

UNIVERSIDAD POLITECNICA SALESIANA
SEDE CUENCA

FACULTAD DE INGENIERIAS

CARRERA DE INGENIERIA DE SISTEMAS

**Tesis previa a la obtención del Título de: Ingeniero de
Sistemas**

**“Análisis, Diseño e Implementación de una aplicación con
realidad aumentada para teléfonos móviles orientada al
turismo”**

AUTORES: **Efraín Galo Cuzco Simbaña.**
 Pablo Rigoberto Guillermo Anguisaca.
 Edison Patricio Peña Guillermo.

DIRECTOR: **Ing. Vladimir Robles.**

Cuenca, 08 de Febrero del 2012

**DECLARATORIA DE
RESPONSABILIDAD:**

Los conceptos desarrollados, los análisis realizados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Cuenca, 02/08/2012

Sr. Galo Cuzco

Sr. Pablo Guillermo

Sr. Edison Peña

Ing. Vladimir Robles

CERTIFICA:

Haber dirigido y revisado prolíjamente
cada uno de los capítulos del informe de
monografía realizada por los Señores
Galo Cuzco S, Pablo Guillermo A y
Edison Peña G.

Cuenca, 02/08/2012

Ing. Vladimir Robles
DIRECTOR

AGRADECIMIENTO:

Agradecemos primeramente a Dios, a nuestros padres, autoridades, profesores de la Universidad Politécnica Salesiana, que durante estos años nos orientaron en el estudio, y de manera muy especial al Ing. Vladimir Robles, quién con gran sentido de responsabilidad orientó eficientemente la presente monografía.

DEDICATORIA:

Quiero dedicar este trabajo de grado a Dios, por todas las bendiciones recibidas, a mis queridos padres María Simbaña P. y Galo Cuzco B., por haberme brindado todo su apoyo sin escatimar sacrificio alguno, a mis hermanos William, Adrián y Jessica, a los pequeños de la casa Danny y Anthony, y en especial a mi pequeño hijo Ismael Cuzco Uzhca, gracias a todos por las palabras de aliento y expresiones de amor recibidas.

La gratitud es un noble sentimiento, por tal razón, también quiero dedicar a mis compañeros de clase Diana, Maritza, Susana, Patricio, José, Edison, Daniel, Saúl, Pablo gracias a todos por su gran apoyo y amistad.

Efraín Galo Cuzco Simbaña

DEDICATORIA:

Mi tesis quiero dedicarla de manera especial a Dios por ser mi guía y fortaleza espiritual, por estar siempre a mi lado como la voz interior que me guía para seguir mi camino hacia un mundo lleno de bondad y sobretodo confianza en mis capacidades.

A mis padres Carmen y José; a mis hermanos Ana, Juan y Emanuel; a mis amigos Edison, Galo, Saúl, Daniel J, Rubén, Daniel B y Liliana, porque son mi apoyo incondicional, siempre buscaron que salga adelante, entregándome ejemplos de superación y entrega, porque gracias a su apoyo, hoy puedo ver alcanzada una más de mis metas.

A ellos dedico mi esfuerzo y superación, que sin lugar a dudas será empleado para el bienestar de una sociedad justa.

**Pablo Rigoberto Guillermo
Anguisaca**

DEDICATORIA:

Esta tesis se la dedico a Dios, a mis queridos padres Carlos y Marina, que con su esfuerzo, sacrificio y apoyo me han encaminado para seguir siempre adelante.

A mis hermanos Juan, Janeth, sobrinas Belén, María Eduarda y amigos Saúl, Daniel J, Daniel B, Pablo, Galo, Rubén, Patricio, Cecy, Diana, Daysi, Anita, Norma, por sus consejos brindados y sobre todo los momentos que hemos pasado juntos harán que siempre los lleve en mi corazón.

Edison Patricio Peña Guillermo

INDICE

CAPITULO 1	1
INTRODUCCION	2
1.1 Antecedentes	4
1.2 Situación Actual	7
1.3 Justificación.....	8
1.4 Objetivos	8
General	8
Específicos	9
CAPITULO 2	10
SISTEMAS OPERATIVOS PARA MÓVILES	11
2.1 Introducción y fundamentos teóricos	11
2.1.1 Introducción	11
2.2 Tipos de OS	12
2.2.1 Fundamentos	12
Android.....	12
RIM Blackberry.....	13
iOS.....	15
Symbian OS	16
Windows phone OS.....	17
2.2.1.1 Características	18
2.2.1.2 Compatibilidad.....	20
Android	20
RIM Blackberry	20
iOS	20
Symbian	21
Windows Phone.....	21
2.2.1.3 Funcionalidades.....	22
2.2.1.4 Ventajas	23
2.2.1.5 Desventajas	26
Android	26
iOS	27
RIM Blackberry	27

Symbian	27
Windows Phone.....	27
2.2.2 Comparación de Funcionalidades	28
2.2.3 Justificación de la selección de los OS	28
CAPITULO 3	30
LENGUAJES DE PROGRAMACIÓN Y HERRAMIENTAS DE DESARROLLO	31
3.1 Conceptos Básicos	31
3.2 Tipos.....	31
Lenguajes de programación	31
JAVA.....	34
C	34
C++.....	35
Objetive-C	36
Lua.....	36
Herramientas de desarrollo	37
iOS SDK	37
Android SDK	38
Wikitude SDK.....	38
QCAR SDK.....	39
Unifeye Mobile SDK	39
Corona SDK.....	40
Librerías de desarrollo.....	41
Look	41
Cocoa Touch.....	41
ARToolKit	42
ATOMIC Authoring Tool	42
ATOMIC Web Authoring Tool.....	42
OSGART	43
ARTag	43
ARToolKitPro	43
Studierstube Tracker.....	43
FLARToolKit	43
NyARToolkit	44

ARDesktop.....	44
AndAR	44
ARToolKit Plus	44
3.2.1 Características	45
Lenguajes de programación	45
JAVA.....	45
C	45
C++.....	46
Objetive-C	47
Lua.....	48
Herramientas de desarrollo	48
iOS SDK	48
Android SDK	49
Wikitude SDK.....	50
QCAR SDK.....	50
Unifeye Mobile SDK	51
Corona SDK.....	51
Librerías de desarrollo.....	52
Look	52
API CocoaTouch	52
NyARToolkit.....	53
Artoolkit Plus	53
3.2.2 Funcionamiento.....	54
Lenguajes de programación	54
JAVA.....	54
C	55
C++.....	55
Objetive-C	56
Lua.....	57
3.2.3 Compatibilidad.....	57
Lenguajes de programación	57
Herramientas de desarrollo: SDK	58
Librerías de desarrollo.....	58

3.3 Análisis y comparación: Ventajas y desventajas	59
3.4 Selección del lenguaje de programación para realidad aumentada.....	60
3.5 Introducción al lenguaje de programación seleccionado	60
3.5.1 Funciones básicas.....	61
JAVA.....	61
Objective-C	61
3.5.2 Manejo de conexiones con base de datos.....	61
3.5.3 Manejo de animaciones.....	62
CAPITULO 4	63
ANALISIS, DISEÑO DE MARCAS Y MODELOS DE INFORMACION	64
4.1 Análisis de los sitios turísticos de la ciudad de Cuenca	64
Iglesia de Santo Domingo	64
Iglesia Carmen de la Asunción.....	65
Iglesia de San Francisco.....	66
Iglesia la Catedral Vieja	67
Monumento de Abdón Calderón	67
Monumento del Padre Carlos Crespi	68
Monumento de Simón Bolívar	69
4.2 Diseño y creación de las marcas de reconocimiento.....	69
Diseño	70
Creación	70
4.3 Diseño y creación de los modelos de información.....	73
Diseño	73
Creación	74
4.4 Pruebas de las marcas de reconocimiento creadas	75
Estrategia.....	75
Resultado.....	76
CAPITULO 5	79
ANALISIS, DISEÑO E IMPLEMENTACION DE LA APLICACIÓN EN REALIDAD AUMENTADA	80
5.1 Análisis y diseño de la aplicación en realidad aumentada	80
5.1.1 Análisis de los requerimientos de la aplicación	80
Especificación de requerimientos funcionales y no funcionales.....	80

Requerimientos funcionales	80
Reconocimiento de marcas	80
Visualización de la información.....	80
Manual de usuario para la utilización de la aplicación	81
Descripción a detalle sobre los requerimientos funcionales	81
Requerimientos no funcionales	83
Fiabilidad	83
Facilidad de uso.....	83
5.1.2 Diseño del modelado de datos para la aplicación	84
Diagrama de Clases.....	84
Diagrama de Contexto.....	85
5.2 Instalación y configuración de la base de datos	86
5.2.1 Instalación de la base de datos	86
Instalación	86
5.2.2 Configuración y pruebas de la base de datos	88
Configuración.....	88
Pruebas	90
5.3 Instalación y configuración del lenguaje de programación y entorno de desarrollo	91
5.3.1 Instalación del lenguaje de programación y el entorno de desarrollo	91
Android	91
Instalación del SDK	92
Instalación de Eclipse.....	92
Instalación del plugin ADT	92
iOS	93
5.3.2 Configuración del lenguaje de programación y el entorno de desarrollo	94
Android	94
Configuración de las variables de entorno para el lenguaje de programación.	94
Configuración del plugin ADT	95
iOS	96
5.4 Implementación de los módulos de la aplicación	96
5.4.1 Programación del módulo de descarga de actualización y nuevas marcas	96
5.4.2 Programación del módulo principal	97

5.4.2.1 Programación de la interfaz entre la base de datos y la aplicación	97
5.4.2.2 Programación de la interfaz entre la aplicación y el entorno	97
5.4.2.3 Programación para el reconocimiento de las marcas	98
5.4.2.4 Programación para la interacción entre la base de datos y las marcas.....	99
Android	100
iOS	101
5.5 Desarrollo de manuales de usuario y documentación	102
CAPITULO 6	103
ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL PORTAL WEB	104
6.1 Análisis de los requerimientos del portal	104
6.2 Diseño del modelado de datos para el portal	104
6.3 Implementación del portal web.....	105
6.3.1 Implementación del módulo de descarga de la aplicación.....	105
6.3.2 Implementación del módulo de descarga de marcas	106
6.3.3 Implementación del módulo de actualización de marcas.....	106
6.4 Desarrollo de manuales de usuario y documentación	107
CAPITULO 7	108
INSTALACIÓN Y PRUEBAS DE LA APLICACIÓN EN EL TELÉFONO MÓVIL	109
7.1 Instalación de la aplicación	109
Android	109
iOS	110
7.2 Pruebas de usabilidad.....	113
7.3 Pruebas de funcionalidad	116
7.4 Pruebas de fiabilidad.....	116
7.5 Depuración y corrección de errores	117
7.5.1 Corrección de errores	117
7.5.2 Depuración de la aplicación	117
7.6 Desarrollo de manuales de instalación.....	118
Conclusiones	119
Recomendaciones.....	121
Referencias	122
ANEXO 1	131

ANEXO 2.....	150
ANEXO 3.....	158
ANEXO 4.....	167

CAPITULO 1

INTRODUCCIÓN

1 Capítulo I INTRODUCCION

En la actualidad la importancia que tienen los avances tecnológicos para la humanidad ha alcanzado niveles sorprendentes, un caso claro de ello lo constituyen los dispositivos móviles, que se han convertido en el mayor ejemplo de convergencia tecnológica¹. Ahora no adquirimos un teléfono móvil simplemente por sus bondades a la hora de hacer llamadas, se tienen en cuenta otros aspectos importantes como: la resolución de su cámara, la capacidad de almacenamiento de archivos MP3, si tiene o no acceso a Internet, la posibilidad de usar aplicaciones con alto rendimiento gráfico, servicio de posicionamiento global (GPS), etc.

En esta última década se han venido desarrollado aplicaciones que a través de un dispositivo móvil ofrecen una visión del mundo real, por medio de las imágenes capturadas por la cámara que se mezclan con una capa virtual en la que se puede visualizar desde información sobre nuestro entorno real hasta gráficos 3D. A esta tecnología la denominamos realidad aumentada y poco a poco está llegando a nuestra vida diaria.

Como hemos comentado para mostrar la realidad aumentada hace falta un dispositivo y como mínimo este debe contar con una cámara y con un cierto nivel computacional.

Este proyecto tiene como objetivo crear una aplicación que utilice la Realidad Aumentada, de manera que podamos observar lo que captura la cámara de vídeo con una capa virtual superpuesta, y donde al aproximarse a los marcadores creados para la aplicación, aparezca la información descriptiva de dicho objeto. En nuestro caso estos marcadores pertenecerán a los lugares turísticos más importantes de la ciudad de Cuenca.

¹ Convergencia Tecnológica.- Es la tendencia de diferentes sistemas tecnológicos en la evolución hacia la realización de tareas similares.

Las plataformas móviles, elegidas para el desarrollo de la aplicación de realidad aumentada, serán elegidas de acuerdo a un análisis que se realizará en capítulos posteriores.

El proyecto está dividido en 7 capítulos que se estructuran de la siguiente manera:

- En el primer capítulo haremos una breve introducción a lo que es la realidad aumentada y el proyecto a desarrollar.
- En el segundo, explicaremos los sistemas operativos disponibles para teléfonos móviles, y así mismo justificaremos la selección de los sistemas operativos para los cuales desarrollaremos la aplicación.
- En el tercero, haremos un estudio de los lenguajes de programación para poder desarrollar aplicaciones en teléfonos móviles, como también herramientas y librerías que nos permitirán desarrollar realidad aumentada.
- En el cuarto haremos un análisis de los principales lugares turísticos de la ciudad de Cuenca, y procederemos a la creación de las marcas de reconocimiento.
- En el quinto realizaremos la implementación de la aplicación.
- En el sexto procederemos a la creación del portal web destinado a descargas y actualizaciones de la aplicación.
- En el séptimo nos concentraremos en lo que se refiere a las diferentes pruebas de funcionalidad de la aplicación.

1.1 Antecedentes

La Realidad Aumentada (RA) es una tecnología que permite agregar elementos virtuales a un entorno real, mediante la utilización de un conjunto de dispositivos como son la cámara, una computadora y el monitor, para ampliar la información que el usuario percibe del ambiente que lo rodea.

Es decir, con la RA conseguimos agregar información virtual a la información ya existente en el mundo real, que ya es conocida e interpretada por el usuario. En tal virtud, la RA no pretende sustituir la realidad física sino que sobreimprime los datos generados por un computador, agregándolos al mundo real del usuario.

El usuario está rodeado por mucha información, la cual muchas veces es difícil de interpretar y/o simular, es por ello que los ambientes creados con realidad virtual pueden llegar a ser muy simples y con falta de información del entorno que se pretende modelar, es en este aspecto donde existe una gran ventaja de la realidad aumentada sobre la realidad virtual, ya que el entorno del usuario, que está lleno de información no se altera, y en lugar de ello se amplía con conocimientos y/o información que retroalimentan la escena que se pretende representar [1].

Dentro del marco turístico de la ciudad de Cuenca se ha visto la necesidad de desarrollar una aplicación para dispositivos móviles que integre tecnología de realidad aumentada, para proveer al usuario una descripción de los lugares turísticos de Cuenca a los que visite, debido a que no se cuenta siempre con un guía turístico o alguien que proporcione información acerca del sitio.

Mediante el desarrollo de esta aplicación se pretende solucionar este inconveniente. La aplicación instalada en un dispositivo móvil, será la encargada de capturar, mediante la cámara, el entorno real y realizar el

reconocimiento de los marcadores en tiempo real y ofrecer al usuario la información necesaria del sitio que este visitando y tenga diseñado una marca para este fin.

Para poder implementar la aplicación que use la RA es necesario el uso de componentes como son:

Monitor: Dispositivo donde se verá reflejado la combinación de lo real y lo virtual que conforman la realidad aumentada, el monitor de nuestro proyecto será la pantalla del dispositivo móvil.

Cámara Web: Dispositivo que toma la información del mundo real y la transmite al software de realidad aumentada. En la actualidad, los dispositivos móviles, por lo general, todos incorporan cámaras de alta resolución.

Software: Programa que toma los datos reales y los combina con datos virtuales dependiendo de algún tipo de condición o interpretación, convirtiéndolos en realidad aumentada.

Marcadores: “Los marcadores son básicamente hojas de papel con símbolos que el software interpreta y de acuerdo a un marcador específico realiza una respuesta específica, mediante el software”².

En el proyecto que se pretende desarrollar, los marcadores servirán para ofrecer información a los usuarios de los lugares turísticos de Cuenca, para los cuales se hayan creado los respectivos marcadores.

² BETANCOURTH Santiago Bernal, ¿Qué es la realidad aumentada?, 29/10/2009, p.1, <http://www.maestrosdelweb.com/editorial/que-es-realidad-aumentada>, 18/06/2011.

A continuación se muestra una imagen (ver figura 1.1), donde se observa cómo debe funcionar la aplicación en el dispositivo móvil, esta se presenta de una manera muy general:

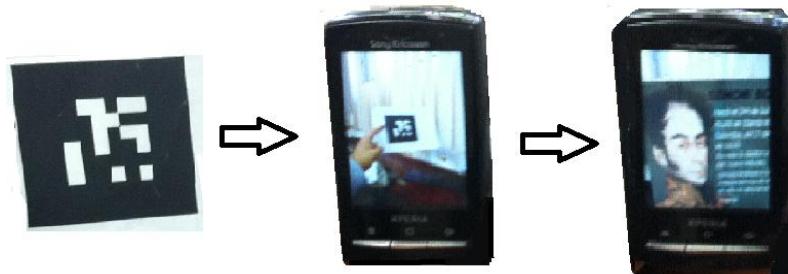


Figura 1.1 Funcionamiento de la aplicación de RA

Como se aprecia en la imagen, se puede ver el marcador, que mediante la cámara del teléfono, es capturado y se obtiene la información correspondiente que se muestra en la pantalla del mismo.

Además del campo turístico, la RA tiene diversos campos de aplicación y sirven de apoyo a varias disciplinas y servicios: mantenimiento y reparación industrial, aprendizaje, entretenimiento, guía turística, localización GPS, publicidad, negocios. Es así que, ahora podemos encontrar sistemas con RA que nos ilustren como realizar un mantenimiento a nuestro vehículo, o que nos indiquen el lugar donde nos encontramos y que nos guíen por la ruta hasta llegar a un determinado destino.

El desafío de nuestro proyecto es fomentar la competitividad de la actividad turística, mediante una aplicación innovadora con el uso de una tecnología aun no muy aplicada en nuestro medio como lo es la RA en teléfonos móviles. Asimismo es posible aprovechar muchos dispositivos ya existentes (por ejemplo: PDAs, TabletPC). Estos dispositivos integran los diversos componentes requeridos y pueden ser adaptados para incorporarles contenidos con realidad aumentada, lo que ayudará a posicionar al turismo como eje estratégico del desarrollo económico.

1.2 Situación Actual

Dentro del marco turístico en el que se encuentra la ciudad de Cuenca, se ha visto la problemática de que los turistas no tienen una correcta descripción del lugar que están visitando. Por experiencia propia, al visitar otro lugar, nos agrada conocer un poco de la historia que encierra un determinado lugar, como por ejemplo al visitar una catedral, sería interesante saber el nombre de la misma, el año de fundación, entre otra información importante. A pesar de que existen páginas web con esta información, es un proceso más tedioso realizar la búsqueda en internet, por lo que podemos contar con tecnologías como la RA, que nos pueden hacer la vida más sencilla, evitando la pérdida de tiempo y posibles errores que el usuario pueda cometer al momento de la búsqueda.

Al vivir en la era de la información y gracias a los avances tecnológicos es posible la utilización de la RA, permitiéndonos desarrollar aplicaciones innovadoras como el proyecto que se pretende implementar.

Otro inconveniente es la carencia de software libre o privativo que permitan la visualización de información en tiempo real utilizando un teléfono móvil, el cual sea fácil de usar y permita recuperar la información de una forma rápida y eficiente.

Es importante mencionar que en la actualidad el proceso de promoción de los principales lugares turísticos de Cuenca se lo hace mediante páginas webs, en las que el usuario realiza las búsquedas respectivas y obtiene información descriptiva del lugar.

Por ello, se ha visto la necesidad de proveer una aplicación que permita realizar una visualización más efectiva, eficiente y amigable en tiempo real de dichos lugares. Se ha optado por la utilización de software libre, para la implementación del sistema que permita visualizar información descriptiva en

tiempo real, de un objeto seleccionado utilizando un teléfono móvil.

1.3 Justificación

En la actualidad en la ciudad de Cuenca no existe ninguna aplicación que brinde información descriptiva de un lugar u objeto específico en tiempo real, que se encuentran dentro de los sectores turísticos de la ciudad, y que permita al usuario la visualización de una forma clara, rápida y sencilla.

Hasta el momento lo que se dispone son descripciones estáticas, cargadas en sitios Web, aunque esta información se la pueda acceder en tiempo real a través del navegador web de un dispositivo móvil, el usuario dispondrá de esta información de manera global, sin reconocer rápidamente la información que pertenece a cada objeto (parques, monumentos, iglesias, etc.).

En vista de la situación mencionada, planteamos como solución la implementación de una aplicación de realidad aumentada, que permita la visualización de la descripción de los diferentes objetos por medio de la captura de marcas, mediante la cámara de un dispositivo móvil en tiempo real, que servirá como herramienta para fomentar el turismo de nuestra ciudad.

1.4 Objetivos

a) General:

- Análisis, diseño e implementación de una aplicación con realidad aumentada en teléfonos móviles orientada al turismo.

b) Específicos:

- Crear una aplicación que sirva como herramienta para promocionar el turismo en la ciudad de Cuenca.
- Dar a conocer los lugares turísticos de Cuenca a partir de las marcas captadas en tiempo real con un dispositivo móvil.
- Analizar el mejor sistema operativo de los dispositivos móviles que permitan implementar realidad aumentada.
- Familiarizarse con el uso de librerías que permitan implementar realidad aumentada.
- Analizar las librerías existentes para la implementación de la aplicación con realidad aumentada.
- Diseñar las marcas de reconocimiento necesarias para la implementación de la aplicación.
- Implementar la aplicación de realidad aumentada.
- Crear un portal web para la actualización y descarga de las marcas y sus respectivas descripciones virtuales.
- Implementar el complemento de descarga de la aplicación en el portal web.

CAPITULO 2

SISTEMAS OPERATIVOS PARA MÓVILES

2 Capítulo II SISTEMAS OPERATIVOS PARA MÓVILES

2.1 Introducción y fundamentos teóricos

2.1.1 Introducción

Estamos rodeados de tecnología que evoluciona día tras día y el sector de los dispositivos móviles no es la excepción. Al momento en este campo se está dando el mayor cambio de su historia en cuanto a hardware y software se refiere, ya que ahora es posible tener dispositivos con capacidades de realizar tareas o actividades mucho más complejas que antes solo era posible ejecutar en computadores de escritorio o portátiles.

Esta revolución de los dispositivos móviles hace que cada día, estos cuenten con mejores procesadores que aumenten el rendimiento y sistemas operativos que ofrezcan una mayor funcionalidad.

Hasta el momento se han desarrollado varios sistemas operativos entre los que destacan Microsoft Windows Mobile, Blackberry OS, Symbian OS, iOS que son software propietario, pero también existe una plataforma libre basada en GNU/Linux y con licencia GPL (General Public License) llamada Android.

Esta gran variedad de sistemas operativos hace que resulte difícil inclinarnos por uno o por otro, más adelante en este capítulo se hablará de cada uno de ellos y de sus principales características, funcionalidades, ventajas y desventajas, que nos ayudará a seleccionar los sistemas operativos adecuados para los cuales se realizará la implementación de la aplicación.

2.2 Tipos de OS

2.2.1 Fundamentos

Para conocer acerca de los diferentes sistemas operativos con los que contamos y en cuales podemos implementar la RA, exponemos a continuación una descripción de los sistemas operativos más adecuados para nuestro proyecto. Esto nos permitirá seleccionar el sistema con mejores prestaciones y características para el desarrollo del presente trabajo.

A continuación se hará una pequeña explicación de cada uno de estos OS:

Android



Figura 2.1 Imagen del logo representativo del OS Android [2].

Sistema operativo basado en GNU/Linux diseñado originalmente para dispositivos móviles, el cual fue desarrollado inicialmente por Android Inc., una empresa que fue comprada por Google en el 2005 [3].

Como Android se encuentra basado en software libre, cuenta con una gran cantidad de desarrolladores que se encuentran creando nuevas aplicaciones, para añadir más funcionalidades a los dispositivos.

La mayoría de aplicaciones son programadas a través del Android SDK³, el cual utiliza como base el lenguaje Java, aunque se puede ejecutar código C directamente a través del NDK⁴.

A continuación se presentará una tabla con la información general del OS Android:

Última versión estable	2.3.4 Gingerbread ⁵
Escrito en	C (núcleo), C++ (algunas bibliotecas de terceros), Java (UI)
Núcleo	Linux
Tipo de núcleo	Monolítico ⁶
Licencia	GNU GPL 2 ⁷
Estado Actual	En desarrollo
Idioma	Multilenguaje

Tabla 2.1 Información General del OS Android [3].

RIM Blackberry



Figura 2.2 Imagen del logo representativo del OS RIM Blackberry [2].

³SDK.- Es un kit de desarrollo de software, que contiene archivos de cabecera, bibliotecas, muestras, documentación y herramientas que utilizan las API necesarias para desarrollar aplicaciones.

⁴NDK.- Es un conjunto de herramientas que permite incorporar los componentes que hacen uso de código nativo en las aplicaciones de Android.

⁵Gingerbread.- Este es el nombre de la versión 2.4 de Android.

⁶Monolítico.- Es un tipo de núcleo o kernel de un sistema operativo. Como ejemplo de sistema operativo de núcleo monolítico están UNIX, Linux.

⁷GNU GPL 2.- GNU General Public License protege la libre distribución, modificación y uso de software. Su propósito Es declarar que el software cubierto por esta licencia es software libre

El Blackberry OS es un sistema operativo móvil, utilizado por la línea de teléfonos móviles del mismo nombre Blackberry, desarrollado por una compañía canadiense fundada en 1984 llamada RIM (Research In Motion) y está orientada al uso profesional [4].

El sistema operativo de Blackberry, trabaja en un entorno multitarea esto permite un uso intensivo de los diferentes dispositivos de entrada adoptados por la empresa, como son en particular la rueda de desplazamiento y el trackpad⁸, entre otras funcionalidades permitir el acceso a correo electrónico, navegación web y sincronización con programas como por ejemplo, Microsoft Exchange. El Blackberry OS está diseñado para dar soporte a Java MIDP⁹ 1.0 y WAP¹⁰ [5].

A continuación se presenta una tabla con el resumen de la información más importante del Blackberry OS:

Última versión estable	7.0 (Blackberry Bold, Agosto del 2011)
Escrito en	Java, C++
Tipo de núcleo	Basado en Java
Interfaz gráfica por defecto	GUI ¹¹
Plataformas soportadas	Línea de smartphones de BlackBerry
Licencia	Propietaria
Estado Actual	Activo

Tabla 2.2 Información General del OS RIM Blackberry [4].

⁸Trackpad.- Dispositivo táctil que permite controlar un cursor a través desde cualquier interfaz gráfica.

⁹MIDP.-Mobile Information Devicepro file es una versión de J2ME (Java 2 Micro Edition) integrada en el hardware de celulares relativamente modernos.

¹⁰WAP.- Wireless Application Protocol es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas.

¹¹GUI.- Proporciona un entorno visual y sencillo para permitir la comunicación con el sistema operativo de un computador y otros dispositivos.

iOS



Figura 2.3 Imagen del logo representativo de iOS [2].

Apple Inc., una empresa la cual se caracteriza por desarrollar software y hardware muy novedosos, lanza al mercado de los móviles un sistema operativo llamado iOS (iPhone OS).

iOS es un sistema operativo móvil, que fue desarrollado inicialmente para el iPhone, y después usado en el iPod Touch e iPad [6].

La mayoría de aplicaciones desarrolladas para iOS, son programadas a través del iOS SDK, el cual utiliza como base el lenguaje de programación Objective-C.

A continuación se presentará una tabla con la información general del iOS:

Última versión estable	4.3.5, (iPad e iPhone-iPod Touch respectivamente)
Escrito en	C, C++, Objective-C
Tipo de núcleo	Núcleo híbrido
Licencia	APSL ¹² y Apple EULA ¹³
Idiomas	Multilenguaje
Estado actual	Activo

Tabla 2.3 Información General del iOS [6].

¹²APSL.- Es la licencia de código fuente abierto y licencia de software libre.

¹³EULA.- Es una licencia por la cual el uso de un producto sólo está permitido para un único usuario.

Symbian OS



Figura 2.4 Imagen del logo representativo de Symbian OS [2].

Es un sistema operativo multitarea de 32 bits diseñado para los teléfonos móviles, planteado para ejecutar aplicaciones en un entorno con recursos reducidos, es decir, está diseñado especialmente para su uso en pequeños dispositivos alimentados con baterías [7].

Para ofrecer la máxima robustez y una rápida capacidad de respuesta su kernel¹⁴, posee una arquitectura de micro-kernel que contiene solo lo mínimo de órdenes primitivas básicas, que es altamente modular y ofrece numerosas APIs (Application Programming Interfaces) para el desarrollo de aplicaciones de comunicaciones y soporta los principales estándares de la industria inalámbrica¹⁵.

Symbian ofrece distintos lenguajes de programación para el desarrollo de aplicación Visual Basic, Java, OPL y C++, siendo esta última su lenguaje nativo [8].

A continuación se presenta una tabla con el resumen de la información más importante del Symbian OS:

Última versión estable	Symbian Belle (Symbian OS 10.1)
Núcleo	Micro-kernel

¹⁴Kernel.- Se refiere al núcleo de un sistema operativo .

¹⁵Estándares inalámbricos.- WAP, XHTML, J2ME, MIDP, MMS, Bluetooth, GPRS, CDMA, SyncML, IPv6, IPsec

Interfaz gráfica por defecto	S60 ¹⁶ , UIQ ¹⁷ , MOAP ¹⁸
Licencia	NSL ¹⁹
Idiomas	Multilenguaje
Estado actual	En desarrollo

Tabla 2.4 Información General del OS Symbian [9].

Windows phone OS



Figura 2.5 Imagen del logo representativo de Windows phone OS [2].

Windows Phone llamado al inicio como Windows Mobile, es la propuesta de la empresa Microsoft dirigida por Bill Gates [2].

Este OS cuenta con una interfaz natural de usuario, es decir permite la interacción entre el usuario y el sistema directamente sin la intervención de un dispositivo de entrada.

Se basa en el núcleo del OS Windows CE²⁰ lo que le permite contar con un repositorio de aplicaciones básicas utilizando las API²¹ de Microsoft Windows [10].

¹⁶S60.-Interfaz de usuario de serie 60 es una plataforma para terminales móviles que utilicen el sistema operativo Symbian OS.

¹⁷UIQ.- Es una plataforma para terminales móviles, basada en el sistema operativo Symbian. Se caracteriza por agregar soporte para pantallas táctiles

¹⁸MOAP.-Es compatible con el sistema operativo Symbian. A diferencia de la Serie 60 y UIQ , MOAP no es una plataforma de desarrollo abierto.

¹⁹NSL.- Licensia Nokia Symbian es una licencia por la cual el uso de un producto, se tendrá que pagar una licencia.

²⁰Windows CE.- Es un sistema operativo desarrollado por Microsoft para sistemas diseñados para realizar una o algunas pocas funciones.

²¹API.- Es el conjunto de funciones y procedimientos o métodos, que ofrece cierta biblioteca para ser utilizado por otro software.

Su diseño tanto estético, organizacional y funcional es similar a la de una máquina de escritorio con OS Windows.

A continuación se presentará una tabla con la información general del OS Windows Phone:

Última versión estable	7
Escrito en	C++
Núcleo	Windows CE
Licencia	Microsoft CLUF (EULA)

Tabla 2.5 Información General del OS Windows Phone [10].

2.2.1.1 Características

A continuación se presentará una tabla con las características que sobresalen en cada uno de los sistemas operativos en estudio:

OS	Características
Android	<ul style="list-style-type: none">• Sus principales entornos de desarrollo son Eclipse y Netbeans.• Almacenamiento removable.• Integración con redes sociales.• Organización de Aplicaciones por carpetas.• Market de Aplicaciones (más de 100.000 aplicaciones)• Soporta Widgets
	<ul style="list-style-type: none">• Orientado a un uso profesional (administrar correo electrónico y

RIM Blackberry	<p>agenda).</p> <ul style="list-style-type: none"> • Pueden crearse programas independientes básicos. • Compatibilidad con redes sociales.
iOS	<ul style="list-style-type: none"> • Interfaz natural de usuario (multitáctil²²). • Organización de Aplicaciones por carpetas. • Market de Aplicaciones (más de 300.000 aplicaciones). • Interfaz intuitiva.
Symbian	<ul style="list-style-type: none"> • Su arquitectura es altamente modular (dividido en capas). • Diseñado para ahorro de energía. • Soporte a Widgets
Windows phone	<ul style="list-style-type: none"> • Apariencia similar a las versiones de escritorio de Windows. • Posee Microsoft office móvil nativamente. • Es relativamente el más nuevo

Tabla 2.6 Características de los sistemas operativos.

²²Multitáctil.- Es el nombre con el que se conoce a una técnica de interacción persona-computador y al hardware que la implementa.

2.2.1.2 Compatibilidad

Es importante conocer la compatibilidad que tiene un sistema operativo con el hardware que posee el dispositivo, para así aprovechar todas las potencialidades que tienen tanto los dispositivos como un determinado sistema operativo.

A continuación, se presentan a los sistemas operativos objetos de nuestro estudio, con una lista de sus principales marcas de celulares con los que actualmente son compatibles.

Android

En la actualidad hay una amplia gama de marcas de celulares que soportan Android, entre las que se destacan:

1. Samsung.
2. HTC.
3. Motorola.
4. Sony Ericsson.
5. LG.

RIM Blackberry

Este sistema operativo es compatible solo con la gama de productos Blackberry.

iOS

Este sistema operativo es compatible solo con los diferentes dispositivos desarrollados por Apple como son:

1. iPhone
2. iPod Touch
3. iPad

Symbian

Este sistema operativo es compatible con una amplia gama de marcas de celulares entre las cuales se encuentran:

1. Nokia
2. Sony Ericsson,
3. PSION
4. Samsung
5. Siemens
6. Arima
7. Benq
8. Fujitsu
9. Lenovo
10. LG
11. Motorola
12. Mitsubishi Electric
13. Panasonic
14. Sharp

Windows Phone

En el mercado se encuentran a disposición las siguientes marcas con sistema operativo Windows Phone:

1. Samsung.
2. LG.

3. HTC.
4. Acer.
5. Fujitsu.
6. ZTE.
7. Nokia.

2.2.1.3 Funcionalidades

A continuación se presentará una tabla con las principales funcionalidades de cada uno de los sistemas operativos en estudio:

OS	Funcionalidad
Android	<ul style="list-style-type: none"> • Posee soporte a Flash, Java y html5. • Soporta varias tecnologías de conectividad. • Es multitarea. • Sincronización multimedia (Transferencia directa de archivos y Aplicaciones de terceros)
RIM Blackberry	<ul style="list-style-type: none"> • Es multitarea. • Tiene soporte a Java y html5. • Compatible con redes CDMA²³. • Se limita el uso de ciertas funcionalidades del OS.
	<ul style="list-style-type: none"> • No soporta páginas creadas con Adobe Flash o Java.

²³ CDMA.- Tecnología de comunicación que emplea el acceso múltiple por división de código.

iOS	<ul style="list-style-type: none"> • Soporta HTML5²⁴. • Es multitarea (Desde iOS 4 o superior). • Sincronización multimedia (iTunes)
Symbian	<ul style="list-style-type: none"> • Es multitarea. • Soporte JAVA. • Orientado a comunicaciones.
Windows phone	<ul style="list-style-type: none"> • Es Multitarea. • No tiene soporte a Java. • Tethering (usar el teléfono como modem para compartir internet) • Sincronización con PC

Tabla 2.7 Principales funcionalidades de los sistemas operativos.

2.2.1.4 Ventajas

Luego de haber visto los principales fundamentos teóricos, de los sistemas operativos más usados en los dispositivos móviles, hemos visto conveniente realizar la comparación de las ventajas que cada plataforma presenta con respecto a las otras, las cuales detallaremos en la siguiente tabla:

²⁴HTML5.- HyperTextMarkupLanguage, versión 5, es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML.

	Android	RIM Blackberry	iOS	Symbian	Windows Phone
Android	<ul style="list-style-type: none"> - Código abierto. - Gran cantidad de desarrolladores. - Amplia gama de dispositivos con Android. 	<ul style="list-style-type: none"> - Multitarea desde la versión 1. - Código abierto. - Gran comunidad de desarrolladores. - Amplia gama de dispositivos con Android. - Memoria expandible. 	<ul style="list-style-type: none"> - Interfaz y uso. - Gran cantidad de desarrolladores. - Conexión web más natural. - Interfaces personalizadas. 	<ul style="list-style-type: none"> - Código abierto. - Gran cantidad de desarrolladores. -Amplia gama de aplicaciones. -Costo del producto. 	
RIM Blackberry	<ul style="list-style-type: none"> - Mayor cantidad de Aplicaciones empresariales disponibles. - Mayor uso como herramienta de trabajo. -Seguridad. 		<ul style="list-style-type: none"> - Mayor cantidad de Aplicaciones empresariales disponibles. - Mayor uso como herramienta de trabajo. 	<ul style="list-style-type: none"> - Mayor cantidad de Aplicaciones empresariales disponibles. - Mayor uso como herramienta de trabajo. 	<ul style="list-style-type: none"> - Mayor cantidad de Aplicaciones empresariales disponibles. -- Mayor uso como herramienta de trabajo.

iOS	<ul style="list-style-type: none"> - Actualización automática. - Diseño de los dispositivos. - Seguridad. 	<ul style="list-style-type: none"> - Gran cantidad de aplicaciones. - Conexión web más natural. - Mejor desempeño de Hardware. 		<ul style="list-style-type: none"> - Gran cantidad de aplicaciones. - Conexión web más natural. - Mejor desempeño de Hardware. 	<ul style="list-style-type: none"> - Gran cantidad de aplicaciones. - Hardware superior. - Mejor desempeño de Hardware.
Symbian	<ul style="list-style-type: none"> - Administración mucho más intuitiva. - Facilidad de sincronización con a PC. 	<ul style="list-style-type: none"> - Amplia gama de dispositivos con Symbian. 	<ul style="list-style-type: none"> - Amplia gama de dispositivos con Symbian. 		<ul style="list-style-type: none"> - Amplia gama de dispositivos con Symbian.
Windows Phone	<ul style="list-style-type: none"> - Seguridad - Microsoft Office nativo. 	<ul style="list-style-type: none"> - Uso intuitivo. - Microsoft Office nativo. 	<ul style="list-style-type: none"> - Memoria expandible - Conectividad con diferentes dispositivos. - Microsoft Office nativo. 	<ul style="list-style-type: none"> - Uso intuitivo - Microsoft Office nativo. 	

Tabla 2.8: Tabla comparativa de ventajas entre los sistemas operativos.

2.2.1.5 Desventajas

A continuación presentaremos las desventajas que influyen en cada uno de los sistemas operativos objetos de estudio [11]:

Android

- Duración de la batería es corta, cuando un usuario está acostumbrado a utilizar mucho las aplicaciones.
- Su interfaz es poco intuitiva, al usuario le puede tomar mucho tiempo poder configurarlo.
- Se necesita instalar aplicaciones de terceros para solucionar problemas de funcionalidad.
- Existen problemas de incompatibilidad entre versiones, debido a que el OS Android está totalmente fragmentado.

RIM Blackberry

- Precio.
- El uso para el cual está diseñado por ejemplo (A muchas personas no les interesa manejar correo electrónico).
- Su interfaz de usuario tiene la misma base de experiencia que las versiones anteriores.
- Limitada las funcionalidades a los desarrolladores (Obligatoriamente se necesita certificar).
- No permite instalar aplicaciones desarrolladas por terceros.

iOS

- Precio.
- Su navegador web posee limitaciones para ciertas tecnologías (Flash, Java, Real Media, Windows media).
- Costo del hardware.
- No tiene sincronización inalámbrica.
- Es una plataforma cerrada.

Symbian

- Su funcionamiento es lento.
- Escasas aplicaciones gratuitas disponibles.
- Consume demasiados recursos.

Windows Phone

- No es tan popular en el mercado de nuestro país.
- Es un sistema cerrado.
- No se pueden ejecutar varias aplicaciones simultáneamente [12].

2.2.2 Comparación de Funcionalidades

En la siguiente tabla se realiza una comparación entre las diferentes funcionalidades de los sistemas operativos.

	Android	iOS	Windows Phone	RIM BlackBerry	Symbian
Soporte a Flash	SI	NO	NO	NO	SI
Soporte Java	SI	NO	NO	SI	SI
Soporte html5	SI	SI	NO	SI	NO
Soporta tecnologías de conectividad	SI	SI	SI	SI	SI
Es multitarea.	SI	SI	SI	SI	SI
Sincronización multimedia	NO	SI	NO	NO	NO
Permite acceso a las funcionalidades del OS.	SI	NO	NO	NO	SI
Tethering ²⁵	NO	NO	SI	NO	SI

Tabla 2.9 Comparación de funcionalidades.

2.2.3 Justificación de la selección de los OS

Después de haber revisado todas las características, funcionalidades, ventajas y desventajas de cada uno de los sistemas operativos, se pudo observar el claro domino de Android e iOS.

Además según estadísticas y estimaciones realizadas por la empresa GartnerInc [13] desde el 2010, el sistema operativo Android ha tenido un rápido crecimiento, como se puede apreciar en la tabla siguiente, según estimaciones, para el 2015 los dos sistemas más populares y que estarán en la mayoría de los dispositivos serán Android e iOS, en tal virtud se ha visto conveniente el desarrollo de la aplicación para estos dos sistemas operativos.

²⁵ Tethering.- Proceso por el cual un dispositivo móvil actúa como puente para ofrecer acceso inalámbrico a la red a otros dispositivos.

OS	2010	2011	2012	2015
iOS	14,766	47,964	68,670	138,497
Market Share (%)	83.9	68.7	63.5	47.1
Android	2,502	13,898	26,382	113,457
Market Share (%)	14.2	19.9	24.4	38.6
MeeGo	107	788	1,271	3,057
Market Share (%)	0.6	1.1	1.2	1.0
WebOS	0	2,796	4,245	8,886
Market Share (%)	0.0	4.0	3.9	3.0
QNX	0	3,901	7,134	29,496
Market Share (%)	0.0	5.6	6.6	10.0
Other Operating Systems	234	432	510	700
Market Share (%)	1.3	0.6	0.5	0.2
Total Market	17,610	69,780	108,211	294,093

Source: Gartner (April 2011)

Tabla 2.10: Estimaciones del uso de sistemas operativos [13].

CAPITULO 3

LENGUAJES DE PROGRAMACIÓN Y HERRAMIENTAS DE DESARROLLO

3 Capítulo III LENGUAJES DE PROGRAMACIÓN Y HERRAMIENTAS DE DESARROLLO

3.1 Conceptos Básicos

Al pensar en teléfonos móviles que soportan sistemas operativos tanto Android, como iOS, surge una importante problemática para la persona que programa tales dispositivos, ya que cada uno de ellos está implementado en un lenguaje de programación distinto y utilizan técnicas de desarrollo diferentes.

Para solucionar esta problemática surge la necesidad de su desarrollo en escritorio, que es la forma más común a la hora de programar un teléfono móvil. Esta técnica consiste en realizar la aplicación a través de alguna herramienta, y depurarla sobre sus sistemas operativos, todo esto ejecutado en un PC.

Estas herramientas pueden ser extensiones para los lenguajes y entornos habituales de los PC's o bien entornos de desarrollo específicamente creados para generar código ejecutable sobre los teléfonos móviles.

Los lenguajes de programación son muy complicados de definir, lo que tenemos claro, es que con ellos podemos desarrollar muchas cosas, como aplicaciones militares, juegos, desarrollar bases de datos, etc. El uso que se les puede dar es infinito, aunque existen lenguajes específicamente para cada uno de ellos.

Entre los lenguajes de programación utilizados para los sistemas operativos antes mencionados podemos mencionar Java, C, C++, Objective C, Lua.

3.2 Tipos

Lenguajes de programación

Si queremos comprender los diferentes tipos de los lenguajes de programación, primero se comenzará explicando que es un lenguaje de programación.

El lenguaje de programación es básicamente códigos que son interpretados por un ordenador como órdenes e instrucciones [14].

Las instrucciones son las que inician o finalizan las acciones que debe realizar un dispositivo, ya que el mismo no hace nada por si solo a menos que se le ordene mediante programación inmediata o previa.

Hoy en día la tecnología orientada a objetos ya no se aplica solamente a los lenguajes de programación, además se viene aplicando en el análisis y diseño con mucho éxito.

La programación orientada a objetos es una de las formas más populares de programar y sigue teniendo gran acogida en el desarrollo de proyectos de software [15].

A continuación se presenta una tabla con los mejores lenguajes de programación:

Positio n Aug 2011	Positio n Aug 2010	Delta in Position	Programmig Language	Ratings Aug 2011	Delta Aug 2010	Status
1	1	=	Java	19.409%	+1.42%	A
2	2	=	C	17.390%	-0.48%	A
3	3	=	C++	8.433%	-1.23%	A
4	4	=	PHP	6.134%	-3.05%	A
5	6	↑	C#	6.042%	+1.06%	A
6	9	↑↑↑	Objective-C	5.494%	+2.34%	A
7	5	↓↓	(Visual) Basic	5.013%	-0.40%	A
8	7	↓	Python	3.415%	-0.81%	A
9	8	↓	Perl	2.315%	-1.11%	A

10	11		JavaScript	1.557%	-0.84%	A
11	23		Lua	1.362%	+0.83%	A
12	12		Ruby	1.329%	-0.65%	A
13	10		Delphi/Object Pascal	1.076%	-1.35%	A
14	16		Lisp	0.905%	+0.28%	A
15	22		Transact-SQL	0.823%	+0.27%	A-
16	28		Ada	0.699%	+0.30%	B
17	19		RPG (OS/400)	0.660%	+0.05%	B
18	17		Pascal	0.659%	+0.04%	A--
19	46		F#	0.604%	+0.37%	B
20	-		Assembly*	0.599%	-	B

Tabla 3.1 Listado de los mejores lenguajes de programación [16].

Como se puede ver el primer puesto lo ocupa el lenguaje Java con respecto a otros lenguajes, es el líder en el entorno empresarial. Es el lenguaje que se utiliza principalmente para trabajar en el API de Android.

En segundo puesto tenemos a C, un lenguaje clásico en el mundo de la informática.

Podemos ver que C++ ocupa el tercer puesto, se mantendrá en este puesto por muchos años debido a su gran importancia en aplicaciones hechas con APIs multimedia.

Luego podemos encontrar a PHP, Perl y Python, estos se mantienen estancados debido al medio interés de desarrollar servicios web dinámicos.

También podemos ver que C# está en constante subida, debido a que Microsoft utiliza el IDE Visual Studio y esto ha ayudado muchísimo a su popularidad.

Finalmente podemos observar claramente la subida de Objective-C y Lua, debido al éxito de iOS.

A continuación se dará una breve descripción de los lenguajes de programación más importantes en el desarrollo de aplicaciones para Android e iOS.

JAVA



Figura 3.1 Logo representativo del lenguaje JAVA [17].

Java es un lenguaje de programación orientado a objetos, desarrollado por la Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Las aplicaciones Java están típicamente compiladas en un bytecode²⁶, aunque la compilación en código máquina nativo también es posible [18].

C



Figura 3.2 Letra representativa del lenguaje C [19].

C es un lenguaje de programación creado en 1972 por Dennis M.

²⁶Bytecode.-es un código intermedio más abstracto que el código máquina.

Ritchie como evolución del anterior lenguaje B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. Se trata de un lenguaje débilmente tipificado de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero a su vez dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos [20].

El lenguaje de programación C está estructurado en tres partes fundamentales las cuales son, una librería estándar, un programa compilador y un preprocesador.

La librería estándar no es más que librerías realizadas en código objeto y puede haber sido realizada en otro lenguaje diferente que el C. El programa compilador es el que tiene como función traducir las instrucciones del programa fuente en C. El programa compilador, depura y detecta los posibles errores en el lenguaje fuente, y es capaz de notificar el error ocurrido al programador mediante un mensaje de texto [21].

C++



Figura 3.3 Letra representativa del lenguaje C++ [22].

C++ fue creado con la intención de extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban

admitidos. Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Una particularidad del C++ es la posibilidad de redefinir los operadores y de poder crear nuevos tipos que se comporten como tipos fundamentales [21].

Objetive C

Objective-C

Figura 3.4 Representación del lenguaje Objective-C [23].

Objective-C es un lenguaje de programación orientado objetos creado como un superconjunto de C. En 1992 fue liberado bajo licencia GPL y actualmente se usa como lenguaje de programación en Mac OS X [23].

Lua



Figura 3.5 Representación del lenguaje Lua [24].

Lua es un lenguaje extensible, suficientemente compacto para usarse en diferentes plataformas. La semántica de Lua puede modificada redefiniendo funciones de las estructuras de datos utilizando metatables (diccionario de datos) [25].

Algo interesante sobre Lua es que fue implementado al 100% en ANSI C, por lo que compila en prácticamente cualquier cosa bajo el sol, lo que lo hace una excelente herramienta educacional a nivel universitario para aprender sobre teoría de lenguajes, compilación, interpretación, etc., [26].

Herramientas de desarrollo

Es algo tan sencillo como una interfaz de programación de aplicaciones o API (application programming interface) creada para permitir el uso de cierto lenguaje de programación o puede también incluir hardware sofisticado para comunicarse con un determinado sistema embebido. Las herramientas más comunes incluyen soporte para la detección de errores de programación como un entorno de desarrollo integrado o IDE (Integrated Development Environment) y otras utilidades [27].

iOS SDK



Figura 3.6 Representación del iOS SDK [28].

El iPhone SDK es un Kit de desarrollo de software para la plataforma iOS de Apple Inc., con el objetivo de permitir a terceros desarrollar aplicaciones nativas para el iOS. Como iPhone comparte base con Mac OS X, la cadena de instrumentos para desarrollar aplicaciones para iPhone está también basada en Xcode e incluye compiladores cruzados para el procesador ARM y un emulador de iPhone llamado Aspen. El lenguaje de programación principal para iPhone OS, al igual que en Mac OS, es Objective-C. El SDK se puede descargar gratis, pero para publicar el software es necesario registrarse en el Programa de Desarrollo del iPhone, un paso que requiere el pago y la aprobación por parte de Apple. Durante el proceso, se entregan al desarrollador unas claves firmadas que permiten subir una aplicación a la tienda de aplicaciones de Apple [28].

Android SDK



Figura 3.7 Representación del Android SDK [29].

Android SDK es el kit de desarrollo necesario para programar e implementar todo tipo de aplicaciones para Android, el sistema operativo para teléfonos móviles propuesto por Google. Este paquete o kit de desarrollo incluye las APIs y herramientas necesarias para desarrollar las aplicaciones utilizando JAVA como lenguaje de programación y testear el código, respectivamente [30].

Wikitude SDK



Figura 3.8 Representación del Wikitude SDK [31].

El Wikitude SDK es un kit de desarrollo de software de gran alcance que permite una buena experiencia en el desarrollo de realidad aumentada, proporcionando a los desarrolladores las herramientas para crear ya sea sus propias aplicaciones para Android o mejorar sus actuales aplicaciones.

Wikitude se encuentra disponible para iOS, Android, Symbian y Blackberry.

QCAR SDK



Figura 3.9 Representación del Qualcomm SDK [32].

Qualcomm, una compañía norteamericana conocida por desarrollar CDMA, está ofreciendo plataformas para el desarrollo de aplicaciones que tengan inmersa la posibilidad de la realidad aumentada para celulares con sistema operativo Android e iOS del iPhone, iPods e iPads de Apple. La forma de vivir, de ver las cosas, de procesar información, de relacionarse con otras personas, de comunicarse, de entretenerte ha cambiado tanto gracias a las nuevas tecnologías, que es muy difícil imaginarse cómo va a ser este mundo en algunas décadas. La realidad aumentada es una de ellas [33].

La Plataforma de realidad aumentada de Qualcomm ofrece una amplia gama de experiencias de entretenimiento que involucran e informan a los consumidores mediante una nueva forma de comunicación interactiva. El conjunto de características de la plataforma permite a los desarrolladores construir aplicaciones de realidad aumentada de alto rendimiento y experiencias sobre las imágenes del mundo real, ya sea en medios impresos (libros, revistas, folletos, entradas, carteles) o en los envoltorios o embalajes de producto [34].

Unifeye Mobile SDK



Figura 3.10 Representación del Unifeye Mobile SDK [35].

Metaio lanza durante el MWC2010 su kit de desarrollo de software (SDK) Unifeye Mobile, con una plataforma para creación de aplicaciones de realidad aumentada en dispositivos Android, Symbian, iPhone y Windows Mobile. Aparentemente Unifeye Mobile es el primer kit de desarrollo de software para la creación de aplicaciones de realidad aumentada apoyando las principales plataformas móviles y características de las últimas tecnologías de reconocimiento de imágenes, generación de imágenes 3D para animaciones con interacción en tiempo real y componentes optimizados para el hardware móvil [36].

El código base está preparado para migrar fácilmente a nuevas plataformas que van surgiendo. Incluye marcador basado en 2D y robusto, textura de seguimiento de imágenes arbitrarias. Se puede incluso probar el poder de la SDK, sin necesidad de programación [37].

Corona SDK



Figura 3.11 Logo representativo de la empresa de Corona SDK [38].

El SDK Corona de Ansca Mobile sustituye los frameworks de Apple y el lenguaje de programación Objective-C por el lenguaje de scripts Lua y cuyo uso resulta más natural, equiparable en cierto modo al propio ActionScript empleado en los desarrollos de Flash o al lenguaje JavaScript utilizado en el desarrollo de aplicaciones Web, es más flexible pues nos permite emplear cualquier editor de textos para escribir el programa propiamente dicho, mientras que la ejecución del mismo se realizará sobre un emulador de iPhone que recuerda mucho al que proporciona Apple como parte del conjunto de herramientas dirigidas al desarrollo de aplicaciones para el iPhone, iPod touch e iPad [39].

Librerías de desarrollo

Look



Figura 3.12 Librería look [40].

En su fase Beta “Look” permite el desarrollo de aplicaciones en 2D y 3D con la posibilidad de integrar la imagen con la cámara, permite la persistencia de datos para transmitir estos de forma transparente a servidores externos, proporciona la posibilidad de localización en interiores y permite la interacción con objetos, además nos brinda una completa documentación en español, así como una librería de ejemplos de las aplicaciones que podremos descargar e integrar de forma inmediata a nuestros dispositivos con Andriod [41].

Cocoa Touch

Originariamente, Cocoa era el nombre de una aplicación para niños con la que podían crear sus primeros proyectos multimedia. La marca fue adquirida por Steve Jobs y a partir de entonces se conoce en el ámbito informático para el método de programación en Mac [42].

Cocoa Touch es un API para la creación de programas para el iPad, iPhone y iPod Touch de la compañía Apple Inc. Se basa en el set de herramientas que proporciona el API de Cocoa para crear programas sobre la plataforma Mac OS X [37].

ARToolKit



Figura 3.13 Librería Artoolkit [43].

ARToolKit es una biblioteca que permite la creación de aplicaciones de realidad aumentada, en las que se sobrepone imágenes virtuales al mundo real. Para ello, utiliza las capacidades de seguimiento de vídeo, con el fin de calcular, en tiempo real, la posición de la cámara y la orientación relativa a la posición de los marcadores físicos. Una vez que la posición de la cámara real se sabe, la cámara virtual se puede colocar en el mismo punto y modelos 3d son sobrepuertos exactamente sobre el marcador real.

Así ARToolKit resuelve dos de los principales problemas en la realidad aumentada, el seguimiento de punto de vista y la interacción objeto virtual. Fue desarrollado originalmente por Hirokazu Kato en 1999 y fue publicado por el HIT Lab de la Universidad de Washington [44].

Variantes

- **ATOMIC Authoring Tool** - es un software Multi-plataforma para la creación de aplicaciones de realidad aumentada, el cual es un Front end para la biblioteca ARToolkit. Fue Desarrollado para no-programadores, y permite crear rápidamente, pequeñas y sencillas aplicaciones de Realidad Aumentada. Está licenciado bajo la Licencia GNU GPL [44].
- **ATOMIC Web Authoring Tool** es un proyecto hijo de ATOMIC Authoring Tool que permite la creación de aplicaciones de realidad aumentada para exportarlas a cualquier sitio web. Es un Front end para la biblioteca Flartoolkit. Está licenciado bajo la Licencia GNU

GPL [44].

- **OSGART** integra el tracking de ARToolKit con las poderosas bibliotecas de gráficos avanzados OpenSceneGraph, por lo que osgART es una solución ideal para el desarrollo rápido de aplicaciones ricas en funcionalidades de realidad aumentada. Las avanzadas técnicas de visualización son compatibles, incluyendo los efectos de representación foto-realistas, como sombras, mapas de entorno y los reflejos, la oclusión, la refracción, así como representaciones no realistas como sombreado de dibujos animados [45].
- **ARTag** - alternativa a ARToolKit que usa un procesamiento de imagen y de símbolos digitales, más complejo, para mayor fiabilidad y resistencia a la luz. Licenciado solo para fines no-comerciales [44].
- **ARToolKitPro** - es la versión profesional de ARToolKit, la librería de reconocimiento óptico de patrones para realidad aumentada más utilizada en el mundo. ARToolKit Pro proporciona un desempeño confiable, de baja demanda, adaptable, escalable y de seguimiento para aplicaciones de realidad aumentada y realidad virtual. ARToolKit Pro es una librería para visión computarizada que se ejecuta en Windows, Mac OS X, Linux y sistemas operativos Irix. Cuenta con una interfaz simple basada en C que permite a los desarrolladores comenzar a construir sus propias aplicaciones en minutos. Tiene una serie de características avanzadas no disponibles en la versión GPL de ARToolKit [45].
- **Studierstube Tracker** - Sucesor de ARToolKitPlus, muchas características, pero ya no es libre [44].
- **FLARToolKit** - FlarToolkit es la versión de ARToolKit para Actionscript Flash (v3) que se pueden utilizar para desarrollar

rápidamente experiencias de realidad aumentada basadas en la Web. Es la librería de RA más utilizada basado en Flash, con el apoyo de una gran comunidad de desarrolladores y muchos sitios web con aplicaciones de ejemplo. FlarToolkit reconoce un marcador visual de una imagen de entrada y luego calcula la orientación de la cámara y la posición en el mundo 3D y capas de gráficos virtuales en la imagen de vídeo en directo. FLAToolKit cuenta con el apoyo de todos los principales motores gráficos 3D flash (Papervision3D, Away3D, arena, Alternativa3D) [45].

- **NyARToolkit** - Al igual que ARToolKit, NyARToolKit proporciona un seguimiento del marcador de AR base. Sin embargo, el software ha sido optimizado para una fácil portabilidad entre diferentes lenguajes de programación. Si desea desarrollar una aplicación AR que puede ejecutarse en diferentes plataformas y sistemas operativos entonces NyARToolKit es una opción ideal [45].
- **ARDesktop** - biblioteca de clase de ARToolKit que crea interfaces de escritorio con controles y widgets [44].
- **AndAR** - es una biblioteca de software de Java que permite realidad aumentada en la plataforma Android [45].
- **ARToolKit Plus** - ofrece la potencia y la simplicidad de ARToolKit Professional para plataforma iOS de Apple, faculta a los desarrolladores hacer aplicaciones revolucionarias de realidad aumentada para móviles. Muchas aplicaciones de estas ya están disponibles en la App Store. Incluye el código fuente completo de aplicaciones para el iPhone compilables, con clases que pueden ser reutilizadas en las aplicaciones propias de los desarrolladores, lo que permite un despliegue rápido de un amplio espectro de aplicaciones de realidad aumentada incluyendo demostraciones de

productos, visualizaciones y juegos [45].

3.2.1 Características

Lenguajes de programación

JAVA

A continuación se presentan las siguientes características [46]:

- Debería usar la metodología de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

C

A continuación se presentan las siguientes características [47]:

- Posee un núcleo del lenguaje simple.
- Programación estructurada.
- Lenguaje muy flexible.

- Utiliza lenguaje de preprocesado²⁷.
- Acceso a memoria de bajo nivel mediante el uso de punteros²⁸.
- Tipos de datos agregados (struct²⁹).

C++

A continuación se presentan las siguientes características [48]:

- Orientado a objetos.- Permite diseñar aplicaciones desde un punto de vista más cercano a la vida real. Además, permite la reutilización del código de una manera más lógica y productiva.
- Portabilidad.- El código fuente puede ser compilado en casi todo tipo de ordenadores y sistemas operativos sin hacer apenas cambios.
- Brevedad.- El código fuente es muy corto, porque en este lenguaje es preferible el uso de caracteres especiales que las "palabras clave".
- Programación modular.- Un cuerpo de aplicación en C++ puede estar hecho con varios ficheros de código fuente que son compilados por separado y después unidos.

²⁷Preprocesado.-Es una fase que comparten algunos lenguajes de programación, anterior a la fase de compilación.

²⁸Punteros.- Son herramientas utilizadas para señalar un punto o región.

²⁹Struct.- Es un tipo de valor utilizado para encapsular pequeños grupos de variables relacionadas.

- Velocidad.- El código resultante de una compilación en C++ es muy eficiente, gracias a su capacidad de actuar como lenguaje de alto y bajo nivel y a la reducida medida del lenguaje.
- Sobrecarga de operadores.- Los operadores intrínsecos³⁰ del lenguaje se les puede redefinir la semántica: se pueden escribir funciones que en vez de tener un nombre, se asocian a un operador, que debe tener por lo menos un parámetro de tipo clase.

Objetive C

- Está orientado a objetos pero es menos utilizado que el lenguaje C++, ya que se utiliza en Mac OS X.
- Está formulado de manera más simple y es mucho menos extenso que el lenguaje C++.
- Tiene herencia simple, es decir una clase puede tener un único parente.
- Se aplican los enlaces dinámicos donde las variables son identificadas por el runtime³¹ y sólo si existen punteros de objetos.
- Memoria dinámica.- Los objetos se crean siempre en memoria dinámica para darle menos peso y que se demore menos.

³⁰intrínsecos.-Características que define la naturaleza de una cosa.

³¹Runtime.- Es el tiempo que demora un sistema operativo en ejecutar el programa.

Lua

A continuación se presentan las siguientes características [25]:

- Compacto para usarse en diferentes plataformas.
- Las variables no tienen tipo, sólo los datos y pueden ser lógicos, enteros, números de coma flotante o cadenas.
- Estructuras de datos como vectores, conjuntos, tablas hash, listas y registros pueden ser representadas utilizando la única estructura de datos de Lua: la tabla.
- Ofrece soporte para funciones de orden superior, recolector de basura.
- Es posible utilizar Lua en programación orientada a objetos.

Herramientas de desarrollo

iOS SDK

- Consta de 4 herramientas principales:
 - Xcode: es la herramienta más utilizada para el desarrollo de software en Mac.
 - iPhone Simulator: simulador del iPhone para poder probar las aplicaciones realizadas.

- Instruments: es un conjunto de herramientas útiles para sacar el máximo partido a sus nuevas Apps³².
 - Interface Builder: para el diseño de las interfaces de usuario.
- Utiliza las mismas herramientas y APIs de Apple.
- El SDK se comercializa gratuitamente.
- Tiene una API que brinda una serie de librerías/frameworks para crear potentes y dinámicas aplicaciones para el iPhone.
- El SDK y el IDE Xcode son herramientas que se distribuyen conjuntamente.

Android SDK

A continuación se presentan las siguientes características [49]:

- Entorno de desarrollo completo incluyendo emulador, herramientas de depuración, análisis de memoria y rendimiento y plugin para el IDE Eclipse.
- Gráficos optimizados por una librería gráfica 2D propia; gráficos 3D basados en la especificación OpenGL ES 1.0.
- Framework de aplicación que habilita la reutilización y remplazo de componentes.

³²Apps.- App Store permite a los usuarios buscar y descargar aplicaciones informáticas, desarrolladas con el iPhone SDK y publicadas por Apple.

- Máquina virtual Dalvik para probar las aplicaciones.
- Base de datos SQLite.
- Soporte para aplicaciones telefónicas GSM o 3G.
- Compatibilidad para desarrollar aplicaciones que utilicen Bluetooth, GPS o Wi-Fi.
- Soporte nativo para YouTube, Google Mail.

Wikitudo SDK

- Multiplataforma.
- Su ambiente de desarrollo es Eclipse.
- Puede ser desarrollado, en lenguajes tales como: C, C++, Java.
- Su licencia es de código abierto y permite a cualquier persona tener acceso al programa.

QCAR SDK

- Multiplataforma.
- Su ambiente de desarrollo es Eclipse en Android y Xcode en iOS.
- Los desarrolladores tienen más opciones para el desarrollo de aplicaciones por sus APIs en C, C++, Lua, Java.

Unifeye Mobile SDK

A continuación se presentan las siguientes características [37]:

- Renderizado³³ de animaciones 3D en tiempo real basado en OpenGL ES³⁴ 1.X.
- Soporte para los desarrolladores mediante una API.
- Formatos 3D nativos: md2 y obj.
- Rastreo (tracking) basado en marcadores ID y plantillas.
- Soporte para el acceso a captura de vídeo.
- Multiplataforma.

Corona SDK

- Permite crear el código del programa en cualquier editor de textos y la ejecución se efectúa en el emulador de iPhone.
- Permite manipular la cámara, biblioteca fotográfica, soporta el acelerómetro, GPS y brújula (iPhone 3GS).
- Ofrece simplicidad en la integración de imágenes.
- Ofrece la depuración estándar.

³³OpenGL-ES.- Proceso de generar una imagen desde un modelo.

³⁴Renderizado.- Es una variante simplificada de la API gráfica OpenGL diseñada para dispositivos integrados tales como teléfonos móviles.

- Su tiempo de compilación es cero, el código desarrollado puede funcionar inmediatamente en el simulador.

Librerías de desarrollo

Look

- Posee herramientas para el dibujado de elementos tanto en 2D como 3D.
- Ofrece una gran variedad de herramientas para la definición de colores y texturas.
- Provee de funcionalidades geométricas las cuales facilitan el desarrollo de gráficos como son puntos, vectores, matrices, planos y rayos.

API CocoaTouch

- Posee una capa de abstracción del sistema operativo para iPhone.
- Esta agrupados en Foundation Framework³⁵ y UIKit Framework³⁶.
- Desarrollada en Objective-C, permitiendo así una mayor flexibilidad.

³⁵FoundationFramework.- Define las clases básicas, un API usado por cualquier tipo de programa Cocoa..

³⁶UIKit Framework.-Provee de todas las clases que una aplicación necesita para construir y gestionar su interfaz de usuario.

- Permite combinar el código con otros lenguajes, de preferencia C y C++, debido a que Objective-C esta basa en estos.

NyArtoolkit

A continuación se presentan las siguientes características [1]:

- Framework sencillo para el desarrollo de aplicaciones de RA.
- Soporte para librerías de vídeo y diversos formatos de imagen.
- Funciones de calibración de la cámara.
- Soporte con gran variedad de librerías gráficas.
- API modular y bien estructurada.
- Soporte para dispositivos Android

Artoolkit Plus

A continuación se presentan las siguientes características [50]:

- Multiplataforma.
- API basada en clase.
- Soporta la identificación de hasta 4096 marcadores.
- Posee un nuevo algoritmo de detección de pose.
- Soporte para formatos de píxel de cámaras de dispositivos móviles, como RGB565 y YU.

- Umbralización automática a partir del último patrón detectado.
- Tablas de búsqueda para funciones trigonométricas y corrección de distorsión.

3.2.2 Funcionamiento

Lenguajes de programación

JAVA

Empecemos describiendo el funcionamiento del lenguaje de programación Java, este lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores al momento de la programación, como la manipulación directa de punteros o memoria [46].

La arquitectura básica de un programa en java es la siguiente:

```
Impor java.lang.*;
Class HolaMundo{
    Public static void main (String [] args){
        System.out.println("Hola Mundo");
    }
}
```

Donde la sentencia import, indica que tenga en cuenta todas las clases de la librería java.lang, la clase java inicia con la palabra reservada class y después se indica el nombre como se puede apreciar todo el código de la clase o del método está encerrado entre llaves. En nuestro ejemplo podemos ver un método llamado main, que es el que primero se ejecutara cuando el programa inicie su ejecución, y dentro de este tenemos solo una sentencia System.out.println que recibe como

parámetro una cadena de texto y este imprimirá el mensaje hola mundo.

C

Para entender mejor el funcionamiento del lenguaje de programación C veamos la estructura básica de un programa en C

```
#include <stdio.h>
main( )
{
    printf("Hola amigos!\n");
}
```

Con el visualizamos el mensaje Hola amigos! en el terminal. En la primera línea indica que se tengan en cuenta las funciones y tipos definidos en la librería stdio (standardinput/output). Estas definiciones se encuentran en el fichero header stdio.h. Ahora, en la función main se incluye una única sentencia que llama a la función printf. Esta toma como argumento una cadena de caracteres, que se imprimen van encerradas entre dobles comillas " ". El símbolo \n indica un cambio de línea [51].

C++

Este lenguaje es un derivado de C tiene algunas ventajas sencillas pero estas lo hacen muy poderoso.

Declaración de clases

Para declarar una clase, todo lo que se necesita es escribir una definición de estructura y sustituir la palabra reservada struct por class. Por ejemplo, una clase “empleado” con campos como el

nombre, el departamento, la posición, el salario y una función que nos imprima la información de este quedaría así [52]:

```
class Alumno {  
    char* m_nombre;  
    char* m_curso;  
    char* m_paralelo;  
    long m_nota;  
    void Imprimir( Alumno infoAlumno);  
}
```

Objetive C

Empecemos por decir que Objective-C es un superconjunto del lenguaje de programación C, a continuación veamos un programa básico de Objective-C.

```
// First program example  
#import <Foundation/Foundation.h>  
int main (int argc, const char * argv[]) {  
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];  
    // insert code here...  
    NSLog(@"Hello, World!");  
    [pool drain];  
    return 0;  
}
```

Una de las particularidades que tiene la programación en Objective-C, es que es distintivo escribir en mayúsculas o minúsculas, es decir, que un comando no tiene el mismo funcionamiento dependiendo de cómo lo escribamos con mayúsculas o minúsculas (o incluso puede no funcionar al escribirlo de forma distinta) por lo que debemos respetar estas cuando escribamos un programa [53].

Lua

Los programas en Lua no son interpretados directamente, sino compilados a código bytecode, que es ejecutado en la máquina virtual de Lua. El proceso de compilación es normalmente transparente al usuario y se realiza en tiempo de ejecución, pero puede hacerse con anticipación para aumentar el rendimiento y reducir el uso de la memoria al prescindir del compilador [25].

A continuación se presenta un ejemplo:

```
#import <Cocoa/Cocoa.h>
#import "segmento.h"
@interface SVista : NSView {
    IBOutlet Segmento *segment;
    IBOutlet id muestraDistancia;
    NSPoint C;
    float escalaX;
    float escalaY;
    float desplazamientoX;
    float desplazamientoY;
}
- (NSPoint)escalar:(NSPoint)punto;
- (NSPoint)reducir:(NSPoint)punto;
@end
```

3.2.3 Compatibilidad

Lenguajes de programación

A continuación se muestra una tabla donde se indica si el lenguaje de programación en cuestión es o no compatible con los sistemas operativos para los cuales se desarrollará nuestro proyecto, cabe

aclarar, que se ha realizado una clasificación según nuestra experiencia y según si el lenguaje de programación es necesario, para la realización del proyecto en el sistema operativo indicado:

	Android	iOS
Java	SI	NO
C	SI	SI
C++	NO	SI
Objetive C	NO	SI
Lua	NO	NO

Tabla 3.1 Compatibilidad con los sistemas operativos seleccionados

Herramientas de desarrollo: SDK

En la tabla que se presenta a continuación, se hace una clasificación acerca de los SDK disponibles y si son o no compatibles para el sistema operativo correspondiente:

	Android	iOS
iPhone SDK	NO	SI
Android SDK	SI	NO
Wikitude SDK	SI	SI
QCAR SDK	SI	SI
Unifeye Mobile SDK	SI	SI
Corona SDK	SI	SI

Tabla 3.2 Compatibilidad con los sistemas operativos seleccionados

Librerías de desarrollo:

Existen varias librerías de desarrollo entre las principales podemos destacar las que se muestran en la tabla que se presenta a continuación y para que sistemas operativos son compatibles, esta clasificación se

realizó dependiendo de si la librería está disponible o no para alguna versión de los sistemas operativos:

	Android	iOS
Look	SI	NO
API CocoaTouch	NO	SI
ANDAR	SI	NO
NyArtoolkit	SI	SI
ArtoolkitPlus	NO	SI

Tabla 3.3 Compatibilidad con los sistemas operativos seleccionados

3.3 Análisis y comparación: Ventajas y desventajas

A continuación se compila una tabla comparativa en donde se aprecia un análisis de las ventajas y desventajas que posee cada lenguaje de programación estudiado.

	Ventajas	Desventajas
Java	<ul style="list-style-type: none"> • Multiplataforma. • Es libre. • Orientado a objetos • Permite integrar código nativo. • Muy Robusto. • Manejo de excepciones. • Portable. 	<ul style="list-style-type: none"> • Usa máquina virtual. • Sintaxis compleja. • Algunas herramientas tienen un costo adicional.
C	<ul style="list-style-type: none"> • Multiplataforma. • Mayor velocidad de ejecución. • Estructurado. • Numerosas bibliotecas. • Portable, • Permite desarrollar sistemas operativos y compiladores. 	<ul style="list-style-type: none"> • Poco robusto. • Muy complejo para principiante. • Mantenimiento más difícil y costoso. • No hay control automático. • No dispone de seguridad.

C++	<ul style="list-style-type: none"> • Orientado a objetos. • Muy potente. • Permite desarrollar sistemas operativos y mucho más. • Puede compilar y ejecutar código de C. • Muchos códigos fuentes y tutoriales. 	<ul style="list-style-type: none"> • Uso de DLL's. • No es recomendable para desarrollo de páginas Web. • Algunos elementos tienen costo. • No es multiplataforma.
Objetive C	<ul style="list-style-type: none"> • Multiplataforma. • Orientada a objetos. • Más rápido y fácil de aprender. • Lenguaje dinámico. • No haya ambigüedad. • Permite manejo de excepciones. 	<ul style="list-style-type: none"> • No hay recolector de basura. • No existe control de errores inmediato. • Tarda mucho tiempo en llamadas a métodos virtuales.

Tabla 3.4 Ventajas y desventajas de los lenguajes de programación.

3.4 Selección del lenguaje de programación para realidad aumentada

Luego de hacer el análisis comparativo de todos los leguajes de programación investigados anteriormente, que nos permitirán implementar la realidad aumentada en los teléfonos móviles se ha llegado a la conclusión de utilizar los lenguajes de programación Java y Objetive C, tanto para la aplicación en Android como para la aplicación en iOS.

3.5 Introducción al lenguaje de programación seleccionado

En este punto se dará a conocer porque se decidió seleccionar Java y Objetive C como lenguajes de programación.

3.5.1 Funciones básicas

JAVA

Java es un lenguaje muy potente y conocido por todos, una de sus principales funciones es que nos permite trabajar con librerías de realidad aumentada tanto para PC como para dispositivos móviles por lo cual hemos decidido utilizar dicho lenguaje. Otra razón de utilizar este lenguaje es porque la aplicación será desarrollada en el SDK Android, este kit de desarrollo permite implementar aplicaciones para el sistema operativo Android, utilizando Java.

Objective-C

Objective C es un lenguaje sencillo y muy fácil de aprender, es utilizado principalmente para desarrollar aplicaciones para Mac. Una de sus principales funciones es que permite implementar realidad aumentada en teléfonos móviles haciendo uso de la herramienta de desarrollo XCode propiamente de Mac por ende se decidió optar por este lenguaje de programación para desarrollar la aplicación para iOS [54].

3.5.2 Manejo de conexiones con base de datos

Existen muchos mecanismos para almacenar datos en nuestras aplicaciones, como por ejemplo los ficheros de preferencias compartidas o las bases de datos SQLite. Estos mecanismos son perfectos para almacenar datos estructurados, pero en ocasiones nos seguirá siendo útil poder disponer también de otros ficheros auxiliares de datos, probablemente con otro tipo de contenidos y formatos. La forma que manejaremos para almacenar datos será incluirlos como

recurso en la propia aplicación. Aunque este método es útil en muchos casos, sólo debemos utilizarlo cuando no necesitemos realizar modificaciones sobre los ficheros, ya que tendremos limitado el acceso a sólo lectura. Para incluir un fichero como recurso de la aplicación debemos colocar la carpeta con todos los ficheros a utilizar en nuestro proyecto, esta carpeta la podemos crear con el nombre que se desee [55].

Para nuestro caso las carpetas que almacenarán los objetos a visualizar tendrán el nombre de “raw” para Android y “3DModels” para iOS.

3.5.3 Manejo de animaciones

En computación, una animación 3d hace referencia a un tipo de animación que simula las tres dimensiones. Se trata de la descripción de los objetos de un modelo 3d a lo largo del tiempo. Para que exista animación, esa descripción debe variar en algo con respecto al tiempo: movimiento de objetos y cámaras, cambio de luces y formas, etc. Puede tratarse de una animación que se renderiza en tiempo real cuando se está ejecutando, o una animación que utiliza tres dimensiones pero ya ha sido renderizada previamente, por lo tanto sólo se trata de un video. La principal diferencia entre ambas radica en el momento de renderizado de la animación, es decir, el proceso de convertir las fórmulas matemáticas en imágenes digitales [56].

Las animaciones son un campo extraordinario y estas fusionadas con la realidad aumentada dotan al usuario la posibilidad de interactuar de una forma más amplia que la convencional.

En la actualidad existen los archivos 3ds y obj que son incapaces de realizar una animación, con el paso del tiempo aparecen los archivos md2 que son capaces de realizar animaciones en 3d y eso nos facilita todo el trabajo.

CAPITULO 4

ANALISIS, DISEÑO DE MARCAS Y MODELOS DE INFORMACION

4 Capítulo IV ANALISIS, DISEÑO DE MARCAS Y MODELOS DE INFORMACION

4.1 Análisis de los sitios turísticos de la ciudad de Cuenca

Cuenca es la capital de la provincia del Azuay y se caracteriza por ser uno de los lugares más visitados y atractivos de nuestro país. Actualmente es la tercera ciudad en importancia del Ecuador, tiene un promedio de medio millón de habitantes y está ubicada al sur del país.

Cuenca es una ciudad con historia viva, es decir coexiste un pasado, presente, y futuro que residen en sus habitantes y se expresan en su estructura física.

El visitante se siente cautivado por sus iglesias, museos y espacios que guardan historia, arte, cultura y tradición; todas estas conjugándose para revivir culturas lejanas.

Por su belleza y todo su valor histórico y cultural fue nombrada Patrimonio Cultural de la Humanidad el 1 de diciembre de 1999.

De esta gran extensión de historia haremos énfasis en algunos sitios y personajes, que son de relevancia en la ciudad. Asimismo daremos a conocer una pequeña reseña histórica, dependiendo de su ubicación y significado en la ciudad de Cuenca.

Iglesia de Santo Domingo



Figura 4.1 Imagen de la iglesia de Santo Domingo [57].

El Cabildo de la ciudad de Cuenca en noviembre de 1557, contribuyó para la construcción del convento e iglesia del “Señor Santo Domingo” [58]. Su construcción demoró 20 años, considerada como la segunda iglesia de la ciudad y su protectora es la imagen de la Virgen del Rosario (Patrona de la provincia desde 1936). A cargo de este templo se encuentran los dominicanos, los cuales están en Cuenca desde hace cuatro siglos y medio; su estructura dominica es igual a la de todas las iglesias dominicas del mundo [59].

Otra característica interesante es que siempre encontramos la imagen de San Francisco por la amistad que tuvo con Santo Domingo.

Iglesia Carmen de la Asunción



Figura 4.2 Imagen de la Iglesia Carmen de la Asunción [60].

El Monasterio Carmen de la Asunción es una de las majestuosas estructuras religiosas de Cuenca. Sus edificaciones fueron construidas en 1680 y fundadas por las sobrinas de Santa Marianita de Jesús en 1862 [59].

Este monasterio se encuentra ubicado en el centro histórico de la ciudad junto a la Plaza de las Flores y está resaltado por una portada tallada en mármol, que constituye un magnífico ejemplo del arte barroco. Cuenta con una estructura de estilo renacentista, en la cual se puede observar interesantes esculturas de tipo barroco [59]. La iglesia fue construida en honor a la Virgen de la Asunción.

Durante los días de la Novena de la Virgen del Carmen, miles de fieles guiados por los Párrocos vienen en peregrinación desde las

distintas Parroquias a participar de las Celebraciones Eucarísticas a nuestra Iglesia del Carmen de la Asunción.

Iglesia de San Francisco



Figura 4.3 Imagen de la Iglesia de San Francisco [60]

Esta iglesia es una de las más importantes de la ciudad. Desde la misma fundación de Cuenca, en abril de 1557, Gil Ramírez Dávalos entregó una cuadra para el convento del “Señor San Francisco”, cuyo lugar perteneció a los franciscanos por tres siglos y posteriormente pasó a manos de la Curia [59].

Preserva dos joyas del tallado colonial, el retablo de altar mayor y el púlpito.

En su fachada constan dos lápidas con sendas efigies e inscripciones latinas, pero lo que más llama la atención es su torre principal, y su interior está dividido en tres naves en donde se admira dos obras de tallado colonial revestidas de pan de oro, además presenta importantes obras de arte [59].

El Retablo barroco del siglo XVII es el elemento de mayor valor de la iglesia, es decir una poesía mística hecha obra de arte.

Iglesia la Catedral Vieja



Figura 4.4 Imagen de la iglesia de la Catedral Vieja [60]

Conocida como iglesia del Sagrario, fue levantada en 1557. Por su conjunto de características de valor espiritual viene a constituir uno de los grandes símbolos religiosos que posee la ciudad [59].

Fue uno de los primeros templos que se edificaron en la ciudad con piedras labradas del Tomebamba.

Es el santuario más célebre de la ciudad ya que una de sus torres fue utilizada por la misión Geodésica Francesa para la medición del arco terrestre. En su torre frontal se encuentra colocado el primer reloj de la ciudad [59].

En la actualidad sus instalaciones son utilizadas como museo, pero la iglesia mantiene su estructura básica colonial a pesar de varias remodelaciones realizadas en los siglos XIX, y XX.

Monumento de Abdón Calderón



Figura 4.5 Imagen del monumento de Abdón Calderón [60]

Abdón Calderón Garaicoa nació el 30 de julio de 1804, y murió en Quito, después de haber participado heroicamente en la batalla de Pichincha, en los últimos días de Mayo de 1822 [61].

Su monumento se encuentra ubicado en el Parque Central del mismo nombre, inaugurado el 24 de mayo de 1931, su autor fue el artista europeo Carlos Mayer. Es una escultura de bronce de más de tres metros de altura, se encuentra rodeada de ocho pilastras de mármol en forma de bala y dos cabezas de león que emanen agua por la boca, todo esto forma en conjunto la admirable escultura del héroe niño [62].

Constituye una importante cédula de identidad de la ciudad, por su valentía a tan corta edad al formar parte de las tropas que nos dieron la libertad.

Monumento del Padre Carlos Crespi



Figura 4.6 Imagen del monumento del Padre Carlos Crespi [60]

Padre Carlos Crespi nació el 29 de Mayo de 1891, y murió víctima de una penosa y larga enfermedad, en Cuenca el 30 de abril de 1982.

Su monumento se encuentra localizado en el parque de María Auxiliadora inaugurada el 27 de enero de 1989 en el barrio de los salesianos, y el autor de esta figura fue Paul Palacios [62].

Fue el padre Crespi un hombre excepcional en todo el sentido de la palabra. Sus investigaciones lo llevaron a develar grandes secretos de la historia, su actividad pedagógica orientó y educó a varias

generaciones, y su ministerio pastoral redimió el alma de todos quienes acudieron a él en busca de ayuda espiritual [61].

Monumento de Simón Bolívar



Figura 4.7 Imagen del monumento de Simón Bolívar [60]

Simón Bolívar nació 24 de Julio de 1783, y murió en Santa Marta, Colombia, el 17 de Diciembre de 1830 [61].

El monumento está localizado en Avenida de las Américas, intersección con la calle Gran Colombia y Av. Ordóñez Lazo. Inaugurado el 9 de Noviembre de 1985, y el autor de la escultura fue Wolfgang Palacio.

Su vida la dedicó a la libertad del Nuevo Mundo, consagrándose a luchar por la Independencia de América y en ello no descansó hasta su muerte [62].

Para probar la aplicación se crearán 5 objetos diferentes que pertenecen a las de Iglesias de Santo Domingo y la Catedral Vieja, y a los monumentos de Abdón Calderón, Simón Bolívar y Padre Carlos Crespi.

4.2 Diseño y creación de las marcas de reconocimiento

Un marcador, marca o patrón es una imagen (impresa en una hoja) que el dispositivo móvil procesa, y de acuerdo a la programación especificada para esa imagen, le incorpora los objetos 3D.

Estos marcadores que serán utilizados para Realidad Aumentada están realizados en un archivo con extensión .pat o .patt que guarda la codificación de la imagen.

- **Diseño**

Partiremos el diseño de nuestros marcadores desde un marcador de inicio creado por la herramienta GIMP, el cual consiste en 3 cuadrados:

- El primer cuadrado blanco servirá como fondo para evitar que la cámara tenga problemas en el reconocimiento del marcador.
- El cuadrado interno negro representará el marco de reconocimiento.
- El cuadrado interno blanco servirá para agregar los diferentes detalles de reconocimiento.

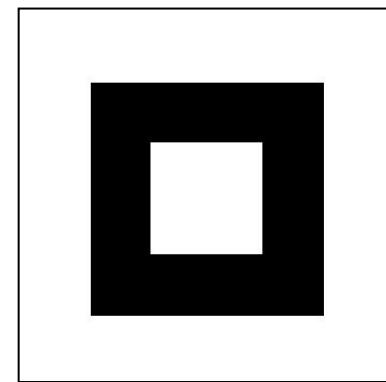


Figura 4.8 Imagen del marcador de inicio.

- **Creación**

Para generar el archivo .pat³⁷ del marcador que hemos diseñado debemos ir a: <http://flash.tarotaro.org/blog/2008/12/14/artoolkit->

³⁷.pat archivos con la codificación de la imagen en mapa de bits.

[marker-generator-online-released/](#) para abrir la aplicación ARToolKit Marker Generator Online con la cual convertiremos nuestra imagen a código.



Figura 4.9 Imagen de la aplicación ARToolKit Marker Generator

Una vez abierta la aplicación seguiremos los siguientes pasos:

- Seleccionar la opción "Permitir" en la siguiente ventana.

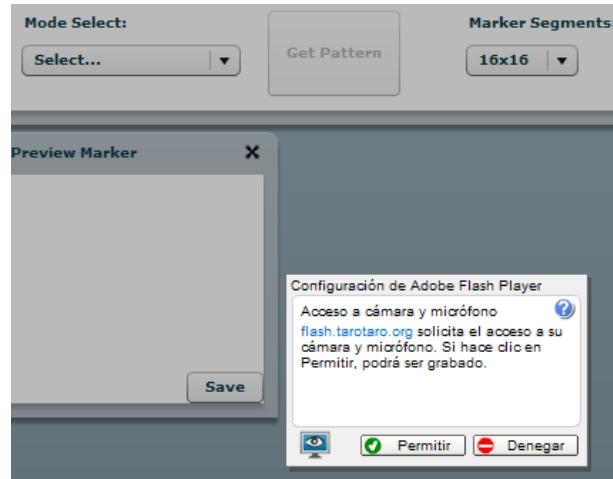


Figura 4.10 Imagen al iniciar ARToolKit Marker Generator

- Dentro de la opción "Mode Select" (modo de selección), seleccionamos "Load marker image" (cargar imagen como marcador) y buscamos la imagen .jpeg creada en GIMP del nuevo marcador:

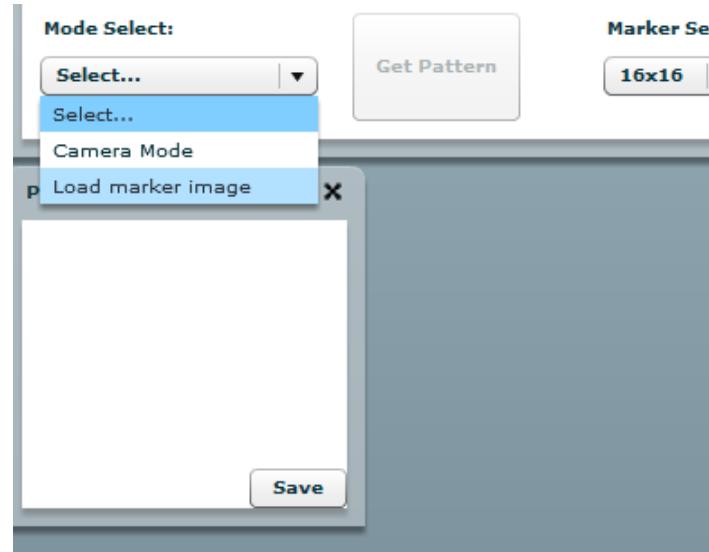


Figura 4.11 Imagen al seleccionar un nuevo marcador

- La opción "Marker Segments" (segmentos del marcador), nos da la posibilidad de escoger las posiciones en que nuestro marcador puede ser convertido, es decir, si seleccionamos 4x4 no es más que las 4 posiciones del marcador convertido a código, en nuestro caso será 16x16.

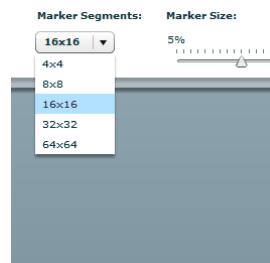


Figura 4.12 Imagen de la opción Marker Segments

- La opción "Marker Size" (Tamaño del marcador) nos da la posibilidad de escoger el tamaño del marcador el cual va desde el 5% al 100%, lo recomendable es dejarlo en el 50%, esto para que el programa pueda detectar al marcador sin inconvenientes.

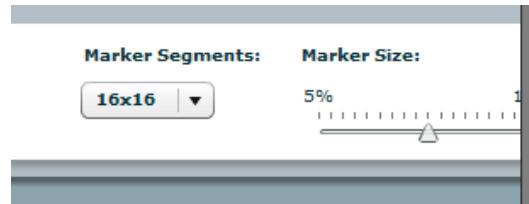


Figura 4.13 Imagen de la opción Marker Size

- Al momento que la imagen esté remarcada en color rojo seleccionamos la opción "Get Pattern" (obtener patrón). Y finalizamos, presionando "Save" (guardar), además asignamos un nombre al archivo para identificar nuestro nuevo marcador.

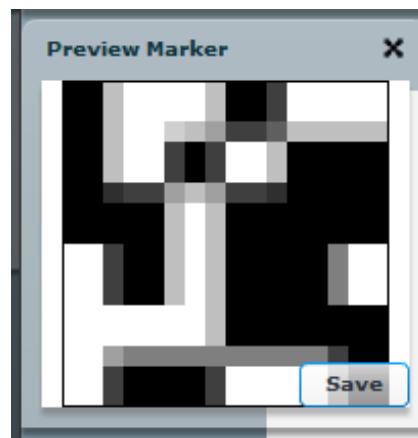


Figura 4.14 Imagen del marcador codificado

4.3 Diseño y creación de los modelos de información

Un modelo de información es la representación explícita de una categoría determinada, de tal forma que podamos trasmisir mediante el uso de entidades, atributos y relaciones el comportamiento de un sistema.

- **Diseño**

Partiremos el diseño de nuestro modelo de información desde el funcionamiento general de una aplicación de realidad aumentada que utiliza ARToolKit, el cual será el siguiente:

- Inicializamos la cámara.

- Se realiza la captura del marcador del mundo real mediante la cámara.
 - Se procede al reconocimiento del marcador capturado realizando el siguiente procedimiento:
 - La imagen se umbraliza de tal forma que los pixeles cuya intensidad superen el valor del umbral son transformados en pixeles de color negro. Los que no superan este umbral se transforman en pixeles blancos.
 - Con el procedimiento de la umbralización el cuadro blanco aparece en negro. Esto con la finalidad de buscar y encontrar todos los marcos negros como los de la plantilla.
 - Se realiza la comparación del marco para analizar la plantilla capturada con las plantillas almacenadas, de tal forma que si coinciden se pueda calcular la posición, orientación relativas, y guardar los valores en una matriz.
 - Con los valores de la matriz se transforma las coordenadas del objeto virtual a dibujar.
 - Se muestra el objeto virtual.
 - Se cierra la cámara.
- **Creación**

De acuerdo al proceso de diseño procedimos a establecer el siguiente modelo de información:

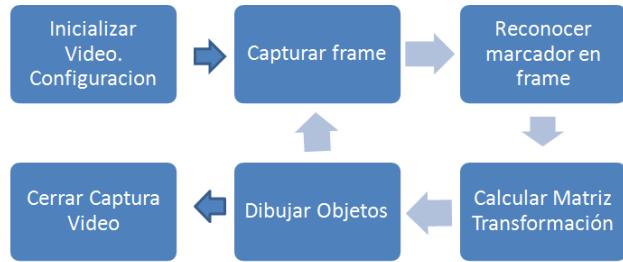


Figura 4.15 Modelo de información [2].

4.4 Pruebas de las marcas de reconocimiento creadas

Para poder realizar las diversas pruebas de las marcas creadas y poder comprobar la eficiente funcionalidad del sistema desarrollado, procedimos a la aplicación de estrategias que ayudarán a proporcionar resultados adecuados para su debida evaluación.

a) Estrategia

Se realizarán las pruebas en ambientes, con diferentes niveles de iluminación, distancia de la cámara del dispositivo con respecto a la marca y diferentes características de los marcadores.

Para esto se utilizaron 2 dispositivos móviles con las siguientes características:

- **Sony Ericsson XPERIA e iPhone 4G**

Los dos dispositivos cuentan con una cámara incorporada de 5 mega píxeles con una resolución de 3008 x 1960.

Los sistemas operativos con los cuentan los dispositivos son Android OS e iOS, respectivamente.

Además, se realizó la impresión de los marcadores con las siguientes características:

- **Marcadores impresos.**

Los marcadores han sido impresos en material de lona y hojas de papel, con un tamaño de 15cm x 15cm.

Los objetos 3D que serán visualizados presentan las siguientes características:

- **Objetos 3D**

Los objetos fueron creados con un tamaño de 720x576 pixeles.

Al realizar las pruebas respectivas, nos brindó los resultados que se muestra a continuación.

b) Resultado

En la tabla 4.1 se presentan los resultados que se obtuvieron al realizar las pruebas en un ambiente abierto:

	Resultado de visualización en		Observaciones
	Android	iOS	
Iluminación			
Alta (sol intenso)	INCORRECTO	INCORRECTO	El marcador posee un fondo blanco y este al contacto con una luz intensa tiende a rebotar
Media	CORRECTO	CORRECTO	La marca es detectada correctamente
Baja (oscuro)	INCORRECTO	INCORRECTO	No se puede visualizar con claridad el marcador
Distancia			
Entre 1 a 20cm	INCORRECTO	INCORRECTO	La marca no puede ser visualizada totalmente por la cámara
Entre 21 a 75cm	INCORRECTO	INCORRECTO	El objeto es demasiado grande para visualizarse correctamente
Entre 76 a 1.1m	CORRECTO	CORRECTO	El objeto se visualiza claramente
Entre 1.11m a 1.5m	INCORRECTO	INCORRECTO	El objeto no se visualiza correctamente
Mayor a 1.6m	INCORRECTO	INCORRECTO	La marca no puede detectarse
Material Marcador			
Lona (con luz intensa)	INCORRECTO	INCORRECTO	Causa mucho brillo
Papel (deteriorado)	INCORRECTO	INCORRECTO	La marca tiende a no ser detectada
Lona (con luz media)	CORRECTO	CORRECTO	La marca es detectada correctamente
Papel (no deteriorado)	CORRECTO	CORRECTO	La marca es detectada correctamente
Modelo Marcador			
	CORRECTO	CORRECTO	La marca es detectada correctamente
	INCORRECTO	INCORRECTO	Este modelo tiende a confundirse con los patrones que se encuentran en el entorno y con otras marcas similares

Tabla 4.1 Resultados obtenidos en un ambiente abierto

Las pruebas también se realizaron en un ambiente cerrado, obteniendo resultados muy similares, con la única diferencia en la iluminación. Como se muestran en la tabla 4.2:

	Resultado de visualización en		Observaciones
	Android	iOS	
Iluminación			
Alta (luz intensa)	INCORRECTO	INCORRECTO	El marcador posee un fondo blanco y este al contacto con una luz intensa tiende a rebotar
Media	CORRECTO	CORRECTO	La marca es detectada correctamente
Baja	CORRECTO	CORRECTO	La marca es detectada correctamente

Tabla 4.2 Resultados obtenidos en un ambiente cerrado

Los diversos marcadores que fueron creados para la aplicación de realidad aumentada funcionan bien en los dispositivos de prueba.

A continuación exponemos las conclusiones obtenidas de las pruebas realizadas:

- La marca debe enfocarse completamente en la cámara del dispositivo.
- Se debe usar un material que no genere demasiado brillo.
- La distancia de la cámara a la marca, debe estar entre 75cm y 1.1m, teniendo en cuenta que el tamaño del marcador es directamente proporcional a la distancia entre la cámara y la marca.
- El modelo del marcador debe ser complejo, para no confundir al sistema con los patrones del ambiente.
- Las marcas funcionan correctamente en las dos plataformas seleccionadas.
- Las marcas deben estar sujetas en un material firme para evitar problemas de reconocimiento.

CAPITULO 5

ANALISIS, DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN EN REALIDAD AUMENTADA

5 Capítulo V ANALISIS, DISEÑO E IMPLEMENTACION DE LA APLICACIÓN EN REALIDAD AUMENTADA

5.1 Análisis y diseño de la aplicación en realidad aumentada

En la ingeniería de software el análisis y diseño de aplicaciones son de gran utilidad para el desarrollo eficiente de sistemas, y que mediante una adecuada ingeniería de requisitos se podrán comprender las necesidades y condiciones a satisfacer por el software a desarrollar.

5.1.1 Análisis de los requerimientos de la aplicación

Especificación de requerimientos funcionales y no funcionales

➤ Requerimientos funcionales

A continuación se detallarán los requerimientos para la aplicación en realidad aumentada con los cuales se obtendrá una mayor comprensión y claridad en cuanto a su funcionamiento.

✓ Reconocimiento de marcas

Para este requerimiento lo que se desea es que la aplicación reconozca automáticamente marcas por medio de la utilización de la cámara del teléfono móvil y capture el entorno real en el que estén colados los mencionados marcadores para su respectivo procesamiento (ver tabla 5.1).

✓ Visualización de la información

Con este requerimiento lo que se desea es que la aplicación muestre en pantalla el objeto guardado y

asignado a una marca respectiva, una vez reconocida y procesada dicha marca (ver tabla 5.2).

✓ **Manual de usuario para la utilización de la aplicación**

Se deberán realizar las guías para el usuario como ayuda para que pueda manipular la aplicación de manera correcta. y no Estas guías serán realizadas para usuarios con conocimiento mínimo en el manejo de programas de este tipo (ver tabla 5.3).

➤ **Descripción a detalle sobre los requerimientos funcionales**

Reconocimiento de marcas	
Descripción:	A través del uso de la cámara del teléfono móvil, se realizará la captura del entorno para el reconocimiento de las marcas.
Entradas:	Escena capturada por la cámara.
Fuente:	Los diferentes archivos .pat que se encontrarán almacenados dentro de un directorio específico de la aplicación.
Salidas:	Marca identificada.
Requerimientos:	Se debe tener guardada la marca dentro de la aplicación.
Precondición:	La aplicación debe ejecutar el modo video.
Postcondición:	Enfocar la marca a reconocer.

Tabla 5.1 Requerimiento para el reconocimiento de marcas.

Visualización de la información	
Descripción:	Visualizar objetos asignados a la marca en tiempo real.

Entradas:	La marca reconocida.
Fuente:	Por medio de los diferentes archivos .pat que se encontrarán almacenados dentro de la aplicación, mostrar el objeto asignado a esa marca en pantalla.
Salidas:	Visualizar en pantalla el objeto 3D asignado a la marca en tiempo real.
Requerimientos:	Se debe tener el objeto almacenado en un directorio dentro de la aplicación.
Precondición:	La aplicación debe procesar y reconocer la marca a través de la cámara de video en tiempo real.
Postcondición:	Mantener enfocada la marca mientras se desee visualizar el objeto.

Tabla 5.2 Visualización de la información.

Manual de usuario para la utilización de la aplicación	
Descripción:	Se creará un manual para el usuario, con la finalidad de que este posea una ayuda y guía en cuanto al manejo de la aplicación.
Entradas:	Se establecerán las capturas de pantallas de la interfaz gráfica de la aplicación.
Fuente:	La información para el manual de usuario se basará en el funcionamiento de la aplicación realizada.
Salidas:	Se tendrá una guía para el usuario con las diferentes capturas explicando a detalle cómo se debe utilizar la aplicación desde su descarga, instalación y funcionamiento.
Destino:	Este manual estará aplicado para

	personas con poca experiencia en el manejo de aplicaciones en teléfonos móviles.
Requerimientos:	Se requiere que el manual sea entendible y detallado, debe incluir todos los pasos para el manejo total de la aplicación.
Precondición:	Tener la aplicación funcionando correctamente.
Postcondición:	El usuario se encuentra capacitado en el uso de la aplicación.

Tabla 5.3 Requerimiento para el manual de usuario

➤ Requerimientos no funcionales

✓ Fiabilidad

El sistema deberá ofrecer un óptimo rendimiento al momento que el usuario final requiera hacer uso del mismo.

Los recursos de hardware y software que el sistema desarrollado ocupe en el teléfono móvil no deberán generar conflictos con otras aplicaciones instaladas en el mismo.

El sistema en ejecución no debe colapsar al momento de su uso.

✓ Facilidad de uso

La aplicación a realizar será de fácil uso, con lo cual el usuario final podrá manejar el sistema de manera correcta.

Además de esto, la aplicación contará con un manual de usuario en donde estará detallado el funcionamiento del sistema.

5.1.2 Diseño del modelado de datos para la aplicación

Nuestra aplicación utilizará como ya se explicó en capítulos anteriores, librerías que nos permiten trabajar con realidad aumentada y objetos 3D³⁸, de manera fácil.

A continuación se presenta el diagrama de clases de la aplicación en desarrollo.

Diagrama de Clases

Para nuestro proyecto utilizaremos las librerías NYARToolkit y ARToolkitPlus, para Android e iOS, respectivamente, las cuales contienen las clases que nos ayudarán a trabajar con realidad aumentada (ver figura 5.1).

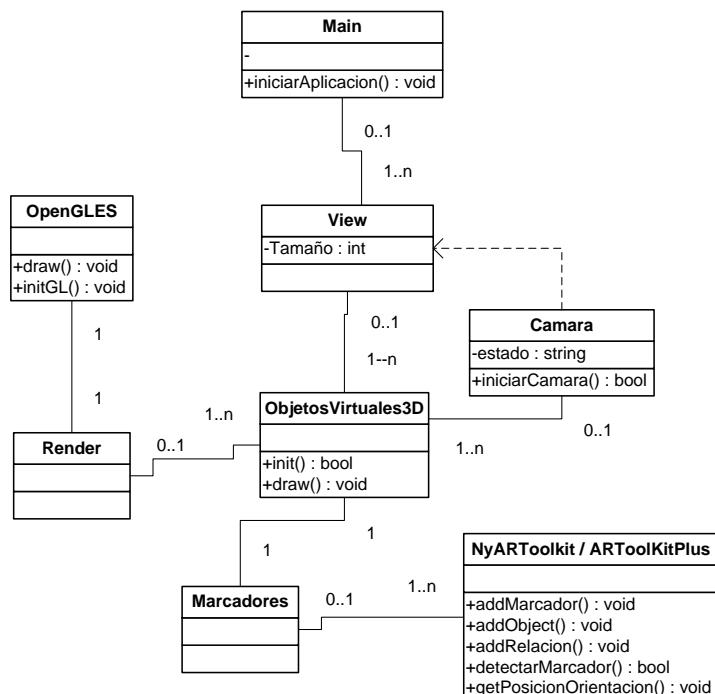


Figura 5.1 Modelo de clases de la aplicación de realidad aumentada

³⁸ Objetos 3D.- Imágenes en tres dimensiones

Diagrama de Contexto

El siguiente diagrama está enfocado a reflejar el flujo del funcionamiento software, elementos de control (entidades externas como los usuarios), que interactúan con el sistema (ver Figura 5.2).

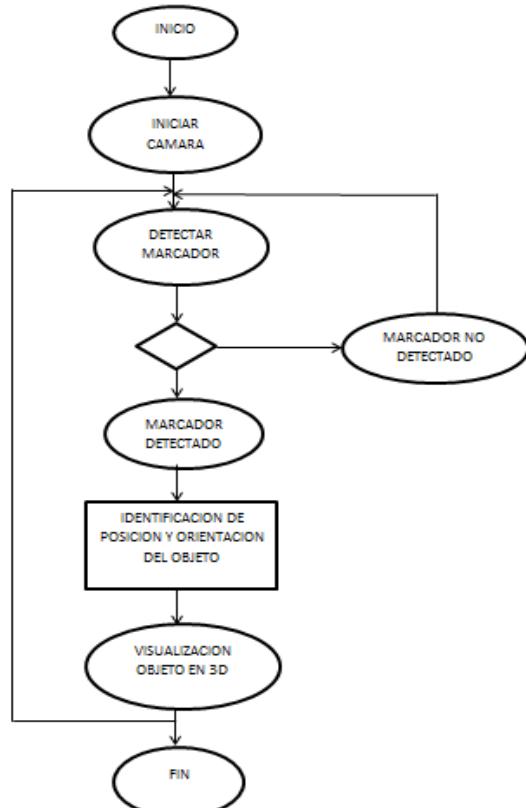


Figura 5.2 Diagrama de contexto

Según el análisis realizado en capítulos anteriores podemos determinar:

- La aplicación se la realizará para los sistemas operativos Android y iOS en el lenguaje de programación Java y Objective C respectivamente, apoyados en sus SDK (kit de desarrollo de software).
- Los entornos de desarrollo para la implementación de la aplicación que prestan todas las facilidades y la compatibilidad necesaria para la utilización de las librerías mencionadas con anterioridad serán:

- Para el caso de Android utilizaremos Eclipse.
- Para el caso de iOS utilizaremos Xcode.
- Los objetos 3D que serán incorporados, se los creará con la aplicación Blender que es el software que más se adapta a las necesidades del proyecto.

5.2 Instalación y configuración de la base de datos

5.2.1 Instalación de la base de datos

Haciendo uso del software de libre distribución, y referencia al tipo de información que necesitamos almacenar hemos decidido utilizar la base de datos MySQL versión 5.0 la cual cuenta con las siguientes características:

- Velocidad: posee un nivel alto de rendimiento al momento de resolver consultas, ya que está orientada a aplicaciones donde la actualización e integridad es más importante.
- Herramientas gráficas, documentación: proporciona una amplia documentación y multitud de herramientas gráficas de gestión y de conectividad.

Instalación

Para realizar esta tarea se necesita disponer del programa de instalación el cual se puede descargar gratuitamente de <http://dev.mysql.com/downloads>.

Una vez descargado el programa procedemos a seguir las instrucciones que nos muestra el asistente de instalación (ver figura 5.3).



Figura 5.3 Asistente de instalación de MySQL

En la ventana de tipo de instalación marcamos "Custom" (ver figura 5.4).

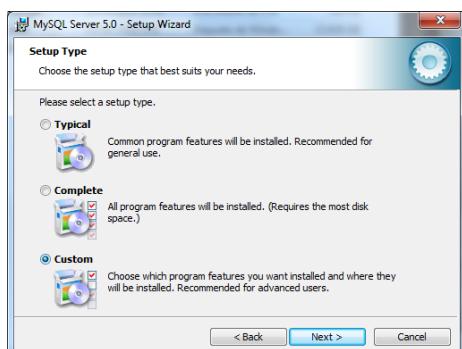


Figura 5.4 Tipo de instalación

A continuación seleccionamos las utilidades a instalar, por defecto se instalará todo salvo las herramientas para desarrolladores (ver figura 5.5).

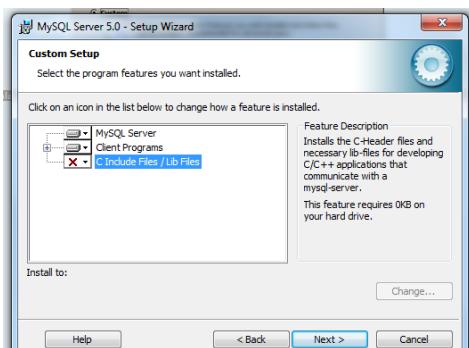


Figura 5.5 Selección de utilidades

A continuación pulsamos en "Install" (ver figura 5.6).

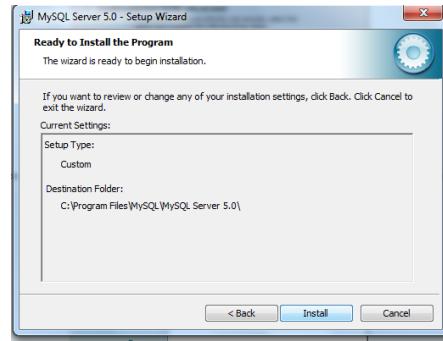


Figura 5.6 Instalando MySQL

5.2.2 Configuración y pruebas de la base de datos

Configuración

En el asistente para la configuración "MySQL Server Instance Configuration Wizard" y pulsaremos en "Next" (ver figura 5.7).



Figura 5.7 Asistente de configuración de MySQL

Marcamos la opción "Standard Configuration" (ver figura 5.8).

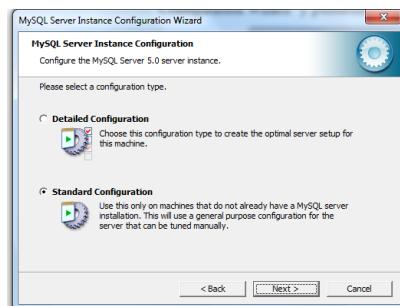


Figura 5.8 Selección de la configuración estándar

Ahora especificamos el tipo de arranque de MySQL Server, en

este paso la opción recomendada es "Install As Windows Service" (ver figura 5.9).



Figura 5.9 Selección del tipo de arranque

Introduciremos la contraseña para el usuario administrador (root) y marcaremos la opción "Enable root access from remote machines" si queremos que se pueda acceder como administrador desde otros equipos (ver figura 5.10).



Figura 5.10 Ingreso de contraseña del administrador root

Por último pulsaremos en "Execute" para finalizar la configuración de MySQL (ver figura 5.11).

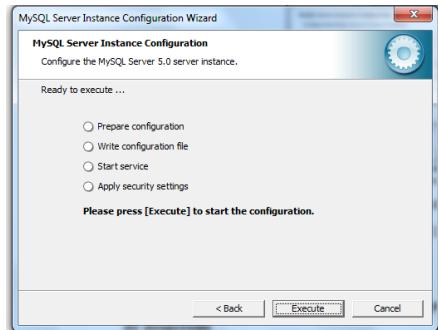


Figura 5.11 Finalización de la configuración

Pruebas

Para comprobar que el servicio de MySQL se está ejecutando podemos usar el administrador de tareas (ver figura 5.12).

Aparecerá un servicio con el nombre "mysqld-nt.exe" que, como se puede observar, usa unos 12 MB de memoria RAM (sin conexiones de clientes).



Figura 5.12 Servicio de MySQL.

Para comprobar el funcionamiento correcto de MySQL iremos a "Inicio" - "Programas" - "MySQL" - "MySQL Server 5.0" - "MySQL Command Line Client". Nos pedirá una contraseña (la que hayamos introducido en la instalación).

Como ejemplo para comprobar que la instalación ha sido correcta nos hemos conectado a la base de datos que MySQL crea automáticamente llamada "mysql", y a su vez hemos ejecutado una consulta sobre la tabla "user" para mostrar el nombre de los usuarios de la BD (ver figura 5.13).

A captura de tela da interface de linha de comando MySQL. O prompt é 'DEBATO_DICE-COMPARTIR_III>'. A saída mostra: 'C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe', 'Your MySQL connection id is 2', 'Server version: 5.0.67-community-nt MySQL Community Edition <GPL>', 'Type 'help'; or 'h' for help. Type 'x' to clear the buffer.', 'mysql> connect mysql', 'Connection id: 3', 'Current database: mysql', 'mysql> select user from user;', ' +-----+ | user | +-----+ | root | | root | +-----+', '2 rows in set (0.00 sec)', 'mysql>'.

Figura 5.13 Comprobación del funcionamiento correcto de MySQL

5.3 Instalación y configuración del lenguaje de programación y entorno de desarrollo

5.3.1 Instalación del lenguaje de programación y el entorno de desarrollo

Para la instalación del lenguaje de programación y el entorno de desarrollo según el sistema operativo en el que se desea desarrollar la aplicación se realizará lo siguiente:

Android

Instalación del SDK

Para su instalación debemos realizar su descarga del siguiente enlace: <http://developer.android.com/sdk/index.html>.



Figura 5.14 Descarga del Android SDK

Descomprimimos el SDK en una carpeta, como sugerencia la dirección de esta debe ser fácil de recordar.

Para el desarrollo de la aplicación en el sistema operativo Android, además de la instalación del SDK es necesario instalar Eclipse como ambiente de desarrollo y el plugin ADT para su interacción.

En nuestro caso la versión utilizada de Eclipse fue Galileo en cual se recomienda utilizar la distribución clásica o la versión

para desarrolladores Java, que corresponda con la arquitectura donde se desee instalar.

Su descarga la podemos realizar desde la siguiente ubicación:

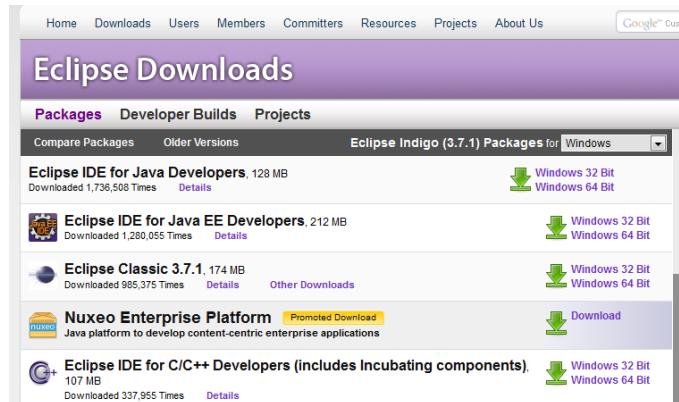


Figura 5.15 Eclipse downloads

Instalación de Eclipse

Su instalación consiste únicamente en descomprimir el archivo y ubicarlo en la ruta donde se desea que este resida.

Instalación del plugin ADT (Android Development Tool)

Iniciamos Eclipse y dentro del menú Help seleccionamos la opción "Install new software".

En la parte superior derecha presionamos en el botón Add y especificamos los siguientes valores (ver figura 5.16).

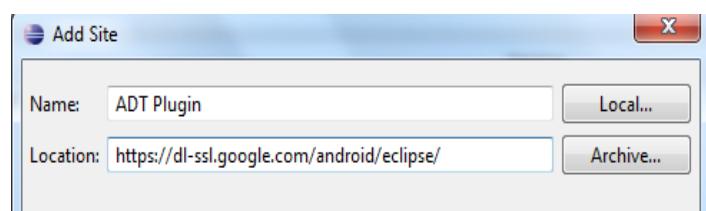


Figura 5.16 Instalación del plugin ADT

Seleccionamos la casilla de verificación de la rama Developer Tools (ver figura 5.17).

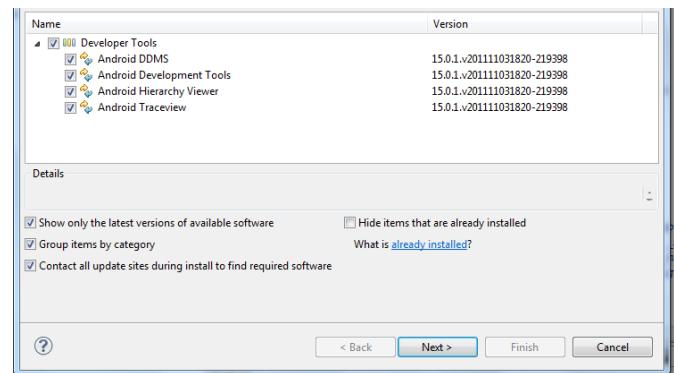


Figura 5.17 Verificación de la rama Developer Tools

Presionamos el botón "Next" en los dos diálogos siguientes. Seleccionamos la opción "I accept the terms of the license agreements" y presionamos el botón **Finish** para terminar el procedimiento.

Reiniciamos Eclipse cuando este lo solicite.

iOS

Para el desarrollo de la aplicación en el sistema operativo iOS es necesaria la descarga del IDE de desarrollo que proporciona Apple llamada Xcode. Este por lo general ya incluye el iOS SDK y se descarga directamente de la página Apple previo registro como se indicó en el capítulo 3.



Figura 5.18 Descarga del Xcode

La versión utilizada para nuestro proyecto es el Xcode 3.2.6 con iOS SDK 4.3, el cual se instala bajo la versión de Snow Leopard.

Con el instalador del Xcode descargado procedemos a dar doble clic sobre el mismo y seguimos el asistente de instalación.

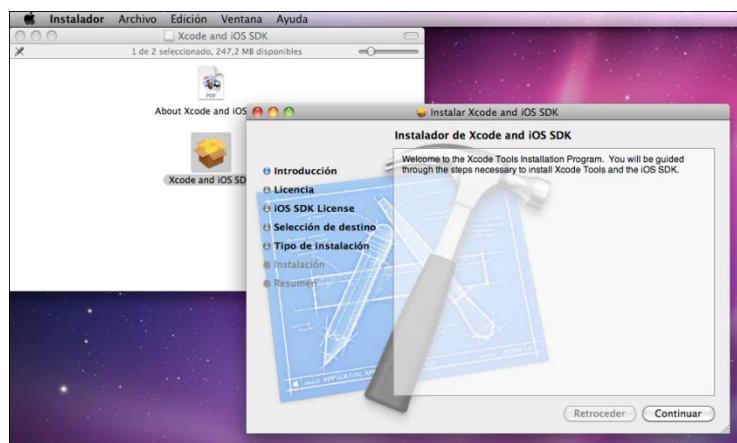


Figura 5.19 Instalación del Xcode

5.3.2 Configuración del lenguaje de programación y el entorno de desarrollo

Android

Configuración de las variables de entorno para el lenguaje de programación

Para que Java funcione correctamente con el SDK de Android hay que configurar las variables de entorno.

Para configurar las variables de entorno tenemos que dar clic derecho en Mi PC, o en Equipo si estamos usando alguna de las versiones de Windows.

Damos clic en "Propiedades", seleccionamos en "Configuración Avanzada del Sistema" y luego clic en el botón de "variables de entorno", en el cuadro de variables del sistema buscamos la variable "Path", damos clic en el botón "editar".

En la ventana de configuración buscamos el campo "valor de la variable", nos desplazamos al final de todas las sentencias y agregamos un punto y coma (;) para separar las sentencias.

Agregamos las rutas del SDK y Tools de Android de la siguiente manera: C:\android-sdk-windows; C:\android-sdk-windows\tools. Finalmente damos clic en aceptar y reiniciamos el equipo.

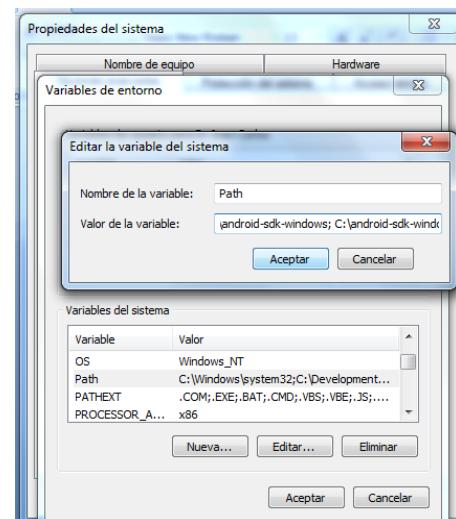


Figura 5.20 Configuración de las variables de entorno

Configuración del plugin ADT

En Eclipse accedemos a preferencias a través del menú "Window- Preferences".

A continuación seleccionamos la opción "Android" en el árbol del lado izquierdo.

Presionamos el botón "Browse..." y ubicamos el directorio de instalación del Android SDK.

Presionamos el botón "Apply" y finalmente "Ok".

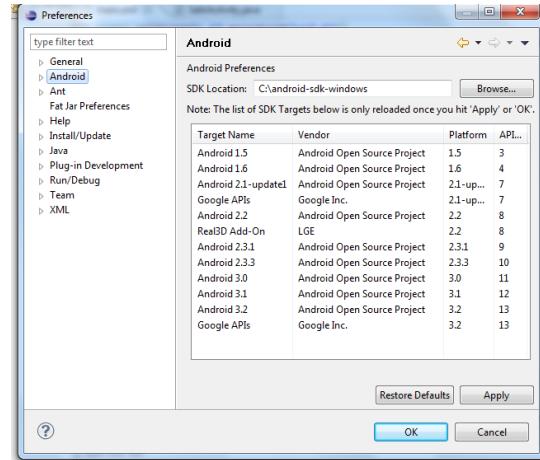


Figura 5.21 Configuración del ADT

iOS

Para iniciar el desarrollo de nuestra aplicación fue solamente necesario instalar el IDE de desarrollo de Apple Inc.

Xcode es una herramienta que no necesita mayores configuraciones, ya que por defecto se instala lo necesario para comenzar el desarrollo de una aplicación, como lo es el iOS SDK, y el Interface Building que nos ayudará en la implementación de la interfaz gráfica.

5.4 Implementación de los módulos de la aplicación

5.4.1 Programación del módulo de descarga de actualización y nuevas marcas

La descarga de actualizaciones del sistema de realidad aumentada tanto para el sistema operativo Android como para el iOS, se podrá realizar mediante la utilización del portal web destinado para este fin.

El portal web dispondrá de un módulo que permitirá la descarga de ficheros .apk (Android Package, paquete Android), e .ipa (Iphone Package, paquete Iphone).

Estos ficheros podrán ser descargados directamente en el celular, como también a través de una PC.

5.4.2 Programación del módulo principal

5.4.2.1 Programación de la interfaz entre la base de datos y la aplicación

La aplicación de realidad aumentada, necesita de recursos que deben estar almacenados dentro de los directorios creados por la misma aplicación, por esta razón se ha decidido no trabajar directamente con la base de datos del teléfono.

Para la interacción con la base de datos, la metodología con la que manejaremos los datos, será incluirlos como recursos dentro de los directorios de la propia aplicación, es decir, manejaremos un sistema de archivos que estará conformado por los marcadores y los objetos virtuales con sus respectivas texturas.

5.4.2.2 Programación de la interfaz entre la aplicación y el entorno

Para realizar la interfaz de la aplicación se utilizarán los SDK de Android e iOS, los cuales permiten realizar interfaces de una manera muy fácil, ya que para nuestro caso solo necesitaremos mostrar en la pantalla del celular las imágenes capturadas en tiempo real por la cámara del celular.

En la figura 5.22 podemos apreciar un ejemplo de lo que deberá ser la interfaz de la aplicación con el entorno una vez

culminada.



Figura 5.22 Interfaz entre la aplicación y el entorno

5.4.2.3 Programación para el reconocimiento de las marcas

Para el reconocimiento de las marcas o marcadores, nosotros analizamos el código de las funciones que nos proporcionan las librerías NyArtoolkit y ARToolkitPlus.

Para facilitar la comprensión se realizó un diagrama de estructura, en el que se representa las funciones y datos intercambiados que toman parte en la identificación de los patrones.

Este funcionamiento se encuentra estructurado de la siguiente manera:

1. El sistema de realidad aumentada utiliza una función o método para capturar una imagen de la cámara.
2. Se lleva cabo una llamada al método arDetectMarker que recibirá como parámetro la imagen capturada y nos devolverá el número de patrones detectados.
3. La función arDetectMarker invoca a una subfunción de etiquetado (labeling) que recibe la imagen capturada y

devuelve la imagen etiquetada con las zonas detectadas.

4. La subfunción labeling se encargará de comparar cada punto de la imagen con un cierto umbral de intensidad, y obtendrá todas las zonas de la imagen que cumplen con cierta condición de intensidad.
5. La función arDetectMarker se encargará de utilizar la información, e invocará a la función arDetectMarker2, este devolverá el número de zonas encontradas que se consideran válidas.
6. La función arDetectMarker2 obtendrá la información de contorno de las zonas detectadas. Y descartará aquellas zonas que no son cuadradas, zonas que son demasiado grandes o demasiado pequeñas.
7. Finalmente arDetectMarker comprobará si en su interior se encuentra alguna de las plantillas almacenadas.

5.4.2.4 Programación para la interacción entre la base de datos y las marcas

En nuestro caso se utilizará una base de datos que estará formada por un conjunto de archivos con extensión .pat, almacenados en su respectivo subdirectorio.

Cada archivo guardará la codificación de la marca en un mapa de bits, las mismas que fueron diseñadas y creadas en el capítulo anterior.

Estos subdirectorios son generados por defecto al momento de

crear un proyecto Android o iOS como explicaremos a continuación:

Android

Entre los directorios que se crean por defecto tenemos: **src**³⁹, **gen**⁴⁰, **assets**⁴¹, y **res**.

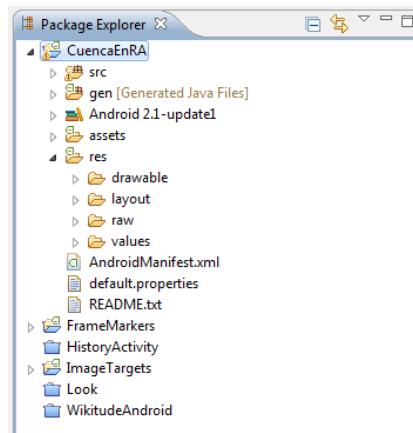


Figura 5.23 Directorios generados por defecto

El directorio **res**, es donde mantendremos la mayor parte de los recursos de nuestra aplicación, la cual tiene una estructura específica para mantener organizados cada uno de los recursos utilizados.

Dentro de la carpeta **res** encontramos al directorio **raw**, el cual guarda archivos de cualquier tipo y pueden ser accedidos de manera directa.

El subdirectorio **raw** guardara los archivos .pat y .md2⁴² que serán utilizados para la aplicación.

³⁹src.- Directorio que el almacena los paquetes que contienen las clases a utilizar.

⁴⁰gen.- Directorio que guarda los archivos creados directamente por ejemplo el archivo de recursos.

⁴¹assets.-Directorio para guardar recursos no compilados del proyecto.

⁴².md2.- Es un formato de archivo diseñado para guardar modelos 3D (objetos en tres dimensiones).

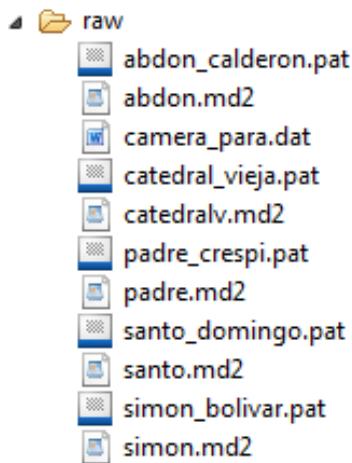


Figura 5.24 Archivos utilizados en la aplicación

iOS

Al iniciar nuestro proyecto existen directorios que se crean por defecto como los son: **Classes**⁴³, **Resources**⁴⁴, **Frameworks**⁴⁵, **Other Sources**⁴⁶, **Products**⁴⁷.

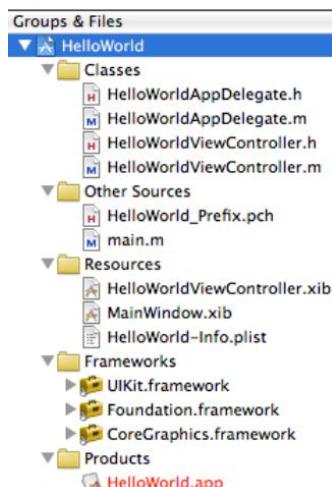


Figura 5.25 Directorios generados por defecto en Xcode

⁴³Classes.- Directorio que contiene las clases a utilizar.

⁴⁴Resources.- Directorio que contiene los ficheros de interfaz de usuario.

⁴⁵Frameworks.- Directorio que contiene las librerías que se usan en la aplicación.

⁴⁶Other Sources.- Directorio que contiene ficheros que no son generados por el desarrollador.

⁴⁷Products.- Directorio que contiene la aplicación una vez se ha compilado el proyecto.

Para almacenar nuestros objetos dentro del sistema de archivos en nuestro proyecto Xcode se crea un nuevo directorio llamado Objetos.

El directorio "objetos 3D" almacenará los archivos .obj⁴⁸.

Las marcas a reconocer serán almacenadas en otro directorio llamado "Marcadores", que será creado para el efecto.

5.5 Desarrollo de manuales de usuario y documentación

El manual y documentación de usuario será desarrollado con la herramienta Microsoft Office Word y está pensado en personas que tengan conocimientos en el desarrollo de aplicaciones para celulares con sistema operativo Android e iOS.

Este manual está expuesto en el Anexo 1 y fue desarrollado con el máximo detalle posible para que sirva de apoyo al momento de realizar una aplicación de realidad aumentada.

⁴⁸.obj.- Es un formato de archivo diseñado para guardar modelos 3D (objetos en tres dimensiones).

CAPITULO 6

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL PORTAL WEB

6 Capítulo VI ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL PORTAL WEB

6.1 Análisis de los requerimientos del portal

El portal web contendrá toda la información necesaria para el uso y la descarga de la aplicación, así como también debe proporcionar una interfaz para la carga de nuevas versiones de la aplicación de realidad aumentada.

A continuación se listan los requerimientos más importantes, con los que debe cumplir la aplicación tanto para Android como iOS:

- Mostrar información acerca de la aplicación y de los objetos que contiene la aplicación.
- Ofrecer la opción de descarga de la aplicación y las diferentes versiones de la misma.
- Cargar nuevas versiones del sistema.
- Poseer una interfaz gráfica amigable y de fácil manejo para el usuario.

6.2 Diseño del modelado de datos para el portal

El modelo de datos para almacenar la información necesaria de la aplicación se presenta en el siguiente diagrama entidad-relación:

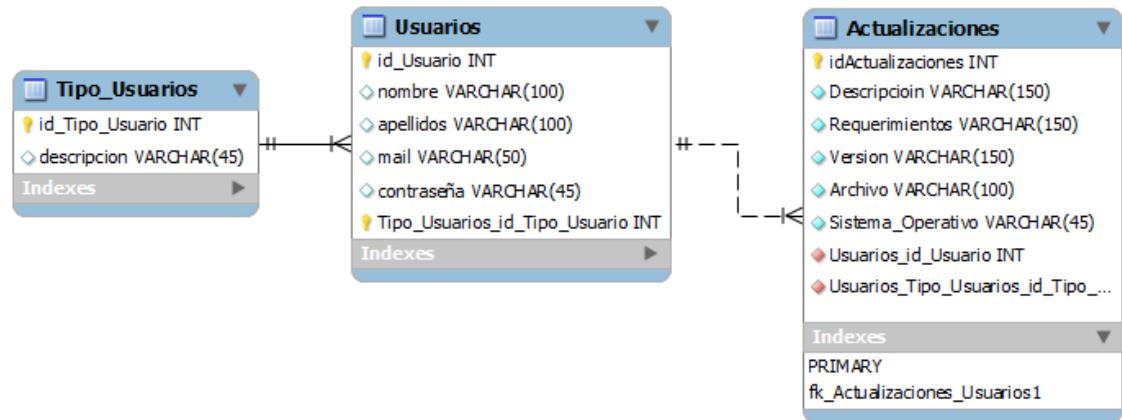


Fig. 6.1 Modelo Entidad-Relación para el portal Web

6.3 Implementación del portal web

6.3.1 Implementación del módulo de descarga de la aplicación

El módulo de descargas de la aplicación en nuestro portal web, presenta la opción de seleccionar la aplicación para las dos plataformas tratadas a lo largo de este documento (Android e iOS).

El usuario podrá descargarse directamente el instalador de la aplicación a su teléfono móvil. Así mismo podrá realizar esta tarea a través de una PC.

El archivo a descargarse tendrá extensión .apk en el caso de Android y .ipa para el sistema operativo iOS.

Esta sección contará con una descripción de todos los objetos disponibles en la aplicación, asimismo se presentará la información necesaria acerca de los requisitos mínimos, tanto de hardware como de software, para poder ejecutar la aplicación sin ningún tipo de inconveniente.

6.3.2 Implementación del módulo de descarga de marcas

Para poder implementar el módulo de descargas se vio necesario utilizar el recurso **resources** (ver figura 6.2), directorio que es creado por defecto al momento de crear un proyecto de aplicación web.

Para poder llamar a cada archivo que se encuentra dentro de este recurso, se procede a realizar una consulta **select** a la base de datos mediante la utilización de una clase conexión.

La base de datos almacena los nombres de los archivos que se encuentra en el directorio **resources** para su respectiva consulta y descarga.

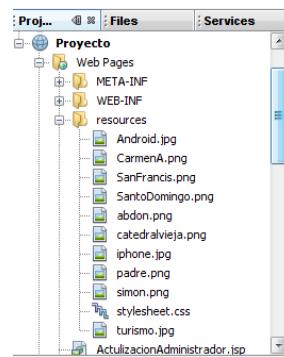


Figura 6.2 Directorio resources

6.3.3 Implementación del módulo de actualización de marcas

Para poder implementar este módulo lo que se realiza es agregar un nuevo archivo al recurso **resources** mediante la función copiar y pegar del sistema operativo.

Para que este archivo nuevo se encuentre disponible se agregará un enlace a este, el cual será guardado en la base de datos y de esta forma poder hacer un correcto mantenimiento del mismo.

6.4 Desarrollo de manuales de usuario y documentación

El manual y documentación de usuario será desarrollado con la herramienta Microsoft Office Word y está pensado en personas que tengan conocimientos mínimos en el desarrollo y administración de aplicaciones web.

Este manual está expuesto en el Anexo 2 y fue desarrollado con el máximo detalle posible para que sirva de apoyo al momento de realizar una aplicación web.

CAPITULO 7

INSTALACIÓN Y PRUEBAS DE LA APLICACIÓN EN EL TELÉFONO MÓVIL

7 Capítulo VII INSTALACIÓN Y PRUEBAS DE LA APLICACIÓN EN EL TELÉFONO MÓVIL

7.1 Instalación de la aplicación

Para la instalación de la aplicación dependiendo de los diferentes ficheros descargados del portal web (.apk para Android, ipa para iOS), lo primero que tenemos que hacer es realizar ciertas configuraciones en el teléfono móvil.

Android

Para habilitar la ejecución de aplicaciones de orígenes desconocidos se debe ir en el terminal al programa de "Ajustes", seleccionar "Aplicaciones" y marcar la casilla correspondiente a "Orígenes Desconocidos" (ver figura 7.1a y 7.1b).

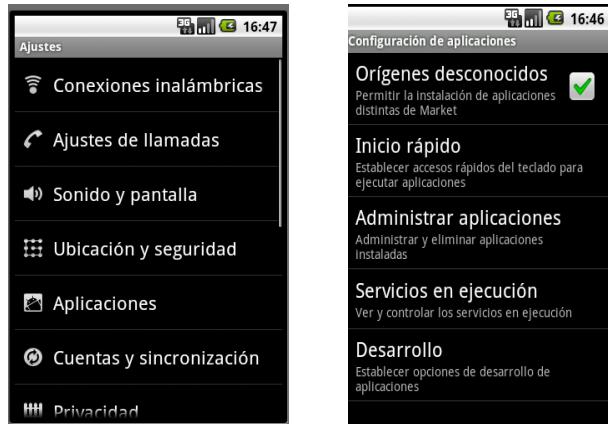


Figura 7.1 Habilitar instalación de aplicaciones de orígenes desconocidos.

Además se necesitará que el teléfono móvil cumpla con ciertos requisitos de hardware y software que se detallan a continuación:

- Sistema operativo Android con una versión superior o igual a 2.1.

- Tener cámara de video
- Procesador superior a 600MHz.

La aplicación fue instalada y probada en tres celulares diferentes:

- **Sony Ericsson XPERIA X10 mini.**- Este celular tiene cámara de 5Mp, Android versión 2.1 update y procesador Qualcomm MSM7227 600MHz. La aplicación funciona perfectamente con objetos que no superen un 1MB en tamaño.
- **Samsung Galaxy Mini.**- Este celular tiene cámara de 5Mp, Android versión 2.2 y procesador Qualcomm MSM7227 ARMv6 600MHz, GPU Adreno 200. La aplicación funciona perfectamente con cualquier tipo de objetos.
- **Motorola.**- Este celular tiene cámara de 5Mp, Android versión 3.0 y procesador Qualcomm. La aplicación funciona perfectamente con cualquier tipo de objetos.

iOS

Para habilitar la ejecución de aplicaciones de orígenes desconocidos en el teléfono móvil, se necesita tener instalado el iPhone Jailbreak para iOS 4.3.3, este permite desbloquear el dispositivo para ejecutar aplicaciones distintas a las alojadas en App Store, y es necesario para poder realizar la simulación con Xcode directamente en el dispositivo.

A continuación los pasos necesarios para instalar el iPhone Jailbreak:

- Descargamos **Redsn0w 0.9.6RC16** ya que este funciona perfectamente en los dispositivos con iOS 4.3.3, iPhone 4, iPhone 3GS, iPad, iPod touch 3G/4G.
- Ejecutamos **RedSn0w 0.9.6RC16** y pulsamos la tecla "Browse" (ver figura 7.2).



Figura 7.2 Ejecución del RedSn0w 0.9.6RC16

- Ejecutamos **RedSn0w 0.9.6RC16** y pulsamos la tecla "Browse" (ver figura 7.2).
- Ahora seleccionamos el archivo **IPSW iOS 4.3.3** para tu dispositivo y pulsamos "Open".
- Pulsamos "Next", aparecerá una pantalla de opciones adicionales, seleccionamos "Install Cydia" y pulsamos "Next" (ver figura 7.3).



Figura 7.3 Selección de la instalación del Cydia

- Pasamos el terminal a **modo DFU**, conectamos el dispositivo al puerto USB y lo apagamos completamente. Pulsamos el botón "Next" y seguimos el siguiente procedimiento: Arrancar con los botones "Home" y "Power" durante 3-4 segundos y luego soltar el botón de encendido, mientras el "Home" se queda apretado por unos segundos y **Redsn0w** lo reconoce.



Figura 7.4 Reconocimiento del Redsn0w

- Ahí comienza el jailbreak propiamente dicho y en cuanto finalice, el terminal se reiniciará automáticamente en modo **jailbreak untethered sobre iOS 4.3.3**.

Además se necesitará que el teléfono móvil cumpla con ciertos requisitos de hardware y software que se detallarán a continuación:

- Sistema operativo iOS con una versión 4.3 o superior.
- Tener cámara de video.

7.2 Pruebas de usabilidad

El objetivo de estas pruebas será la verificación del manejo del sistema en cuanto al grado de satisfacción por parte de los usuarios finales.

Estas pruebas fueron realizadas cuando el sistema fue completamente terminado y enfocadas a verificar el grado de satisfacción de los atributos no funcionales más importantes del sistema.

Para estas pruebas se vio la necesidad de la selección de 10 personas categorizadas en 3 perfiles distintos, los cuales estuvieron compuestos de 3 personas para el perfil de invitado, 2 personas para el perfil de administrador, 5 personas para el perfil de programador.

Cada perfil fue categorizado de acuerdo al nivel de conocimiento de los usuarios, con respecto a tecnologías de información, como se describe a continuación:

- **Perfil de invitado.-** Es un usuario con conocimientos básicos en el manejo de teléfonos móviles, como por ejemplo poder conectarse a internet y navegar sin problema.
- **Perfil de administrador.-** Es un usuario con conocimientos avanzados en el manejo de teléfonos móviles y administración de sistemas.
- **Perfil de programador.-** Es un usuario técnico, con conocimientos avanzados en el desarrollo de aplicaciones para teléfonos móviles.

Para poder evaluar la usabilidad del sistema, se consideró las siguientes preguntas:

- 1) ¿El sistema cumple con su propósito?

2) ¿La presentación de la información es clara?

3) ¿Cómo evalúa el tiempo de respuesta?

En relación a la pregunta 1, el resultado posterior a la encuesta fue un 100% de aceptación por parte de los usuarios invitados, un 96% para los usuarios programadores y 100% los usuario administradores.

Como se puede apreciar en la figura 7.5 existe una aprobación de la aplicación de más del 90% con respecto a la pregunta 1, por lo que se concluye que el sistema realmente cumple con su propósito.

Pregunta 1

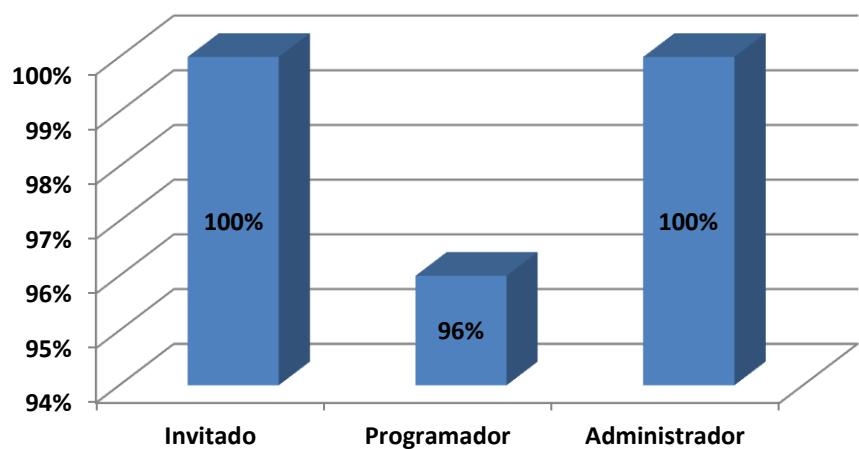


Figura 7.5 Gráfico de usabilidad - Pregunta 1

En relación a la pregunta 2, el resultado posterior a la encuesta fue un 100% para los usuarios invitados, un 100% para los usuarios programadores y 100% los usuario administradores.

De acuerdo a los resultados obtenidos anteriormente (ver Figura 7.6) se concluye que la información es presentada de forma clara.

Pregunta 2

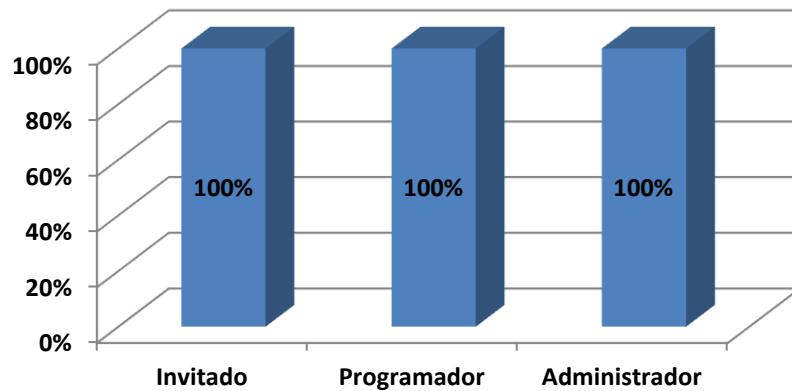


Figura 7.6 Gráfico de usabilidad - Pregunta 2

En relación a la pregunta 3, el resultado posterior a la encuesta fue un 100% para los usuarios invitados, un 100% para los usuarios programadores y 100% los usuario administradores.

De acuerdo a los resultados obtenidos anteriormente (ver Figura 7.7) se concluye que el tiempo de respuesta del sistema desarrollado es el adecuado.

Pregunta 3

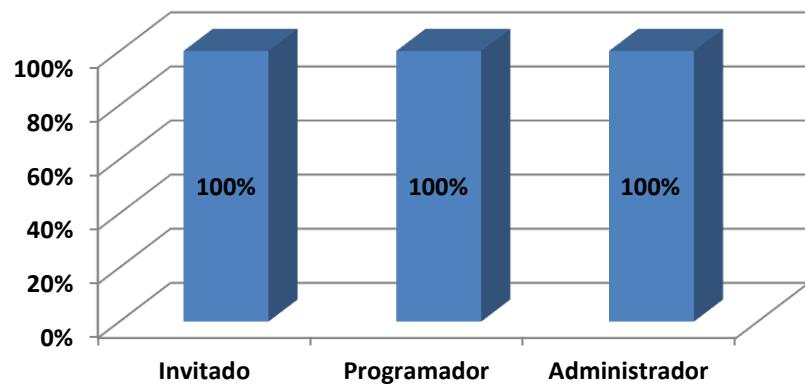


Figura 7.7 Gráfico de usabilidad - Pregunta 3

De acuerdo a los resultados obtenidos anteriormente se concluye que el sistema desarrollado es de fácil manejo para los usuarios finales.

7.3 Pruebas de funcionalidad

El objetivo de estas pruebas será la de comprobar que las funcionalidades requeridas en la especificación de requerimientos que se encuentran en el sistema (ver Capítulo 5).

En la tabla 7.1 se listan las pruebas de sistema y su estado:

Referencia	Requerimiento	Estado
Tabla 5.1	Reconocimiento de marcas	Correcto
Tabla 5.2	Visualización de la información	Correcto
Tabla 5.3	Manual de usuario	Correcto

Tabla 7.1 Pruebas de sistema

7.4 Pruebas de fiabilidad

Para medir la fiabilidad del sistema desarrollado se han tomado en cuenta diferentes aspectos en los cuales consideramos que el sistema desarrollado debe ofrecer un óptimo rendimiento.

Los aspectos que fueron analizados son los siguientes:

- La ejecución de la aplicación en el teléfono una vez instalada estará disponible cuando y cuantas veces lo desee el usuario, sin mayores requerimientos, teniendo como único requisito que el teléfono y la cámara estén funcionales.
- El uso de recursos del teléfono por parte de la aplicación, tanto de hardware como de software no colapsa el sistema al momento que este se encuentre en ejecución. Es importante observar que esto se cumplirá siempre y cuando el teléfono cumpla con los requerimientos

mínimos de hardware y software, para los cuales está diseñada una versión específica del sistema.

- La distancia del teléfono con respecto a la marca para un apropiado reconocimiento no es fija, es decir, esta puede variar. Para la visualización completa de la información la distancia mínima sugerida debe ser de unos 75 cm y para evitar distorsiones no debe superar los 130 cm, mientras que se cumplan estos parámetros la aplicación funcionará de manera correcta y 99% fiable
- El sistema desarrollado no genera conflictos con otras aplicaciones instaladas en el teléfono.

7.5 Depuración y corrección de errores

7.5.1 Corrección de errores

Luego de realizar todas las pruebas detalladas en los puntos anteriores se logró resolver los inconvenientes que presentaba la aplicación tanto para la plataforma Android como para iOS.

El error más común presentado fue el reconocimiento de la marca para la visualización de la información, razón por la cual fue necesario realizar pruebas con diferentes tipos de marcadores con diferente grado de complejidad.

7.5.2 Depuración de la aplicación

Para la depuración de la aplicación tanto para Android como para iOS, se utilizará las propias opciones de los respectivos IDE's de desarrollo para cada una de las plataformas.

Estas opciones nos facilitan el trabajo al momento de eliminar variables y/o métodos declarados que no fueron utilizados, así como

también paquetes importados innecesariamente. De esta forma se podrá optimizar el uso de la memoria de los dispositivos.

7.6 Desarrollo de manuales de instalación

El manual de instalación, está pensado en personas que tengan conocimientos básicos en la instalación de aplicaciones para teléfonos móviles con sistema operativo Android e iOS. Teniendo en cuenta que el instalador de la aplicación será descargado vía PC o directamente al teléfono desde un repositorio web.

Este manual está expuesto en el Anexo 3 y fue desarrollado con el máximo detalle posible para que sirva de apoyo al momento de la descarga e instalación de la aplicación en el teléfono móvil.

Conclusiones

Al terminar el proyecto propuesto se pudo desarrollar la aplicación de realidad aumentada que ayudará a dar a conocer los lugares turísticos y emblemáticos de nuestra ciudad a cualquier usuario de dicho sistema. Este programa podrá ser usado no solo por los turistas sino por todas las personas interesadas en saber algo más sobre nuestra ciudad, dando un énfasis a los lugares donde se encuentra gran parte de la rica historia de Cuenca.

A continuación se presentan las principales conclusiones que obtuvimos al desarrollar este proyecto:

- La realidad aumentada es una herramienta que puede ser utilizada en infinidad de campos como: programas educativos de estimulación para niños con parálisis cerebral, sistemas educativos, aplicaciones de entretenimiento, etc.
- Los dos mejores sistemas operativos para dispositivos móviles en la actualidad son Android e iOS, por todas las innovaciones y funcionalidades que brindan. Asimismo, es importante anotar que según estimaciones realizadas serán las dos plataformas que lideren el mercado a nivel mundial.
- Con la aplicación desarrollada se puede promocionar a Cuenca de una manera diferente mediante el uso de una nueva tecnología.
- Es fundamental conocer las funcionalidades del dispositivo móvil para poder obtener el mayor provecho posible.
- Existen varias librerías gratuitas que nos ayudan en la implementación de aplicaciones de realidad aumentada, el uso de NyArtoolkit y ArtoolkitPlus aplicadas en Android OS e iOS respectivamente, facilitaron mucho el desarrollo del software planteado.

- En cuanto al diseño de los marcadores se sugiere que estos sean de forma compleja y no utilizar únicamente una palabra.
- Las marcas deben estar impresas en un material que no genere demasiado brillo con la luz del ambiente, además se debe considerar que la luz solar no caiga perpendicularmente hacia la cámara del dispositivo al enfocar a la marca.
- Se implementó un portal web, donde el usuario podrá descargar la aplicación y las diferentes versiones del mismo.
- La aplicación es de fácil uso, y de acceso gratuito a través del portal web.
- Se comprobó que en la plataforma iOS el rendimiento de la librería ArtoolkitPlus es mucho mejor que NyArtoolkit.
- El uso de la tecnología JSF (Java Server Faces) es de gran ayuda al momento de crear páginas web para navegadores de escritorio y móviles.

Recomendaciones

- Para iniciarse en el mundo del desarrollo de aplicaciones móviles para Apple se recomienda contar con una máquina Macintosh y disponer de los dispositivos móviles para realizar pruebas reales.
- Como recomendación para una segunda fase del proyecto, se pueden añadir mayores funcionalidades a la aplicación como por ejemplo:
 - Incorporar un menú en la pantalla principal de la aplicación, que nos permita escoger varias opciones como:
 - Elegir el idioma de la información que se visualiza.
 - Manipular el tamaño del objeto 3D, mediante esta opción hacer que el objeto visualizado pueda aumentar o disminuir su tamaño.
 - A demás de los objetos 3D que se visualizan, podríamos agregar la opción de audio.

Referencias

- [1] ALCARRIA IZQUIERDO Carlos, Desarrollo de un sistema de realidad aumentada en dispositivos móviles, 2010, p.5,
<http://riunet.upv.es/bitstream/handle/10251/8597/PFC%20-%20Desarrollo%20de%20un%20sistema%20de%20Realidad%20Aumentada%20en%20dispositivos%20m%C3%B3viles.pdf>, 18/06/2011
- [2] MACHADO Carlos Marco, Aplicación de la realidad aumentada para un sistema de entrenamiento, 2011, p.21,
<http://upcommons.upc.edu/pfc/bitstream/2099.1/12455/1/73347.pdf>, 09/06/2011
- [3] WIKIPEDIA FOUNDATION, Inc., Android, 06/06/2011, p.1,
<http://es.wikipedia.org/wiki/Android>, 11/06/2011
- [4] WIKIPEDIA FOUNDATION, Inc., BlackBerry OS, 07/06/2011, p.1,
http://es.wikipedia.org/wiki/BlackBerry_OS, 11/06/2011
- [5] WIKIPEDIA FOUNDATION, Inc., BlackBerry, 08/06/2011, p.1,
<http://es.wikipedia.org/wiki/BlackBerry>, 13/06/2011
- [6] WIKIPEDIA FOUNDATION, Inc., iOS (sistema operativo), 04/06/2011, p.1, [http://es.wikipedia.org/wiki/IOS_\(sistema_operativo\)](http://es.wikipedia.org/wiki/IOS_(sistema_operativo)), 13/06/2011
- [7] UNIVERSIDAD DE MÁLAGA, Symbian C++, 03/08/2007, p.63,
<http://www.lcc.uma.es/~pedro/mobile/Projects/pdfs/Tutorial%20symbian.pdf>, 14/06/2011

[8] DÍAZ Almudena, Análisis de Symbian OS para desarrollar aplicaciones distribuidas sobre terminales GPRS, 11/02/2009, p.6,

<http://www.lcc.uma.es/~pedro/mobile/Publications/pdfs/1520.pdf>, 15/06/2011

[9] WIKIPEDIA FOUNDATION, Inc., Symbian OS, 11/06/2011, p.1,

http://es.wikipedia.org/wiki/Symbian_OS, 15/06/2011

[10] WIKIPEDIA FOUNDATION, Inc., Windows Phone, 16/06/2011, p.1,

http://es.wikipedia.org/wiki/Windows_Phone, 20/06/2011

[11] FONSI, Ventajas e inconvenientes de tener Android, 14/12/2010, p.1,

<http://movilae.com/ventajas-inconvenientes-android-google>, 22/06/2011

[12] TECNOCREATIVOS, Ventajas y desventajas del SO Windows Phone,

01/02/2011, p.1, [http://tecnocreativos.com/ventajas-y-desventajas-del-so-](http://tecnocreativos.com/ventajas-y-desventajas-del-so-windows-phone-7)

[windows-phone-7](#), 23/06/2011

[13] ALEXI'S TECHBLOG, Estadísticas! Tablets, previsión de cuotas de

mercado global hasta 2015, 11/04/2011, p.1,

<http://alexistechblog.com/2011/04/11/estadisticas-tablets-prevision-de-cuotas-de-mercado-global-hasta-2015>, 25/06/2011

[14] PATTI Danny, Tipos de lenguajes de programación, 02/12/2009, p.2,

<http://www.monografias.com/trabajos73/tipos-lenguajes-programacion/tipos-lenguajes-programacion.shtml>, 26/07/2011

- [15] CIBERAULA, Lenguajes de programación orientada a objetos, 08/02/2010, p.1, http://java.ciberaula.com/articulo/tecnologia_orientada_objetos, 27/07/2011
- [16] TIOBE SOFTWARE, Measuring Software Code Quality, 18/06/2011, p.1, <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>, 27/07/2011
- [17] ESCUELA DE INGENIERÍA, Java (Certificación Internacional), 18/04/2010, p.1, http://ingenieria.pucesi.edu.ec/site/index.php?option=com_content&view=article&id=63&Itemid=54, 30/07/2011
- [18] WIKIPEDIA FOUNDATION, Inc., Java (lenguaje de programación), 28/07/2011, p.1, [http://es.wikipedia.org/wiki/Java\(lenguaje_de_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java(lenguaje_de_programaci%C3%B3n)), 30/07/2011
- [19] WIKIPEDIA FOUNDATION, Inc., Java (lenguaje de programación), 31/07/2011, p.1, http://es.wikipedia.org/wiki/C%28lenguaje_de_programaci%C3%B3n%29, 01/08/2011
- [20] JHORDI, Programación en C, 28/11/2010, p.1, <http://www.lenguajes-de-programacion.com/programacion-en-c.shtml>, 06/08/2011

[21] WIKIPEDIA FOUNDATION, Inc., C++, 02/08/2011, p.1,
<http://es.wikipedia.org/wiki/C%2B%2B>, 10/08/2011

[22] JHORDI, Programación en c++, 21/08/2011, p.1,
<http://programixe.blogspot.com/p/tutoriales.html>, 22/08/2011

[23] WIKIPEDIA FOUNDATION, Inc., Objective-C, 03/08/2011, p.1,
<http://es.wikipedia.org/wiki/Objective-C>, 22/08/2011

[24] IERUSALIMSCHY Roberto, The programming Language Lua, 2011, p.1,
<http://www.lua.org>, 26/07/2011

[25] WIKIPEDIA FOUNDATION, Inc., “Lua”, 2011, p.1,
<http://es.wikipedia.org/wiki/Lua>, 26/07/2011

[26] ELIAS José, El nuevo lenguaje de programación Lua, 03/04/2007, p.1,
http://eliasx.com/?post_id=2676, 27/07/2011

[27] WIKIPEDIA FOUNDATION, Inc., “Kit de desarrollo de software”,
2008, p.1, http://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software,
28/07/2011

[28] WIKIPEDIA FOUNDATION, Inc, “iPhone SDK”, 2008, p.1,
http://es.wikipedia.org/wiki/iPhone_SDK, 30/07/2011

[29] PAL Kiran, Aava Mobile Releases ‘Virta Android SDK’ for Android Apps Development on Intel ATOM Platform, 2010, p.1,
<http://www.tech2day.com/aava-mobile-releases-virta-android-sdk-for-android-apps-development-on-intel-atom-platform>, 01/08/2011

[30] EUROPAPRESS, Google lanza el SDK de Android 3.2, 2011, p.1,
<http://www.europapress.es/portaltic/movilidad/software/noticia-google-lanza-sdk-android-32-20110718134212.html>, 02/08/2011

[31] SCOTT, Wikitude 4 Now Available, 2009, p.1,
<http://www.androidguys.com/2010/01/15/wikitude-4-now-available>, 03/08/2011

[32] BAUTISTA MONTEJO José Antonio, Qualcomm y la realidad aumentada, 2011, p.1, <http://mundo-tablet.webs-interesantes.es/Noticias/Realidad-aumentada-Qualcomm.htm>, 03/08/2011

[33] SANTOS CALDERÓN Guillermo, Realidad aumentada, 2011, p.1,
http://www.eltiempo.com/opinion/columnistas/guillermosantoscalderon/ARTICULO-WEB-NEW_NOTA_INTERIOR-10111650.html, 05/08/2011

[34] EUROPAPRESS, El SDK de Realidad Aumentada de Qualcomm para iOS ya está disponible, 2011, p.1,
<http://www.europapress.es/portaltic/movilidad/software/noticia-sdk-realidad-aumentada-qualcomm-ios-ya-disponible-20110728160011.html>, 06/08/2011

[35] METAIO, Inc., Documentation Unifeye SDK Mobile, 2010, p.1,
http://doxygen.metaio.com/UnifeyeSDKMobile/page_sdk.html, 06/08/2011

- [36] UNIFEYE MOBILE, SDK para el desarrollo de aplicaciones de realidad aumentada en móviles, 2010, p.1, <http://www.wayerless.com/2010/02/mwc2010-unifeye-mobile-sdk-para-el-desarrollo-de-aplicaciones-de-realidad-aumentada-en-moviles>, 09/08/2011
- [37] METAIO, Inc., Mobile SDK and AR Browser junaio Will Have Image Recognition and Tracking, 2011, p.1, <http://www.metaio.com/press/press-release/2010/metaio-is-ready-for-the-iphone-os-40>, 09/08/2011
- [38] VOCUS, Inc., Ansca Mobile Partners with Appsfire to Bring Marketing Opportunities to Corona SDK Developers, 2012, p.1, <http://www.prweb.com/releases/2012/1/prweb9080159.htm>, 10/08/2011
- [39] MACSOLUCIONES, Programación para iPhone e iPod touch con Corona, 2010, p.1, <http://macsoluciones.com/iphone/programacion/726-programacion-para-iphone-e-ipod-touch-con-corona>, 11/08/2011
- [40] SILES Fernando, Look! un framework de realidad aumentada para Android hecho por estudiantes, 2011, p.1, <http://www.genbeta.com/desarrollo-aplicaciones-moviles/look-un-framework-de-realidad-aumentada-para-android-hecho-por-estudiantes>, 11/08/2011
- [41] VASQUEZ Deyby, Look: Crea aplicaciones de Realidad Aumentada para Android, 2011, p.1 <http://loquenecesita.com/2011/07/look-crea-aplicaciones-de-realidad-aumentada-para-android>, 12/08/2011

[42] HERRERO Manuel Ibáñez, Programando en Mac, 2006, p.35,
http://www.applenext.com/pdf/AppleNext.com%20Programar_en_Mac.pdf,
13/08/2011

[43] CORRALES GARCÍA Alberto, Probando ARToolKitPlus, 2008, p.1,
<http://pfc-albertocorralesgarcia.blogspot.com/2008/06/probando-artoolkitplus.html>, 15/08/2011

[44] WIKIPEDIA FOUNDATION Inc., ARToolKit, 2009, p.1,
<http://es.wikipedia.org/wiki/ARToolKit>, 16/08/2011

[45] MEDIA SOFT INTERACTIVE, Inc., ARToolKit de IOS, 2011, p.1,
http://www.mediasoftinteractive.com/index.php?option=com_content&view=article&id=74%2F&Itemid=65, 17/08/2011

[46] WIKIMEDIA FOUNDATION, Inc., Java (lenguaje de programación),
17/08/2011, p.1,
http://es.wikipedia.org/wiki/Java_%28lenguaje_de_programaci%C3%B3n%29,
23/08/2011

[47] ALDEA UNIVERSITARIA LICEO FRAY PEDRO DE AGREDA,
Lenguaje de programación C y C++, 27/12/09, p.2,
<http://aldeafraypedroagredaunoc.files.wordpress.com/2008/11/lenguaje-de-programacion-c.pdf>, 24/08/2011

- [48] HUERTA Alberto, Laboratorio De Lenguajes De Programación, 21/09/2010, p.1, <http://albertolaboratoriolenguajesprogra.blogspot.com/2010/11/lenguajes-imperativos.html>, 24/08/2011
- [49] MARTIN Vanesa, Crea aplicaciones para Android con este kit de desarrollo, 24/10/2011, p.1, <http://android.malavida.com>, 01/11/2011
- [50] SANABRA SERNA Luis Francisco, Sistema de navegación para invidentes utilizando técnicas de inteligencia artificial y fusión sensorial, 2009, p.34, <http://biorobotics.fi-p.unam.mx/filesrepository/TesisLuisSanabraRevision7.0.pdf>, 26/08/2011
- [51] FERNANDEZ BAILLO Jaime Carrero, Estructura básica de un programa en C, 18/01/2011, p.1, http://www.programacion.com/articulo/lenguaje_c_202/2, 26/08/2011
- [52] UNIVERSIDAD NACIONAL DE COLOMBIA, CLASES EN C++, 22/10/2010, p.1, <http://dis.unal.edu.co/~fgonza/courses/2003/poo/c++.htm>, 26/08/2011
- [53] MARCELO, Objective-C, Xcode And Interface Builder, 21/01/2011, p.1, <http://marcelopavez.blogspot.com/2011/01/objective-c-xcode-and-interface-builder.html>, 01/09/2011
- [54] APPLE Inc., ¿Por qué Objective-C?, 15/11/2010, p.1, <http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/OOP>

[55] SALVADOR GOMEZ Oliver, Ficheros en Android (I): Memoria Interna, 05/07/2011, p.1, <http://www.sgoliver.net/blog/?p=2019>, 02/09/2011

[56] GIRALDEZ Leandro, Definición de Animación 3d, 02/03/2011, p.1, <http://www.alegsa.com.ar/Dic/animacion%203d.php>, 02/09/2011

[57] HARO Roberto, Iglesia de Santo Domingo Cuenca-Ecuador, 21/12/2010, p.1, <http://www.sisepuedeecuador.com/turismo/turismo-urbano/7948-iglesia-de-santo-domingo-cuenca-m-ecuador.html>, 08/09/2011

[58] UNIVERSIDAD DE AZUAY, Cuenca un tesoro escondido, 04/10/1999, p.18, http://uazuay.edu.ec/estudios/examen_turismo.pdf, 10/09/2011

[59] PREFECTURA DEL AZUAY, “Guía turística de cuenca y el Azuay”, Cuenca, Abril 2011, p.10 13 y 16, 18/09/2011

[60] ENRIQUE RODAS Cía. Ltda., Lugares turísticos de Cuenca, 26/03/1999, p.1, <http://www.cuencanos.com/lugaresturisticos/index.php>, 19/09/2011

[61] CELLOS GARCÍA Gabriel, “Historias del ecuador”, Quito, Marzo 1952, p.49, 22/09/2011

[62] VALLEJO María Luisa, “Libro cabildo de la ciudad de cuenca”, Cuenca, Septiembre 2009, p.38 y 39, 18/09/2011

ANEXO 1

MANUAL DE USUARIO DE LA APLICACIÓN DE REALIDAD AUMENTADA

MANUAL DE USUARIO DE LA APLICACIÓN DE REALIDAD AUMENTADA

1 Creación de la aplicación de RA para Android OS

Creación de un nuevo proyecto en Eclipse.

Ejecutamos la aplicación Eclipse y creamos un nuevo proyecto, File>>New>>Android Project (ver figura A1.1).

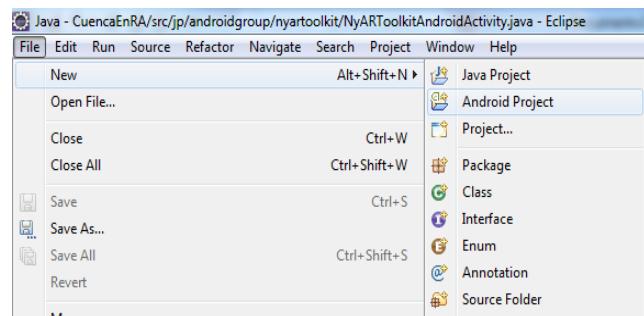


Figura A1.1 Nuevo proyecto en Eclipse

Al elegir que el proyecto sea para Android se nos cargará la siguiente ventana en donde tenemos que ingresar el nombre del proyecto, seleccionar la versión del Android e ingresar el nombre de paquete que contendrá nuestro proyecto y hacemos clic en Finish (ver figura A1.2).

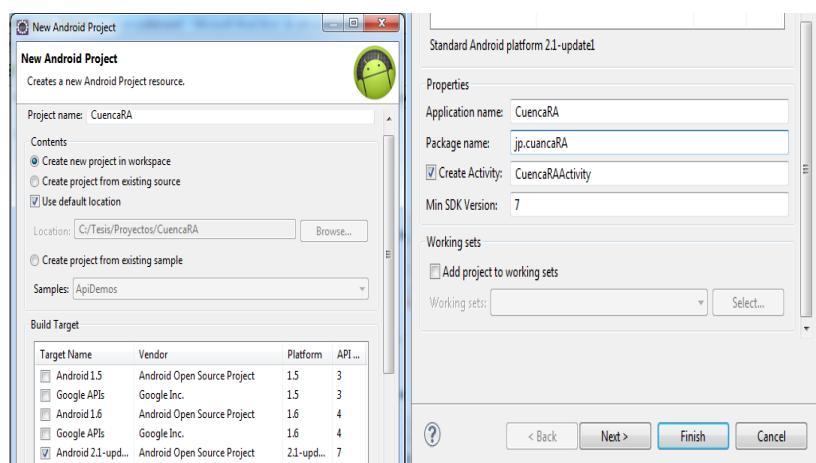


Figura A1.2 Nombre, versión y paquete del proyecto

Una vez concluido los pasos anteriores, se nos presenta una

ventana con los diferentes directorios que se crean por defecto (ver figura A1.3).

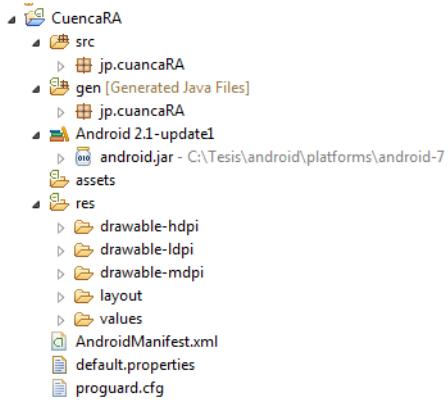


Figura A1.3 Directorios creados por Eclipse

Crear la interfaz de usuario

Dentro del directorio layout damos doble clic en el archivo main para abrir la Interface Graphical Layout (ver figura A1.4).

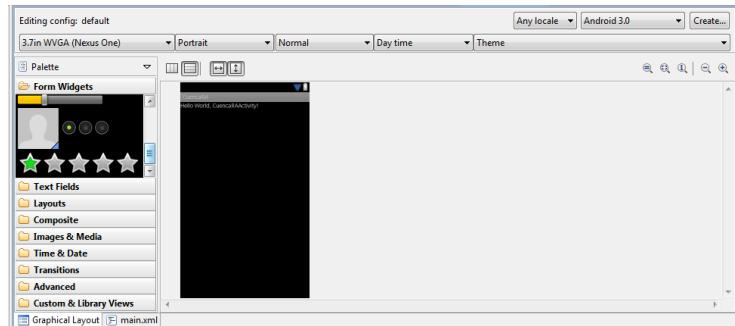


Figura A1.4 Interface Graphical Layout

Para ver los diferentes objetos (Labels, Buttons, Text Fields, etc.) que se pueden utilizar, nos dirigimos a Palette del Graphical Layout (ver figura A1.5).

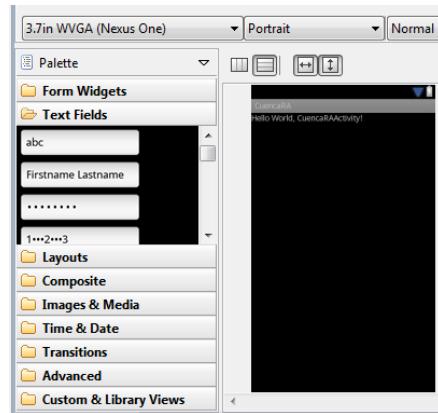


Figura A1.5 Librería de objetos de Interface Graphical Layout

Cargar la librería de realidad aumentada (NyArtoolkit)

Damos clic derecho en el proyecto creado, seleccionamos Import... y damos clic en **File System** (ver figura A1.6).

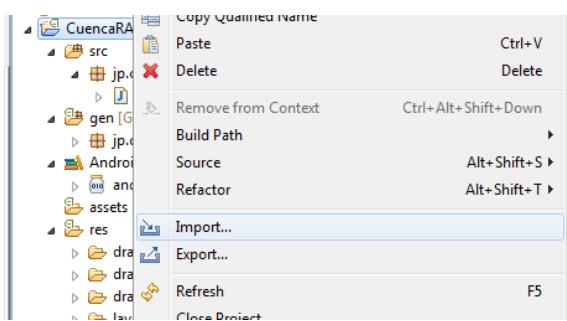


Figura A1.6 Cargar librería existente

A continuación buscamos la ruta donde se encuentra la librería y damos clic en **Finish** (ver figura A1.7).

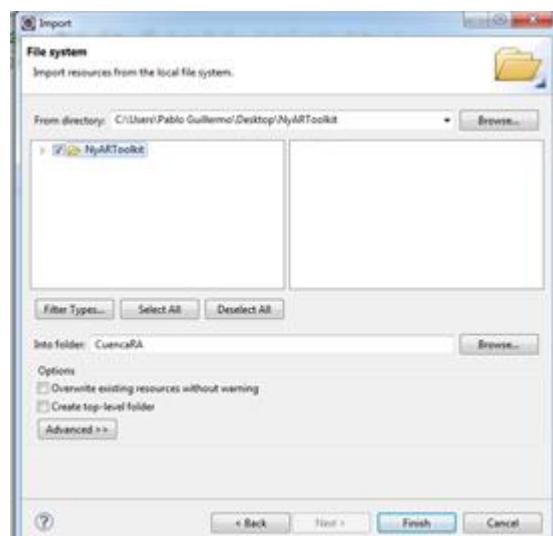


Figura A1.7 Búsqueda de librería

Finalmente podemos ver que la librería está agregada correctamente al proyecto (ver figura A1.8).

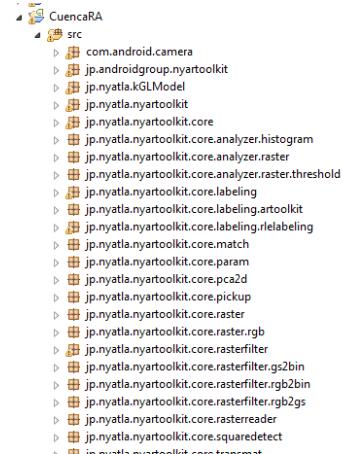


Figura A1.8 Librería NyArtoolkit agregada al proyecto

Crear la base de datos

La base de datos es un sistema de archivos (ver capítulo 5), procedemos a dar clic derecho en el proyecto, agregamos una nueva carpeta y le asignamos un nombre (ver figura A1.9).

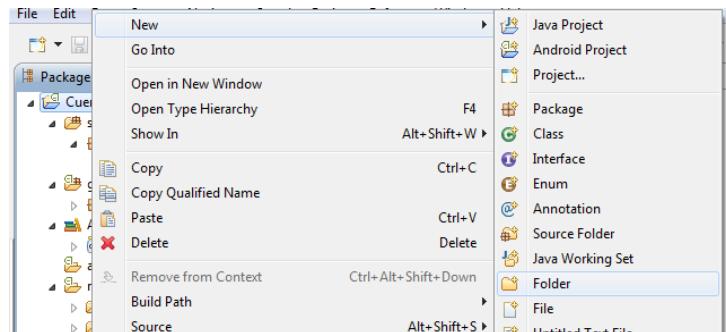


Figura A1.9 Agregar nueva carpeta

Finalmente creado la carpeta con éxito servirá como repositorio para los distintos objetos a mostrar (ver figura A1.10).

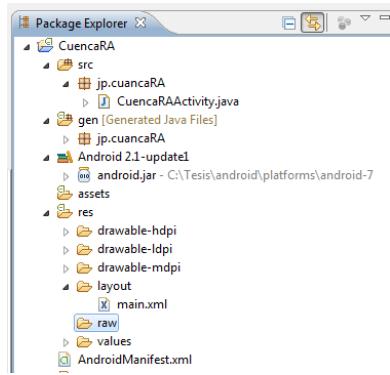


Figura A1.10 Nueva carpeta (raw)

Cargar los objetos 3D

Una vez diseñado los objetos, los agregamos dentro de la carpeta creada en el paso anterior, se puede hacer esto mediante la función copiar-pegar del sistema operativo, aquí se guardaran todos los objetos a visualizar (ver figura A1.11).

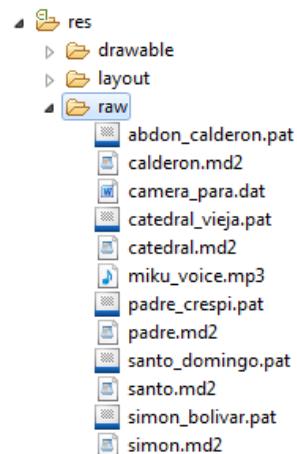
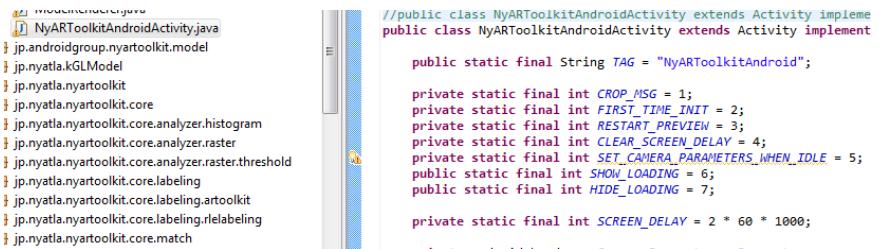


Figura A1.11 Objetos cargados al proyecto

Relacionar el marcador con el objeto

Para relacionar el objeto con el marcador tenemos que modificar el código para lo cual abrimos la clase NyARToolkitAndroidActivity (ver figura A1.12).



```

//public class NyARToolkitAndroidActivity extends Activity implements
public class NyARToolkitAndroidActivity extends Activity implements

    public static final String TAG = "NyARToolkitAndroid";

    private static final int CROP_MSG = 1;
    private static final int FIRST_TIME_INIT = 2;
    private static final int RESTART_PREVIEW = 3;
    private static final int CLEAR_SCREEN_DELAY = 4;
    private static final int SET_CAMERA_PARAMETERS_WHEN_IDLE = 5;
    public static final int SHOW_LOADING = 6;
    public static final int HIDE_LOADING = 7;

    private static final int SCREEN_DELAY = 2 * 60 * 1000;

```

Figura A1.12 Clase NyARToolkitAndroidActivity

Una vez abierta la clase que permitirá relacionar el objeto con el marcador procedemos a agregar unas líneas de código en donde tendremos que poner el nombre del marcador y el nombre del objeto que vamos a utilizar (ver figura A1.13).

```

for (int i = 0; i < 5; i++) {
    width[i] = 80;
}
ArrayList<InputStream> patt = new ArrayList<InputStream>();
//Marcador Monumento de Simón Bolívar
patt.add(getResources().openRawResource(R.raw.simon_bolivar));
//Marcador Monumento de Abdón Calderón
patt.add(getResources().openRawResource(R.raw.abdon_calderon));
//Objeto Monumento de Simón Bolívar
parser = Parser.createParser(Parser.Type.MD2,
    getResources(), "jp.androidgroup.nyartoolkit:raw/simon", true);
parser.parse();
animationObject3d = parser.getParsedAnimationObject();
animationObject3d.rotation().z = -90.0f;
animationObject3d.scale().x = animationObject3d.scale().y = animationObject3d.scale().z = 1.0f;
//Objeto Monumento de Abdón Calderón
parser = Parser.createParser(Parser.Type.MD2,
    getResources(), "jp.androidgroup.nyartoolkit:raw/calderon", false);
parser.parse();
animationObject3d = parser.getParsedAnimationObject();
animationObject3d.rotation().z = -90.0f;
animationObject3d.scale().x = animationObject3d.scale().y = animationObject3d.scale().z = 1.0f;
scene.addChild(animationObject3d);

```

Figura A1.13 Objetos y marcadores relacionados

Compilar y ejecutar

Una vez que todo esté bien, compilamos y ejecutamos el proyecto para ello hacemos clic derecho sobre el nombre y escogemos la opción **Run As** y luego **Android Application** (ver figura A1.14).

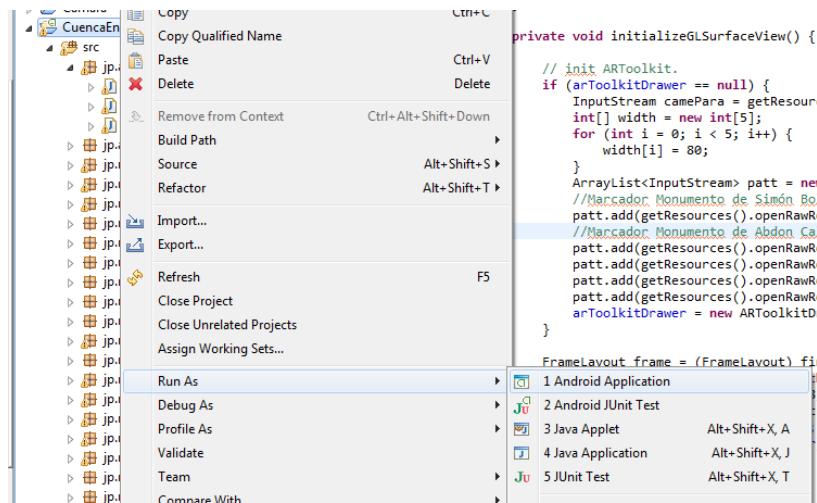


Figura A1.14 Compilación y ejecución del proyecto

2 Creación de la aplicación de RA para iOS

Creación de un nuevo proyecto en Xcode.

Ejecutamos la aplicación Xcode y creamos un nuevo proyecto (ver figura A1.15).

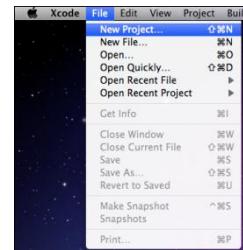


Figura A1.15 Nuevo proyecto en Xcode

Al elegir el tipo de aplicación que deseamos desarrollar, seleccionamos aplicación para iOS y escogemos la opción View-based Application (ver figura A1.16).



Figura A1.16 Selección del tipo de aplicación para iOS

A continuación damos clic en siguiente, le asignamos un nombre al proyecto y seleccionamos ubicación del mismo (ver figura A1.17).

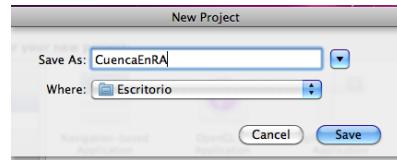


Figura A1.17 Nombre y ubicación del proyecto

Una vez concluidos los pasos anteriores, se nos presenta una ventana con los diferentes directorios que se crean por defecto (ver figura A1.18).

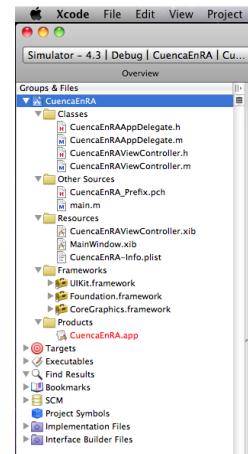


Figura A1.18 Directorios creados por Xcode

Crear la interfaz de usuario

Dentro del directorio resources damos doble clic en el archivo ViewController para abrir el Interface Builder (ver figura A1.19).

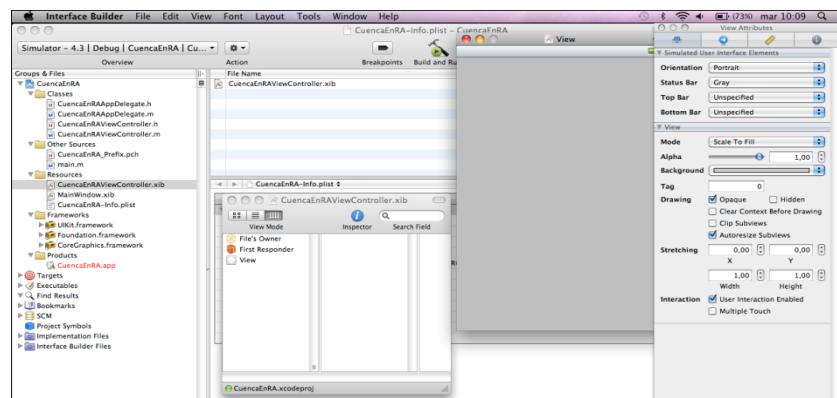


Figura A1.19 Interface Builder

Para ver los diferentes objetos (Labels, Buttons, Text Fields, etc.) que se pueden utilizar, nos dirigimos a la barra de menú del Interface Builder, damos clic en Tools y buscamos la opción Library (ver figura A1.20).

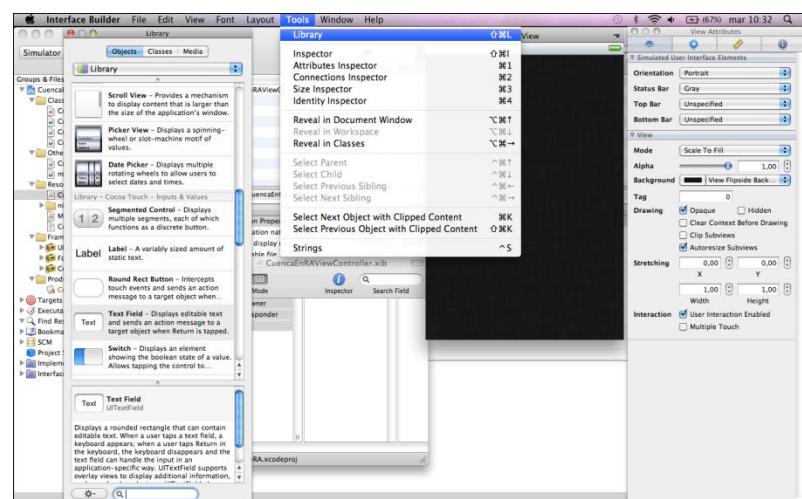


Figura A1.20 Librería de objetos de Interface Builder.

Cargar la librería de realidad aumentada (ArtoolkitPlus)

Damos clic derecho en el proyecto creado, seleccionamos Add y damos clic en "Existing Files" (ver figura A1.21)

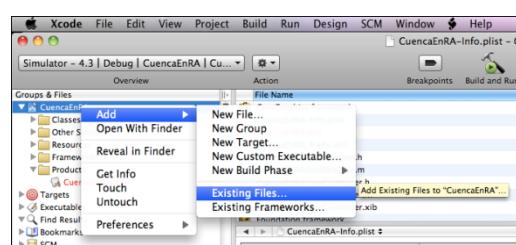


Figura A1.21 Cargar librería existente.

A continuación buscamos la ruta donde se encuentra la librería y damos clic en Add (ver figura A1.22).

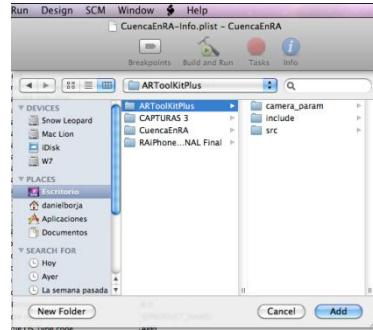


Figura A1.22 Búsqueda de librería.

Seleccionamos la opción "Copy items into destination group's folder" y damos clic en Add (ver figura A1.23).

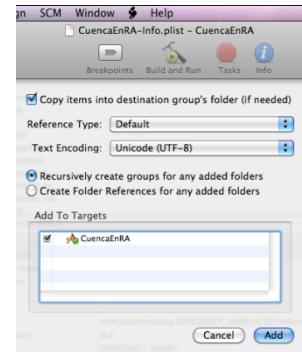


Figura A1.23 Selección de opciones

Finalmente podemos ver que la librería está agregada correctamente al proyecto (ver figura A1.24).

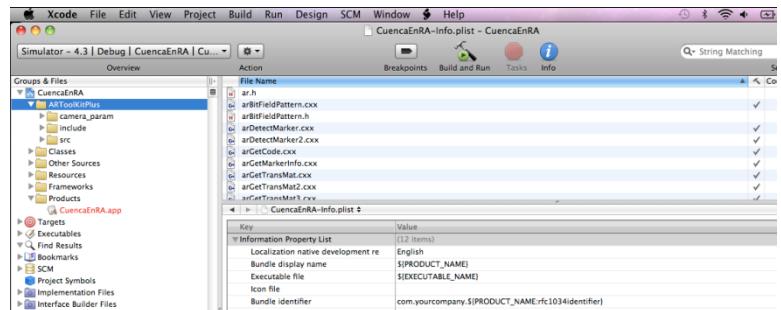


Figura A1.24 Librería ArtoolkitPlus agregada al proyecto

Crear la base de datos

La base de datos es un sistema de archivos (ver capítulo 5), procedemos a dar clic derecho en el proyecto, agregamos un

nuevo grupo o directorio y le asignamos un nombre (ver figura A1.25).

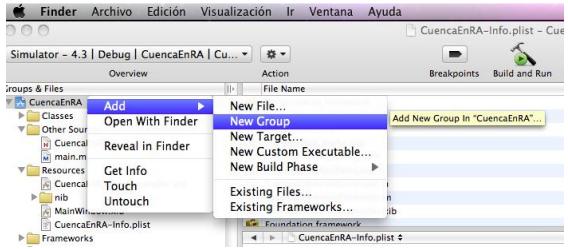


Figura A1.25 Agregar nuevo directorio

Finalmente creado el directorio con éxito servirá como repositorio para los distintos objetos a mostrar (ver figura A1.26).

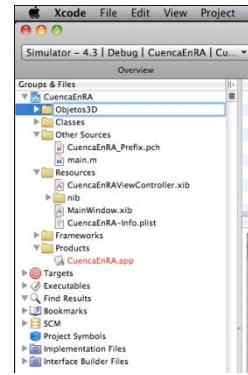


Figura A1.26 Nuevo directorio (Objetos3D)

Cargar los objetos 3D

Una vez diseñado los objetos, agregamos un nuevo directorio dentro del grupo creado en el paso anterior, para distinguir a cada objeto que será cargado. Arrastramos los objetos a la carpeta que se acaba de crear (ver figura A1.27).

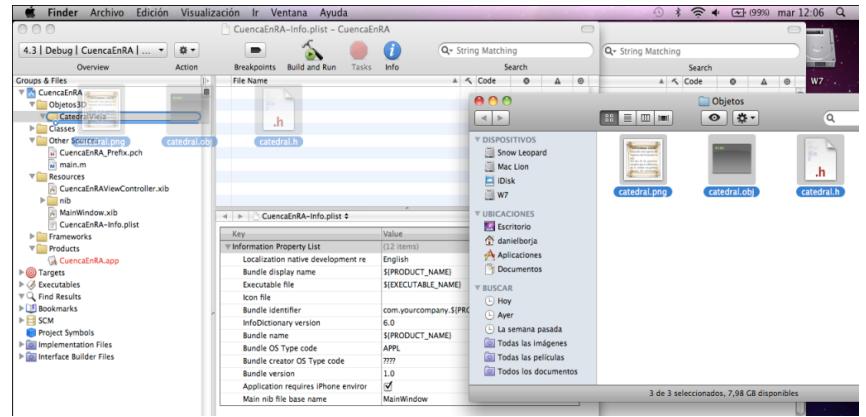


Figura A1.28 Cargar objetos

Seleccionamos la opción "Copy items into destination group's folder" y damos clic en Add (ver figura A1.29).

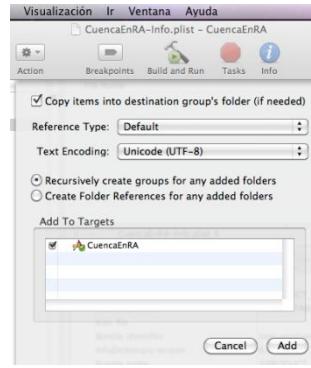


Figura A1.29 Selección de opciones

Una vez cargado los objetos necesitamos agregar un nuevo archivo de tipo .xml que será utilizado para relacionar con el marcador (ver figura A1.30).

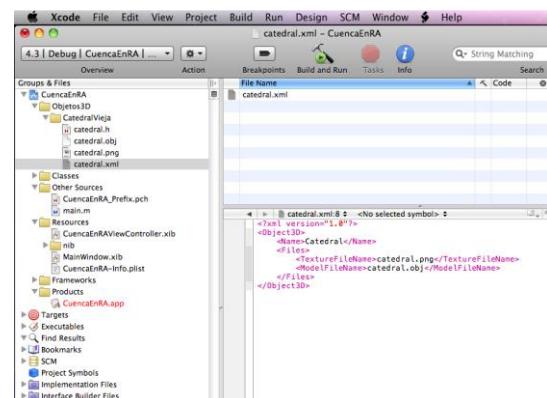


Figura A1.30 Objetos cargados al proyecto

Relacionar el marcador con el objeto

Para relacionar el objeto con el marcador a continuación

describimos los pasos necesarios:

- Crear un nuevo archivo .plist (ver figura A1.31).

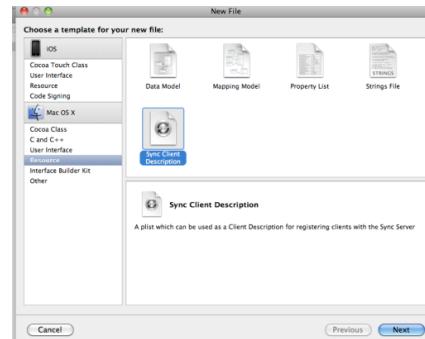


Figura A1.31 Creación del archivo .plist

- Asignamos un nombre (ver figura A1.32).

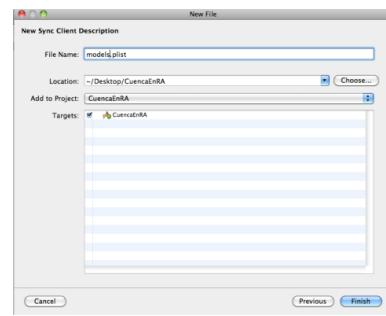


Figura A1.32 Asignar nombre al archivo .plist

- Relacionar el archivo .xml creado en pasos anteriores con el número del identificador del marcador (ver figura A1.33).

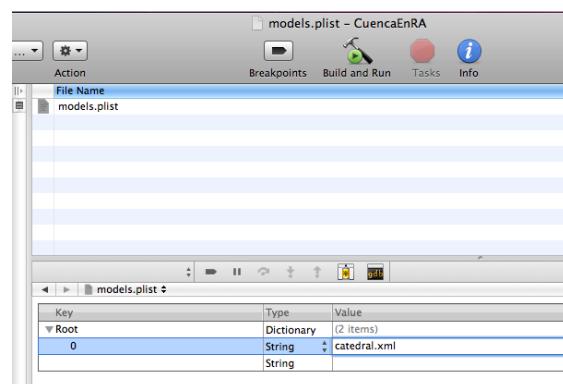


Figura A1.33 Relación marcador-objeto

Compilar y ejecutar

Para compilar y ejecutar es necesario dirigirse a la barra de

menú de Xcode, dar clic en "Build" y buscar la opción "Build" o "Build and Run" para que se instale y ejecute directamente en el dispositivo (ver figura A1.34).

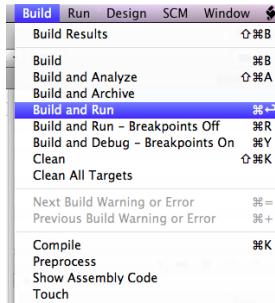


Figura A1.34 Opción para compilar y ejecutar

Cabe recalcar que el dispositivo debe estar conectado e instalado todo lo necesario (ver capítulo 7), además realizar las configuraciones que se indican en la figura A1.35.

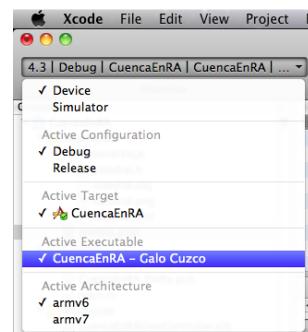


Figura A1.35 Opciones de ejecución y simulación

Es importante mencionar que, para que la aplicación se instale y se ejecute en el dispositivo se debe simular un certificado digital y agregarlo a la aplicación de Xcode.

A continuación se describe los pasos necesarios:

- Debemos ejecutar la aplicación "**Acceso a llaveros**" la cual se encuentra dentro de *Finder/Applicaciones/Utilidades*. Aquí podremos crear un certificado mediante el "**Asistente de Certificados**" (ver figura A1.36).

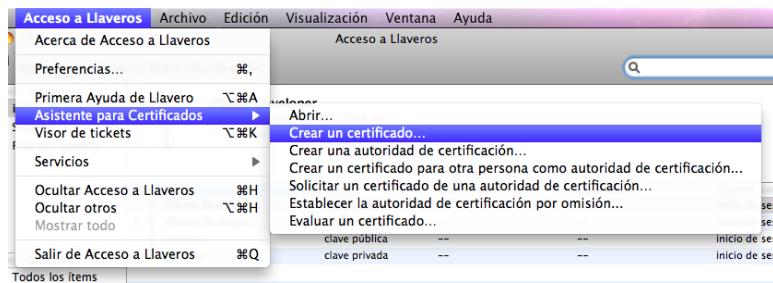


Figura A1.36 Crear nuevo certificado

- Se abrirá el asistente para certificados y seguiremos las instrucciones necesarias, las cuales se muestran en las siguientes figuras:



Figura A1.37 Nombre del certificado



Figura A1.38 Período de validez del certificado



Figura A1.39 Información común

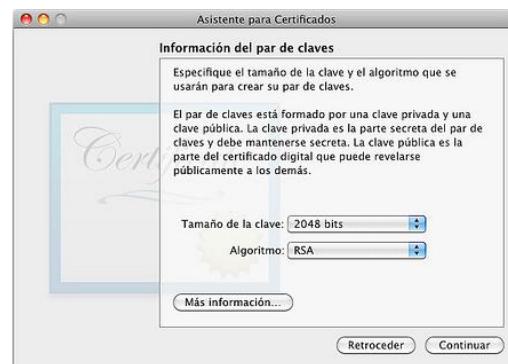


Figura A1.40 Tamaño y algoritmo aplicado al certificado

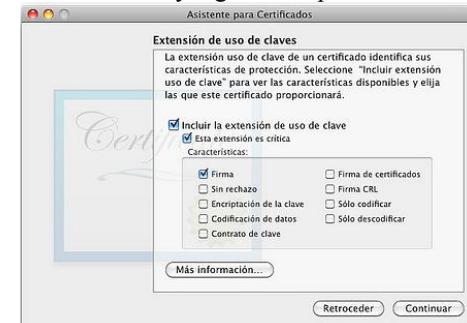


Figura A1.41 Extensión de uso de claves

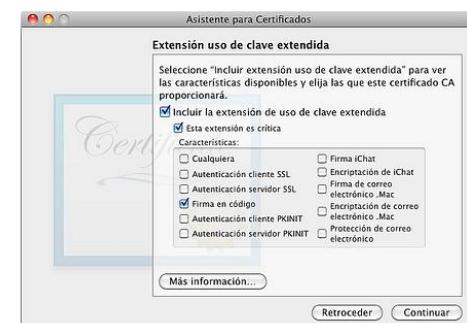


Figura A1.42 Extensión de uso de claves extendida



Figura A1.43 Extensión de restricciones básicas

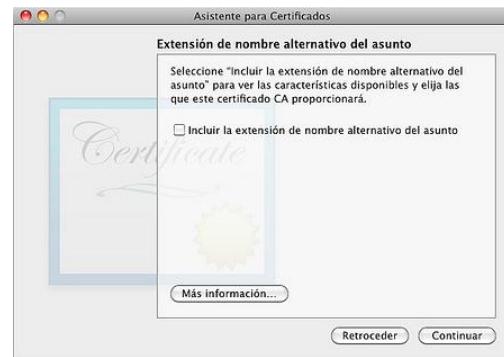


Figura A1.44 Extensión de nombre alternativo de usuario



Figura A1.45 Ubicación del certificado

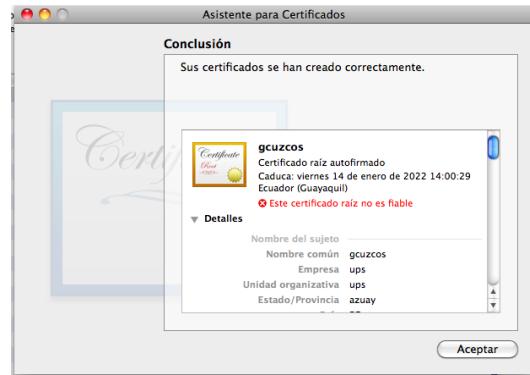


Figura A1.46 Certificado creado

Una vez creado el certificado, será necesario que Xcode permita la utilización de certificados autofirmados. Para esto tendremos que ejecutar el siguiente comando⁴⁹ en el Terminal (ver figura A1.47).



Figura A1.47 Comando ejecutado en la terminal

Ahora debemos asignar el certificado a la aplicación que estamos creando en Xcode, para esto damos clic derecho sobre el proyecto y seleccionamos la opción Get Info (ver figura A1.48).

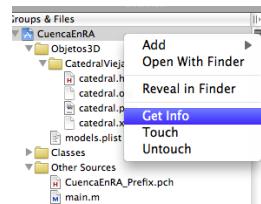


Figura A1.48 Ver preferencias del proyecto

Por ultimo asignamos el certificado autofirmado al proyecto con las opciones que indica la siguiente figura (ver figura A1.49).

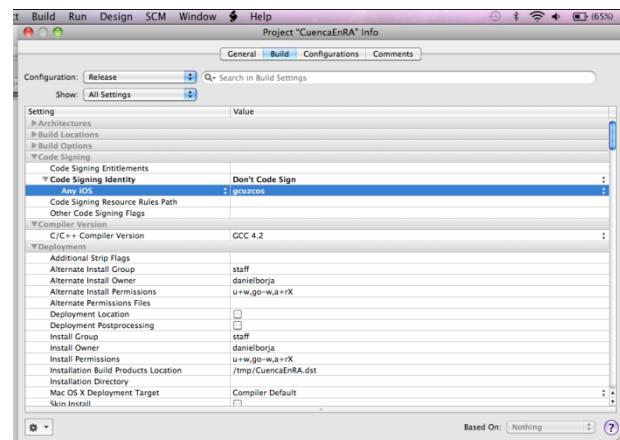


Figura A1.49 Asignación del certificado al proyecto

⁴⁹ Ejecutar en el terminal.- sudo /usr/bin/sed -i .bak 's/XCiPhoneOSCodeSignContext/XCCodeSignContext/' /Developer/Platforms/iPhoneOS.platform/Info.plist

ANEXO 2
MANUAL DE USUARIO DEL PORTAL
WEB

MANUAL DE USUARIO DEL PORTAL WEB

1 Creación del portal web

Creación de un nuevo proyecto en Netbeans

Ejecutamos la aplicación Netbeans y creamos un nuevo proyecto (ver figura A2.1).

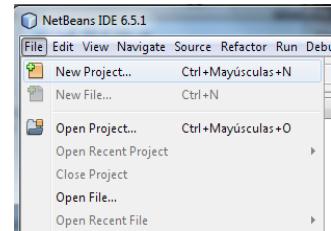


Figura A2.1 Nuevo proyecto en Netbeans

Al elegir la categoría escogemos Java Web y en el tipo de proyecto seleccionamos la opción Web Application (ver figura A2.2).

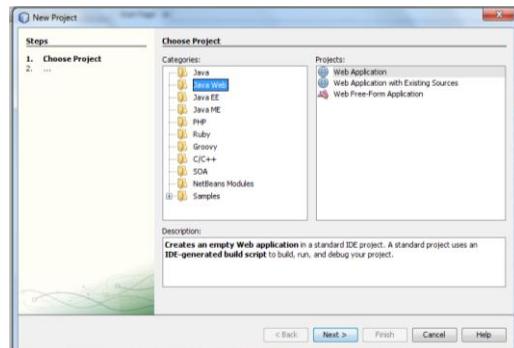


Figura A2.2 Selección del tipo de aplicación

A continuación damos clic en siguiente, le asignamos un nombre al proyecto y seleccionamos la ubicación del mismo (ver figura A2.3).

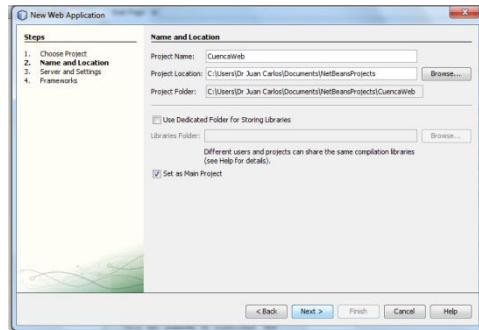


Figura A2.3 Nombre y ubicación del proyecto

Escogemos el servidor con el que vamos a trabajar, para el proyecto se ha decidido por Apache Tomcat (ver figura A2.4).

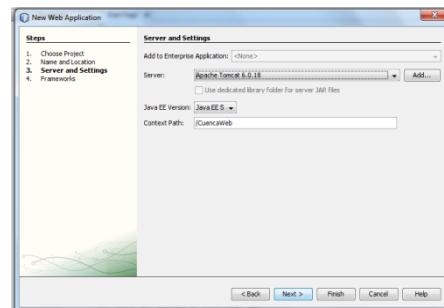


Figura A2.4 Selección del servidor

Ahora debemos escoger el framework con el que vamos a trabajar, para el proyecto se utilizó Visual JSF⁵⁰ (ver figura A2.5).

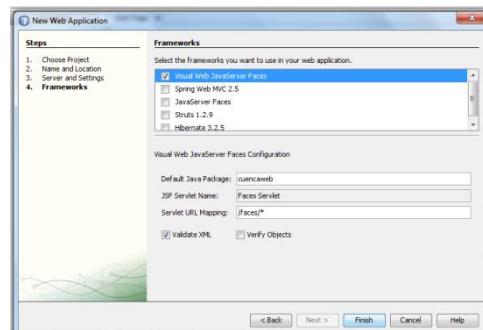


Figura A2.5 Selección del framework

Una vez concluidos los pasos anteriores, se nos presenta una ventana con los diferentes directorios que se crean por defecto (ver figura A2.6).

⁵⁰ JSF.- Java Server Face, es un framework para aplicaciones web en Java EE.

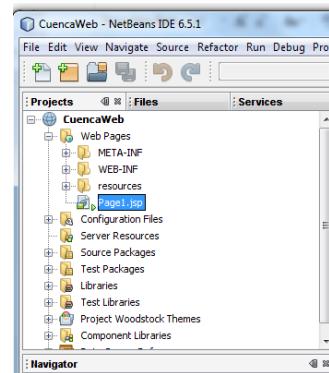


Figura A2.6 Directorios creados por defecto

Crear la interfaz de usuario

Para ver los diferentes objetos (Labels, Buttons, Text Fields, etc.) que se pueden utilizar, nos dirigimos a *pallette* (paleta de objetos) y arrastramos los componentes hacia al panel de diseño (ver figura A2.7).

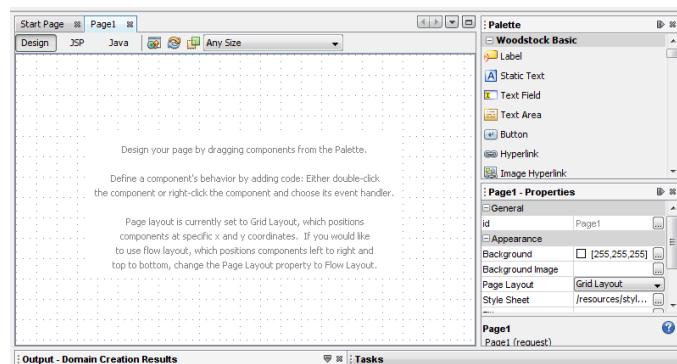


Figura A2.7 Panel de diseño con la paleta de objetos

Instalar y configurar la base de datos

La base de datos con la que trabajaremos es MySQL, para su instalación y configuración ver capítulo 5.

Crear la base de datos

En este punto procedemos a ejecutar el script generado a partir del modelo de datos del capítulo 6.

Para ejecutar el script realizamos lo siguiente:

- Utilizar la línea de comandos de MySQL, ingresar la contraseña (ver figura A2.8).

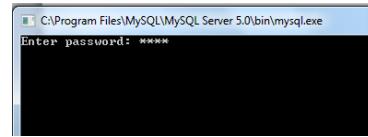


Figura A2.8 Ingreso de contraseña

- Copiar y pegar contenido del script a la línea de comandos de MySQL (ver figura A2.9).

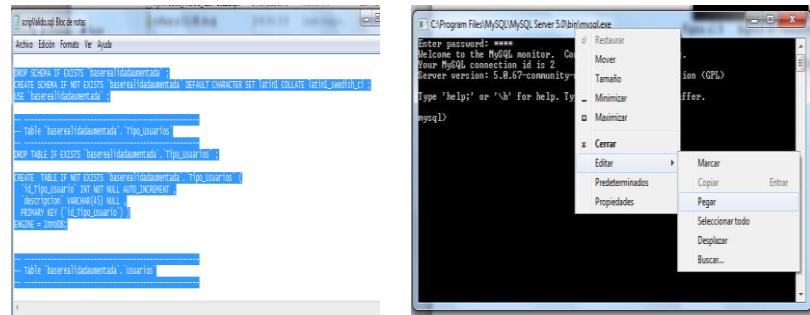


Figura A2.9 Ejecución de script

Creación de la interfaz para la conexión entre la base de datos y la aplicación

Agregamos una nueva clase .java a nuestro proyecto donde tendremos los métodos necesarios para conectarnos a una instancia de la base de datos MySQL (ver figura A2.10)

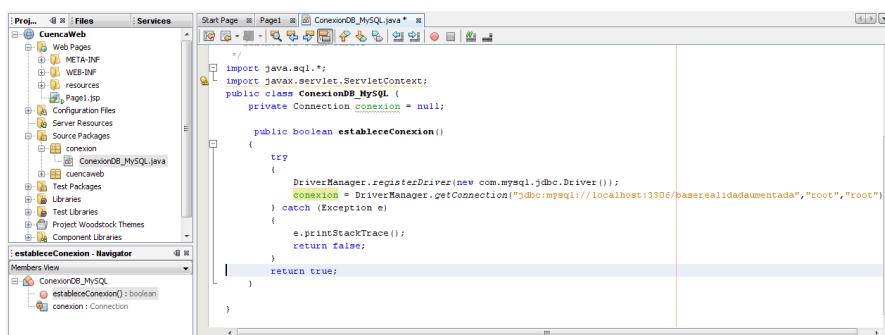


Figura A2.10 Clase Interfaz de conexión

Compilar y ejecutar la aplicación

Una vez que todo esté bien, compilamos y ejecutamos el proyecto para ello hacemos clic derecho sobre el proyecto y

escogemos la opción Run (ver figura A2.11).

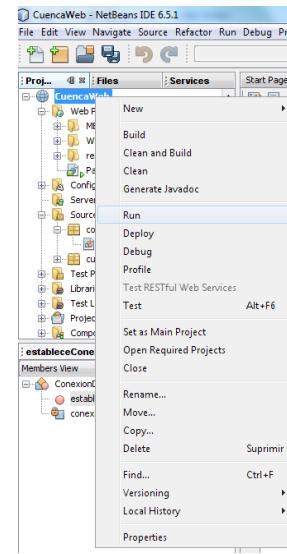


Figura A2.11 Compilación y ejecución de la aplicación

A continuación se muestra la aplicación en ejecución:



Figura A2.11 Compilación y ejecución de la aplicación

Cargar la aplicación al repositorio web

Para subir el instalador de la aplicación de realidad aumentada tendremos que autenticarnos con un usuario con privilegios de administrador, en nuestro caso estos usuarios ya están ingresados previamente.

Usaremos el usuario: edipat86@hotmail.com y la clave: patricio para ingresar a la página que permitirá subir el instalador (ver figura A2.12)

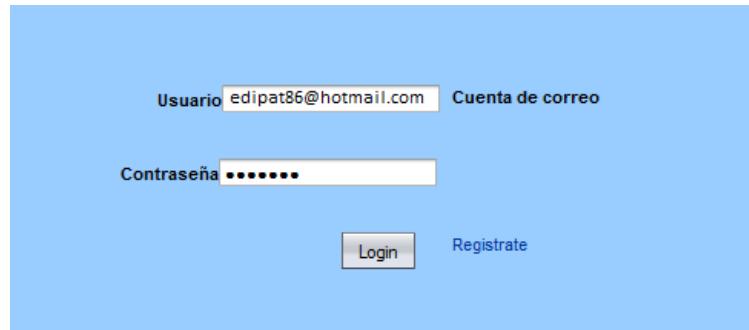


Figura A2.12 Inicio de sesión como administrador

Una vez autenticado procedemos a llenar los datos solicitados en la página como son: una descripción de la aplicación, requerimientos de la aplicación para que pueda ser instalada en el celular, versión de la aplicación, el sistema operativo para el cual estará disponible la aplicación y finalmente el nombre del archivo a subir. Luego hacemos clic en GUARDAR y todos los datos serán guardados en la base de datos (ver figura A2.13).

The form is titled "Descarga de tus aplicaciones virtuales". It features a logo of a castle and a night sky. At the top right are tabs: "INICIO", "ACTUALIZACION", and "DESCARGAS", with "INICIO / ACTUALIZACION" highlighted. The user is identified as "Usuario activo: Patricio". The main area contains fields for "Descripción" (with a note about containing 5 objects, 3 being monuments), "Requerimientos" (mentioning camera and Android 2.1 update), "Versión" (set to 1.0), "Sistema Operativo" (set to Android), and "Archivo" (containing "CuencaEnRA.apk"). A "Guardar" button is at the bottom.

Figura A2.13 Página para subir el archivo

Para comprobar que el archivo se subió correctamente hacemos clic en el botón DESCARGAS, aparecerá un listado de las aplicaciones subidas las mismas que podrán ser descargadas por los usuarios normales (ver figura A2.14).

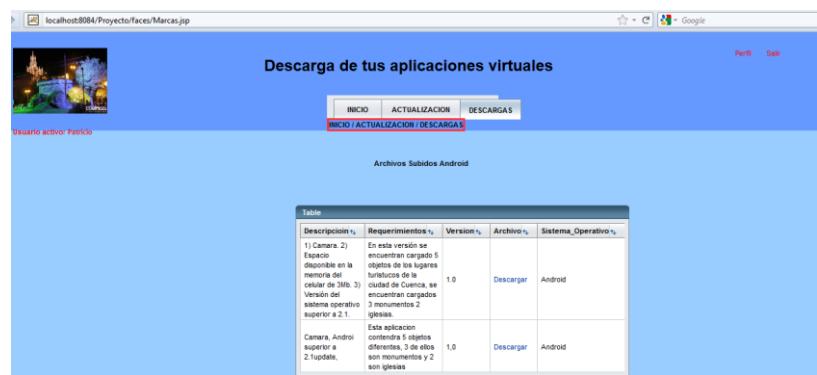


Figura A2.14 Listado de las aplicaciones subidas

ANEXO 3

MANUAL DE USUARIO DE LA INSTALACIÓN Y FUNCIONAMIENTO DE LA APLICACIÓN EN EL DISPOSITIVO

MANUAL DE USUARIO DE LA INSTALACIÓN Y FUNCIONAMIENTO DE LA APLICACIÓN EN EL DISPOSITIVO

1 Descarga de la aplicación

La aplicación se la puede descargar tanto para la plataforma Android como para iOS a través del portal web diseñado para este propósito. A continuación se describen los pasos necesarios para realizar la descarga.

- Registrarse en el portal web.

Datos Personales

Nombres: * Jose

Apellidos: * Granda

Mail: * pepito@hotmail.com

Contraseña: * *****

Guardar

Figura A3.1 Pantalla de registro

- Acceder a la cuenta, ingresamos el usuario y la contraseña e iniciamos sesión (Clic en Login).

Usuario pepito@hotmail.com Cuenta de correo

Contraseña *****

Login Registrate

Figura A3.2 Inicio de sesión

- Ingresar en el módulo de descargas.

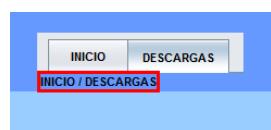


Figura A3.3 Selección del módulo de descargas.

- Escoger el sistema operativo.



Figura A3.4 Selección del OS.

- Seleccionar la versión que soporte nuestro dispositivo de acuerdo al OS escogido.

Descargas SO Android					
Descripción	%	Requerimientos	%	Versión	Sistema_Operativo
1) Camara 2) Espacio disponible en la memoria del celular de 30B. 3) Versión del sistema operativo superior a 2.1.		En esta versión se encuentran cargado 5 objetos de los lugares turísticos de la ciudad de Cuenca, se encuentran cargados 3 monumentos 2 iglesias.		1.0	Descargar Android
Camara, Androi superior a 2.1 update,		Esta aplicación contiene 5 objetos diferentes, 3 de ellos son monumentos y 2 son iglesias		1.0	Descargar Android

Figura A3.5 Listado de aplicaciones para Android.

- Presionar descargar y guardar en el lugar deseado.



Figura A3.6 Descarga de la aplicación.

2 Instalación de la aplicación en el dispositivo

Para poder instalar la aplicación de realidad aumentada, en los distintos sistemas operativos se necesita cumplir los siguientes requisitos:

Dispositivos con Android

- Realizar las configuraciones para la instalación de aplicaciones de orígenes desconocidos en el teléfono móvil (ver capítulo 7).
- Descargar el instalador para el Android OS del portal web (ver punto anterior).
- Realizar los siguientes pasos:
 - Una vez descargado el archivo y guardado en el teléfono móvil, se procederá a buscarlo, para ello nos vamos a menú>>Application>>CuencaEnRA (ver figura A3.7).



Figura A3.7 Preinstalación de la aplicación

- Luego seleccionamos CuencaEnRA y lo mandamos a instalar, nos preguntará si queremos instalar la aplicación y seleccionamos INSTALAR (ver figura A3.8).



Figura A3.8 Confirmación de la instalación.

- Ahora nos aparecerá un mensaje diciendo "APLICACIÓN INSTALADA", allí nos da dos opciones ABRIR y ACEPTAR (ver figura A3.9).



Figura A3.9 Mensaje de aplicación instalada.

Dispositivos con iOS (iPhone)

- Tener instalado el jailbreak y AppSync en el dispositivo (para su instalación, ver capítulo 7).
- Descargar e instalar iTunes51 en la PC de escritorio.
- Descargar el instalador para el iPhone del portal web (ver punto anterior).
- A continuación se realiza los siguientes pasos:
 - Ejecutar iTunes

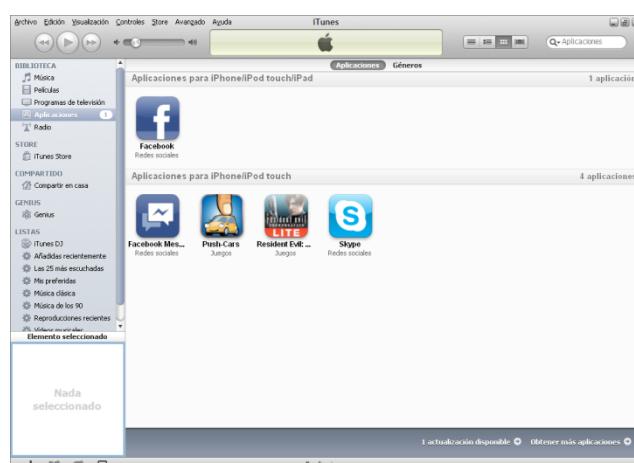


Figura A3.10 Pantalla de inicio del iTunes

⁵¹ iTunes.- Reproductor de medios y tienda de contenidos multimedia desarrollado por Apple para reproducir, organizar y sincronizar iPods, iPhones, iPad.

- Conectar el dispositivo (el iPhone), una vez conectado se podrá visualizar el dispositivo como se indica en la siguiente figura:



Figura A3.11 Pantalla de inicio del iTunes con el dispositivo conectado

- Ubicarnos en el directorio donde nos descargamos el instalador de la aplicación y drag and drop el archivo (.app) a la biblioteca itunes.

Esto también nos servirá para crear el archivo con extensión .ipa (clic derecho mostrar en el escritorio) que se subirá al repositorio web.

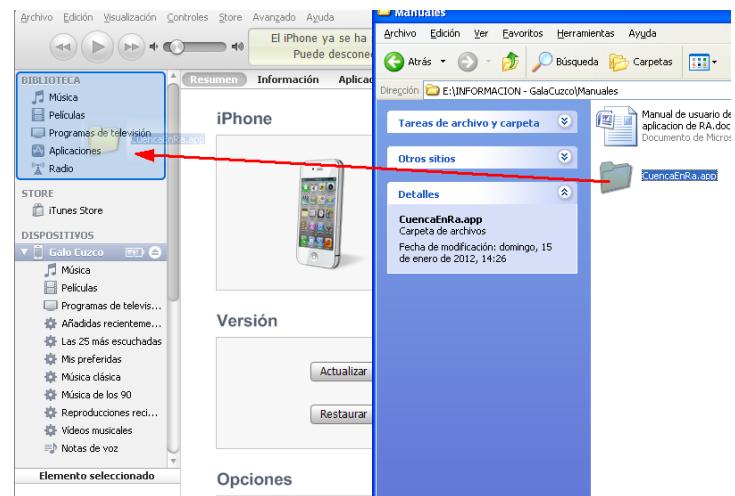


Figura A3.12 Arrastrar (Drag and Drop) de la aplicación a la biblioteca iTunes

- Por último sincronizar el dispositivo para instalar la aplicación, con la opción que se indica en la siguiente figura:



Figura A3.13 Sincronización del iPhone con la biblioteca iTunes

- ✓ Cuando inicia la sincronización se instalará la aplicación en el dispositivo.



Figura A3.14 Inicio de la Sincronización

- Cuando termine la sincronización ya tendremos la aplicación lista para su utilización, el ícono se mostrará en la pantalla del dispositivo, como se muestra en la siguiente figura.



Figura A3.15 Ícono de la aplicación de RA en la pantalla del iPhone

3 Funcionamiento de la aplicación

El funcionamiento de la aplicación es similar tanto en los dispositivos con Android e iOS, además es muy sencilla su utilización, solo se debe

seguir los siguientes pasos:

- Ejecutar la aplicación en el dispositivo, dando un clic en el icono de la aplicación.



Figura A3.16 Ejecutar la aplicación en los dispositivos con iOS y Android

- Una vez se haya ejecutado la aplicación, se encenderá la cámara y podremos visualizar nuestro entorno en la pantalla del dispositivo:



Figura A3.17 La aplicación en ejecución en iOS y Android

- Se debe enfocar completamente las marcas con la cámara, y podremos visualizar la información correspondiente:



Figura A3.18



Visualización de objetos

- Para poder visualizar de mejor manera los objetos en la pantalla se debe tener en cuenta la distancia especificada en el capítulo 4.

ANEXO 4

MANUAL DE USUARIO DE LA CREACIÓN DE OBJETOS 3D

MANUAL DE USUARIO DE LA CREACIÓN DE OBJETOS 3D

1 Creación de texturas para los objetos 3d

Creación de un nuevo proyecto en Photoshop.

Ejecutamos la aplicación Adobe Photoshop y creamos un nuevo proyecto, **Archivo>>Nuevo** (ver figura A4.1).

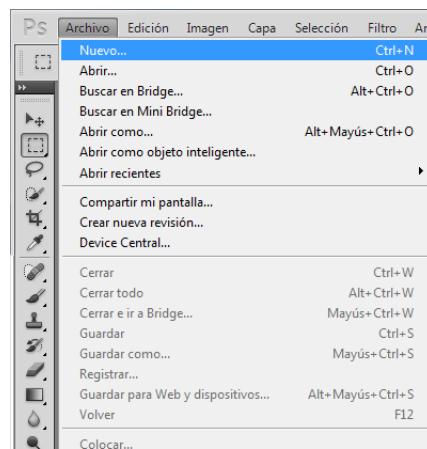


Figura A4.1 Nuevo proyecto en Photoshop

Luego de elegir nuevo se cargará una ventana en donde tenemos que ingresar el nombre del proyecto, el tamaño de la imagen y los demás campos los podemos dejar con los valores que vienen por defecto (ver figura A4.2).

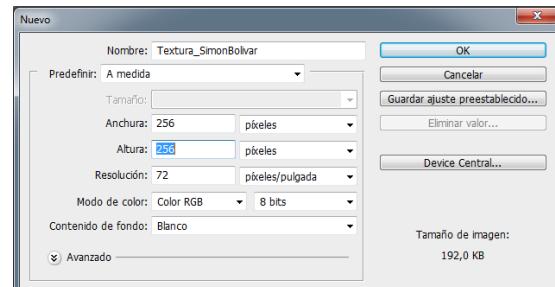


Figura A4.2 Nombre y tamaño del contenedor

Una vez concluido los pasos anteriores, se nos presenta una ventana que ya nos permite crear nuestra textura con colores, texto, transparencia, etc. (ver figura A4.3).

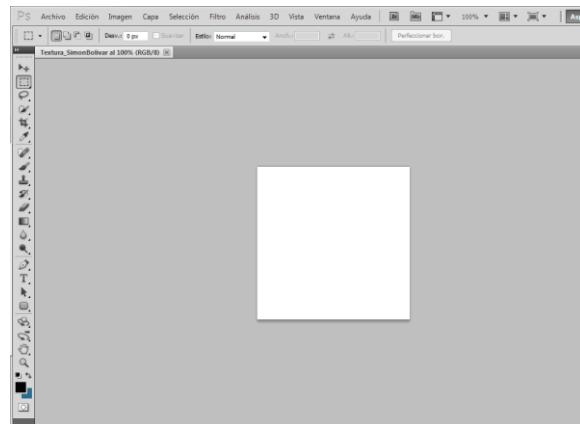


Figura A4.3 Proyecto creado por Photoshop

Colocar color y texto

Para poner color a la capa seleccionamos **Capa>>Nueva capa de relleno>>Color uniforme**, ahora colocamos el nombre a la nueva capa y luego clic en ok (ver figura A4.4).

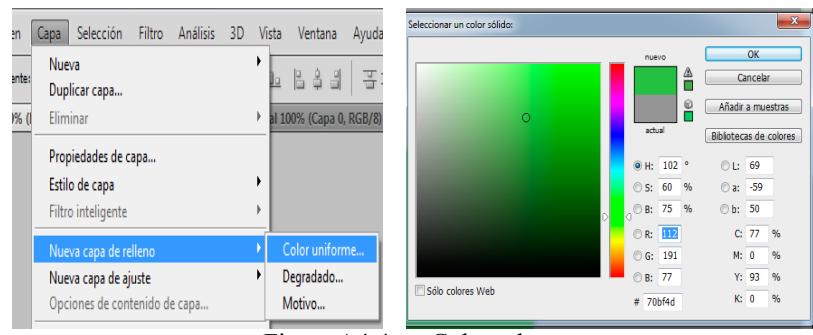


Figura A4.4 Color a la capa

Luego de seleccionar el color apropiado para la capa, hacemos clic en ok y obtendremos una nueva capa con el color seleccionado (ver figura A4.5).

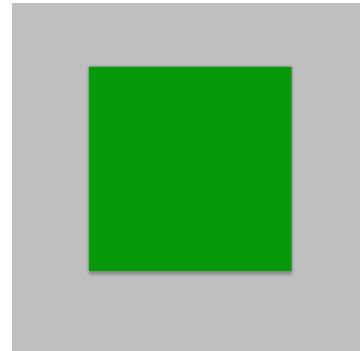


Figura A4.5 Capa con el color seleccionado

Ahora procedemos a colocar el texto a la capa, para lo cual escogemos **Herramienta texto horizontal** de la barra de herramientas (ver figura A4.6).



Figura A4.6 Herramienta texto horizontal

Una vez seleccionada la herramienta hacemos clic sobre la capa con color que fue creada anteriormente y procedemos a ingresar el texto deseado. Al texto ingresado le podemos poner un color, tipo de letra, darle un tamaño, etc., (ver figura A4.7).

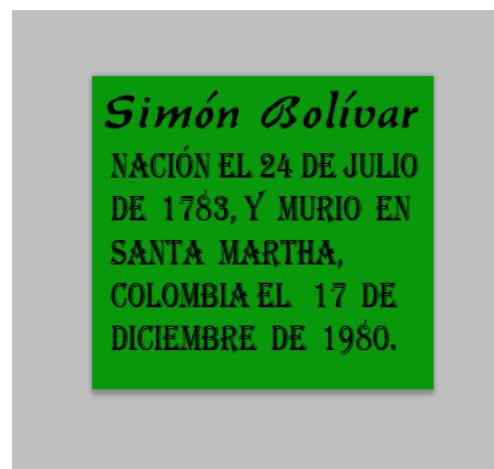


Figura A4.7 Capa con texto

Finalmente luego de haber creado todas las capas podemos jugar con su transparencia dependiendo la necesidad (ver figura A4.8).

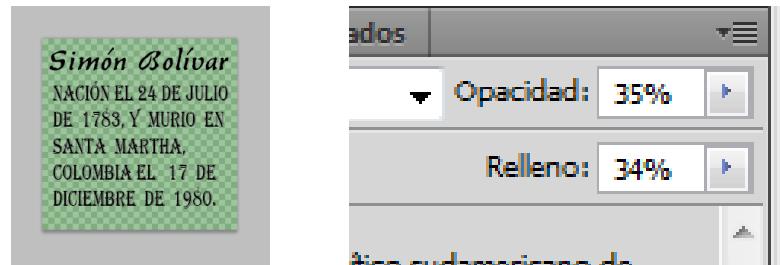


Figura A4.8 Transparencia de la textura

Generar la textura

Una vez que todo esté bien, procedemos a generar la textura para ello nos vamos a **Archivo>>Guardar como...**, en la ventana que se nos muestran elegimos el formato .png y hacemos clic en guardar (ver figura A4.9).

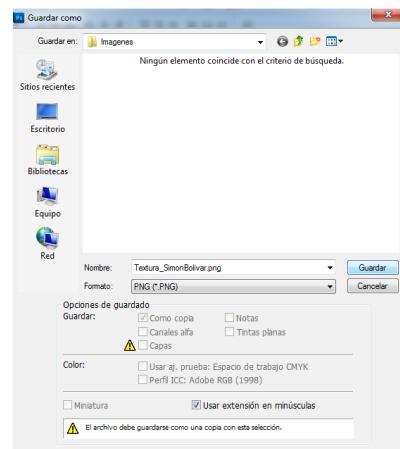


Figura A4.9 Guardado de la textura

2 Creación del objeto 3d

Creación de un nuevo proyecto en Blender.

Ejecutamos la aplicación Blender y creamos un nuevo proyecto (ver figura A4.10).

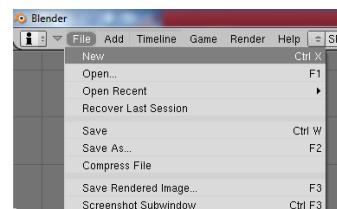


Figura A4.10 Nuevo proyecto en Blender

Ahora agregamos un objeto al proyecto en este caso será un plano, **Add>>Mesh>>Plane** (ver figura A4.11).

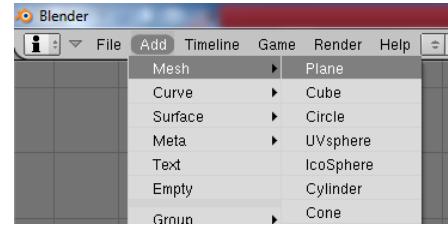


Figura A4.11 Agregar un objeto

Al objeto insertado lo podemos hacer que rote pulsando la tecla **r** y cambiar su tamaño pulsando la tecla **s** (ver figura A4.12).

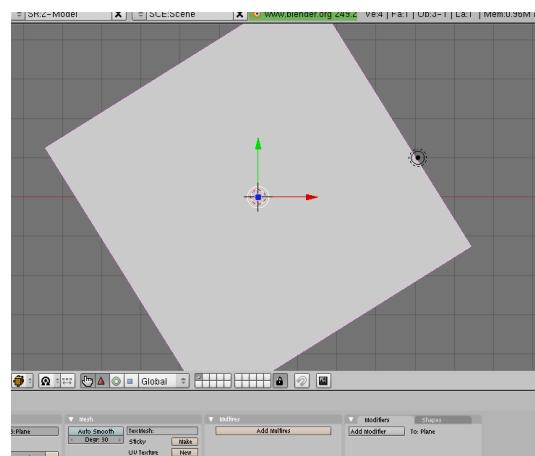


Figura A4.12 S cambiar de tamaño, R rotar

Ahora utilizando la herramienta **Split Area** dividimos el área de trabajo en dos partes, para cargar la textura al objeto en la parte derecha hacemos clic en **Displays Current Window Type>>UV/Image Iditor** (ver figura A4.13).

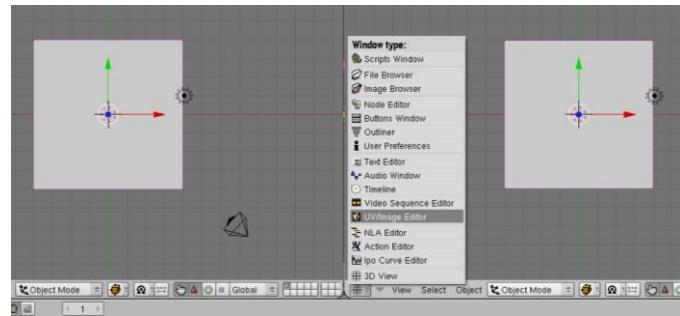


Figura A4.13 Dos áreas de trabajo

Una vez dado clic en **UV/Image Iditor** se nos abrirá una ventana en donde podemos cargar la textura al objeto 3d (ver figura A3.14).

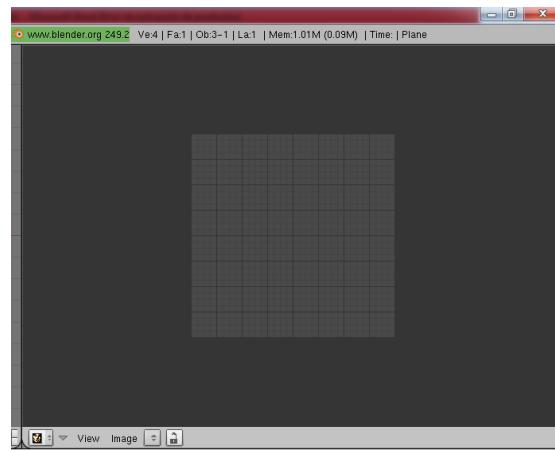


Figura A4.14 Editor de imágenes

Ahora cargaremos la textura al objeto para ello tenemos que ir a la dirección donde este guardado la imagen, **Image>> Open**, y buscamos la imagen (ver figura A4.14).

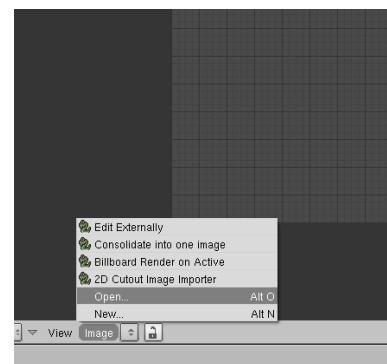


Figura A4.14 Selección de la imagen

Una vez seleccionada la textura podemos ver que se carga en una de las áreas de trabajo (ver figura A4.15)

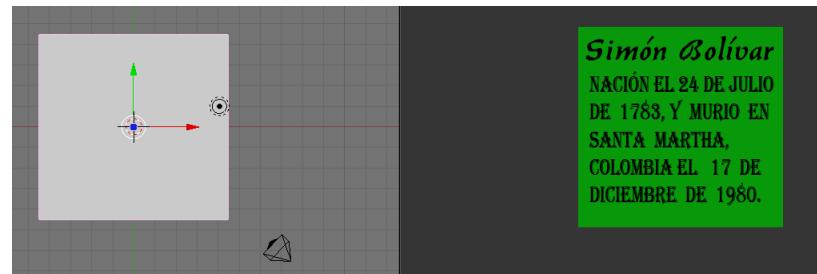


Figura A4.15 Objeto 3d y Textura

A continuación colocaremos la textura al objeto 3d, para ello en la parte izquierda de la área de trabajo hacemos clic en **Draw type>>Textured** (ver figura A4.16).

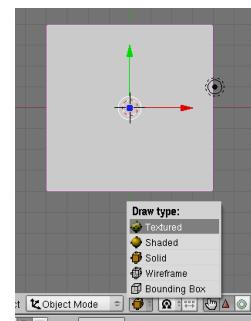


Figura A4.16 Texturado

Ahora hacemos clic en **Mode** y tendremos que ir seleccionando una a una sus opciones (ver figura A4.17).

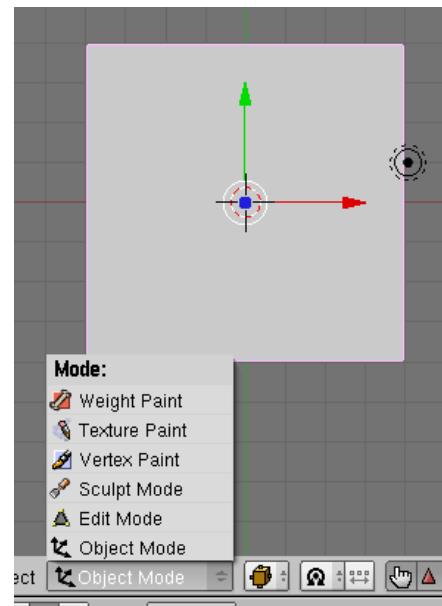


Figura A4.17 Selección de opciones

A continuación se presenta una imagen con las opciones de

Mode seleccionadas (ver figura A4.18).

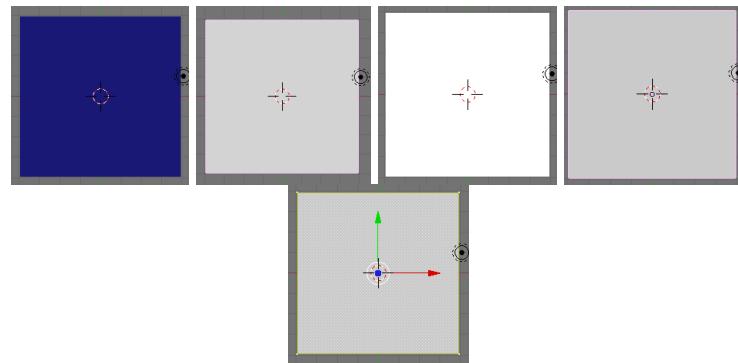


Figura A4.18 Opciones de Mode ya aplicadas al objeto 3d

Luego de aplicar la última opción del **Mode** podemos ver que el objeto 3d ya aparece con la textura (Si no se carga la imagen tendremos que volver a abrirlo) (ver figura A4.19).



Figura A4.19 Objeto 3d con la textura

Para agregar la textura al objeto 3d, pulsaremos f5 y luego haremos clic en **Add New>>TextFace** (ver figura A4.20).

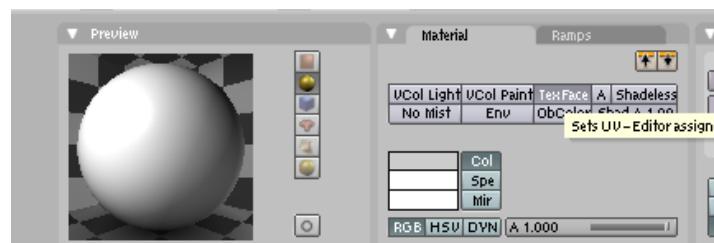


Figura A4.20 Agregar la textura

Finalmente, para verificar que la textura se agregó correctamente al objeto 3d, pulsamos f12 para que Blender ejecute la orden renderizar⁵² (ver figura A4.21).

⁵² Renderizar.- es un término usado en jerga informática para referirse al proceso de generar una imagen desde un modelo.

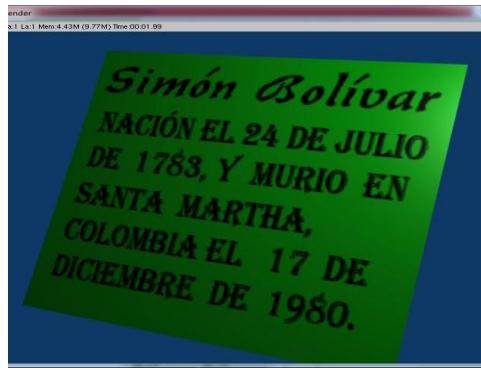


Figura A4.21 Objeto renderizado