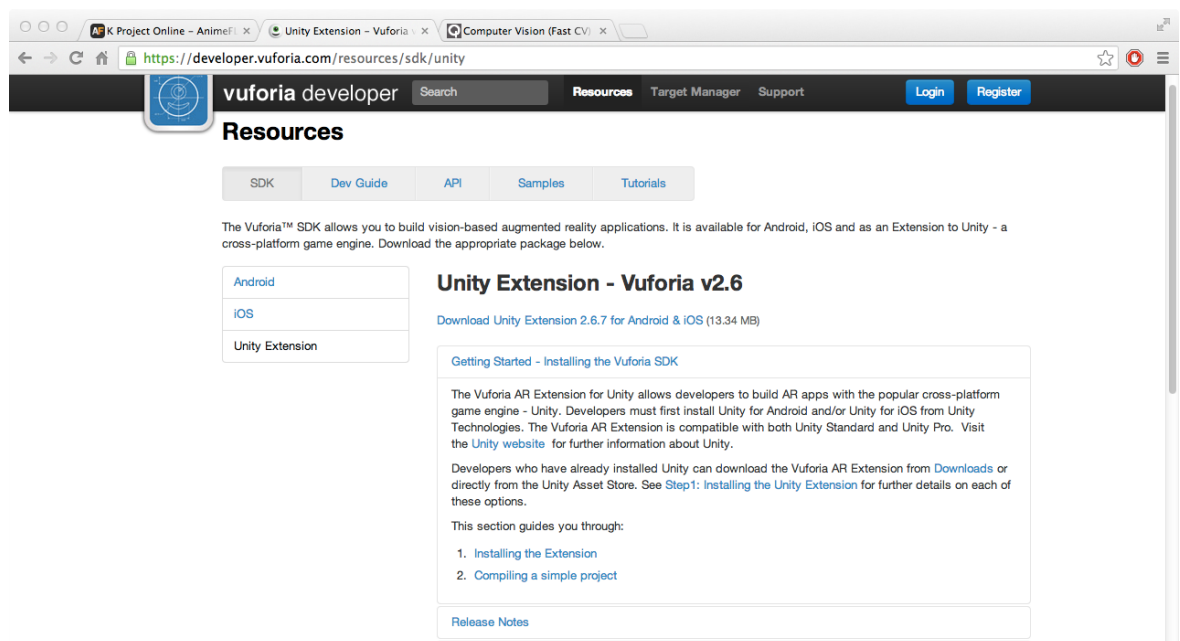


Figura 31. Página oficial de Vuforia.

- ✓ Agregar el paquete de Vuforia a su proyecto en Unity.

### 4.3 Construcción de la aplicación móvil en Unity

La aplicación fue desarrollada en Unity, con la ayuda de la librería de Realidad Aumentada Vuforia y con la herramienta MonoDevelop para la creación de las clases en C#, que son las siguientes:

- ✓ **location\_script:** Obtiene la geolocalización en la que se encuentra el usuario, para determinar la distancia al punto de interés más cercano.
- ✓ **scrollview:** Esta clase presenta en pantalla la información del punto de interés ya sea por Realidad Aumentada o geolocalización.
- ✓ **shoot\_button:** Controla la captura de pantalla de la aplicación (foto).
- ✓ **Multiple\_image:** Al iniciar la aplicación, activa todos los image target para que la aplicación pueda reconocer las edificaciones.

#### 4.3.1 Consideraciones de la programación de RA utilizando Vuforia

Para entender cómo funciona la realidad aumentada se debe tomar en cuenta los siguientes términos:

- ✓ **Marcador (Marker):** La imagen que va a reconocer la aplicación (Figura 32).

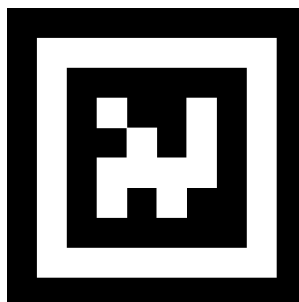


Figura 32. La imagen representa un marcador de realidad aumentada.

- ✓ **Características (Features):** El grado de detalles que tiene un marcador, se facilita el reconocimiento a mayor número de detalles (Figura 33).



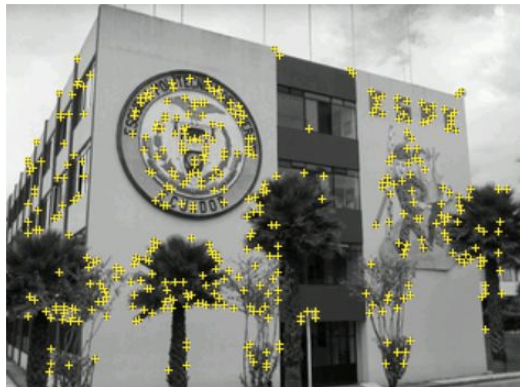


Figura 33. Muestra las características del bloque central ESPE.

- ✓ **Objeto RA:** Objeto digital que la aplicación va a posicionar encima del marcador (Figura 34).



Figura 34. Muestra un objeto RA al ser reconocido su marcador.

Uno de los primeros retos que se afrontó para realizar esta aplicación, fue el hecho de que el SDK Vuforia está diseñado para reconocer imágenes, no estructuras físicas reales (edificaciones), para que la aplicación pueda reconocer estos objetos, se tuvo que alimentar a la aplicación con un gran número de fotografías de la edificación tomados desde diferentes ángulos, con esto la aplicación va comparando cada una de las fotografías con la imagen capturada por la cámara del dispositivo móvil en tiempo real. Esto lo hace hasta encontrar la fotografía más parecida a la imagen de la cámara, así la aplicación sabe que edificación es y procede a cargar el Objeto RA que en nuestro caso es el nombre de la estructura y una leyenda informativa de la misma.

### 4.3.2 Consideraciones especiales de la geolocalización

La geolocalización son las coordenadas geográficas (latitud y longitud) de un punto en el globo. Este punto puede representar en donde se encuentra una persona u objeto en el planeta Tierra. Esto permitió añadir dos características a la aplicación que son las siguientes:

#### 4.3.2.1 Distancia entre dos puntos geográficos

Para calcular la distancia entre 2 puntos geográficos (latitud, longitud) se utilizó la fórmula de Haversine: (Abramowitz & Stegun, 1972)

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (1)$$

$$c = 2 \cdot \tan^{-1}(\sqrt{a}, \sqrt{1-a}) \quad (2)$$

$$d = R \cdot c \quad (3)$$

Donde:

$\varphi$ : latitud.

$\lambda$ : longitud.

R: radio de la Tierra (6371 km).

Con esto se logra calcular la distancia entre el usuario y el punto de interés más cercano.

#### 4.3.2.2 Ruta para llegar a un punto geográfico específico

La aplicación posee una flecha que apunta hacia el punto de interés más cercano con relación a la posición actual del usuario, esto se logra calculando el Rumbo (Bearing) del usuario y se lo realiza con la siguiente fórmula (4): (Abramowitz & Stegun, 1972)

$$\theta = \tan^{-1}(\sin(\Delta\lambda) \cdot \cos(\varphi_2), \cos(\varphi_1) \cdot \sin(\varphi_2) - \sin(\varphi_1) \cdot \cos(\varphi_2) \cdot \cos(\Delta\lambda))$$

Donde:

$\varphi$ : latitud.

$\lambda$ : longitud.

Tanto la distancia como el rumbo se ven representados por una flecha en la parte superior izquierda de la pantalla del dispositivo como se muestra en la Figura 35.



Figura 35. Captura de la aplicación que muestra la distancia y el rumbo.

#### 4.4 Pruebas de la Aplicación

Para realizar pruebas a la aplicación se ha tomado el proceso propuesto por Roger S. Pressman en su libro Ingeniería de Software, el cual está orientado a aplicaciones web. Ya que las aplicaciones móviles comparten muchas características con las web entre ellas: operación en múltiples sistemas operativos y hardware, protocolos de comunicación, por lo cual este modelo ayuda probar nuestra aplicación en diferentes aspectos: contenido, función, estructura, usabilidad, navegación, desempeño,

compatibilidad y seguridad. La representación gráfica se encuentra en las Figuras 36 y 37.

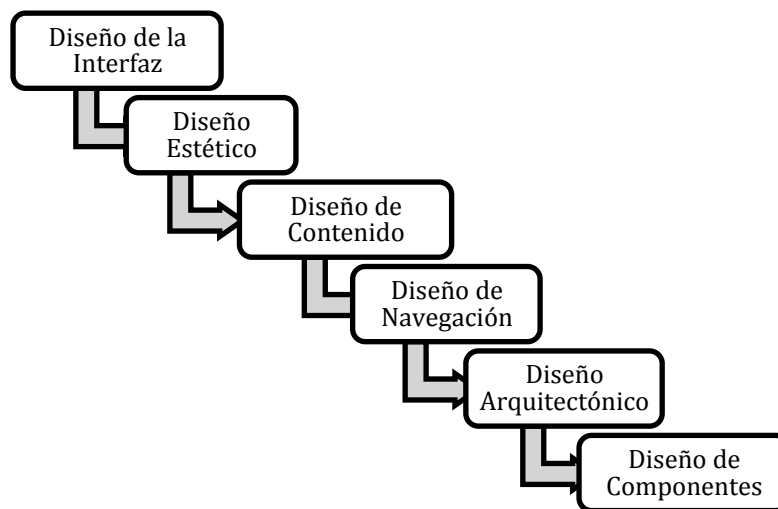


Figura 36. Proceso de prueba – diseño.

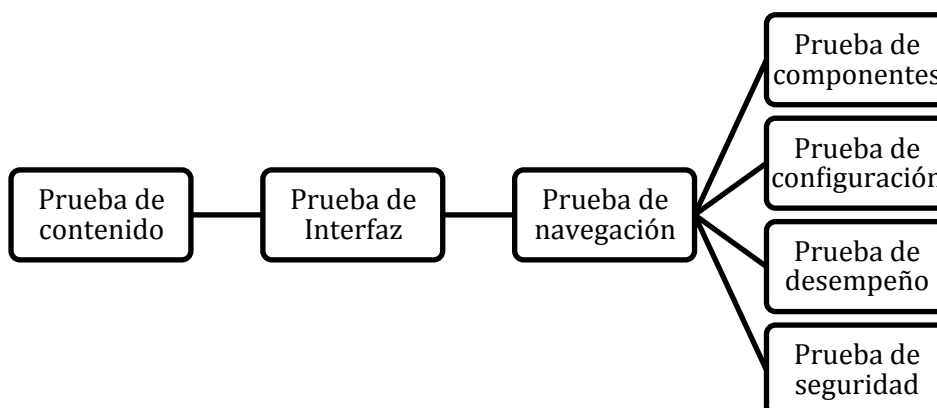


Figura 37. Proceso de prueba – pruebas.

#### 4.5 Prueba de Contenido

Con varios usuarios se ha realizado una revisión a través de la observación y lectura de los siguientes contenidos:

- ✓ **Estático:** Se refiere a la información que permanece estática en la aplicación.
- ✓ **Gramática:** Errores tipográficos o gramaticales.

- ✓ **Semántica:** Información incompleta o ambigua que pueda tener el software.
- ✓ **Orden:** Errores en la organización de la información presentada al usuario.

#### 4.5.1 Pruebas de Usuario

##### 4.5.1.1 Pruebas de Interfaz

- ✓ **Enlaces:** Si cada uno de los enlaces de la aplicación llevan al usuario a donde este lo requiere.
- ✓ **Ventanas Dinámicas:** Si la navegación en cada una de las ventanas garantiza el uso de memoria dinámica para el correcto funcionamiento de la aplicación.

##### 4.5.1.2 Pruebas de facilidad de uso

En este tipo de pruebas se consideraron:

- ✓ **Despliegue:** Que la aplicación use el máximo de la resolución de la pantalla del dispositivo móvil en el que se encuentra.
- ✓ **Interacción:** Menús, enlaces, botones, estética.
- ✓ **Usabilidad:** El grado de dificultad que tiene el usuario para poder usar la aplicación.

##### 4.5.1.3 Pruebas de navegación

Para esta pruebas se ha examinados si todos los enlaces de la aplicación llevan a la pantalla correcta y el contenido deseado por el usuario. Y que sobre todo los enlaces sean entendibles para el usuario en la navegación.

Además se verificaron los nombres de cada uno de los enlaces, que sean significativos para los usuarios, como por ejemplo: a) Iniciar, b) Opciones, c) Ayuda. Con esta prueba se logró que la aplicación sea amigable al usuario.

## **4.5.2 Prueba de componentes**

### **4.5.2.1 Prueba de configuración**

Se ha generado un archivo ejecutable de la aplicación para los sistemas operativos iOS y para la plataforma Android, para cualquier dispositivo que tenga instalado uno de estos sistemas.

### **4.5.2.2 Prueba de seguridad**

La aplicación no implementa ningún sistema de seguridad ya que el objetivo principal de la aplicación es que cualquier persona tenga acceso a ella. Además la aplicación puede ser instalada en cualquier plataforma iOS, Android.

### **4.5.2.3 Prueba de desempeño**

Se han realizado pruebas en diferentes dispositivos iOS, Android, con diferentes capacidades de hardware, manteniendo un nivel de FPS (Frames Per Second) no inferior a los 30 fps que es el nivel recomendado para cualquier aplicación móvil.

## **4.5.3 Deployment de la aplicación**

Para la fase de producción de la aplicación se requiere de lo siguiente:

- ✓ Compilar el software con todos los componentes necesarios para su correcto funcionamiento.

- ✓ Generar el archivo ejecutable según en donde quiera ser instalado .apk para Android que Unity lo genera directamente y .ipa para iOS. En caso de iOS Unity genera un proyecto para Xcode (IDE) y este a su vez genera el archivo ejecutable.

## CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

### 5 Conclusiones

- ✓ La combinación de dispositivos móviles con realidad aumentada da un número ilimitado de oportunidades, ya que el usuario no necesita de un gran equipo computacional para poder experimentar la realidad aumentada en cualquier lugar que se encuentre.
- ✓ El uso de la geolocalización permitió realizar un pequeño sistema de navegación que ayuda al usuario a encontrar los puntos de interés de la aplicación móvil.
- ✓ Unity Engine ha permitido desarrollar una aplicación multiplataforma, usando el lenguaje programación C# que sin la necesidad de reescribir el código fuente se pudo compilar la aplicación móvil para los sistemas operativos requeridos.
- ✓ Vuforia SDK ayudó a que la programación para la realidad aumentada sea fácil y al ser compatible con Unity multiplataforma (iOS, Android). Vuforia SDK esta orientado al reconocimiento de imágenes y no esta diseñado para la detección de objetos físicos, por lo cual se requirió realizar algunas modificaciones para que esta librería sea de utilidad.
- ✓ Para poder compilar la aplicación para los dispositivos iOS se requirió combinar Unity con Xcode. Unity genera un proyecto de Xcode, con el cual se configura y se compila la aplicación móvil.
- ✓ Al utilizar la metodología OOHDM y el patrón de diseño MVC, facilitó el desarrollo de la aplicación y el proceso de control de calidad de la misma.



- ✓ La aplicación QuitoAR no solo puede estar enfocada en el turismo, puede ser usada como un medio de orientación e información de cierta área geográfica, como por ejemplo podría ayudar a los estudiantes nuevos de una institución educativa a encontrar las dependencias de la institución con facilidad.

### **5.1 Recomendaciones**

- ✓ Al desarrollar una aplicación móvil es muy importante tener en cuenta siempre utilizar una programación óptima, ya que los dispositivos móviles cuentan con limitado hardware lo que puede afectar el desempeño de la aplicación.
- ✓ Antes de utilizar un paquete para Unity, se debe revisar si existe documentación del mismo y los comentarios en la web de personas que ya lo hayan utilizado, con esto el tiempo de desarrollo se acorta considerablemente.
- ✓ Ya que Vuforia SDK no está orientado al reconocimiento de edificaciones, se recomienda buscar en el mercado librerías de realidad aumentada especializadas en este tipo de reconocimiento.
- ✓ Se recomienda realizar un paquete de UNITY que maneje la navegación con respecto al usuario y que puede ser fácilmente utilizado en cualquier tipo de proyecto.
- ✓ Como futuro proyecto utilizando esta tecnología, se puede realizar una aplicación que al apuntar la cámara del dispositivo a un texto, el software dé lectura del mismo, lo cual puede ayudar a una persona invidente.

## GLOSARIO DE TÉRMINOS

**Smartphone:** Teléfono celular que posee conexión a internet y permite ejecutar aplicaciones tales como: navegador, editor texto, juegos , etc.

**Aplicación móvil:** Es un software creado para ser ejecutado en plataformas móviles (smart-phone, tablet).

**Sistema operativo:** Es un software que gestiona los recursos de hardware y da servicios mediante los programas de aplicación.

**iOS:** Sistema operativo (software propietario) de los dispositivos móviles producidos por Apple Inc.

**Android:** Sistema operativo de código abierto creado por Google Inc. El cual puede ser instalado en dispositivos de diferente compañía.

**Realidad Aumentada:** Combinación indirecta del mundo digital con el mundo real.

**Geolocalización:** Coordenadas geográficas que representan un punto en el planeta Tierra.

**Marcador:** Imagen con ciertas características visuales, utilizado para cargar modelos tridimensionales en la pantalla del dispositivo móvil.

**Markerless:** Marcador de Realidad Aumentada que no sigue un patrón específico, puede ser una imagen ú objeto.

**Facebook:** Es un sitio web de redes sociales que permite a cualquier persona que tenga una cuenta compartir fotos, comentarios, videos con las personas que desee.

**Twitter:** Es una red social en la cual se pueden escribir mensajes de un máximo de 140 caracteres y compartirlo con sus seguidores.

**Foursquare:** Es un servicio basado en la localización web en donde sus usuarios pueden marcar los lugares que han visitado y compartirlo con sus amigos.

## BIBLIOGRAFÍA

Peralta, E. (2003). *Quito Patrimonio Cultural de la Humanidad*. Quito: Trama.

Empresa Pública Metropolitana Quito Turismo. (2013). *Quito en Cifras*. Quito: Quiti Turismo.

Gottesman, B. (2013). *Readers' Choice Award: PCMag Difital Group*. Retrieved 9 de Febrero de 2014 from PCMag Difital Group: <http://www.pcmag.com/article2/0,2817,2416521,00.asp>

Haag, J. (2013). *Using Augmented Reality for Contextual Mobile Learning*. Learning Solutions Magazine.

Schiller, J., & Voisard, A. (2004). *Location-Based Services*. San Francisco: Morgan Kaufmann.

Cámara Provincial Turismo Pichincha. (1 de Agosto de 2013). *Estadísticas Turísticas*. Retrieved 9 de Febrero de 2014 from captur: [http://www.captur.travel/web2011/estadisticas\\_turisticas/estadistica.html](http://www.captur.travel/web2011/estadisticas_turisticas/estadistica.html)

Pressman, R. (2010). *Software Engineering: a practitioner's approach*. New York: McGraw-Hill.

Noboa, G. (2001). *Reglamento General a la Ley Especial de Telecomunicaciones*. Registro Oficial 404.

Registro Oficial N°. 996. (1992). *Ley Especial de Telecomunicaciones reformada(Ley N°. 184)*.

Asokan, M. (2013). *Android vs iOS - an Analysis*. Nehru Memorial College, PG and Research department of Computer Science. Puthanampatti: ISSN.

Microsoft. (2014). *Novedades en Windows Phone 8*. Retrieved 9 de Febrero de 2014 from windowsphone: <http://www.windowsphone.com/es-es/how-to/wp8/start/whats-new-in-windows-phone>

Ferguson, J., Brian, P., Beres, J., Boutquin, P., & Gupta, M. (2003). *La biblia de C#*. Madris: Ed. Anaya Multimedia.

Apple Inc. (n.d.). *Developer Tool Features*. Retrieved 9 de Febrero de 2014 from Apple Developer: <https://developer.apple.com/technologies/tools/features.html>

Apple Inc. (2014). *Features in iOS*. Retrieved 9 de Febrero de 2014 from Apple Developer: <https://developer.apple.com/technologies/ios/features.html>

Smith, M., & C, Q. (2013). *Unity 4.x Cookbook*. Birmingham: Pack Publishing.

Abramowitz, M., & Stegun, I. (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover.

acrossair company. (2014). *acrossair*. Retrieved 15 de Febrero de 2014 from acrossair:  
[http://www.acrossair.com/acrossair\\_app\\_augmented\\_reality\\_nearesttube\\_london\\_for\\_iPhone\\_3GS.htm](http://www.acrossair.com/acrossair_app_augmented_reality_nearesttube_london_for_iPhone_3GS.htm)

IEEE Org. (2008). *Estándar IEEE STD 830 - 1998*. IEEE.

iSO-FORM, LLC. (2014). *iSO-FORM*. Retrieved 17 de Febrero de 2014 from AR Liver Viewer: <http://www.iso-form.com/apps/ARLiver/>

IKEA Inc. (2014). *IKEA*. Retrieved 17 de Febrero de 2014 from 2014 IKEA Catalogue Comes To Life with Augmented Reality:  
[http://www.ikea.com/ca/en/about\\_ikea/newsitem/2014catalogue](http://www.ikea.com/ca/en/about_ikea/newsitem/2014catalogue)

Apple Inc. (2014). *Apple*. From iPhone 5s: <https://www.apple.com/es/iphone-5s/>

Apple Inc. (2014). *Apple*. From Ipad Air: <https://www.apple.com/es/ipad-air/>

Samsung Group. (2014). *Samgung*. From Samsung Galaxy S4:  
<http://www.samsung.com/us/guide-to-galaxy-smart-devices/galaxy-s-4-smartphone.html>

Samsung Group. (2014). *Samsung*. From Samsung Galaxy Tab 3:  
<http://www.samsung.com/us/mobile/galaxy-tab/SM-T110NDWAXAR>

Rossi, G. S. (2000). *User interface patterns for hypermedia applications*. Palermo, Italia.

Google Inc. (2014). *What it does*. Retrieved 9 de Febrero de 2014 from Google Glass: <http://www.google.com/glass/start/what-it-does/>

Qualcomm Inc. (2014). *Vuforia Developer*. Retrieved 9 de Febrero de 2014 from Vuforia Developer: [www.developer.vuforia.com](http://www.developer.vuforia.com)

Aurasma Inc. (2014). *Supported Devices*. Retrieved 9 de Febrero de 2014 from Aurasma: <http://www.aurasma.com/supported-devices/>

PAR Works Inc. (2014). *PAR Works AR*. From Par Works:  
<http://parworks.com/>

Silva, A. (2002). *Construyendo Aplicaciones Web con una Metodología de Diseño Orientado a Objetos*. Retrieved 2014 from UNAB:  
[http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22\\_art5\\_c.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_c.pdf)

**HOJA DE LEGALIZACIÓN DE FIRMAS**

**ELABORADO POR**

VERA YÁNEZ DANIEL AUGUSTO

---

Sr. Daniel Augusto Vera Yánez

**DIRECTOR DE LA CARRERA**

ING. MAURICIO CAMPAÑA

---

Sr. Ing. Mauricio Campaña

Sangolquí, Febrero del 2014