

UNIVERSIDAD DEL BÍO-BÍO

Facultad de Ciencias Empresariales
Departamento de Sistemas de Información



Proyecto de título presentado en conformidad a los requisitos para obtener el título de
Ingeniero de Ejecución en Computación e Informática

**Recorrido virtual animado basado en realidad
aumentada sobre sistema android a través de las
dependencias y lugares de interés de la Universidad del
Bío-Bío, campus Concepción**

Alumno: Juan Riquelme López

Profesor Guía: Juan Parra Márquez

Concepción, 2016

Escuela de Ingeniería de Ejecución en Computación e Informática

AGRADECIMIENTOS

En primer lugar mi gratitud está dirigida a Dios, quién me ha otorgado la vida, salud y fortaleza necesaria en todo momento, especialmente en mis años de estudio.

Agradezco a mi familia y a mi novia quienes han dedicado gran esfuerzo en apoyarme, comprenderme, acompañarme y amarme incondicionalmente en altos y bajos, la compañía de las personas más importantes de mi vida fue un pilar fundamental para lograr mis metas y superar desafíos.

A las personas que me tendieron una mano para seguir con mis estudios en momentos de angustia, a los profesores de la universidad que me brindaron su ayuda y no escatimaban su voluntad en muchas ocasiones, doy las gracias a mi profesor guía, Juan Carlos Parra Márquez y a los profesores Marco Iturra Mella, Manuel Crisosto Muñoz, Karina Rojas Contreras y Pedro Campos Soto que marcaron mi paso por la universidad con valiosos consejos y conocimientos invaluable en mi formación.

Mi gratitud a todos ellos.

RESUMEN

Este proyecto se presenta en conformidad a los requisitos exigidos por la Universidad del Bío-Bío en proceso de titulación de la carrera de Ingeniería de Ejecución en Computación e Informática. El proyecto “Recorrido virtual animado basado en realidad aumentada sobre sistema Android a través de las dependencias y lugares de interés de la Universidad del Bío-Bío, Campus Concepción” tiene la intención de desarrollar una herramienta de información bajo el término de realidad aumentada utilizable en distintos lugares dentro del campus a modo de aplicación móvil para dispositivos Android.

En este caso, se utiliza la realidad aumentada para mostrar aspectos importantes de lugares reconocidos por el sistema de detección de imágenes mediante la orientación del dispositivo por parte del usuario a edificios específicos dentro del área y proyectando, sobre la realidad captada por la cámara, una visión aumentada virtualmente con contenido informativo.

Se pretende que el uso de esta aplicación acerque a las personas y la institución en un ambiente atractivo, cercano e interactivo al alcance de un gran número de personas que acuden al campus.

La aplicación mencionada utiliza características que actualmente se encuentran en casi la totalidad de los dispositivos que posee la población, considerando características nativas como la cámara trasera y el sensor GPS.

Su construcción se realiza bajo el motor de contenido interactivo Unity 3D, programado en lenguaje C# y utilizando como base el kit de desarrollo de software Vuforia.

ABSTRACT

This project is presented in accordance with the requirements demanded by the University of Bío-Bío in the titling process for Execution Engineering in Computing and Informatics. The project "Animated virtual tour based on augmented reality on Android system through the facilities and places of interest of the Bio-Bio University, Campus Concepción" intends to develop an information tool under the augmented reality term usable in different places within the campus as a mobile application for Android devices.

In this case, augmented reality is used to show important aspects of places recognized by the image detection system by orienting the device for the user to specific buildings within the area and projecting, on the reality captured by the camera, a virtually augmented vision with informative content.

It is intended that the use of this application bring people and the institution together, in an attractive, close and interactive environment available to a large number of people who come to the Campus.

The aforementioned application uses features that are currently found in almost all the devices that the population owns, considering native features such as the rear camera and GPS sensor.

It is built under the interactive content engine Unity 3D, programmed in C# language and based on the software development kit Vuforia.

ÍNDICE GENERAL

AGRADECIMIENTOS	2
RESUMEN	3
ABSTRACT	4
<u>INTRODUCCIÓN</u>	<u>13</u>
<u>CAPÍTULO 1 - DEFINICIÓN DE LA INSTITUCIÓN</u>	<u>16</u>
1.1 DESCRIPCIÓN DE LA INSTITUCIÓN	16
1.1.1 ANTECEDENTES GENERALES DE LA INSTITUCIÓN	16
1.1.2 ENTORNO.....	16
1.1.3 VISIÓN	16
1.1.4 MISIÓN	17
1.1.5 ORGANIGRAMA	18
1.2 Descripción del ÁREA DE ESTUDIO	19
1.2.1 ÁREA DE ESTUDIO.....	19
1.2.2 OBJETIVO DEL ÁREA	19
1.3 DESCRIPCIÓN DE LA PROBLEMÁTICA	20
1.3.1 INTRODUCCIÓN.....	20
1.3.2 DESARROLLO DE LA PROBLEMÁTICA	20
1.3.3 SOLUCIÓN.....	21
<u>CAPÍTULO 2 – ESTADO DEL ARTE.....</u>	<u>22</u>
2.1 INTRODUCCIÓN.....	22
2.2 DISPOSITIVOS MÓVILES INTELIGENTES	22
2.2.1 PLATAFORMAS.....	22
2.2.2 JUSTIFICACIÓN DE PLATAFORMA	23
2.3 SISTEMA OPERATIVO ANDROID	26
2.3.1 ANDROID	26

2.3.2	VERSIONES DEL SISTEMA	28
2.3.3	COMPONENTES Y HERRAMIENTAS FUNDAMENTALES	29
2.3.4	Estructura del proyecto Android en el motor Unity	32
2.4	REALIDAD AUMENTADA.....	35
2.4.1	INTRODUCCIÓN.....	35
2.4.2	DESARROLLO BÁSICO	36
2.5	MOTOR DE DESARROLLO UNITY	37
2.6	SDK VUFORIA.....	37
2.6.1	ARQUITECTURA DEL SDK	38
2.6.2	INSTALACIÓN EN UNITY	40
2.7	APLICACIONES SIMILARES	41
<u>CAPÍTULO 3 – DEFINICIÓN DEL PROYECTO.....</u>		42
3.1	JUSTIFICACIÓN.....	42
3.2	OBJETIVOS.....	43
3.2.1	OBJETIVOS GENERALES	43
3.2.2	OBJETIVOS ESPECÍFICOS	44
3.3	AMBIENTE DE INGENIERÍA DE SOFTWARE	45
3.3.1	METODOLOGÍA DE DESARROLLO	45
3.3.2	TÉCNICAS Y NOTACIONES	46
3.3.3	ESTÁNDARES DE DOCUMENTACIÓN	46
3.3.4	HERRAMIENTAS DE APOYO AL DESARROLLO	47
3.4	DEFINICIONES, SIGLAS Y ABREVIACIONES.....	47
<u>CAPÍTULO 4 – ESPECIFICACIÓN DE REQUERIMIENTOS.....</u>		51
4.1	ALCANCES.....	51
4.2	OBJETIVOS DE LA APLICACIÓN.....	55
4.2.1	OBJETIVOS GENERALES	55
4.2.2	OBJETIVOS ESPECÍFICOS	55

4.3	DESCRIPCIÓN GLOBAL	56
4.3.1	INTERFAZ DE USUARIO.....	56
4.3.2	INTERFAZ DE HARDWARE.....	56
4.3.3	INTERFAZ DE SOFTWARE.....	57
4.3.4	CONSIDERACIONES AMBIENTALES	57
4.4	REQUERIMIENTOS ESPECÍFICOS	58
4.4.1	REQUERIMIENTOS FUNCIONALES	58
4.4.2	ESPECIFICACIÓN DE REQUERIMIENTOS.....	59
4.4.3	INTERFACES EXTERNAS DE ENTRADA	63
4.4.4	INTERFACES EXTERNAS DE SALIDA	63
4.4.5	ATRIBUTOS DE LA APLICACIÓN.....	63
<u>CAPÍTULO 5 – ESTUDIO DE FACTIBILIDAD</u>		67
5.1	FACTIBILIDAD TÉCNICA	67
5.2	FACTIBILIDAD OPERATIVA.....	70
5.2.1	IMPACTOS POSITIVOS	70
5.2.2	IMPACTOS NEGATIVOS	71
5.3	FACTIBILIDAD ECONÓMICA.....	71
5.3.1	INVERSIÓN DE TANGIBLES	71
5.3.2	INVERSIÓN DE INTANGIBLES	71
5.3.3	INVERSIÓN DE RECURSOS HUMANOS.....	73
5.4	CONCLUSIÓN ESTUDIO DE FACTIBILIDAD	73
<u>CAPÍTULO 6 - ANÁLISIS</u>		74
6.1	DIAGRAMA DE CASOS DE USO	74
6.1.1	ACTORES	74
6.1.2	CASOS DE USO	75
6.1.3	DESCRIPCIÓN GENERAL	76
6.1.4	ESPECIFICACIÓN DE CASOS DE USO	76

<u>CAPÍTULO 7 - DISEÑO</u>	88
7.1 DISEÑO DE ARQUITECTURA FUNCIONAL	88
7.2 DISEÑO DE INTERFAZ Y NAVEGACIÓN	90
7.3 ICONOGRAFÍA	90
7.4 ESPECIFICACIÓN DE MÓDULOS.....	93
<u>CAPÍTULO 8 - PRUEBAS</u>	95
8.1 INTRODUCCIÓN	95
8.2 ELEMENTOS DE PRUEBAS	95
8.3 ESPECIFICACIÓN DE PRUEBAS	97
8.4 RESPONSABLES	100
8.5 CALENDARIO DE PRUEBAS	101
8.6 DETALLE DE PRUEBAS.....	101
8.7 CONCLUSIÓN DE PRUEBAS	101
<u>CAPÍTULO 9 – PLAN DE IMPLEMENTACIÓN</u>	102
<u>CAPÍTULO 10 – RESUMEN DE ESFUERZO REQUERIDO</u>	104
<u>CAPÍTULO 11 – CONCLUSIONES</u>	105
REFERENCIAS	107
ANEXO 1 – PLANIFICACIÓN INICIAL DEL PROYECTO.....	109
ANEXO 2 – POTENCIALES USUARIOS	119
ANEXO 3 – ESPECIFICACIÓN DE PRUEBAS.....	128
ANEXO 4 – CÓDIGO RELEVANTE	134

ÍNDICE DE TABLAS

Tabla 1: Versiones de S.O. Android _____	28
Tabla 2: Alcances de información en la aplicación _____	54
Tabla 3: Requerimientos funcionales _____	58
Tabla 4: ERF.01 – Especificación de requerimientos _____	59
Tabla 5: ERF.02 – Especificación de requerimientos _____	59
Tabla 6: ERF.03 – Especificación de requerimientos _____	60
Tabla 7: ERF.04 – Especificación de requerimientos _____	60
Tabla 8: ERF.05 – Especificación de requerimientos _____	61
Tabla 9: ERF.06 – Especificación de requerimientos _____	62
Tabla 10: Interfaces externas de salida _____	63
Tabla 11: Software – Factibilidad técnica _____	67
Tabla 12: Plugins – Factibilidad técnica _____	68
Tabla 13: Dispositivos – Factibilidad técnica _____	69
Tabla 14: Distribución propia – Inversión intangible _____	72
Tabla 15: Distribución Google Play – Inversión intangible _____	72
Tabla 16: Descripción de actores – Casos de uso _____	74
Tabla 17: M01. Mostrar mapa – Especificación de módulos _____	93
Tabla 18: M02. Consultar ubicación – Especificación de módulos _____	93
Tabla 19: M03. Consultar estado de sensores – Especificación de módulos _____	93
Tabla 20: M04. Desplegar información en pantalla – Especificación de módulos _____	94
Tabla 21: M05. Ocultar ítems – Especificación de módulos _____	94
Tabla 22: M06. Cambiar vista Autoridades – Especificación de módulos _____	94
Tabla 23: Especificación de pruebas _____	97
Tabla 24: Actividades de pruebas _____	98
Tabla 25: Calendario de pruebas _____	101
Tabla 26: Horas Esfuerzo requerido _____	104
Tabla 27: Clasificación de actores - UAW _____	111

Tabla 28: Clasificación de casos de uso - UUCW	111
Tabla 29: Peso de los factores técnicos - TCF	113
Tabla 30: Ponderación de factores - TCF	113
Tabla 31: Peso y ponderación Factores relevantes - TCF	114
Tabla 32: Peso y ponderación Factores ambientales - EF	115
Tabla 33: Cálculo LOE - Esfuerzo	116
Tabla 34: Scripts relevantes	117
Tabla 35: E.P1 – Especificación de pruebas	128
Tabla 36: E.P2 – Especificación de pruebas	129
Tabla 37: E.P3 – Especificación de pruebas	131
Tabla 38: E.P4 – Especificación de pruebas	131
Tabla 39: E.P5 – Especificación de pruebas	132
Tabla 40: E.P6 – Especificación de pruebas	132
Tabla 41: E.P7 – Especificación de pruebas	133

ÍNDICE DE FIGURAS

Figura 1: Organigrama Universidad del Bío-Bío _____	18
Figura 2: Mercado global de dispositivos móviles entre 2009 y 2014 _____	24
Figura 3: Mercado latinoamericano de dispositivos móviles por plataforma _____	25
Figura 4: Creación de modelo 3D - Blender _____	30
Figura 5: UV mapping, modelado 3D - Blender _____	30
Figura 6: SDK Vuforia 6.1 _____	31
Figura 7: SDK Android – Android Studio _____	31
Figura 8: Importar SDK – Unity _____	32
Figura 9: Configuración Player – Unity _____	33
Figura 10: Configuración Icon – Unity _____	33
Figura 11: Otras configuraciones – Unity _____	34
Figura 12: Realidad mixta – Milgram Virtuality Continuum _____	35
Figura 13: Diagrama de flujo de datos SDK Vuforia en una aplicación _____	39
Figura 14: Importar paquetes (1) – Unity _____	40
Figura 15: Importar paquetes (2) – Unity _____	40
Figura 16: Augment 3D – Aplicaciones similares _____	41
Figura 17: ARcuerpohumano – Aplicaciones similares _____	42
Figura 18: RAInfinitem – Aplicaciones similares _____	42
Figura 19: Diagrama de Casos de uso _____	75
Figura 20: CU01 Flujo eventos básico _____	77
Figura 21: CU02 Flujo eventos básico _____	78
Figura 22: CU02 Flujo eventos alternativo _____	79
Figura 23: CU03 Flujo eventos básico _____	81
Figura 24: CU04 Flujo eventos básico _____	83
Figura 25: CU05 Flujo eventos básico _____	85
Figura 26: CU06 Flujo eventos básico _____	86
Figura 27: Diagrama de descomposición funcional Vista general–Arquitectura funcional _____	88

Figura 28: Diagrama de eventos – Arquitectura funcional	89
Figura 29: Logo de la aplicación	90
Figura 30: Imagen de la aplicación	90
Figura 31: Figuras de la aplicación	90
Figura 32: Esquema de navegación 1	91
Figura 33: Esquema de navegación 2	91
Figura 34: Esquema de navegación 3	92
Figura 35: Planificación del proyecto	109
Figura 36: Carta Gantt	110
Figura 37: Encuesta pregunta 1 – Análisis potenciales usuarios	120
Figura 38: Encuesta pregunta 2 y 3 (a) – Análisis potenciales usuarios	121
Figura 39: Encuesta pregunta 2 y 3 (b) – Análisis potenciales usuarios	122
Figura 40: Encuesta pregunta 4 – Análisis potenciales usuarios	123
Figura 41: Encuesta pregunta 5 – Análisis potenciales usuarios	124
Figura 42: Encuesta pregunta 6 – Análisis potenciales usuarios	125
Figura 43: Encuesta pregunta 7 y 8 – Análisis potenciales usuarios	126
Figura 44: Encuesta pregunta 9 – Análisis potenciales usuarios	127
Figura 45: ConexionInt.cs – Código relevante	134
Figura 46: Gps1.cs – Código relevante	135
Figura 47: DatabaseLoadBehaviour.cs – Código relevante	140
Figura 48: DefaultTrackerEventHandler.cs – Código relevante	141

INTRODUCCIÓN

Actualmente vivimos un presente colmado de avances y ambiciosa evolución tecnológica que nos lleva, a veces sin darnos cuenta, a adherirnos al deseo de la gran mayoría de las personas queriendo hacer nuestra vida más fácil y cómoda recurriendo a la tecnología que tenemos al alcance.

Hablar de realidad aumentada no sólo es hablar de sustituir la realidad percibida como lo hace la realidad virtual, sino que se trata de nutrir los elementos físicos tangibles con elementos virtuales para lograr una realidad mixta que permite experimentar un escenario que nace a partir del escenario real pero que en conjunto enriquecen lo que percibimos de ellos, ya sea para divertirnos, experimentar o informarnos.

Tomando este concepto, el presente proyecto busca utilizar las herramientas tecnológicas necesarias para crear una aplicación de realidad aumentada que nos informe sobre el entorno dentro del campus de una forma accesible, fácil y atractiva.

El documento se encuentra dividido en 11 capítulos más anexos en los cuales se describe lo realizado para cumplir los objetivos planteados, considerando herramientas utilizadas, investigaciones relativas a los temas implicados, lenguajes de programación, entre otros aspectos de relevancia.

A continuación se presenta una breve descripción de los capítulos que conforman este proyecto de título:

- **Capítulo 1. Definición de la institución:** Se presenta una descripción general de la institución para la cual se destina la aplicación y también se muestra la definición de la problemática a resolver.

- **Capítulo 2. Estado del arte:** Este capítulo introduce al tema de la realidad aumentada y muestra estadísticas sobre herramientas a utilizar, especifica decisiones tomadas en pos de un mejor desarrollo en cuestión de software y plataformas, además profundiza levemente en las herramientas base del proyecto y sus interacciones entre sí.
- **Capítulo 3. Definición del proyecto:** Se definen acá los parámetros del ambiente de ingeniería de software correspondientes, metodología a utilizar y notaciones importantes.
- **Capítulo 4. Especificación de requerimientos:** Se definen los alcances que tiene la aplicación, objetivos generales y específicos justo a sus descripciones.
- **Capítulo 5. Estudio de factibilidad:** Se desarrolla el estudio de factibilidades técnicas, operativas y económicas del proyecto.
- **Capítulo 6. Análisis:** Muestra el diagrama de casos de uso, describe actores correspondientes y desarrolla el análisis de cada caso en particular.
- **Capítulo 7. Diseño:** Muestra la arquitectura funcional de la aplicación y detalles sobre aspectos visuales y navegación.
- **Capítulo 8. Pruebas:** En este capítulo se mencionan las pruebas que se realizan a la aplicación en busca de fallos en el funcionamiento, también se detallan plazos y calendarización.
- **Capítulo 9. Implementación y puesta en marcha:** Se explican los procesos posteriores al desarrollo de la aplicación relacionados a adoptar una estrategia para llevar el software a los dispositivos móviles de los usuarios finales.

- **Capítulo 10. Resumen de esfuerzo requerido:** Se realizan análisis de tiempos requeridos por el desarrollador durante el proyecto.
- **Capítulo 11. Conclusiones:** Se mencionan las conclusiones que deja el proyecto en vistas al cumplimiento efectivo de los objetivos y requerimientos alcanzados.
- **Referencias:** Se definen las referencias de apoyo utilizadas para el proyecto.
- **Anexos:** Se añaden además 4 anexos que pueden resultar de interés para el lector, estos corresponden a: Planificación inicial del proyecto, Potenciales usuarios, Especificación de pruebas y Scripts relevantes.

CAPÍTULO 1 – DEFINICIÓN DE LA INSTITUCIÓN

1.1 Descripción de la institución

1.1.1 Antecedentes generales de la institución

- Nombre: Universidad del Bío-Bío.
- Dirección: Calle Collao 1202, Concepción, Octava región, Chile (Campus Concepción).
- Rubro: Educación.
- Productos-Servicios ofrecidos: Carreras de pregrado y postgrado.

1.1.2 Entorno

- Competencia directa: Instituciones universitarias de educación superior.

1.1.3 Visión

Ser reconocida a nivel nacional e internacional como una Universidad pública, responsable socialmente y regional que, comprometida con su rol estatal, desde la Región del Biobío, forma personas integrales de excelencia y aporta a través de su quehacer al desarrollo sustentable de la región y el país.

1.1.4 Misión

La Universidad del Bío-Bío, a partir de su naturaleza pública, responsable socialmente y estatal, tiene por misión, desde la Región del Biobío, aportar a la sociedad con la formación de personas integrales, a través de una Educación Superior de excelencia. Comprometida con los desafíos de la región y del país, contribuye a la movilidad e integración social por medio de; la generación y transferencia de conocimiento avanzado, mediante la docencia de pregrado y postgrado de calidad, la investigación fundamental, aplicada y de desarrollo, la vinculación bidireccional con el medio, la formación continua y la extensión. Asimismo, impulsa el emprendimiento y la innovación, el fortalecimiento de la internacionalización y el desarrollo sustentable de sus actividades, basada en una cultura participativa centrada en el respeto a las personas.

1.1.5 Organigrama

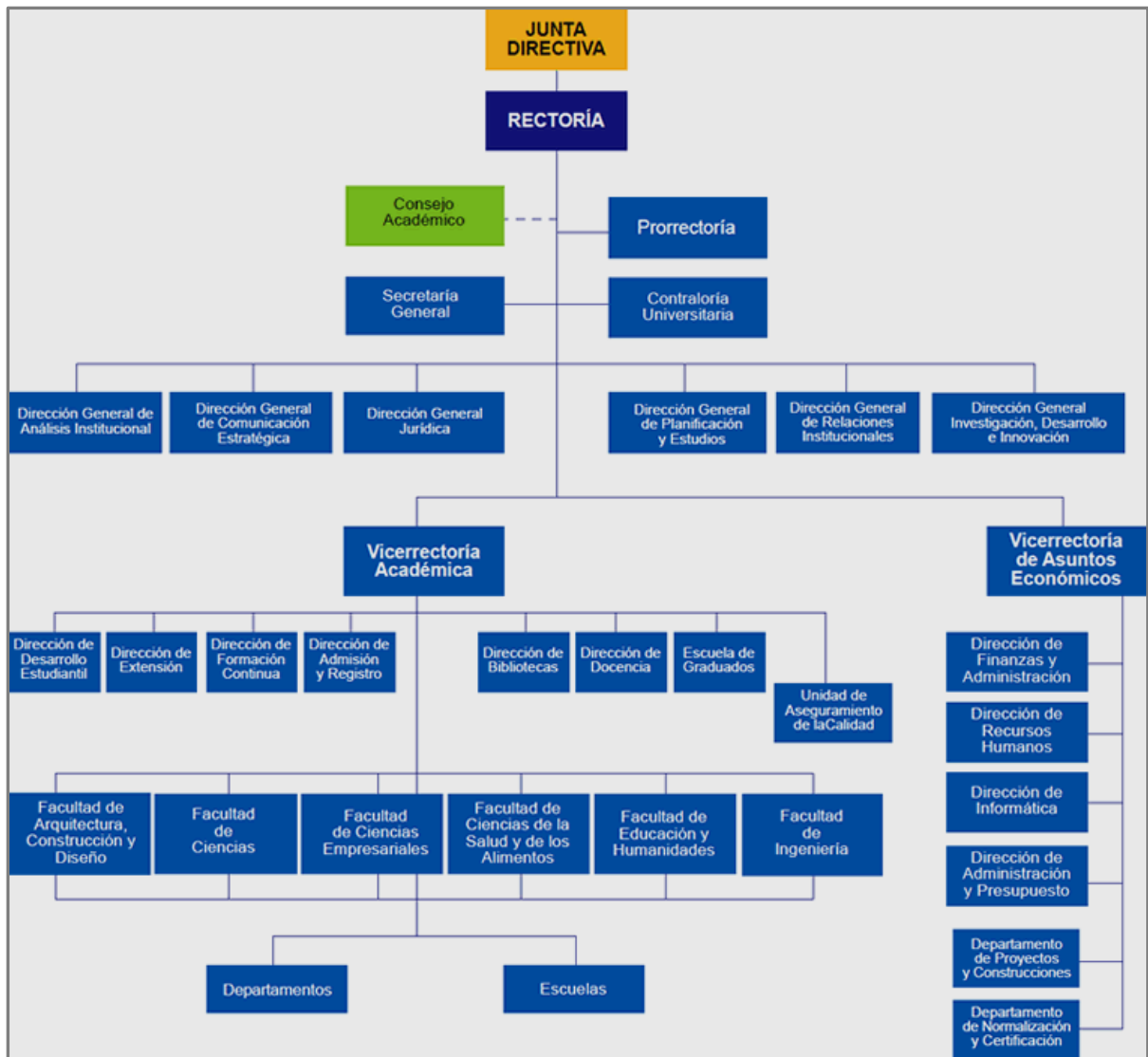


Figura 1: Organigrama Universidad del Bío-Bío.
Fuente: ubiobio.cl

1.2 Descripción del Área de estudio

1.2.1 Área de estudio, funciones y cargos relacionados al proyecto

- Área: La persona en el rol de cliente para este proyecto pertenece a la Dirección General de Comunicación Estratégica de la Universidad del Bío-Bío.
- Persona asesora en forma de cliente: Pía Lindemann Varoli, Directora de la Dirección General de Comunicación Estratégica, Universidad del Bío-Bío.
- Unidad académica participante: Facultad de Ciencias Empresariales, Universidad del Bío-Bío.
- Académico Guía: Profesor Juan Carlos Parra Márquez.

1.2.2 Objetivo del área

La DIGCE está al servicio de los objetivos estratégicos de la Universidad, proyectando la identidad de la Universidad con una imagen que produzca cercanía y confianza en su entorno relevante, y adhesión en los públicos objetivos.

En consideración a este proyecto la meta es utilizar las herramientas disponibles de parte de dirección en conjunto al desarrollo de los alumnos de informática para desarrollar productos innovadores apuntando hacia el progreso positivo de la imagen de la institución en la población.

1.3 Descripción de la Problemática

1.3.1 Introducción

El modo de vivir del ser humano parece ir cambiando a pasos agigantados a medida que progresa la tecnología, en nuestra época se ha visto reflejada en la medicina, equipos electrónicos, la ciencia, la comunicación, entre muchos otros ámbitos que en conjunto pretenden darnos una mejor calidad de vida. La tecnología avanza y nos encanta con sus variadas utilidades, así también la realidad virtual y posterior desarrollo de la realidad aumentada se hacen cabida en nuestros equipos y en nuestras vidas, permitiéndonos experimentar nuevas vivencias que hace unos años se encontraban fuera de nuestros pensamientos.

Hoy en día existen muchas herramientas que nos permiten desarrollar realidad aumentada y herramientas que nos ayudan a implementarla en nuestros equipos personales. Entonces, ¿Por qué no tomar estas herramientas y mejorar la interacción con el entorno?

1.3.2 Desarrollo de la problemática

Cada año llega una gran cantidad de alumnos nuevos para estudiar en nuestra universidad, cada día personas vienen a visitar el campus, trabajar o simplemente pasear. Es fácil distinguir a las personas que aún no están familiarizadas con el lugar, las preguntas por indicaciones se hacen pan de cada día dentro del campus. Claramente el uso de la tecnología para mejorar nuestra calidad de vida se puede aplicar a esta situación, idea de donde se origina la problemática.

Es necesario llevar información a estas personas sobre su entorno, qué función cumplen los distintos edificios que les rodean dentro del campus, su personal, sus horarios, de una manera atractiva y fácil de utilizar. Actualmente gran parte de la población dispone de equipos móviles los cuales podrían servir perfectamente como un medio para obtener dicha información.

1.3.3 Solución

La realidad aumentada puede aplicarse en distintos campos, uno de los principales usos consiste en identificar y localizar todo aquello que nos rodea y obtener información. Con esto en mente, la solución comprende en el desarrollo de una aplicación móvil que utilizará el concepto de realidad aumentada para obtener detalles del entorno, específicamente de algunos edificios dentro del campus y desplegar, a partir de su detección, información sobre estos que, para ser obtenidos de forma normal, se debe consultar los diferentes portales web de la institución, proceso que conlleva un tiempo y dificultad mayor que sólo consultar una sola aplicación que reúna todos estos datos y los despliegue en los dispositivos móviles de cada persona.

CAPÍTULO 2 – ESTADO DEL ARTE

2.1 Introducción

En este capítulo se presentan los estudios realizados considerados relevantes para el proyecto y lograr los objetivos propuestos. Existen aristas de gran importancia como lo son la elección del sistema operativo al que se orientará la aplicación, las condiciones de hardware que establecerán los requerimientos necesarios para utilizarla, las distintas encrucijadas surgidas a lo largo del desarrollo y las decisiones tomadas en pos de un mejor resultado. Se muestra además, detalles a considerar sobre la realidad aumentada, las herramientas utilizadas y aplicaciones similares existentes.

2.2 Dispositivos móviles inteligentes

2.2.1 Plataformas

Los sistemas operativos en dispositivos móviles más utilizados en el último tiempo son los siguientes:

- **Android:** Su desarrollo es liderado por Google y la alianza comercial Open Handset Alliance, basado en un núcleo Linux de código libre, gratuito y multiplataforma diseñado principalmente para dispositivos móviles con pantalla táctil como teléfonos inteligentes, tablets, relojes inteligentes, entre otros. Su versión más reciente corresponde a Android Nougat 7.1. Posee una tienda virtual de descarga gratuita y compra de aplicaciones llamada “Google Play”.

- iOS: Sistema desarrollado por Apple Inc. Originalmente destinado para iPhone como iPhone OS, pero que se ha usado en dispositivos como el iPod touch y el iPad. No permite ser instalado en hardware de terceros. Su versión más reciente corresponde a iOS 10.1. Posee una tienda virtual de descarga gratuita y compra de aplicaciones llamada “App Store”.
- Windows Phone: Desarrollado por Microsoft, sucesor de su antiguo sistema operativo Windows Mobile, aunque la compañía ha anunciado su baja para desarrollar un sistema operativo más unificado y versátil denominado Windows 10 Mobile, disponible para teléfonos inteligentes, tablets y computadoras. Su versión más reciente corresponde a Windows Phone 8.1.2. Posee una tienda de descarga gratuita y compra de aplicaciones llamada “Microsoft Store”.
- Otros: Existen otros sistemas operativos para dispositivos móviles aparte de los mencionados anteriormente, aunque su uso es mucho más bajo, debido a funcionalidad, año, familiaridad, distribución en el mercado, entre otros. Por esto se nombrarán algunos como: Symbian, BlackBerry, Firefox OS y Ubuntu Touch.

2.2.2 Justificación de plataforma

Resulta evidente que uno de los factores primordiales en la elección del sistema operativo para el cual desarrollar corresponde a la preferencia que tiene la población sobre estos, por lo que nos limitaremos a analizar los siguientes puntos sobre los sistemas Android, iOS y Windows Phone.

- Mercado mayor y crecimiento.

Salta a la vista que Android predomina el mercado en nuestro país, es muy común ver dispositivos móviles con este sistema donde quiera que vayamos en nuestra región. Esto se genera porque Android cubre una mayor gama de dispositivos que la competencia, de esta forma es que ha tenido un mayor crecimiento y de manera mucho más rápida sobre Windows Phone y iOS.

En la siguiente tabla podemos analizar información sobre la distribución en el mercado global de estos sistemas operativos.

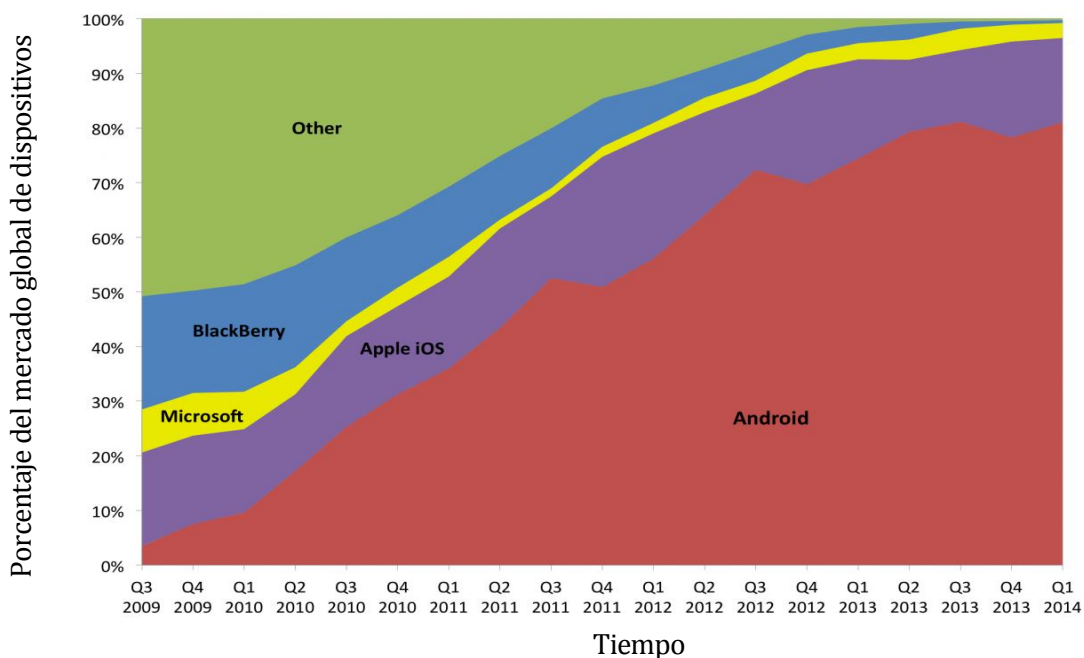


Figura 2: Mercado global de dispositivos móviles entre 2009 y 2014.
Fuente: IDC, Strategy Analytics.

Se puede apreciar la predominancia del mercado por parte de Android entre el tercer trimestre de 2009 y el primer trimestre de 2014. Vemos que sucede algo similar si sólo analizamos Latinoamérica.

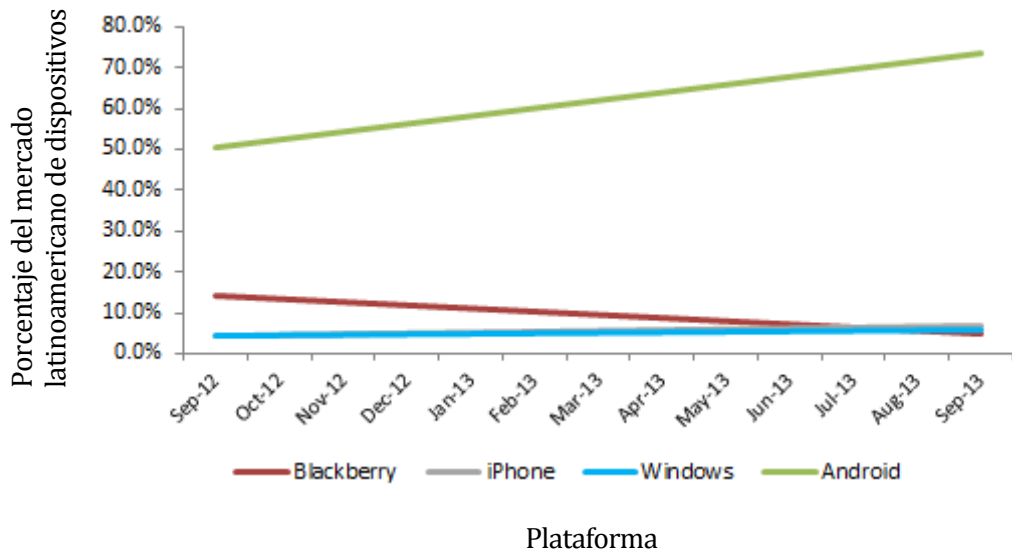


Figura 3: Mercado latinoamericano de dispositivos móviles por plataforma.
Fuente: IDC, Strategy Analytics.

Por tanto, debido al mercado abarcado y su notable crecimiento, desarrollar sobre plataforma Android tiene un mayor número de potenciales usuarios y de esta forma posibilitar a un mayor porcentaje de personas a usar nuestra aplicación.

- Apple Store approval Process

El proceso de aceptación en Apple Store podría suponer alguna dificultad al publicar una aplicación en la tienda de Apple, aunque posee una serie de reglas accesibles las aplicaciones obligatoriamente pasan por este proceso de validación durante varios días antes de estar disponible.

- Disponibilidad de herramientas de desarrollo y material de ayuda

Si bien no es excluyente, la disponibilidad de la herramienta de desarrollo nativo para iPhones xCode sólo está presente para equipos con sistema operativo OS X, eso quiere decir que si no se trabaja en un equipo Mac se tendrán que evaluar alternativas.

Por otro lado, los recursos disponibles para desarrollar sobre el sistema Android son muy variados, desde códigos específicos, tutoriales, SDK's, IDEs. Por este motivo Android supone un punto a favor.

Al momento de hacer una elección el primer punto tiene mucho peso a la hora de crear una aplicación que pretende llegar al mayor número de personas posibles, se podría desarrollar de igual forma en cualquiera de los otros sistemas mencionados, pero más de la mitad de los usuarios a los que está destinada no podrían utilizarla por no cumplir con los requerimientos mínimos, es justo lo que se pretende evitar.

2.3 Sistema operativo Android

2.3.1 Android

Android es un sistema operativo basado en Linux, diseñado para dispositivos móviles de pantalla táctil como teléfonos inteligentes (smartphones), tabletas (tablets), relojes inteligentes (smartwatches), aunque en la actualidad ya se encuentra presente hasta en televisores y automóviles.

Este sistema posee un modelo de desarrollo de código abierto, fue lanzado inicialmente en septiembre de 2008 por Android Inc. Y su última versión estable corresponde a la versión 7.1 de agosto de 2016 ya liderado por Google y el consorcio entre 48 compañías de hardware, software y telecomunicaciones Open Handset Alliance. Se encuentra programado principalmente en lenguajes C, C++ y Java.

Android permite controlar los dispositivos mediante librerías desarrolladas o adaptadas por Google mediante lenguaje de programación Java.

Una de las principales características que hacen lo hacen tan atractivo es su libertad. Es decir, no requiere gastos para instalar ni programar reduciendo los costes de lanzar aplicaciones nuevas e incluso equipos. Cabe señalar que siendo de código abierto permite detectar fallos rápidamente y hasta mejorar los sistemas operativos para los distintos terminales.

2.3.2 Versiones del sistema

Desde el lanzamiento de Android beta en 2007 se han creado un gran número de versiones para este sistema. Desde Abril de 2009 estas versiones se han lanzado bajo nombres clave siguiendo un orden alfabético: Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop, Marshmallow y finalmente la versión más reciente de agosto de 2016 Nougat.

Nombre	Número de versión	Fecha de lanzamiento	Nivel de API
Apple Pie	1.0	23 de septiembre de 2008	1
Banana Bread	1.1	9 de febrero de 2009	2
Cupcake	1.5	27 de abril de 2009	3
Donut	1.6	15 de septiembre de 2009	4
Éclair	2.0 – 2.1	26 de octubre de 2009	5 – 7
Froyo	2.2 – 2.2.3	20 de mayo de 2010	8
Gingerbread	2.3 – 2.3.7	6 de diciembre de 2010	9 – 10
Honeycomb	3.0 – 3.2.6	22 de febrero de 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	18 de octubre de 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	9 de julio de 2012	16 – 18
KitKat	4.4 – 4.4.4	31 de octubre de 2013	19 – 20
Lollipop	5.0 – 5.1.1	12 de noviembre de 2014	21 – 22
Marshmallow	6.0 – 6.0.1	5 de octubre de 2015	23
Nougat	7.0 – 7.1	22 de agosto de 2016	24 - 25

*Tabla 1: Versiones de S.O. Android.
Fuente: Elaboración propia.*

API corresponde a la abreviatura en inglés de Application Programming Interface. Puede verse en detalle en el punto 3.4 de este documento.

2.3.3 Componentes y herramientas fundamentales

En este punto se describen los componentes y herramientas a utilizar en la creación de una aplicación de realidad aumentada para Android utilizando Vuforia en el motor de desarrollo Unity.

- Entorno

Unity por sí solo no será suficiente para crear la base de la aplicación en cuestión, es primordial el uso de dos SDK (Kit de desarrollo de softwares), SDK Vuforia y SDK Android, ambos para ser utilizados en el motor mencionado. Para crear modelos 3D animados se ha optado por Blender, programa de modelado, renderizado, animación y creación de gráficos que compatibiliza perfectamente con Unity. Para este caso se ocupan todos los elementos su versión para Windows.

- **Blender** es un programa multiplataforma, libre y gratuito, ideal para el modelado 3D, con él se pueden crear modelos fácilmente exportables hacia Unity, lo especial de Blender es que posee una gran variedad de primitivas geométricas, curvas y mallas poligonales; características interactivas como detección de colisiones, recreaciones dinámicas y lógica.

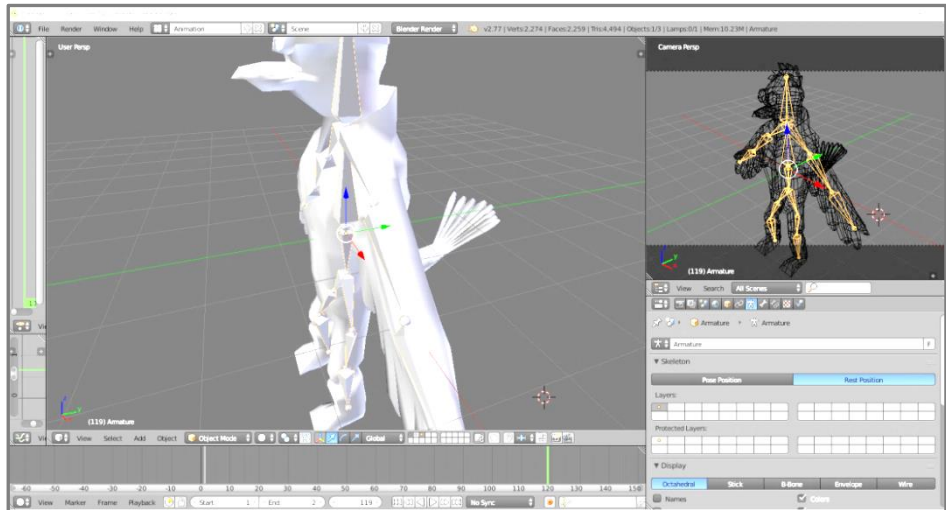


Figura 4: Creación de modelo 3D – Blender.
Fuente: Captura propia en Blender.

- Se necesitan 3 elementos principales para conformar el modelo antes de realizar las animaciones.
 - Modelo 3D.
 - Estructura de huesos y articulaciones que serán utilizadas por blender para darle movimiento al modelo.
 - Texturas: Para darle textura al modelo Blender cuenta con un sistema UV mapping la cual asigna coordenadas por cada vértice creado en el modelo.



Figura 5: UV mapping modelado 3D – Blender.
Fuente: Captura propia en Blender.

- Por su parte, Unity nos permitirá importar y trabajar con los SDK mencionados, es necesario descargarlos desde el motor.

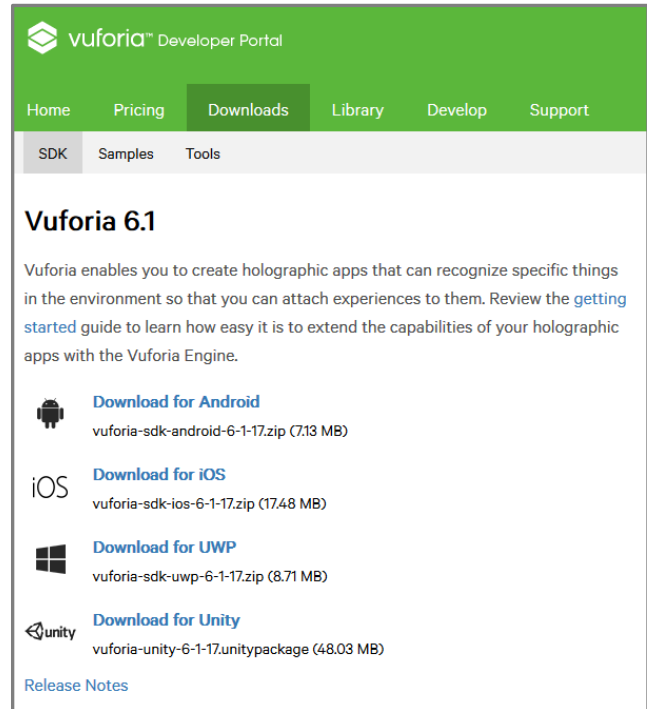


Figura 6: SDK Vuforia 6.1.
Fuente: developer.vuforia.com

Android Studio viene con todas las herramientas que necesitas para la creación de apps de Android.

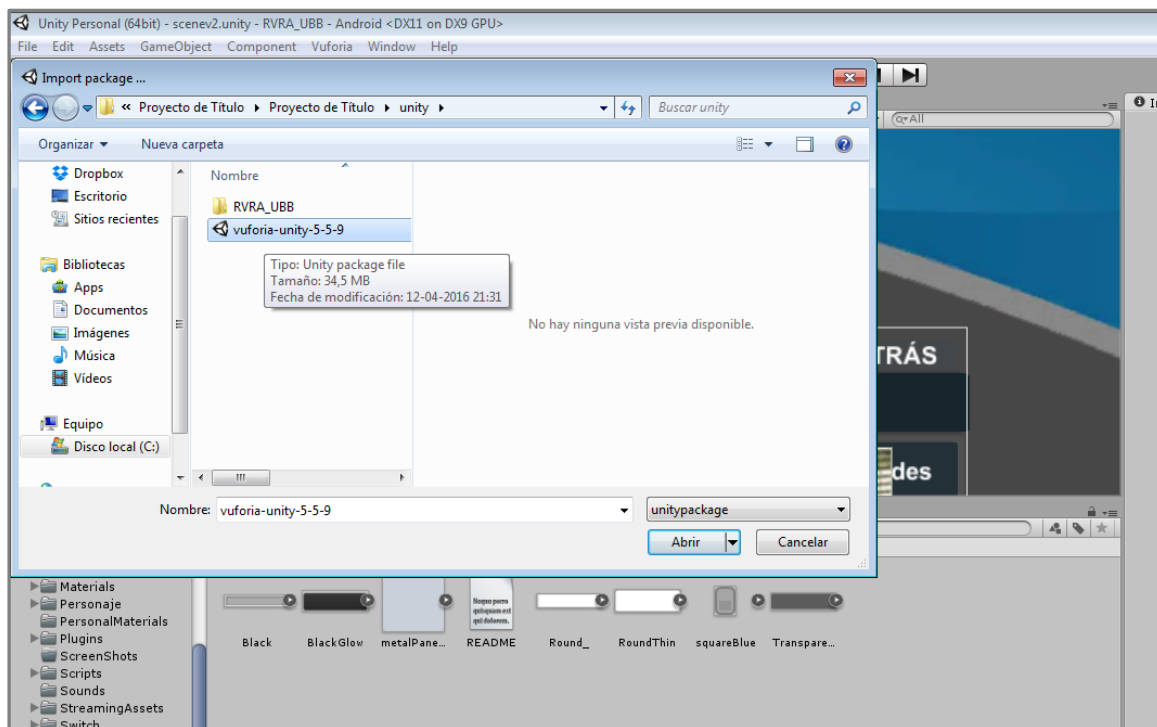
CÓMO DESCARGAR ANDROID STUDIO
2.2.2.0 FOR WINDOWS (1641 MB)

VERSIÓN: 2.2.2.0
FECHA DE LANZAMIENTO: OCTOBER 19, 2016

Selecciona una plataforma diferente

Plataforma	Paquete de Android Studio	Tamaño	Suma de comprobación de SHA-1
Windows	android-studio-bundle-145.3360264-windows.exe Incluye Android SDK (recomendado)	1641 MB (1721650280 bytes)	d879e4bf8cd2530dfa6cc7e176d72bb8dfd37b41
	android-studio-ide-145.3360264-windows.exe Sin Android SDK	423 MB (444308960 bytes)	54c65afe143e87ef40decc720854a9c1f30417d3
	android-studio-ide-145.3360264-windows.zip Sin Android SDK, sin instalador	445 MB (467098338 bytes)	27152fb1cc2b59c0110935c6bdeb2eaa58fa955f
Mac OS X	android-studio-ide-145.3360264-mac.dmg	440 MB (461824413 bytes)	2e89fed3601e5bd19112c29c172cb29be3b34f8e
Linux	android-studio-ide-145.3360264-linux.zip	445 MB (466765476 bytes)	fc63ca247762697c33102a78063a95f8b5ab5dea

Figura 7: SDK Android – Android Studio.
Fuente: developer.vuforia.com



*Figura 8: Importar SDK – Unity.
Fuente: Captura propia en Unity.*

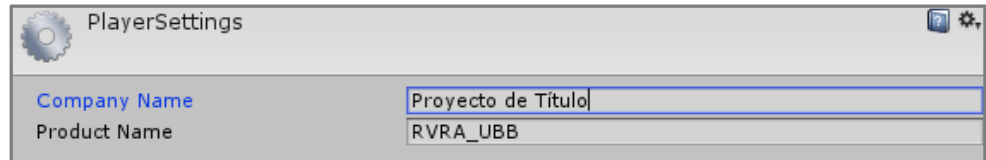
- Unity nos permite importar y trabajar con los SDK mencionados que pueden ser descargados desde el mismo software.
Con todos estos elementos se puede comenzar a desarrollar la aplicación utilizando Vuforia sobre Unity como se describe en los puntos siguientes.

2.3.4 Estructura del proyecto Android en el motor Unity

La creación de una aplicación para Android por medio del motor de desarrollo Unity requiere de una breve configuración que hace posible la correcta configuración dentro del sistema del dispositivo móvil.

En el menú Editar>Opciones de Proyecto>Player se debe especificar lo siguiente:

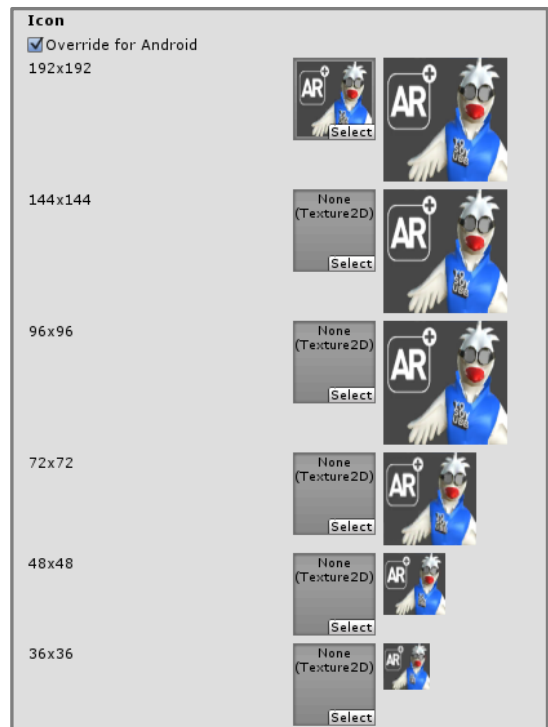
- Player
 - Nombre del producto.
 - Nombre de la compañía o Desarrollador.



*Figura 9: Configuración Player - Unity.
Fuente: Captura propia en Unity.*

- Resolution and Presentation
 - Orientación por defecto de la aplicación en la pantalla.

- Icons and Splash Image



*Figura 10: Configuración Icon – Unity.
Fuente: Captura propia en Unity.*

- Other Settings
 - APIs Gráficas a utilizar.
 - Renderizado multiproceso.
 - Identificación dentro del dispositivo: com.ID.ID.
 - Versión actual.
 - Nivel de API mínima requerida.
 - Accesos de escritura.
 - Nivel de compatibilidad API.

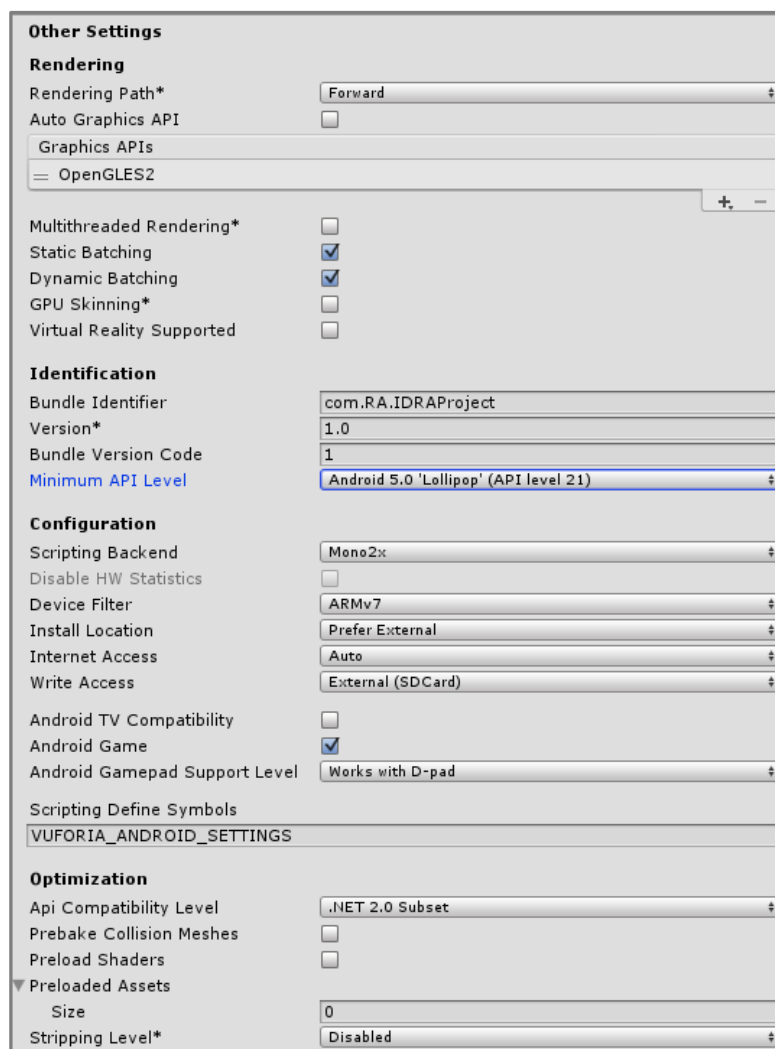


Figura 11: Otras configuraciones – Unity.
Fuente: Captura propia en Unity.

2.4 Realidad Aumentada

2.4.1 Introducción

Mientras que en la realidad virtual el mundo real desaparece, en la realidad virtual este mundo continúa presente, pero además se ve enriquecido con nuevas características e información. Al observar la realidad en tiempo real el dispositivo móvil se encarga de añadir información adicional como: información de lugares, recreación de objetos virtuales, información en tiempo real de los sucesos captados.

En 1994, Paul Milgram y Fumio Kishino formularon la siguiente definición de realidad aumentada llamada Milgram Virtuality Continuum, esta nos dice que entre un entorno real y uno virtual existe una realidad mixta que se encuentra dividida entre dos realidades: la aumentada y la virtualidad aumentada, esta última en lo más próximo a la virtualidad pura.

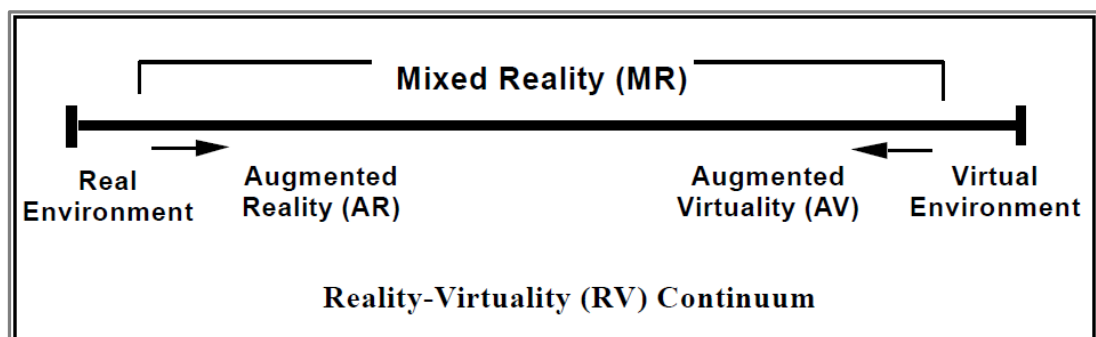


Figura 12: Realidad mixta – Milgram Virtuality Continuum.
Fuente: SPIE Vol. 2351, Telem manipulator and Telepresence Technologies, 1994.

Mientras que en 1997, Ronald Azuma describe la realidad aumentada como aquella que es capaz de cumplir 3 requisitos:

- Combinación de elementos virtuales y reales.
- Interactividad en tiempo real.
- Información almacenada en 3D.

Para hacer posible lo anterior es necesario que el dispositivo cuente con:

- Sistema de visualización: Pantalla, casco o gafas que nos permitan obtener la información que el dispositivo nos entrega.
- Dispositivo de entrada: Como requisito mínimo se debe contar con una cámara, aunque adicionalmente hay otros elementos útiles como el GPS, giroscopios y acelerómetros.

2.4.2 Desarrollo básico

Para obtener la correcta fusión entre lo captado del mundo real con la cámara y las imágenes virtuales 3D se debe crear un sistema de coordenadas que sitúen las imágenes en las posiciones correctas atribuidas a diferentes condiciones leídas por la cámara desde el mundo real, como por ejemplo una casa en particular, un auto específico, una taza de determinadas características.

Generalmente el método consta de dos partes:

- Detección: El dispositivo analiza el mundo real a través de la cámara utilizando detección de esquinas, de bordes, de umbral y métodos de procesado de imágenes. Así el dispositivo puede reconocer cual es el lugar al que está apuntando.
- Posicionamiento de modelos 3D sobre el sistema de coordenadas: A partir de los datos obtenidos en la primera parte el procesador es capaz de establecer la información virtual creada sobre el sistema de coordenadas creado en concordancia con el mundo real para crear la fusión sobre el dispositivo de salida, ya sea pantalla, casco o gafas.

Es en este punto en el que necesitamos, a partir del motor de desarrollo Unity que es capaz de crear el sistema de coordenadas, un método que nos permita crear esta fusión de realidades, el detonante que nos indique en qué momento nos encontramos en un paso de detección válido y cuándo realizar el posicionamiento de modelos sobre las coordenadas.

2.5 Motor de desarrollo Unity

Unity es un motor multiplataforma creado por Unity Technologies compatible con 3ds, Maya, Softimage, Blender, Modo, ZBrush, Cinema 3d, Cheetah3D, Photoshop, Adobe Fireworks y Allegorithmic Substance, los cambios realizados en los objetos del proyecto creados por estos productos son automáticamente actualizados en todas las instancias sin necesidad de volver a importarlos.

El motor gráfico utiliza Direct3D en Windows, OpenGL en Mac y Linux, OpenGL ES en Android y iOS.

El script es basado en Mono, la implementación de código abierto de .NET Framework. Los programadores pueden utilizar UnityScript (lenguaje personalizado inspirado en sintaxis ECMAScript), Boo (sintaxis inspirada en Python) o C#.

2.6 SDK Vuforia

Vuforia es un kit de desarrollo de aplicaciones de realidad aumentada disponible para Android y iOS con extensión para Unity que permite efectuar lo descrito en el capítulo 2.4 sobre realidad aumentada.

2.6.1 Arquitectura del SDK

Una aplicación desarrollada con Vuforia SDK está compuesta por los siguientes elementos:

- Cámara: Esta asegura que la imagen sea captada y procesada por el Tracker.
- Base de datos: Es creada por el dispositivo utilizando el Target Manager y almacena una colección de Targets para ser reconocidos por el Tracker.
- Target: Es utilizado por el Tracker para reconocer el mundo real, los Targets pueden ser de diferentes tipos, los principales son:
 - Image Targets: Imágenes, tales como: fotos, cubiertas de libros, posters, entre muchos otros.
 - Word Targets: Elementos textuales representados por palabras simples o compuestas que pueden ser detectadas en modo de palabra completa o caracter por caracter.
- Tracker: Analiza la imagen de la cámara y detecta objetos del mundo real a través de los frame de la cámara con el fin de encontrar coincidencias en la base de datos.

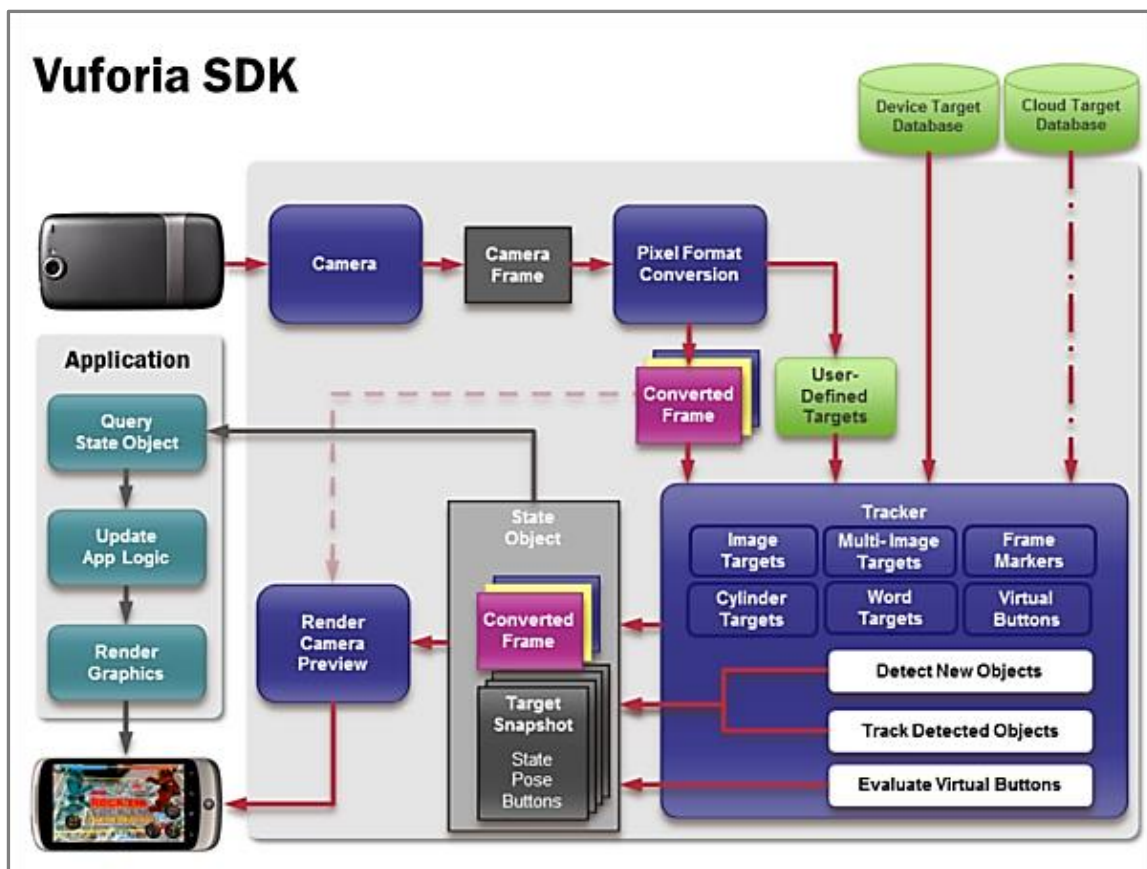


Figura 13: Diagrama de flujo de datos SDK Vuforia en una aplicación.
Fuente: vuforia.com.

En el diagrama presentado el dispositivo capta una escena del mundo real tomada por la cámara, el SDK Vuforia crea un frame (imagen particular dentro de una sucesión de imágenes) desde la escena capturada y convierte la imagen capturada por la cámara a una diferente resolución para ser correctamente tratada por el Tracker.

Vuforia analiza la imagen a través del Tracker y se buscan coincidencias en la base de datos que se encuentra compuesta por Targets.

Por último la aplicación renderiza el contenido virtual creado en el motor de desarrollo hacia la pantalla del dispositivo para crear la realidad mixta con los

elementos virtuales combinados con los elementos reales o lo que se conoce como Realidad aumentada.

2.6.2 Instalación en Unity

Para instalar el SDK se debe importar el paquete descrito al final del punto 2.3.3 de este proyecto, al igual que cualquier otro paquete de Unity.

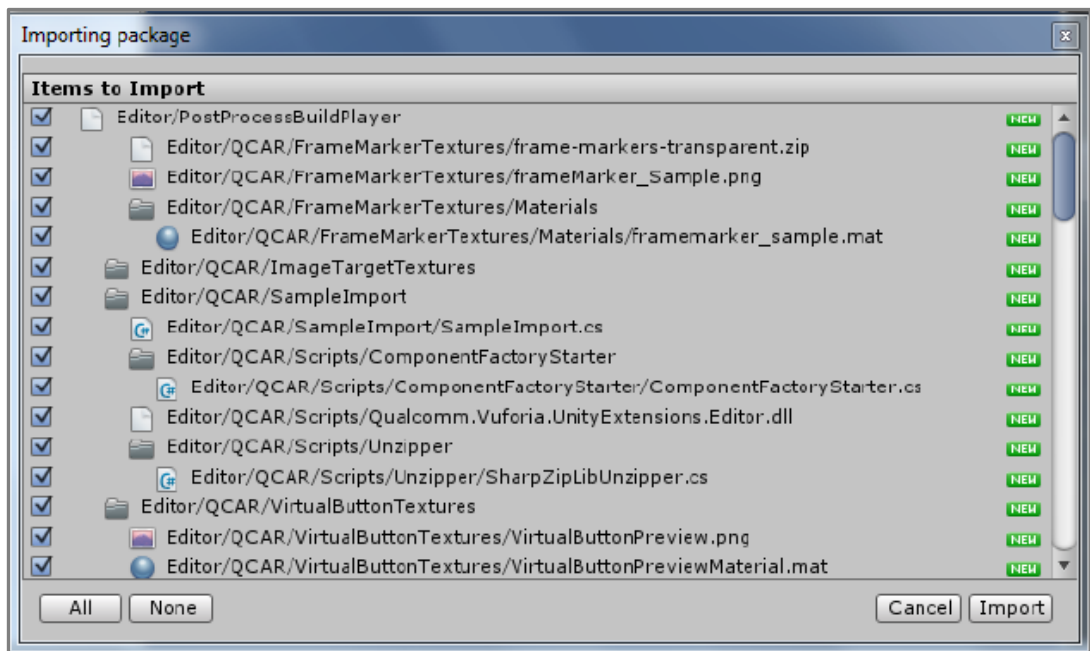


Figura 14: Importar paquetes (1) – Unity.
Fuente: Captura propia en Unity.

Una vez importado el paquete aparecerán nuevos Assets y una opción en el menú para acceder a toda la documentación.

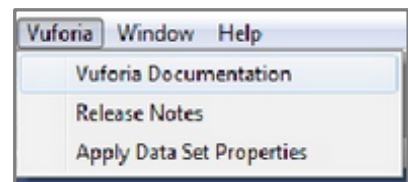


Figura 15: Importar paquetes (2) – Unity.
Fuente: Captura propia en Unity.

2.7 Aplicaciones similares

A continuación se menciona algunas de las aplicaciones de realidad aumentada más descargadas desde la tienda virtual android Google Play.

- Augment, 3D Realidad Aumentada:

Augment permite a las personas visualizar modelos 3D de los productos disponibles en una tienda, permite agregar modelos 3D y marcadores personalizados.



*Figura 16: Augment 3D – Aplicaciones Similares.
Fuente: Tienda virtual Google Play.*

- ARcuerpohumano:

Esta aplicación muestra modelos 3D de las ilustraciones del cuerpo humano presentes en el libro “Cuerpo Humano” de Ediciones Saldaña, se pueden mover o girar a gusto para ver en detalle las diferentes partes de los órganos y huesos.



Figura 17: ARcuerpohumano – Aplicaciones similares.
Fuente: Tienda virtual Google Play.

- RAInfinitem Realidad Aumentada:

Esta aplicación muestra modelos 3D de las siguientes publicaciones:

Revista Solo para ti, Revista Fastmag, Revista Contenido, todas estas de Sanborns.

Revista Asómate y Guía de Claro video y Recibo de Telmex.

Algunos discos musicales en tiendas Sanborns, el catálogo anual y el libro de El Tri: “Que viva el Rocanrol”, entre otros productos de Telmex, Sanborns y ZTE.

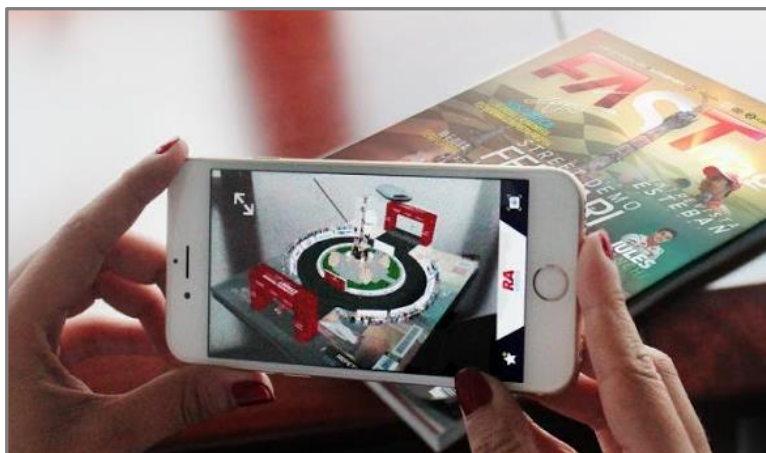


Figura 18: RAInfinitem – Aplicaciones similares.
Fuente: Tienda virtual Google Play.

CAPÍTULO 3 – DEFINICIÓN DEL PROYECTO

3.1 Justificación

El presente proyecto desea llevar a los usuarios más allá de la percepción habitual del ambiente que le ofrece el campus y presentarles una realidad complementaria que muestre a través de sus propios dispositivos móviles aspectos relevantes de lo que ellos estén observando esperando cautivar al usuario con material audiovisual que le permitan conocer más sobre su entorno detallando información sobre un conjunto de lugares cuidadosamente seleccionados que el estudiante deberá visitar o conocer debido al beneficio otorgado por estos durante el transcurso de su etapa universitaria como por ejemplo, facultades, bibliotecas, departamentos educacionales y de salud, casinos, oficinas importantes, entre otra. Información como descripciones, carreras impartidas o actividades realizadas, jefaturas, direcciones o personal importante, datos culturales e históricos sobre edificaciones y esculturas, horarios de atención, entre otros que la persona asesora pueda considerar pertinentes.

A causa de lo anterior se pueden identificar grandes aportes tanto para los usuarios de la aplicación, siendo o no estudiantes de la universidad, como para la institución que abarcan desde la ayuda a la persona que visita la universidad en conocer datos relevantes que no siempre están disponibles de manera fácil y rápida, el incentivo al estudiante para aprender más de su entorno educativo, hasta la culturización sobre lugares, esculturas y dependencias de la universidad y que en conjunto representan una contribución a mejorar la experiencia de sus usuarios.

3.2 Objetivos

3.2.1 Objetivos Generales

Desarrollar una aplicación móvil que permita apreciar al usuario un recorrido interactivo por medio de un personaje animado basada en realidad aumentada que presentará información relevante de un conjunto de edificios y lugares de interés para el estudiante dentro de las dependencias de la Universidad del Bío-Bío campus Concepción.

3.2.2 Objetivos Específicos

- Realizar estudios detallados sobre el uso de lenguajes de programación como Java, Javascript y C#, uso de herramientas de modelado, renderizado y animación 3D Blender y Unity3D, uso e implementación del motor de reconocimiento de imágenes Vuforia y creación de aplicaciones para móviles con sistema android bajo lenguaje Java.
- Modelar los personajes presentes en la guía virtual y desarrollar las distintas animaciones que se deberán adecuar a cada coordenada objetivo presente en la identificación de los edificios y lugares capturados por la cámara.
- Llevar a cabo procesos de investigación y recopilación de información de cada lugar incluido en la aplicación.
- Efectuar la captura de imágenes de los lugares incluidos en la aplicación necesarias para el sistema de reconocimiento.

- Realizar estudios para evaluar la factibilidad de las distintas alternativas de almacenamiento de información y posterior elección considerando que las medidas adoptadas pueden afectar positiva o negativamente en la calidad de la aplicación final.
- Implementar la base de datos para el almacenamiento de la información requerida.
- Desarrollar una aplicación móvil android que permita montar el sistema de reconocimiento y animado 3D en los dispositivos de los usuarios.
- Aplicar técnicas de optimización gráfica de android.

3.3 Ambiente de ingeniería de software

3.3.1 Metodología de desarrollo

Se ha considerado que la metodología SCRUM es la más adecuada para aplicar a este proyecto, la persona asesora puede ver avances al finalizar cada ciclo Sprint con el fin de incorporar o desechar elementos de la aplicación otorgando a esta un mayor grado de adaptabilidad a las expectativas y necesidades reales del usuario, al finalizar cada ciclo se pueden observar funcionalidades concretas lo que permitirá detectar y solucionar posibles errores en la ejecución de la aplicación.

Scrum aporta también un alto grado de compromiso de parte del alumno en cumplir con los tiempos designados hasta el final de cada ciclo sprint, combatiendo los retrasos en la fecha de finalización del proyecto.

Cabe destacar que realizar un trabajo conjunto entre asesor y alumno garantiza claridad en los requisitos y objetivos, verificación de los avances realizados y retroalimentación en pos de una evolución positiva de la aplicación. El hecho de poder adicionalmente utilizar ciclos incrementales da flexibilidad y un importante aporte a la calidad del producto final.

3.3.2 Técnicas y notaciones

En el desarrollo del proyecto se contemplan:

- Modelos UML de diagrama de Casos de Uso.
- Modelos de Arquitectura y descomposición funcional.
- Modelos de secuencias.
- Encuesta online no probabilística.
- Toma de requerimientos a través de entrevistas con el cliente.
- Reuniones programadas y retroalimentación según metodología SCRUM.

3.3.3 Estándares de documentación

Para documentar este proyecto se utiliza la plantilla de desarrollo de software de la Facultad de Ciencias Empresariales de la Universidad del Bío-Bío siguiendo su formato de memorias de titulación en la versión del año 2016, este documento responde a una adaptación de los estándares:

- IEEE Software Test Documentation Std 829 – 1998.
- IEEE Software Requirements Specifications Std 830 – 1998.

3.3.4 Herramientas de apoyo al desarrollo

- Motor de desarrollo de contenido interactivo Unity con SDK Vuforia y SDK Android.
- Motor de desarrollo de gráficos 3D y animación, Blender.
- Herramienta de modelado Power Designer para diagramas UML.
- Editor de imágenes Photoshop CS6.
- Editor de código Sublime Text 3.
- Microsoft Office.
- Microsoft Project.
- Software online de diagramas draw.io
- Emulador android para Windows Bluestacks.

3.4 Definiciones, siglas y abreviaciones

1. **API:** Application Programming Interface, Interfaz de programación de aplicaciones. Es un conjunto de funciones y métodos utilizados generalmente en bibliotecas de programación mediante un software de desarrollo para conseguir abstracción entre niveles inferiores y superiores del software y funciones de otras aplicaciones.
2. **APP Store:** Servicio disponible en dispositivos Apple, representa una tienda virtual que permite a los usuarios la búsqueda y descarga de aplicaciones móviles.
3. **AR:** Augmented Reality, en español Realidad Aumentada (RA). – Término utilizado para definir la visión de un entorno físico del mundo real a través de un dispositivo tecnológico que combina esta visión con elementos virtuales.

4. **C#:** C Sharp, pronunciado en inglés (si sharp), es un lenguaje de programación orientado a objetos, desarrollado por Microsoft, posee una sintaxis derivada de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar a Java, aunque incluye mejoras de otros lenguajes, posee influencias de lenguajes Java, C++, Eiffel, Modula-3 y Pascal.
5. **Google Play:** Servicio brindado a los dispositivos con sistema operativo Android, representa una tienda virtual que permite a los usuarios la búsqueda y descarga de aplicaciones móviles.
6. **GPS:** Global Positioning System, Sistema de posicionamiento global, Es un sistema que permite determinar la posición geográfica del dispositivo por medio de sensores GPS y satélites.
7. **GUI:** Graphical User Interface, Interfaz gráfica de usuario, Conjunto de imágenes y objetos gráficos que representan información y acciones disponibles en la interfaz, proporciona un entorno visual sencillo que permite la comunicación entre el usuario y el sistema.
8. **IDE:** Integrated Development Environment, Entorno de desarrollo integrado, Aplicación informática que proporciona servicios integrales que facilitan al desarrollador o programador el desarrollo de software.
9. **IEEE:** Institute of Electrical and Electronics Engineers, Instituto de Ingeniería Eléctrica y Electrónica, Asociación mundial de ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas.

10. **JDK:** Java Development Kit, Kit de desarrollo de java, Software que provee herramientas de desarrollo para la creación de programas en Java.
11. **Landscape:** Paisaje en su traducción al español, representa la orientación horizontal de un dispositivo (como si se sacara una fotografía a un paisaje), para efectos de este proyecto, landscape hacia la izquierda corresponde a orientar el dispositivo móvil horizontalmente con la parte superior de este hacia el lado izquierdo del usuario.
12. **OpenGL:** Open Graphics Library, especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.
13. **Plugin:** Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.
14. **SDK:** Software Development Kit, Kit de desarrollo de software, Generalmente corresponde a un conjunto de herramientas de desarrollo de software que permiten al desarrollador o programador de software crear aplicaciones para un sistema concreto.
15. **Smartphone:** Teléfono inteligente en español, Teléfono móvil con capacidad para usarse como un computador de bolsillo, posee pantalla táctil, permite al usuario conectarse a internet, gestionar cuentas de correo electrónico e instalar aplicaciones y recursos a modo de pequeño computador.
16. **UML:** Unified Modeling Language, Lenguaje unificado de modelado, corresponde a un lenguaje de modelado de sistemas de software que permite, mediante gráficas, visualizar, especificar, construir y documentar sistemas.

17. **VR:** Virtual Reality, en español Realidad virtual (RV). – Entorno generado mediante tecnología informática que crea en el usuario la sensación de estar inmerso en él.

18. **x64:** Arquitectura de microprocesadores que describen la anchura de registros, bus de direcciones, bus de datos o instrucciones de 64 bits.

CAPÍTULO 4 – ESPECIFICACIÓN DE REQUERIMIENTOS

Adaptación basada en IEEE Software requirements specifications Std 830-1998.

4.1 Alcances

Esta aplicación tiene como propósito entregar información al usuario sobre la estructura dentro del campus a la que está apuntando con su dispositivo móvil en ciertos puntos designados.

Se establece por tanto los siguientes lugares como objetivos que activan el funcionamiento de la aplicación.

Lugar	Información considerada
Rectoría	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Autoridades ▪ Aspectos históricos de la universidad ▪ Aspectos arquitectónicos e históricos del edificio ▪ Visión y misión de la universidad
Facultades	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Facultad de Arquitectura, Construcción y Diseño ▪ Facultad de Ciencias ▪ Facultad de Ciencias ▪ Autoridades y contactos ▪ Departamentos ▪ Carreras en el campus ▪ Horarios de atención

<p>Empresariales</p> <ul style="list-style-type: none"> ▪ Facultad de Educación y Humanidades ▪ Facultad de Ingeniería 	<ul style="list-style-type: none"> ▪ Programas y cursos ▪ Aspectos arquitectónicos e históricos del edificio ▪ Visión y misión de la facultad
<p>Ubicaciones de los Departamentos pertenecientes a las Facultades</p>	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Autoridades y contactos ▪ Aspectos arquitectónicos e históricos del edificio
<p>Edificio de Administración general (Finanzas, Registro académico, Oficina de Partes, Caja)</p>	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Oficinas: <ul style="list-style-type: none"> ▪ Finanzas ▪ Registro académico ▪ Oficina de Partes ▪ Caja ▪ Autoridades y contactos ▪ Horarios de atención
<p>Edificio de pregrado</p>	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Autoridades y contactos ▪ Horarios de atención
<p>Biblioteca</p>	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Autoridades y contactos ▪ Horarios de atención

	<ul style="list-style-type: none"> ▪ Aspectos arquitectónicos e históricos del edificio ▪ Beneficios y programas ▪ Visión y misión
Salas de Estudios Generales	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Autoridades y contactos ▪ Horarios de atención
Casino	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Autoridades y contactos ▪ Horarios de atención ▪ Aspectos arquitectónicos e históricos del edificio
Bienestar estudiantil	<ul style="list-style-type: none"> ▪ Nombre. ▪ Descripción ▪ Autoridades y contactos ▪ Horarios de atención ▪ Aspectos arquitectónicos e históricos del edificio ▪ Beneficios y programas
Aula Magna	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Autoridades y contactos ▪ Horarios de atención ▪ Aspectos arquitectónicos e históricos del edificio ▪ Beneficios y programas
Gimnasio	<ul style="list-style-type: none"> ▪ Nombre

	<ul style="list-style-type: none"> ▪ Descripción ▪ Autoridades y contactos ▪ Horarios de atención
Paraninfo	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Aspectos arquitectónicos e históricos del edificio
Federación de Estudiantes	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Horarios de atención
Plaza de la democracia	<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Aspectos arquitectónicos e históricos del lugar
Salas de estudio y edificios de clases	<ul style="list-style-type: none"> ▪ Nombres
Algunas esculturas dentro del campus	<ul style="list-style-type: none"> ▪ Nombre ▪ Creador ▪ Información del motivo e historia de la obra

*Tabla 2: Alcances de información en la aplicación.
Fuente: Elaboración propia.*

La aplicación será capaz de mostrar la información descrita en la tabla anterior en cada uno de los lugares mencionados. Esta también se encuentra, a modo de programación, segmentada en 3 partes, Introducción, Mapa y Detección de imágenes.

La introducción consta de una serie de imágenes y texto que introduce al usuario hacia el uso correcto de la aplicación y entrega una pequeña presentación de la misma, el mapa entrega una visión aproximada de la ubicación del usuario y una

imagen del edificio en los lugares marcados utilizando el GPS del dispositivo móvil, la detección de imágenes pone en marcha el código de reconocimiento de los objetivos mencionados anteriormente para entregar la información pertinente en casa caso.

4.2 Objetivos de la aplicación

4.2.1 Objetivos generales

La aplicación detectará los lugares objetivo y mostrará un aviso en pantalla cada vez que algún lugar sea identificado por el dispositivo, de este modo el usuario puede decidir si desplegar una lista con los ítems disponibles para ese lugar que al ser pulsados despliegan la información relacionada con el ítem seleccionado.

En la sección de “Mapa” la aplicación mostrará un pequeño mapa del sector con un marcador en caso de que el usuario se encuentre dentro de los límites del campus, se muestra un mensaje en caso de no disponer del GPS del dispositivo.

4.2.2 Objetivos específicos

- Detectar en qué momento el dispositivo se encuentra con la cámara activa hacia uno de los lugares objetivo designados y en qué momento se ha dejado de enfocar el objetivo.
- Por cada lugar objetivo, se mostrará la lista de ítems disponibles correspondiente.

- Detectar la ubicación del dispositivo a través del GPS cuando el usuario esté utilizando la vista de “Mapa”, mostrar el marcador de posición si se encuentra dentro del campus.

4.3 Descripción global

4.3.1 Interfaz de usuario

- Para funcionar de manera adecuada la aplicación mostrará, para cada lugar objetivo disponible, un listado de ítems relacionados al lugar al que apunta la cámara.
- La aplicación funcionará exclusivamente con orientación landscape izquierda (horizontal con la parte superior del dispositivo para el lado izquierdo).

4.3.2 Interfaz de hardware

Bajo el sistema operativo Android se requiere:

- Sistema de Posicionamiento Global (Localizador GPS).
- Cámara posterior.

4.3.3 Interfaz de software

El uso específico de software por parte de la aplicación es el siguiente:

- Sistema operativo
 - Nombre: Sistema operativo Android.
 - Abreviación: S.O. Android.
 - Versión: Android 4.4 (KitKat) o superior.

4.3.4 Consideraciones ambientales

La aplicación necesita el cumplimiento de los siguientes requisitos mínimos de hardware y software en sus dispositivos móviles por parte de los usuarios para su funcionamiento adecuado (también detallados en el punto 5.1 de este documento):

- Procesador de 800MHz o mejor.
- 500 MB de memoria RAM.
- Cámara de 5MP. O superior.
- Pantalla de 4.5 pulgadas, resolución mínima 480x800 o mejor.
- Sistema operativo Android 4.4 Kitkat o superior (ver tabla 13 - Dispositivos).
- Opcionales: Conexión a internet y disponibilidad de sensor GPS.

4.4 Requerimientos Específicos

4.4.1 Requerimientos funcionales

ID	Requerimiento	Descripción
RF.01	Detección de imágenes sobre lugares objetivo.	Mostrar en pantalla cuando la aplicación encuentre un lugar objetivo válido.
RF.02	Desplegar ítems correspondientes al lugar objetivo detectado.	Al pulsar sobre el nombre del lugar detectado se debe desplegar una lista de ítems que contienen la información correspondiente a ese lugar.
RF.03	Mostrar información del ítem seleccionado.	Al pulsar sobre el ítem deseado se debe mostrar en pantalla la información relacionada.
RF.04	Detectar objetivo perdido.	Al dejar de enfocar un lugar objetivo el dispositivo debe reconocer la situación y no seguir mostrando ítems ni información sobre el objetivo que se ha perdido.
RF.05	Desplegar mapa del campus y ubicación dentro de él.	Debe encontrarse disponible un mapa del sector y mostrar la ubicación del dispositivo en este si el usuario se encuentra dentro del rango abarcado.
RF.06	Mostrar estados sobre objetivos rastreados y uso del GPS.	Desplegar mensajes claros para el usuario en las siguientes situaciones: <ul style="list-style-type: none"> • Lugar objetivo encontrado. • Lugar objetivo perdido. • GPS no disponible. • Wifi no disponible.

Tabla 3: Requerimientos funcionales.
Fuente: Elaboración propia.

4.4.2 Especificación de requerimientos

Esp. R.	Req.	Nombre		
ERF.01	RF.01	Detección de imágenes sobre lugares objetivo.		
Descripción		Mostrar en pantalla cuando la aplicación encuentre un lugar objetivo válido.		
Entradas		Salidas	Restricciones	Proceso
Imagen capturada por la cámara.		Recuadro en la interfaz que indica el objetivo válido.	Objetivo debe estar al alcance de la cámara.	Con la aplicación abierta el usuario debe apuntar la cámara del dispositivo a un objetivo válido.

Tabla 4: ERF.01 – Especificación de requerimientos.
Fuente: Elaboración propia.

Esp. R.	Req.	Nombre		
ERF.02	RF.02	Desplegar ítems correspondientes al lugar objetivo detectado.		
Descripción		Al pulsar sobre el nombre del lugar detectado se debe desplegar una lista de ítems que contienen la información correspondiente a ese lugar.		
Entradas		Salidas	Restricciones	Proceso
Botón pulsado con el nombre del lugar objetivo válido.		Listado de ítems disponibles de acuerdo al lugar objetivo identificado.	Objetivo debe estar al alcance de la cámara.	Con la aplicación abierta el usuario debe apuntar la cámara del dispositivo a un objetivo válido y presionar el botón que indica el nombre del lugar reconocido.

Tabla 5: ERF.02 – Especificación de requerimientos.
Fuente: Elaboración propia.

Esp. R.	Req.	Nombre		
ERF.03	RF.03	Mostrar información del ítem seleccionado.		
Descripción		Al pulsar sobre el ítem deseado se debe mostrar en pantalla la información relacionada.		
Entradas		Salidas	Restricciones	Proceso
Botón pulsado del ítem deseado.		Información del ítem seleccionado se muestra en pantalla.	Objetivo debe estar al alcance de la cámara.	Con el objetivo válido reconocido en la cámara y el usuario debe presionar un ítem de información específica.

Tabla 6: ERF.03 – Especificación de requerimientos.
Fuente: Elaboración propia.

Esp. R.	Req.	Nombre		
ERF.04	RF.04	Detectar objetivo perdido.		
Descripción		Al dejar de enfocar un lugar objetivo el dispositivo debe reconocer la situación y no seguir mostrando ítems ni información sobre el objetivo que se ha perdido a excepción de la información detallada de cada ítem. Por ejemplo: ítem seleccionado: Horarios – La información desplegada de horarios se muestra en pantalla (información específica).		
Entradas		Salidas	Restricciones	Proceso
No existe		Ya no se muestra la información en pantalla del objetivo perdido.	El objetivo debió ser reconocido anteriormente.	El usuario deja de apuntar su dispositivo hacia un objetivo válido.

Tabla 7: ERF.04 – Especificación de requerimientos.
Fuente: Elaboración propia.

Esp. R.	Req.	Nombre	
ERF.05	RF.05	Desplegar mapa del campus y ubicación dentro de él.	
Descripción		Debe encontrarse disponible un mapa del sector y mostrar la ubicación del dispositivo en este si el usuario se encuentra dentro del rango abarcado.	
Entradas	Salidas	Restricciones	Proceso
Botón mapa pulsado por el usuario.	El mapa se muestra en la pantalla.	<p>El mapa se muestra al ser requerido por el usuario.</p> <p>La ubicación se muestra siempre y cuando el usuario se encuentre dentro del rango del mapa y con el sensor de GPS activado.</p> <p>La vista del mapa puede experimentar fallos de ubicación leves cuando el dispositivo no se encuentre conectado a internet (en este caso el mapa de GoogleMaps se reemplaza por una imagen estática).</p>	El usuario pulsa el botón de mapa presente en la interfaz, dependiendo de las necesidades puede o no activar su sensor GPS y su receptor Wifi.

Tabla 8: ERF.05 – Especificación de requerimientos.
Fuente: Elaboración propia.

Esp. R.	Req.	Nombre	
ERF.06	RF.06	Mostrar estados sobre objetivos rastreados y uso del GPS.	
Descripción		Desplegar mensajes claros para el usuario en las siguientes situaciones: <ul style="list-style-type: none"> • Lugar objetivo encontrado. • Lugar objetivo perdido. • GPS no disponible. • Wifi no disponible. 	
Entradas	Salidas	Restricciones	Proceso
No existe	Notificaciones en la interfaz	No existe	Las notificaciones se muestran en la interfaz en los casos: <ul style="list-style-type: none"> • La cámara detecta un lugar objetivo válido. • La cámara pierde un lugar detectado anteriormente. • El sensor GPS y/o Wifi no se encuentra disponible.

Tabla 9: ERF.06 – Especificación de requerimientos.
Fuente: Elaboración propia.

4.4.3 Interfaces externas de entrada

Debido a las características de la aplicación no existen interfaces externas de entrada.

4.4.4 Interfaces externas de salida

ID	Nombre Interfaz	Detalle	Medio
IES.1	Nombre del lugar objetivo encontrado.	Se indica en un recuadro el nombre del lugar que ha sido detectado por la cámara.	Pantalla
IES.2	Ítems disponibles relacionados al lugar objetivo encontrado.	Se muestran los ítems disponibles para el lugar encontrado, por ejemplo: Horarios, Autoridades, Programas, entre otros.	Pantalla
IES.3	Información referente al ítem seleccionado.	Muestra la información requerida sobre el ítem seleccionado, por ejemplo: el detalle sobre la Misión del lugar, el detalle de sus Autoridades, entre otros.	Pantalla

*Tabla 10: Interfaces externas de salida.
Fuente: Elaboración propia.*

4.4.5 Atributos de la aplicación

Es de vital importancia que una aplicación destinada a ayudar al usuario y hacer su experiencia más fácil para realizar una tarea cotidiana enfatice los atributos de usabilidad, el producto final debe ser amigable, intuitivo, atractivo, pero además de esto debe otorgar complacencia entre otros aspectos fundamentales detallados a continuación mencionando el grado de relevancia que poseen en este proyecto.

- Funcionalidad: Los aspectos de funcionalidad intentan cumplir con las necesidades del usuario, en este punto, que la aplicación logre ser el aporte que el usuario busca en ella y que cumpla los requerimientos establecidos es esencial.
 - Adecuación: La aplicación se ajusta a los requerimientos establecidos.
 - Idoneidad: Se posee la capacidad para entregar las funciones apropiadas en vista a solucionar la problemática planteada, la aplicación comprende un funcionamiento ideal para detectar objetivos y entregar información utilizando el término de realidad aumentada.
 - Precisión: Dado que la información entregada en gran parte proviene de los portales oficiales de la institución y otra parte es entregada por personal calificado el producto tiene capacidad necesaria para proporcionar resultados correctos a la fecha en la que son consultados.

- Fiabilidad: La aplicación está diseñada para mantenerse operativa dentro de las condiciones normales establecidas en los requisitos mínimos de hardware (Punto 5.1).
 - Madurez: La aplicación se ha desarrollado teniendo en cuenta la prevención de fallos en el software.

- Usabilidad: Durante el desarrollo, uno de los enfoques clave fue darle al producto la habilidad para que el usuario invierta el mínimo esfuerzo al utilizar las funciones disponibles.
 - Inteligibilidad: El usuario puede comprender fácilmente el uso correcto de la aplicación y sus utilidades en las tareas para las que está destinada.
 - Facilidad de aprendizaje: Se ha desarrollado un sistema de navegación intuitivo que no requiere mayor esfuerzo de aprendizaje.
 - Operabilidad: El usuario no requiere mayores conocimientos para operar y controlar la aplicación de manera óptima.
 - Atractividad: Se pretende dar atractivo al producto final a través de la interfaz gráfica, aunque el atractivo mayor lo otorga la incorporación como fundamento de la realidad aumentada.
 - Cumplimiento de la usabilidad: El producto se adhiere a las convenciones de estilo y regulaciones de usabilidad.

- Eficiencia: Se posee la capacidad de responder con adecuada velocidad a las peticiones del usuario.

- Comportamiento en el tiempo: Bajo las condiciones requeridas mínimas es posible operar en tiempos de respuesta y de procesos adecuados.
 - Cumplimiento de la eficiencia: El producto se adhiere a las convenciones de eficiencia establecidas.
-
- Adaptabilidad: La aplicación funcionará correctamente según los requerimientos mínimos establecidos en equipos con sistemas Android de versión 4.4 KitKat o superior sin inconvenientes.
 - Facilidad de instalación: Analizando el punto 5.3.2. Cualquier alternativa que se tome supondrá para el usuario, en el caso más extenso, apretar un botón para descargar la aplicación y otro para aceptar los permisos necesarios del sistema. Instalar la aplicación no lleva más de 1 minuto.
 - Coexistencia: Ya instalada, la aplicación no interfiere con funcionamientos no procesos externos pertenecientes a otras aplicaciones o propiedades del sistema en el dispositivo.
 - Fácilmente actualizable: Actualizar la aplicación sigue un proceso muy similar al de instalación descrito en el punto anterior “Facilidad de instalación” puesto que el sistema android puede sobrescribir los archivos nuevos y crear una versión actualizada de la instalación.

CAPÍTULO 5 – ESTUDIO DE FACTIBILIDAD

Completar un proyecto de manera exitosa y disminuir los riesgos que puedan comprometer el cumplimiento de los objetivos se logra en gran parte al análisis cuidadoso de aspectos técnicos, operativos y económicos. Es necesario considerar el software y hardware mínimo, costos, beneficios y conocimientos previos detallados a continuación.

5.1 Factibilidad técnica

La primera parte del estudio de factibilidad consiste en evaluar la tecnología existente, software, hardware, dispositivos, necesarios para desarrollar y utilizar la aplicación.

Software			
Software	Versión	Tipo de software	Plataforma
Microsoft Windows	7 Ultimate x64	Sistema operativo - Plataforma	
Android	6.0.1 – Marshmallow	Sistema operativo - Plataforma	
Blender	2.78a	Diseño y animación 3D	Windows 10 8 7 vista - x64
Unity	5.3.1 – Personal Edition	Diseño y animación 3D	Windows 10 8 7 vista - x64
Sublime Text	3 – build 3126	Editor de texto para código	Windows x64
Photoshop	CS6 – v13	Editor de imágenes	Windows x64
Android	2.2.2.0	IDE creación	Windows 10 8 7

Studio		aplicaciones Android	x64 – x86
Power Designer	16.1	Modelado	Windows

Tabla 11: Software – Factibilidad técnica.
Fuente: Elaboración propia.

Plugins		
Plugin	Versión	Plataforma
SDK Vuforia	6.1 – Unity	Unity – Windows 7 x64
Android SDK	24.4.1	Unity – Windows 7 x64
Java Development Kit (JDK)	Kit 8	Windows x64

Tabla 12: Plugins – Factibilidad técnica.
Fuente: Elaboración propia.

Las licencias del software y plugins descritos se encuentran dentro de los límites de adquisición del estudiante en situación de desarrollo de proyecto de título, gran parte de ellos se encuentran en internet con licencias gratuitas y el resto con versiones trial o de prueba con periodos suficientes para el margen establecido en la carta Gantt, por lo que se puede desarrollar la aplicación utilizando las herramientas mencionadas con un presupuesto mínimo.

Para los sistemas operativos en la tabla 11 se utilizan licencias activas desde la compra de los dispositivos que las usan: un computador de escritorio y un Smartphone Motorola MotoG de tercera generación.

Dispositivos utilizados y requerimientos mínimos		
Dispositivo utilizado		Requerimientos mínimos
Computador de escritorio		
Componente	Especificaciones	
Procesador	AMD Athlon x3 2.9GHz.	1GHz o más rápido – 64bits (x64). Compatible con instrucciones SSE2.
Memoria RAM	7GB de RAM	2GB de RAM o mejor.
Almacenamiento	500GB de almacenamiento.	Al menos 40GB de almacenamiento.
Tarjeta gráfica	Nvidia GTX 550ti 1GB.	Compatible con DirectX9 con controlador WDDM 1.0 o superior. O DirectX11 con funciones nivel 9.3.
Requisitos adicionales		
Acceso a Internet.		
Smartphone		Requerimientos mínimos
Componente	Especificaciones	
Procesador	1.4 GHz. QuadCore.	800MHz. O más rápido.
Memoria RAM	1GB de RAM.	500MB de RAM o mejor.
Cámara posterior	13 MP.	5 MP o mejor.
Sistema operativo	Android 6.1 Marshmallow	Android 4.4 Kitkat o superior.
Pantalla	5” – 720x1280 pixels.	4.5” o mejor. – 480 x 800 o superior recomendado.
Requisitos adicionales	Conexión a internet.	

Tabla 13: Dispositivos – Factibilidad técnica.
Fuente: Elaboración propia.

Por último, el desarrollador cuenta con los conocimientos mínimos para el desarrollo del proyecto, la mayoría de estos han sido aprendidos del material bibliográfico mencionado en su punto correspondiente.

En los anexos de este proyecto se presenta información adicional correspondiente a encuestas sobre aspectos técnicos de los dispositivos móviles de la población.

5.2 Factibilidad operativa

Se establecen los impactos positivos y/o negativos relacionados con la aplicación.

5.2.1 Impactos positivos

- Disminuir la incertidumbre de los usuarios en variada información sobre los lugares y edificios de la universidad.
- Posee información confiable extraída en su totalidad desde los diferentes portales web de la universidad.
- Visualización de la información de manera fácil y entendible.
- Aplicación amigable para el usuario.
- Método de desarrollo utilizado atractivo para el usuario.
- Obtención de la información en tiempos menores a los que serían necesarios consultando la información en portales o consultas presenciales.

5.2.2 Impactos negativos

- Las personas que no cuenten con los requerimientos mínimos, principalmente que no sean usuarios de un dispositivo con sistema operativo android, no podrán utilizar la aplicación.

Luego de analizar los impactos se destaca que los negativos no logran opacar los beneficios que la aplicación puede entregar a sus usuarios.

5.3 Factibilidad económica

Se especifica inversiones y ganancias que podrían surgir de este proyecto y posterior producto.

5.3.1 Inversión de tangibles

En lo que refiere a equipos y herramientas, tanto por parte del desarrollador como del cliente, la aplicación no considera inversiones de ningún tipo y teniendo en cuenta los impactos positivos mencionados puede llegar a considerarse como un ahorro para el cliente.

5.3.2 Inversión de intangibles

Como se puede ver en punto 5.1 no es necesario invertir en software o comprar herramientas adicionales, pero se debe tener en cuenta que existen dos claros caminos para hacer llegar la aplicación hasta los dispositivos de los usuarios, el cliente deberá

decidir si utilizar ambos o sólo uno de ellos, de esto dependerá la inversión de intangibles.

Alternativa 1: Distribuir la aplicación utilizando los portales web de la universidad	
Inversión	Costo
No requiere inversión	\$0.-

*Tabla 14: Distribución propia – Inversión intangible.
Fuente: Elaboración propia.*

Alternativa 2: Distribuir la aplicación por medio de la tienda virtual Google Play	
Inversión	Costo
Suscripción vitalicia de desarrollador en Google Play	USD\$ 25. (Dólares) – Aproximadamente CLP\$16.840.- (Pesos chilenos).

*Tabla 15: Distribución Google Play – Inversión intangible.
Fuente: Elaboración propia.*

El valor de cambio está sujeto a cambios de acuerdo a la variación del dólar, se ha calculado el valor en pesos chilenos de la siguiente manera al 1ro de Diciembre de 2016:

$$\text{USD\$ 1} = \text{CLP\$ 673,54.-} \quad \text{CLP\$ 673,54} \times 25 = \text{CLP\$ 16.838,5.-}$$

Es importante entender que utilizar la tienda virtual de Android (Alternativa 2) puede ser un plus de confianza para las personas, además de ayudar a alcanzar un mayor número de personas quienes vean la aplicación y sus funciones a causa del alto número de visitas que posee Google Play cada día.

5.3.3 Inversión de recursos humanos

Este proyecto es desarrollado por sólo una persona, alumno memorista de la carrera de Ingeniería de Ejecución en Computación e Informática, no se contempla contratación de terceros ni supone un costo para el proyecto.

5.4 Conclusión estudio de factibilidad

De los puntos anteriores de factibilidad técnica, operativa y económica se puede concluir un índice de factibilidad positivo si consideramos que no requiere inversiones por parte del desarrollador, ni inversiones necesarias en software o equipos y significa un beneficio para la institución y los usuarios finales a costo cero. Alcanzará a la mayor parte de la población gracias a la elección de la plataforma en la que funcionará la aplicación y los requerimientos están, a la fecha actual, muy por debajo de la media presente en los dispositivos móviles utilizados por la población en general.

CAPÍTULO 6 – ANÁLISIS

En este capítulo se muestran los diagramas de casos de uso que representan la forma que tendrán los usuarios de interactuar con el sistema.

6.1 Diagrama de Casos de uso

6.1.1 Actores

Usuario	
Rol	Persona que utiliza la aplicación móvil para obtener información desde su Smartphone.
Nivel de conocimientos técnicos requeridos	Utilización de dispositivos móviles Android.
Nivel de privilegio en la aplicación	Ver la información disponible de los lugares detectados por medio de la cámara de su dispositivo y los detalles de esta, ver su ubicación dentro del mapa.
Cámara del dispositivo	
Rol	Hardware del dispositivo móvil encargado de capturar la imagen del mundo real, enviarla al sistema para indicar hacia dónde apunta el dispositivo con ella.
GPS del dispositivo	
Rol	Hardware encargado de capturar las coordenadas de posicionamiento global del dispositivo móvil (latitud, longitud).

*Tabla 16: Descripción de actores – Casos de uso.
Fuente: Elaboración propia.*

6.1.2 Casos de uso

Se representan las funcionalidades de la aplicación y su interacción con los actores en la siguiente figura.

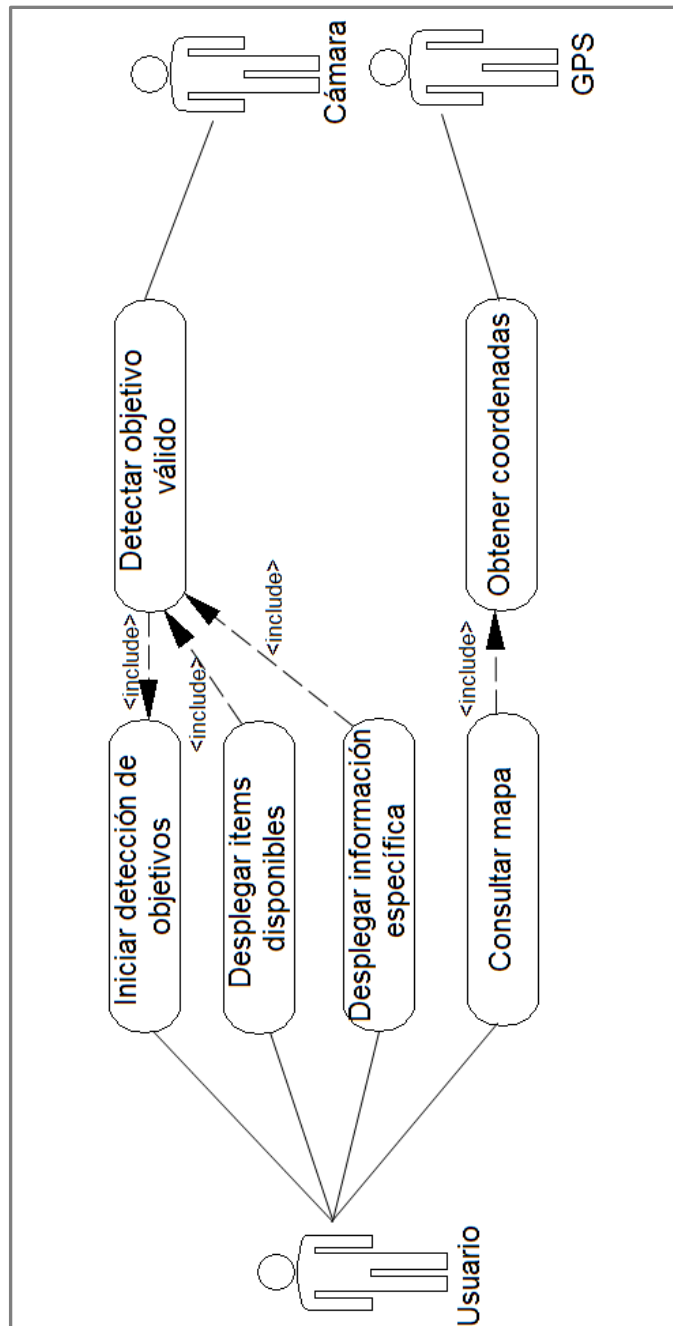


Figura 19: Diagrama de Casos de uso.
Fuente: Elaboración propia.

6.1.3 Descripción general

De la figura anterior se pueden identificar claramente tres actores que interactúan con el sistema, el primer actor “Usuario” corresponde a la persona natural que usa el dispositivo y por tanto, la aplicación móvil. Este puede consultar el mapa y si está disponible, su ubicación dentro de él mediante coordenadas GPS. También puede iniciar la detección de objetivos o en palabras simples iniciar el funcionamiento más importante de la aplicación que consiste en indicar al sistema que muestre un aviso cuando se esté en presencia de algún objetivo válido, tras ello el usuario puede desplegar los ítems del objetivo encontrado y su información específica.

El GPS debe obtener las coordenadas de posicionamiento global cuando sea requerido y la Cámara es la encargada de interactuar con la aplicación cada vez que encuentre un objetivo válido.

6.1.4 Especificación de casos de uso.

[CU01] Caso de uso: Iniciar detección de objetivos:

- **Descripción:** Indica al sistema que inicie la funcionalidad para detectar imágenes desde la cámara.
- **Actor:** Usuario.
- **Pre-Condiciones:**
 - El dispositivo debe contar con una cámara posterior operativa.

▪ **Flujo de eventos básico (paso 1):**

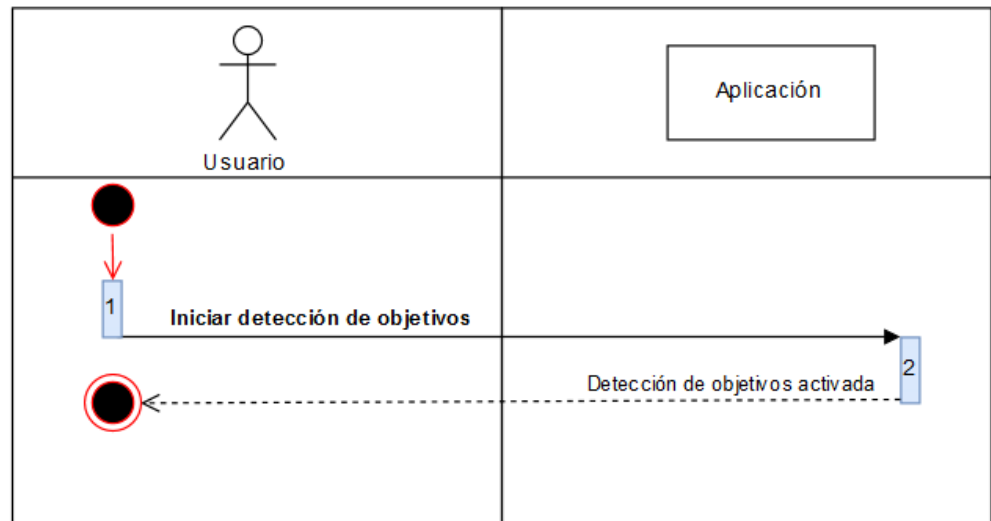


Figura 20: CU01 Flujo eventos básico.
Fuente: Elaboración propia.

▪ **Flujo de eventos alternativo:**

La única manera de no poder cumplir con el flujo de eventos básicos es que la aplicación no se encuentre activa, por lo tanto, utilizando la aplicación siempre se cumplirá el flujo anterior.

▪ **Post-Condiciones:**

- En la pantalla se muestra las imágenes obtenidas por la cámara en tiempo real, en caso de detectar un objetivo se sobrepondrá el nombre del lugar en cuestión.

[CU02] Caso de uso: Desplegar ítems disponibles:

- **Descripción:** Se muestra en la pantalla del dispositivo la lista de ítems que se encuentran disponibles para el objetivo encontrado por la cámara.
- **Actor:** Usuario.
- **Pre-Condiciones:**
 - El dispositivo debe contar con una cámara posterior operativa.
 - El dispositivo debe haber detectado el objetivo y no perderlo para completar este funcionamiento.
- **Flujo de eventos básico (paso 4):**

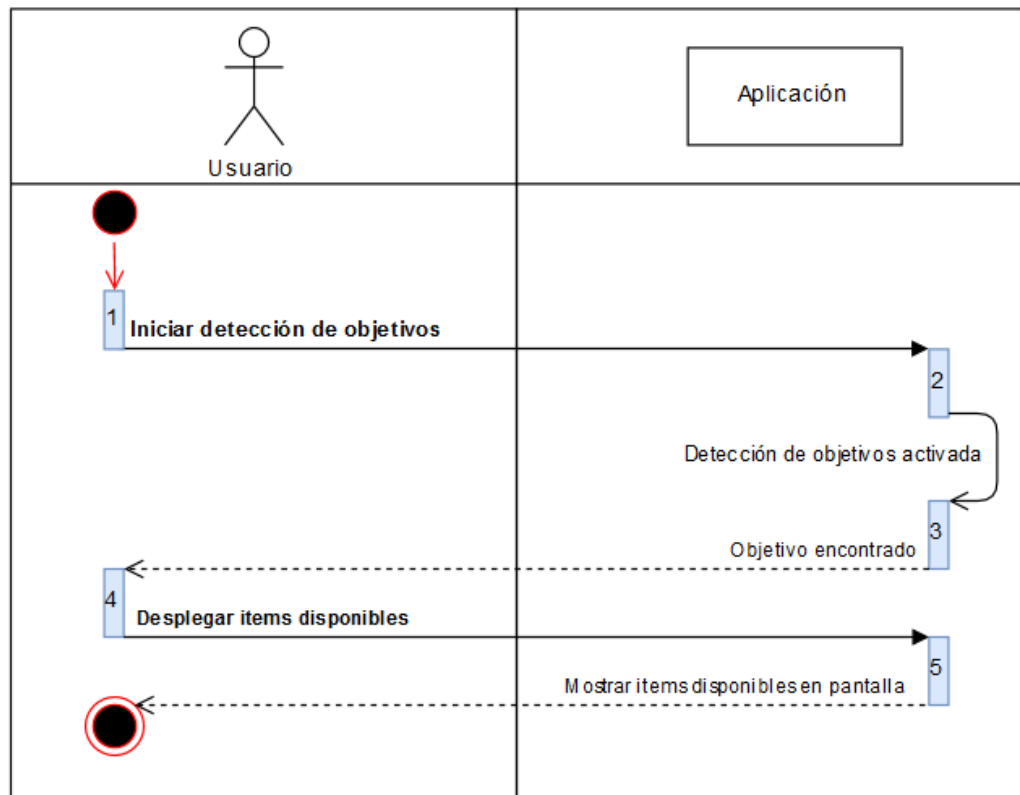


Figura 21: CU02 Flujo eventos básico.
Fuente: Elaboración propia.

▪ **Flujo de eventos alternativo (paso 6):**

Es posible en muchos casos que antes de desplegar los ítems disponibles para el objetivo al cual se está enfocando este se pierda de vista para la cámara, principalmente a causa de que el usuario deje de apuntar en esa dirección con el dispositivo móvil, por esto la cámara perderá el objetivo y deberá detectarlo nuevamente para poder seguir con el funcionamiento básico (fases 3 y 4 de la figura expuesta a continuación), esto puede darse una o más veces hasta que el usuario decida desplegar los ítems pulsando el nombre del lugar que aparecerá en pantalla sin dejar de enfocarlo con la cámara posterior.

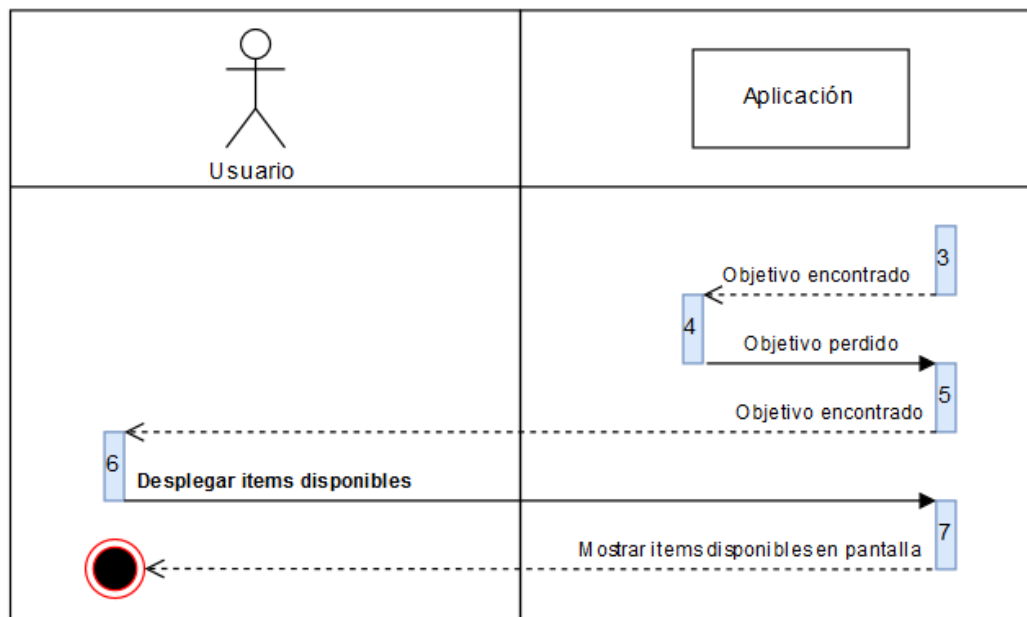


Figura 22: CU02 Flujo eventos alternativo.
Fuente: Elaboración propia.

▪ **Post-Condiciones:**

- En la pantalla se muestra las imágenes obtenidas por la cámara en tiempo real y sobre estas se muestra el listado de ítems disponibles para ver en detalle si se desea.

[CU03] Caso de uso: Desplegar información específica:

- **Descripción:** Se muestra en la pantalla del dispositivo la información de un ítem específico seleccionado sobre un objetivo detectado.

- **Actor:** Usuario.

- **Pre-Condiciones:**
 - El dispositivo debe contar con una cámara posterior operativa.
 - El dispositivo debe haber detectado el objetivo y no perderlo para completar este funcionamiento.
 - Se debe haber desplegado la lista de ítems disponibles y seleccionado uno de ellos para mostrar el detalle de la información.

▪ **Flujo de eventos básico (paso 6):**

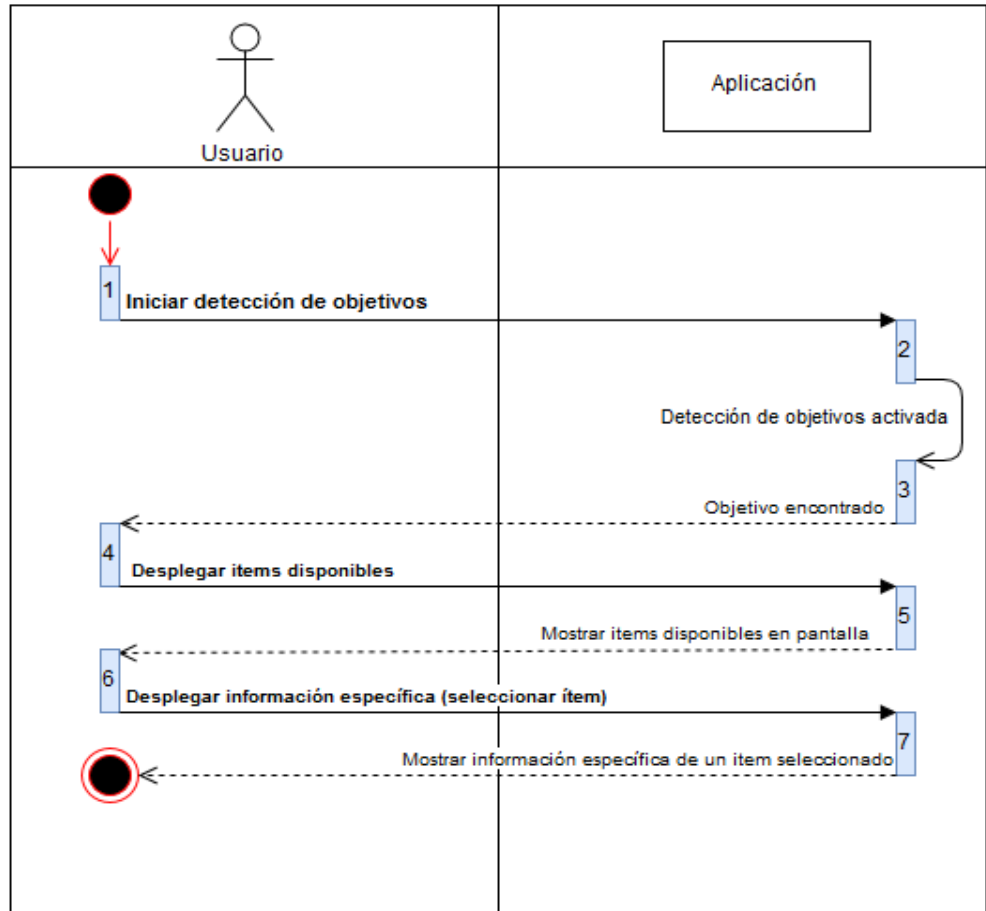


Figura 23: CU03 Flujo eventos básico.
Fuente: Elaboración propia.

▪ **Flujo de eventos alternativo:**

Seleccionar un ítem para mostrar su información específica no posee un flujo alternativo, perder el objetivo con en CU02 no significa perder de vista la información ni lo que se esté mostrando en pantalla en CU03, puesto que muchos de los ítems son desarrollados en textos de mediano y gran tamaño que muchas veces deben seguir disponibles para que el usuario pueda leerlos sin

necesidad de tener levantado el dispositivo en dirección al objetivo mientras lee.

En palabras simples, CU02 perdía la información mostrada en pantalla si el objetivo era perdido por la cámara, CU03 sigue mostrando la información aunque el objetivo ya no se esté enfocando.

▪ **Post-Condiciones:**

- En la pantalla se muestra las imágenes obtenidas por la cámara en tiempo real y sobre estas se muestra la información detallada de uno de los ítems disponibles seleccionados para un objetivo válido sin importar si el usuario sigue o no enfocando el objetivo.

[CU04] Caso de uso: Consultar mapa:

- **Descripción:** Se muestra en la pantalla un mapa del campus correspondiente al sector en el cual opera la aplicación.
- **Actor:** Usuario.
- **Pre-Condiciones:** No posee pre-condiciones.

- **Flujo de eventos básico:**

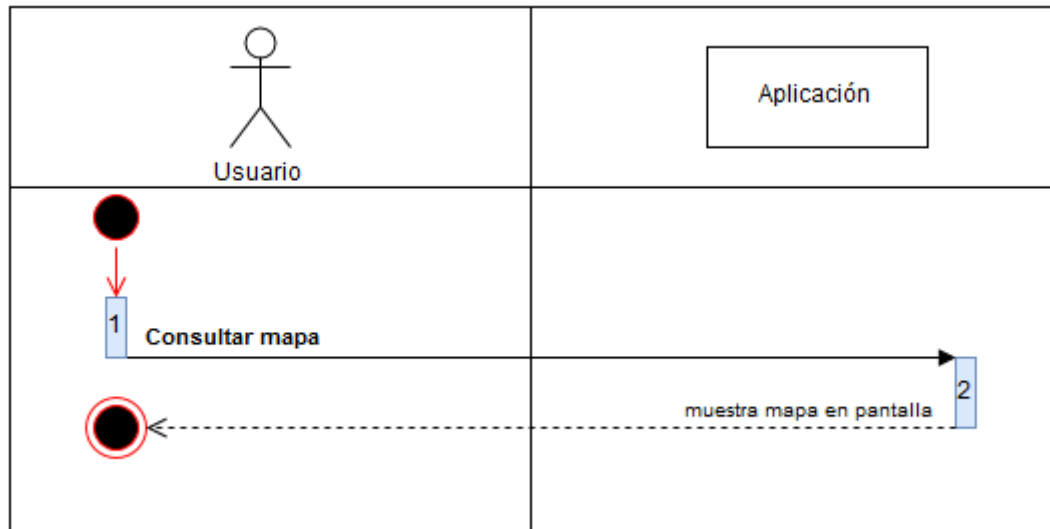


Figura 24: CU04 Flujo eventos básico.
Fuente: Elaboración propia.

- **Flujo de eventos alternativo:**

No posee flujo de eventos alternativo, pero sí se deben cumplir las condiciones indicadas en el punto que continúa.

- **Post-Condiciones:**

El mapa siempre se mostrará en el momento en que el usuario lo consulte, pero el producto final siempre será sólo uno de los puntos detallados a continuación:

- Se muestra el mapa junto a un mensaje de GPS no disponible si el GPS se encuentra desactivado.
- Sólo se muestra el mapa si el usuario se encuentra fuera de los límites del mapa establecido y el GPS está activado.
- Se muestra el mapa y un ícono de ubicación dentro de él si el usuario se encuentra dentro de los límites del mapa establecido con el GPS activado.

[CU05] Caso de uso: Detectar objetivo válido:

- **Descripción:** La cámara indica al sistema cuándo se está en presencia de un objetivo válido.
- **Actor:** Cámara del dispositivo móvil.
- **Pre-Condiciones:**
 - El dispositivo debe contar con una cámara posterior operativa.

- **Flujo de eventos básico:**

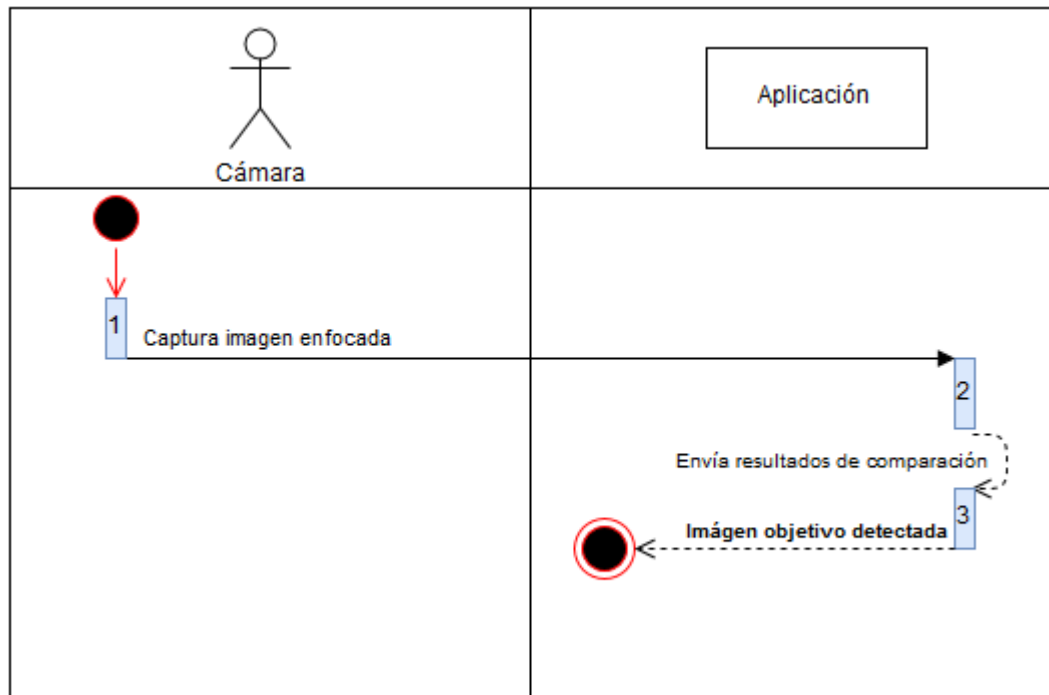


Figura 25: CU05 Flujo eventos básico.
Fuente: Elaboración propia.

- **Flujo de eventos alternativo:**

No posee flujos alternativos.

- **Post-Condiciones:**

- El sistema recibe las imágenes capturadas por la cámara en tiempo real y puede detectar cuáles de ellas corresponden a objetivos válidos.

[CU06] Caso de uso: Obtener coordenadas:

- **Descripción:** El GPS recibe las coordenadas de posicionamiento global y las indica en el sistema.
- **Actor:** Sistema de posicionamiento global, GPS del dispositivo móvil.
- **Pre-Condiciones:**
 - El dispositivo debe contar con un GPS operativo.
 - El GPS debe estar encendido.
- **Flujo de eventos básico:**

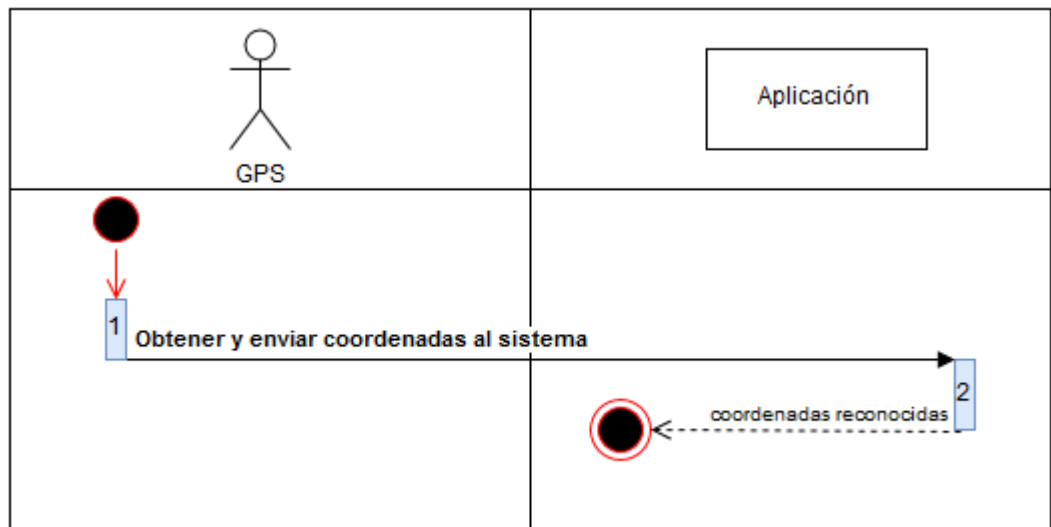


Figura 26: CU06 Flujo eventos básico.
Fuente: Elaboración propia.

- **Flujo de eventos alternativo:**

No posee flujos alternativos.

- **Post-Condiciones:**

- Las coordenadas latitud y longitud son reconocidas por la aplicación para la ubicación actual del usuario.

CAPÍTULO 7 – DISEÑO

En este capítulo se presentan la arquitectura funcional, diseño de interfaces, diseño de elementos visuales y de navegación presente en la aplicación, iconografía, logos y algunas vistas de su funcionamiento.

7.1 Diseño de arquitectura funcional

En el punto actual se plantean módulos y procesos de forma general utilizando diagrama de descomposición funcional.

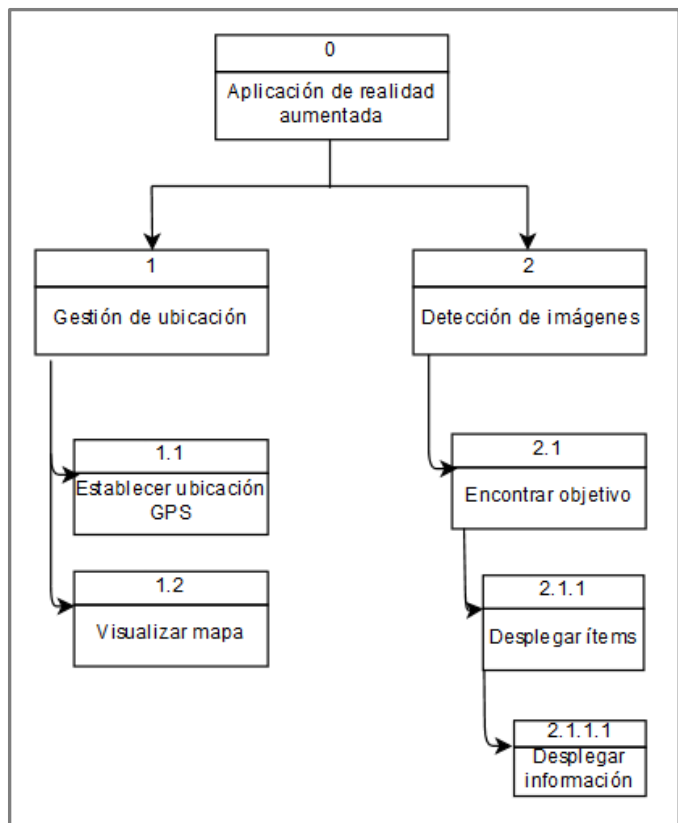


Figura 27: Diagrama de descomposición funcional – Vista general – Arquitectura funcional.
Fuente: Elaboración propia.

El siguiente diagrama de eventos grafica el diseño general de la funcionalidad de la aplicación y las distintas bifurcaciones posibles en cada evento.

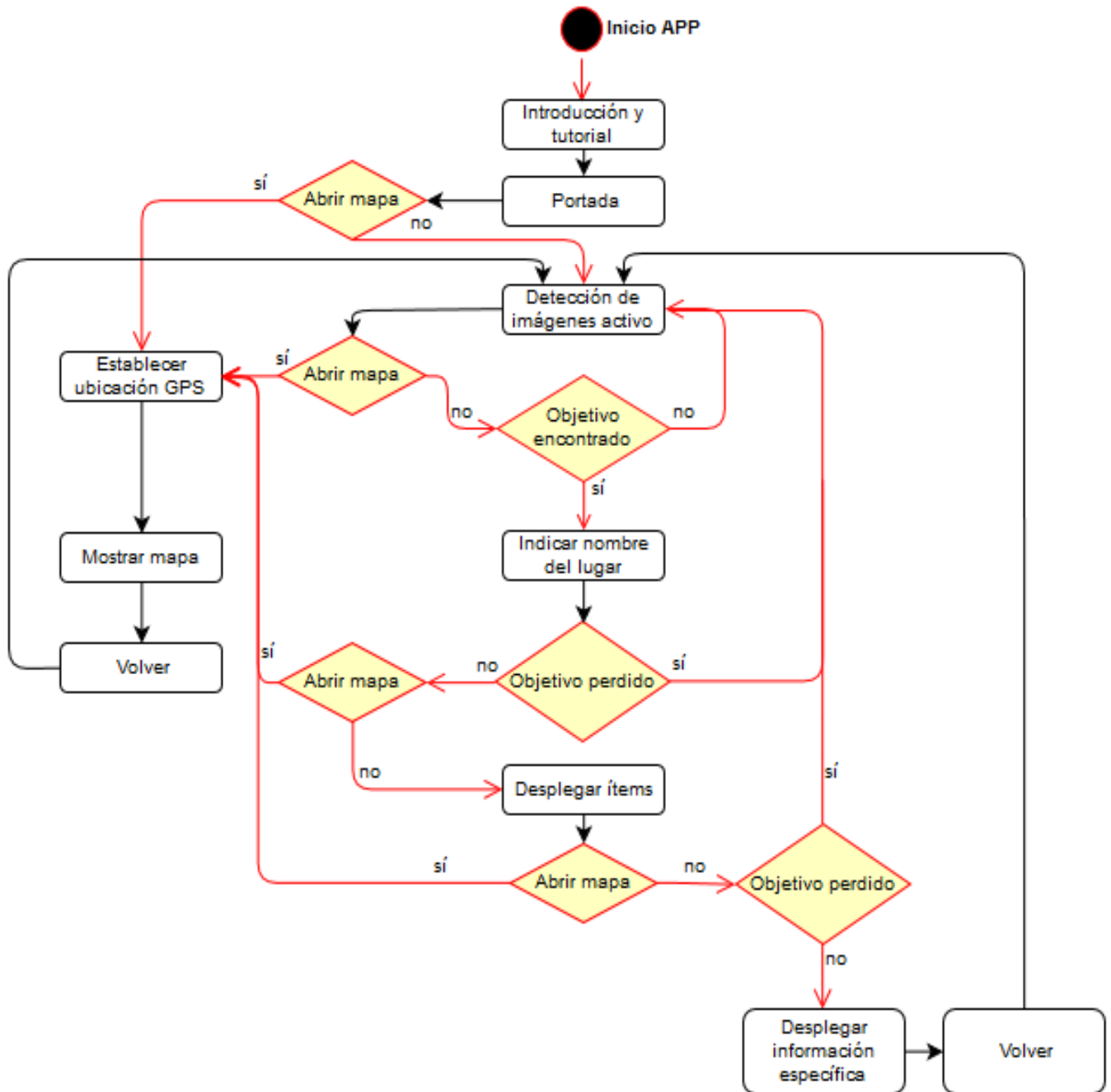


Figura 28: Diagrama de eventos – Arquitectura funcional.
Fuente: Elaboración propia.

7.2 Diseño de interfaz y navegación

Se muestra en este capítulo la interfaz presente en la aplicación considerando posicionamiento de los elementos, iconografías y elementos de navegación.

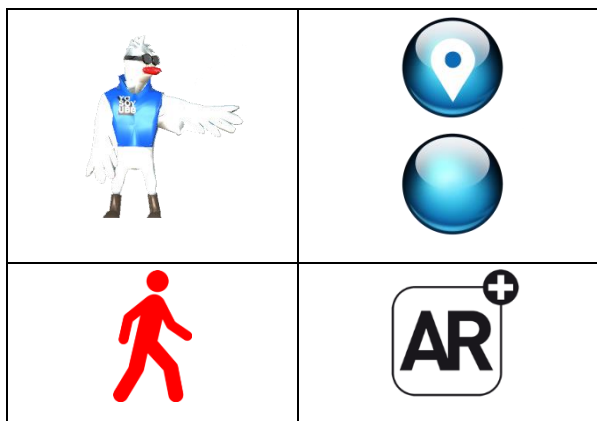
7.3 Iconografía

- Logo de la aplicación e imagen de presentación:



*Figura 29: Logo de la aplicación. Figura 30: Imagen de la aplicación.
Fuente: Elaboración propia*

- Figuras utilizadas:



*Figura 31: Figuras de la aplicación.
Fuente: Elaboración propia.*

- Elementos de navegación:



Figura 32: Esquema de navegación 1.
Fuente: Elaboración propia.

1. Bienvenida al abrir la aplicación, panel desplegable que puede cerrarse al presionar el botón “Entendido”.
2. Indicador de objetivo encontrado o perdido.
3. Botón para abrir el mapa.
4. Indicador del nombre del lugar encontrado, puede pulsarse para abrir la siguiente vista:



Figura 33: Esquema de navegación 2.
Fuente: Elaboración propia.

1. Animación que se posa sobre las estructuras de la realidad capturada por la cámara, puede ver su modelo 3D al enfocar desde distintos ángulos.
2. Botón para ocultar botones de ítems y volver a la vista anterior.
3. Botones de ítems disponibles correspondientes al lugar enfocado, pueden pulsarse para abrir la siguiente vista:

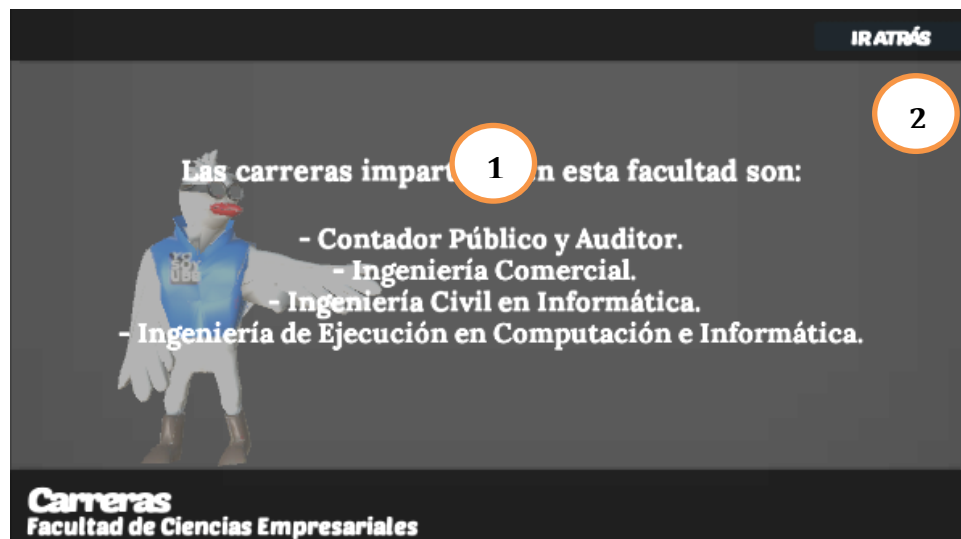


Figura 34: Esquema de navegación 3.
Fuente: Elaboración propia.

1. Información mostrada correspondiente al ítem seleccionado en la vista anterior.
2. Botón para volver a la vista anterior.

7.4 Especificación de módulos

M01	Módulo: Mostrar mapa		
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo	Nombre	Tipo
		Cambio de escena	Int

Tabla 17: M01. Mostrar mapa – Especificación de módulos.
Fuente: Elaboración propia.

M02	Módulo: Consultar ubicación		
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo	Nombre	Tipo
		Latitud	Float
		Longitud	Float

Tabla 18: M02. Consultar ubicación – Especificación de módulos.
Fuente: Elaboración propia.

M03	Módulo: Consultar estado de sensores (Wifi, GPS)		
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo	Nombre	Tipo
		Disponibilidad	Boolean

Tabla 19: M03. Consultar estado de sensores – Especificación de módulos.
Fuente: Elaboración propia.

M04	Módulo: Desplegar información en pantalla		
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo	Nombre	Tipo
Acción (Botón)	Boolean	Información	Text (Plane - Unity)

*Tabla 20: M04. Desplegar información en pantalla – Especificación de módulos.
Fuente: Elaboración propia.*

M05	Módulo: Ocultar ítems		
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo	Nombre	Tipo
Acción (Botón)	Boolean	Reacción	Boolean

*Tabla 21: M05. Ocultar ítems – Especificación de módulos.
Fuente: Elaboración propia.*

M06	Módulo: Cambiar vista Autoridades		
Parámetros de entrada		Parámetros de salida	
Nombre	Tipo	Nombre	Tipo
Acción (Botón)	Int	Reacción	Boolean

*Tabla 22: M06. Cambiar vista Autoridades – Especificación de módulos.
Fuente: Elaboración propia.*

CAPÍTULO 8 – PRUEBAS

Adaptación basada en IEEE Software Test Documentation Std 829-1998.

8.1 Introducción

En el presente capítulo se describen los casos de pruebas elaborados para examinar posibles fallos en la aplicación, los elementos han sido creados arbitrariamente en base a una estimación de los problemas de mayor relevancia que podrían afectar el funcionamiento del software.

8.2 Elementos de pruebas

1. **Nombre:** Funcionamiento de interfaz gráfica en dispositivos con pantallas de 4.5, 5 y 5.5 pulgadas (las más comunes).

Descripción: Se comprueba que los elementos gráficos se ajusten correctamente a las resoluciones de pantallas desde 4.5 a 5.5 pulgadas que corresponden a las más comunes del mercado.

2. **Nombre:** Comunicación con dispositivo GPS.

Descripción: Se comprueba que el dispositivo GPS pueda enviar las coordenadas de ubicación a la aplicación.

3. **Nombre:** Detección de objetivos.

Descripción: Se comprueba que la aplicación pueda detectar correctamente los objetivos incorporados en la base de datos de targets de Vuforia.

4. **Nombre:** Funcionamiento de interfaz en diferentes estados.

Descripción: Se comprueba que la interfaz realice la correcta transición de elementos al cambiar de estados descritos en el punto 7.1, entre ellos: detectar objetivo, perder objetivo, desplegar ítems, desplegar información.

5. **Nombre:** Transición de escenas.

Descripción: Las diferentes escenas dispuestas para el funcionamiento de la navegación funcionan correctamente (el desplazamiento mediante navegación a través de la aplicación utilizando la interfaz gráfica funciona correctamente).

6. **Nombre:** El ícono de ubicación GPS se desplaza correctamente.

Descripción: Se comprueba que el ícono de ubicación sobre el mapa se mueva como es debido cuando el GPS ha detectado la ubicación y movimiento dentro del área establecida.

7. **Nombre:** Visualización de información a ciertas distancias del objetivo.

Descripción: Se comprueba que lo mostrado en la pantalla del dispositivo dependiente de la orientación del móvil hacia un objetivo en particular se mantenga visible a una distancia mínima y máxima establecida.

8.3 Especificación de pruebas

ID	Característica	Nivel de prueba	Objetivo	Enfoque
P1	Interfaz	Unidad	Funcionamiento de interfaz en diferentes resoluciones.	Caja negra
P2	Funcionalidad		Comunicación con GPS.	
P3	Funcionalidad		Detección de objetivos.	
P4	Interfaz		Interfaz correcta en diferentes fases del funcionamiento.	
P5	Funcionalidad		Transición de vistas de la aplicación mediante interfaz.	
P6	Funcionalidad		Movimiento ícono de ubicación.	
P7	Funcionalidad		Funcionamiento a ciertas distancias del objetivo.	

*Tabla 23: Especificación de pruebas.
Fuente: Elaboración propia.*

ID	Técnica de definición de casos de prueba	Actividades	Criterios de cumplimiento
P1	Instalación de aplicación en diferentes dispositivos móviles.	<ol style="list-style-type: none"> 1. Instalar y comprobar el funcionamiento de la aplicación en un dispositivo con pantalla de 4.5 pulgadas. 2. Instalar y comprobar el funcionamiento de la aplicación en un dispositivo con pantalla de 5 pulgadas. 3. Instalar y comprobar el funcionamiento de la aplicación en un dispositivo con pantalla de 5.5 pulgadas. 	<p>La interfaz debe mantener su aspecto y distribución en cada una de las vistas de la aplicación de forma correcta, permitiendo el uso adecuado y la visualización de todas las características.</p>
P2	Consultar el mapa con el GPS activado y desactivado.	<p>Situación A: Abrir el mapa de la aplicación con el GPS activado, luego desactivarlo.</p> <p>Situación B: Abrir el mapa de la aplicación con el GPS desactivado, luego activarlo.</p> <ol style="list-style-type: none"> 1. Ejecutar (A) dentro del área marcada en el mapa. 2. Ejecutar (A) fuera del área marcada en el mapa. 3. Ejecutar (B) dentro del área marcada en el mapa. 4. Ejecutar (B) fuera del área 	<p>Al estar dentro del área del mapa, si el GPS está activado debe mostrar el ícono de ubicación, si el GPS está desactivado se debe indicar el estado de este.</p> <p>Al estar fuera del área se debe</p>

		marcada en el mapa.	indicar que el usuario se encuentra fuera de los rangos esté o no el GPS activado.
P3	Orientación del móvil a lugares aleatorios, algunos pertenecientes a objetivos válidos.	Con la aplicación en funcionamiento: <ol style="list-style-type: none"> 1. Apuntar el móvil hacia un objetivo válido. 2. Apuntar el móvil hacia cualquier objetivo no válido. 	Cuando el objetivo sea válido (1) (se detecte que corresponde a un objetivo en la base de datos de targets de Vuforia) se debe mostrar el nombre del lugar, para (2) la pantalla debe seguir igual (sin cambios).
P4	Orientación del móvil a objetivos válidos y navegación por los botones y gráficos desplegados por los objetivos reconocidos.	Con la aplicación en funcionamiento: <ol style="list-style-type: none"> 1. Apuntar el móvil hacia un objetivo válido. 2. Desplegar ítems (Seleccionar con el dedo sobre el nombre del lugar desplegado). 3. Desplegar información (Seleccionar los ítems para ver la información detallada). 	La interfaz de navegación posee un funcionamiento adecuado según lo detallado en el punto 7.1.

		<p>4. Perder objetivo (dejar de apuntar) entre la fase (2) y (3).</p> <p>5. Perder el objetivo después de la fase (3).</p>	
P5	Navegación en la aplicación.	Navegar por la aplicación utilizando la interfaz.	La interfaz de navegación se comporta adecuadamente.
P6	Movimiento del dispositivo.	Durante el punto [1] detallado en [P4], desplazarse físicamente por los límites mostrados por el mapa.	El ícono de ubicación se mueve correctamente.
P7	Enfoque de objetivos a diferentes distancias.	<p>Con la aplicación en funcionamiento y mostrando que se ha detectado un objetivo válido:</p> <p>1. Apuntar dispositivo al objetivo a diferentes distancias (alejar, acercar).</p>	El objetivo sigue siendo detectado dentro de los rangos de distancia establecidos.

Tabla 24: Actividades de pruebas.

Fuente: Elaboración propia.

8.4 Responsables

Es necesario señalar que el responsable de las pruebas corresponde a la persona desarrolladora de la aplicación, no se han dividido las tareas a terceros.

8.5 Calendario de pruebas

ID	Fecha	Duración
P1	01 de diciembre de 2016	1 día
P2	04 de diciembre de 2016	1 día
P3	06 de diciembre de 2016	1/2 día
P4	06 de diciembre de 2016	
P5	08 de diciembre de 2016	1 día
P6	06 de diciembre de 2016	1/2 día
P7	09 de diciembre de 2016	1 día

*Tabla 25: Calendario de pruebas.
Fuente: Elaboración propia.*

8.6 Detalle de pruebas

El detalle de las pruebas realizadas se ha incorporado en el Anexo 3 de este documento.

8.7 Conclusión de pruebas

La realización de este proceso fundamental para el desarrollo de cualquier software, en este caso en particular se han obtenido resultados satisfactorios y por ello se ha verificado el correcto funcionamiento para los escenarios planteados, aunque existe un periodo de pruebas en el cual se debe monitorear el buen funcionamiento de la aplicación en la totalidad de sus características.

CAPÍTULO 9 – PLAN DE IMPLEMENTACIÓN

En el punto 5.3.2 del presente documento se menciona la contemplación de dos alternativas a utilizar para llevar la aplicación a los dispositivos de los usuarios finales. La implementación y puesta en marcha constará de ciertos pasos básicos para hacer efectivo este proceso.

Al utilizar ambas alternativas se debe considerar los siguientes pasos:

- Descarga de la aplicación a través de portales web de la institución:
 1. Recepción y validación del funcionamiento de la aplicación por parte de personal capacitado de la universidad.
 2. Elaboración de una técnica efectiva de promoción de la aplicación en los portales correspondientes.
 3. Poner el instalador de la aplicación a disposición del público para ser descargada.
 4. [Usuario] El usuario debe tener en cuenta los requisitos mínimos de la aplicación, de cumplirlos, debe abrir el instalador en su dispositivo móvil y asignar los permisos necesarios en la etapa de instalación.
- Descarga de la aplicación a través de la tienda virtual Google Play de Android:
 1. Convertir una cuenta de google en cuenta de desarrollador según lo mencionado en el punto 5.3.2.

2. Visitar la consola de desarrollador de Google Play en play.google.com/apps/publish/signup/ y agregar la nueva aplicación.

2.1. Seleccionar el idioma predeterminado y agregar un título para la aplicación que será con el cual aparecerá en la tienda virtual.

2.2. Cargar archivo APK (instalador).

2.3. Elegir el canal de producción (Alfa o Beta – etapas de lanzamiento).

2.4. Publicar.

3. [Usuario] Luego de esto el usuario puede buscar la aplicación directamente en sus móviles a través de Google Play e instalarla pulsando el botón “instalar” de la interfaz de la tienda.

- En ambos casos se puede monitorear las cantidades de descargas. En el caso de Google Play se dispone de un sistema de valoraciones públicas y comentarios.

CAPÍTULO 10 – RESUMEN DE ESFUERZO REQUERIDO

Se puede ver a continuación las horas de trabajo destinadas a cada una de las tareas del proyecto.

ID	Actividad	Nº de Horas
Ac1	Reuniones con cliente	06
Ac2	Investigación previa y estado del arte	40
Ac3	Investigación de metodología	14
Ac4	Estudio de herramientas y métodos de desarrollo	150
Ac5	Estudio de lenguajes y programación de funciones	520
Ac6	Diseño y modelado de interfaces	40
Ac7	Preparación de informe	100
Total		870 horas.

*Tabla 26: Horas Esfuerzo requerido.
Fuente: Elaboración propia.*

En complemento se puede analizar el cálculo de estimación de horas hombre requeridas determinado por la técnica de Puntos de casos de uso en el Anexo 1 Planificación inicial del proyecto, punto A1.1.

CAPÍTULO 11 – CONCLUSIONES

Según lo analizado a partir de la problemática planteada, los objetivos del proyecto y del software, se ha desarrollado una aplicación que cumple con lo planteado en sus inicios que es capaz de entregar información específica a los usuarios, relacionada a lugares específicos, de una manera sencilla, intuitiva, atractiva y eficaz.

Durante el periodo de estudios en la carrera de Ingeniería de Ejecución en Computación e Informática se pudo adquirir, tanto conocimiento como experiencias, dominio de lenguajes de programación, técnicas y herramientas que permiten el desarrollo de esta aplicación móvil, motivado en todo momento a encontrar la forma más eficiente de converger lo aprendido hacia un producto final de utilidad para sus usuarios resolviendo con innovación e ingenio la problemática latente.

Para completar el proyecto se debió idear técnicas para resolver variadas situaciones de complejidad presentadas durante el desarrollo, investigaciones de programación y utilización de las herramientas de apoyo e implementación de componentes implicados en la arquitectura del software, entre otras actividades necesarias para el cumplimiento de las tareas involucradas, algunas más complejas que otras, pero que conducen finalmente a conocer un universo de posibilidades de desarrollo y utilidades en lo que respecta a la realidad aumentada en la actualidad, aprender sobre el tema y crear nuevo material sobre esto comprende un gran crecimiento profesional y de conocimiento.

La metodología implementada y la retroalimentación significaron en varias oportunidades a replantear ideas y solucionar nuevas problemáticas, lo más complejo fue el manejo de las sombras en la detección de imágenes y las distancias entre los objetivos capturados en relación a los planos virtuales proyectados, complicaciones comunes al trabajar

con realidad aumentada pero que ayudaron a una orientación positiva en busca del producto final adecuado.

Personalmente, haber concluido un periodo de estudios en la Universidad del Bío-Bío significa una muy buena experiencia, que deja una huella de gran valor y que ha entregado lo necesario para afrontar este desafío con motivación y deseos de superación.

REFERENCIAS

1. Universidad del Bío-Bío. Estructura orgánica de la universidad. [en línea]
< http://www.ubiobio.cl/miweb/web2012.php?id_pagina=5152 >
[Consulta: 16 de Agosto de 2016].
2. Universidad del Bío-Bío. Información de Facultades y departamentos. [en línea]
< <http://www.ubiobio.cl> > [Consulta: Segundo semestre 2016].
3. Augmented Reality: A class of displays on the reality-virtuality continuum.
Paul Milgram, Haruo Takemura, Akira Utsumi, Fumio Kishino. [Documento en línea]
< http://etclab.mie.utoronto.ca/publication/1994/Milgram_Takemura_SPIE1994.pdf >
[Consulta: 25 de Agosto de 2016].
4. Manual de desarrollo Unity [en línea]
< <http://docs.unity3d.com/es/current/Manual/UnityManual.html> >
[Consulta: Segundo semestre 2016].
5. Manual de referencia de Blender en español [en línea]
< <http://www.blender.org/manual/es/contents.html> >
[Consulta: Segundo semestre 2016].
6. Guías de desarrollo Vuforia [en línea]
< <http://library.vuforia.com/guide> > [Consulta: Segundo semestre 2016].
7. Documentación Unity3D creación y uso de Scripts. [en línea]
< <http://docs.unity3d.com/Manual/CreatingAndUsingScripts.html> >
[Consulta: Segundo semestre 2016].
8. Manual Programación Android V2 – Salvador Gómez Oliver – Noviembre 2011.

9. Windows. Características de Windows Phone [en línea]
< http://www.microsoft.com/es-es/windows/features?OCID=win10_null_vanity_win10features >
[Consulta: Septiembre de 2016].

10. Apple. iPhone [en línea] < <http://www.apple.com/iphone/> >
[Consulta: Septiembre de 2016].

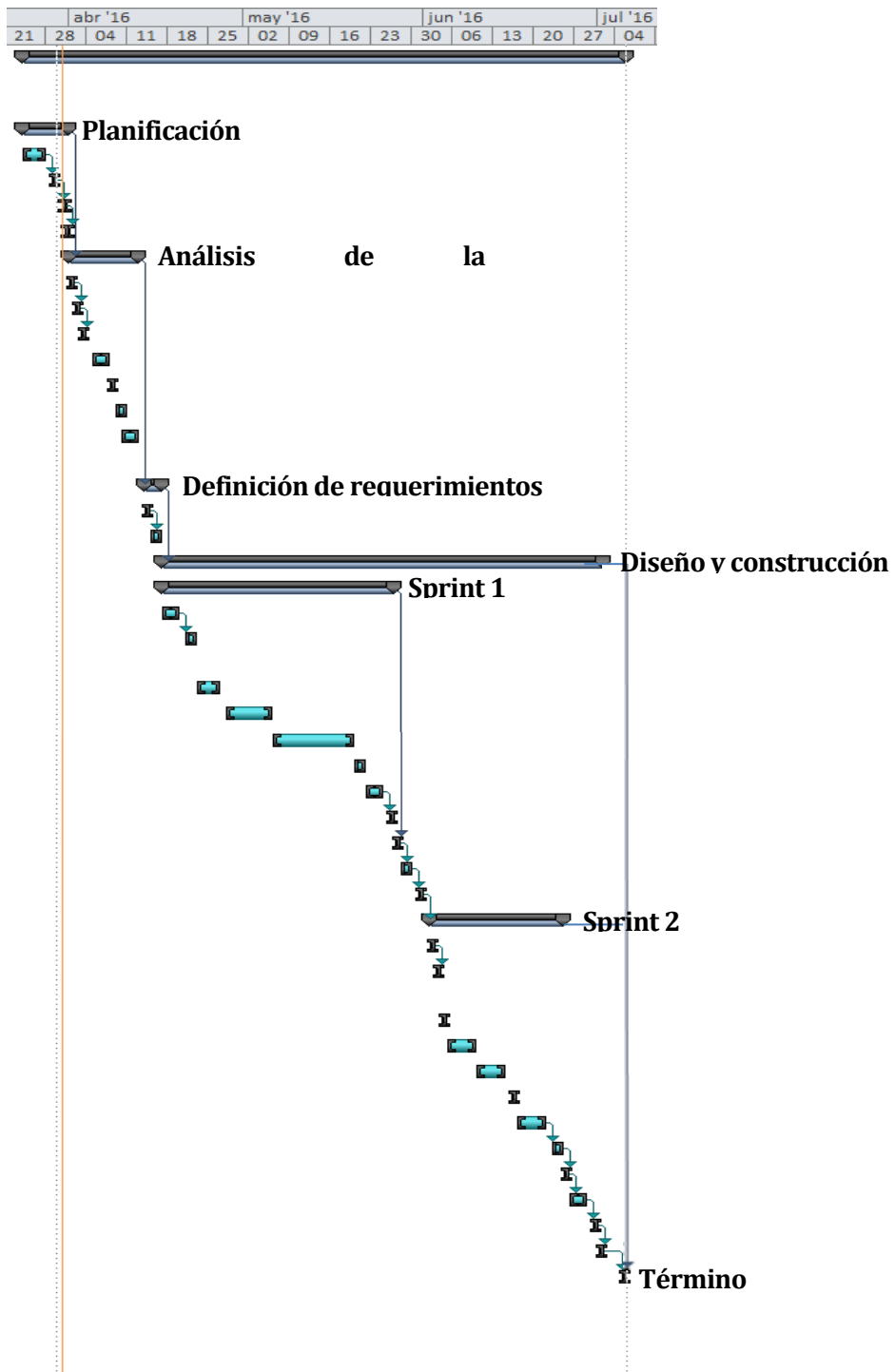
11. Android. Android SDK [en línea]
< <http://developer.android.com/studio/index.html#ExistingIDE> >
[Consulta: Octubre de 2016].

12. Android Developers. Frequently Asked Questions [en línea]
< <http://source.android.com/source/faqs.html> >
[Consulta: Segundo semestre 2016].

ANEXO 1 – PLANIFICACION INICIAL DEL PROYECTO

El plan de trabajo se ajusta a un horario de 5 días semanales incluyendo sábados y domingos pero sustituyendo estos días por lunes y miércoles como días no disponibles por motivos de estudios.

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
Recorrido virtual de Realidad Aumentada sobre sistema android a través de las dependencias y lugares de interés de la Universidad del Bío-Bío campus Concepción.	75 días	jue 24-03-16	mar 05-07-16	
Planificación	6 días	jue 24-03-16	jue 31-03-16	
Elaboración propuesta de proyecto	4 días	jue 24-03-16	dom 27-03-16	
Estudio de herramientas de trabajo	1 día	mar 29-03-16	mar 29-03-16	3
Entrega de propuesta y planificación global al cliente	0,5 días	jue 31-03-16	jue 31-03-16	4
Presentación de la propuesta	0,5 días	jue 31-03-16	jue 31-03-16	5
Análisis de la problemática planteada	9 días	vie 01-04-16	mar 12-04-16	2
Descripción de la aplicación	1 día	vie 01-04-16	vie 01-04-16	
Delimitación de alcances del sistema	1 día	sáb 02-04-16	sáb 02-04-16	8
Propuesta inicial de requerimientos	1 día	dom 03-04-16	dom 03-04-16	9
Identificación del entorno tecnológico	2 días	mar 05-04-16	jue 07-04-16	
Identificación de los usuarios e interfaces	1 día	vie 08-04-16	vie 08-04-16	
Análisis y especificación de lugares de interés a incluir	2 días	sáb 09-04-16	dom 10-04-16	
Estudio de métodos eficientes para el hosting de bases de datos y recursos de la aplicación	2 días	dom 10-04-16	mar 12-04-16	
Definición de requerimientos	3 días	jue 14-04-16	sáb 16-04-16	7
Definición y análisis de requerimientos	1 día	jue 14-04-16	jue 14-04-16	
Especificación de requerimientos	2 días	vie 15-04-16	sáb 16-04-16	16
Diseño y construcción	54 días	dom 17-04-16	vie 01-07-16	15
Sprint 1	28 días	dom 17-04-16	jue 26-05-16	
Recopilación de fotografías de lugares seleccionados	2 días	dom 17-04-16	mar 19-04-16	
Importación de coordenadas e información a partir de fotografías obtenidas	2 días	jue 21-04-16	vie 22-04-16	20
Recopilación de información sobre lugares	3 días	sáb 23-04-16	mar 26-04-16	
Modelado y animación de personaje	6 días	jue 28-04-16	jue 05-05-16	
Diseño y programación de interfaces gráficas	10 días	vie 06-05-16	jue 19-05-16	
Diseño e Implementación de base de datos	2 días	vie 20-05-16	sáb 21-05-16	
Desarrollo prototipo funcional	2 días	dom 22-05-16	mar 24-05-16	
Pruebas	1 día	jue 26-05-16	jue 26-05-16	26
Integración	1 día	vie 27-05-16	vie 27-05-16	19
Documentación	2 días	sáb 28-05-16	dom 29-05-16	28
Reunión y restrospectiva	1 día	mar 31-05-16	mar 31-05-16	29
Sprint 2	17 días	jue 02-06-16	vie 24-06-16	30
Recopilación de fotografías de lugares seleccionados	1 día	jue 02-06-16	jue 02-06-16	
Importación de coordenadas e información a partir de fotografías obtenidas	1 día	vie 03-06-16	vie 03-06-16	32
Recopilación de información sobre lugares	1 día	sáb 04-06-16	sáb 04-06-16	
Modelado y animación de personaje	3 días	dom 05-06-16	jue 09-06-16	
Diseño y programación de interfaces gráficas	4 días	vie 10-06-16	mar 14-06-16	
Diseño e Implementación de base de datos	1 día	jue 16-06-16	jue 16-06-16	
Desarrollo prototipo funcional con interacciones de interfaz	4 días	vie 17-06-16	mar 21-06-16	
Pruebas	2 días	jue 23-06-16	vie 24-06-16	38
Integración	1 día	sáb 25-06-16	sáb 25-06-16	39
Documentación	2 días	dom 26-06-16	mar 28-06-16	40
Reunión y restrospectiva	1 día	jue 30-06-16	jue 30-06-16	41
Integración de aplicación en su versión final	1 día	vie 01-07-16	vie 01-07-16	42
Entrega	1 día	mar 05-07-16	mar 05-07-16	43



Figuras 35 y 36: Planificación del proyecto – Carta Gantt.
Fuente: Elaboración propia.

A1.1 Estimación inicial de tamaño

La medición del tamaño del trabajo a desarrollar se utiliza la estimación de Puntos de Casos de Uso.

A1.1.1 Clasificación de actores

Actor	Tipo	Factor
Usuario	Complejo	3
Cámara	Medio	2
Sensor GPS	Medio	2

*Tabla 27: Clasificación de actores – UAW.
Fuente: Elaboración propia.*

Cálculo del factor de peso de los actores sin ajustar UAW:

$$UAW = 3 + 2 + 2 = 7$$

A1.1.2 Clasificación de casos de uso

Caso de uso	Tipo	Factor
[CU01] Iniciar detección de objetivos	Simple	5
[CU02] Desplegar ítems disponibles	Simple	5
[CU03] Desplegar información	Medio	10

específica		
[CU04] Consultar mapa	Simple	5
[CU05] Detectar objetivo válido	Simple	5
[CU06] Obtener coordenadas	Simple	5

Tabla 28: Clasificación de casos de uso – UUCW.
Fuente: Elaboración propia.

Cálculo del factor de peso los casos de uso sin ajustar UUCW:

$$UUCW = 5 * 5 + 10 = 35$$

A1.1.3 Puntos de casos de uso sin ajustar

Para calcular los puntos de casos de uso sin ajustar debemos sumar UAW (A1.1.1) y UUCW (A1.1.2):

$$UUCP = UAW + UUCW = 7 + 35 = 42$$

A1.1.4 Factores técnicos relevantes

La complejidad técnica es dada por:

$$TCF = 0.6 + (0.01 * TFactor), \text{ donde:}$$

$$TFactor = \sum(Peso * Factor)$$

Asignación de peso a los distintos factores:

Factor	Descripción	Peso
F01	Sistema distribuido	2
F02	Objetivos de performance o tiempo de respuesta	1
F03	Eficiencia del usuario final	1
F04	Procesamiento interno complejo	1
F05	El código debe ser reutilizable	1
F06	Facilidad de instalación	0.5
F07	Facilidad de uso	0.5
F08	Portabilidad	2
F09	Facilidad de cambio	1
F10	Concurrencia	1
F11	Incluye objetivos especiales de seguridad	1
F12	Provee acceso directo a terceras partes	1
F13	Se requiere facilidades especiales de entrenamiento a usuarios	1

*Tabla 29: Peso de los factores técnicos - TCF.
Fuente: Elaboración propia.*

La tabla anterior se debe evaluar con la siguiente ponderación:

Descripción	Factor
Irrelevante	0 – 2
Medio	3 – 4
Esencial	5

*Tabla 30: Ponderación de factores - TCF.
Fuente: Elaboración propia.*

El resultado de esta evaluación queda de la siguiente manera:

Factor	Descripción	Peso	Factor
T01	Sistema distribuido	2	0
T02	Objetivos de performance o tiempo de respuesta	1	4
T03	Eficiencia del usuario final	1	5
T04	Procesamiento interno complejo	1	3
T05	El código debe ser reutilizable	1	1
T06	Facilidad de instalación	0.5	5
T07	Facilidad de uso	0.5	5
T08	Portabilidad	2	0
T09	Facilidad de cambio	1	1
T10	Concurrencia	1	5
T11	Incluye objetivos especiales de seguridad	1	1
T12	Provee acceso directo a terceras partes	1	0
T13	Se requiere facilidades especiales de entrenamiento a usuarios	1	1

*Tabla 31: Peso y ponderación Factores relevantes - TCF.
Fuente: Elaboración propia.*

De lo anterior, $T\mathbf{Factor} = \sum(\mathbf{Peso} * \mathbf{Factor}) = 26$

Por lo tanto, $T\mathbf{CF} = 0.6 + (0.01 * 26) = 0.86$

A1.1.5 Factores ambientales

El factor ambiental está dado por la función:

$$EF = 1.4 + (-0.03 * EFactor), \text{ donde } EFactor = \sum(Peso * Factor)$$

Evaluando los factores de 1 a 5, se obtiene:

Factor	Descripción	Peso	Factor
E01	Familiaridad con el modelo de proyecto utilizado	1.5	2
E02	Experiencia en la aplicación	0.5	0
E03	Experiencia en orientación a objetos	1	2
E04	Capacidad de analista	0.5	4
E05	Motivación	1	5
E06	Estabilidad de los requerimientos	2	4
E07	Personal Part-Time	-1	0
E08	Dificultad del lenguaje de programación	-1	2

Tabla 32: Peso y ponderación Factores ambientales – EF.
Fuente: Elaboración propia.

De lo anterior, $EFactor = \sum(Peso * Factor) = 18$

Por lo tanto, $EF = 1.4 + (-0.03 * 18) = 0.86$

A1.1.6 Puntos de casos de uso ajustados

$$UCP = UUCP * TCF * EF$$

$$UCP = 42 * 0.86 * 0.86 = 31.0632$$

A1.1.7 Cálculo de esfuerzo requerido horas-hombre

Se deben considerar los siguientes puntos:

- Cond01: Factores desde E1 hasta E6 menores a 3.
- Cond02: Factores desde E7 hasta E8 mayores a 3.

- Cond01 = E01, E02 y E03 = 3.
- Cond02 = 0.

- Condiciones: Cond01 + Cond02 = 3

Cálculo del valor LOE o E:

Filtro	LOE - E
20	0 – 2
28	3 – 4
36	Mayor a 4

*Tabla 33: Cálculo LOE – Esfuerzo.
Fuente: Elaboración propia.*

El esfuerzo de horas-hombre está dado por:

$$E = UCP * CF, \text{ por tanto:}$$

$$E = 31.0632 * 28 = 869.7696 \text{ horas hombre.}$$

A1.1.8 Contabilización de líneas de código.

La estructura general de la aplicación se realiza mediante un sistema de coordenadas en 3D en el motor de desarrollo Unity, en las cuales se sitúan los diferentes elementos virtuales que interactúan en cada funcionalidad establecida, es por esto que a diferencia de la programación de software comunes o programación web este proyecto no contiene un gran número de líneas de código, aunque sí es necesario establecer un comportamiento para cada situación programada mediante scripts en lenguaje C# que se encuentran dentro del paquete de la aplicación:

Los scripts más importantes se muestran en la siguiente tabla:

Ruta: APP //	Assets/Scripts/	
Script	Función	Líneas
ConexionInt.cs	Verifica la conexión a internet del dispositivo.	48
Gps1.cs	Establece las coordenadas del ícono de ubicación dentro del mapa.	288
GoogleMap.cs	Genera el mapa a través de Google Maps. (Genérico en plugin GoogleMaps de Unity)	136
DatabaseLoadBehaviour.cs	Carga la base de datos de imágenes de vuforia a la cámara de la escena en Unity.	31
DefaultTrackableEventHandler.cs	Administra los eventos producidos al detectar y perder objetivos.	134
Existen muchos otros pero su participación en la aplicación es mínima o es en su mayoría de interfaz gráfica y navegación.	Total líneas:	637

Tabla 34: Scripts relevantes. Fuente: Elaboración propia.

Los scripts mencionados son descritos en el Anexo 4 de este documento.

A1.1.8 Análisis final del tamaño del software

Como se puede ver en el capítulo 10 y en los puntos anteriores de este anexo, el cálculo estimado del esfuerzo requerido para este proyecto se correlaciona correctamente con el tiempo establecido por la Universidad para el desarrollo de un proyecto de estas características.

Se requieren alrededor de 870 horas-hombre aproximadamente, para un desarrollador trabajando de 7 a 8 horas diarias, 5 días a la semana, el tiempo requerido para completar la totalidad de las tareas establecidas será cercano a los 6 meses, 1 semestre.

ANEXO 2 – POTENCIALES USUARIOS

A2.1 Introducción

Se ha realizado una encuesta entre 120 personas al azar del ambiente universitario para determinar algunas preferencias en dispositivos móviles, conocimientos e intereses.

A2.2 Encuesta

- Universo: Potenciales usuarios de la aplicación móvil.
- Unidad de muestreo: Cantidad de personas.
- Fecha: Entre noviembre y diciembre de 2016.
- Tipo de muestreo: No probabilístico.
- Técnica de recolección: Encuesta online.
- Tamaño de la muestra: 120 personas.
- Número de preguntas: 9 preguntas.
- Objetivos:
 - Evaluar variados aspectos como la disponibilidad técnica de los potenciales usuarios, intereses en el área, tipo de uso de sus dispositivos móviles, conocimientos en el sistema Android, entre otros.

A2.3 Análisis

- Pregunta 1:

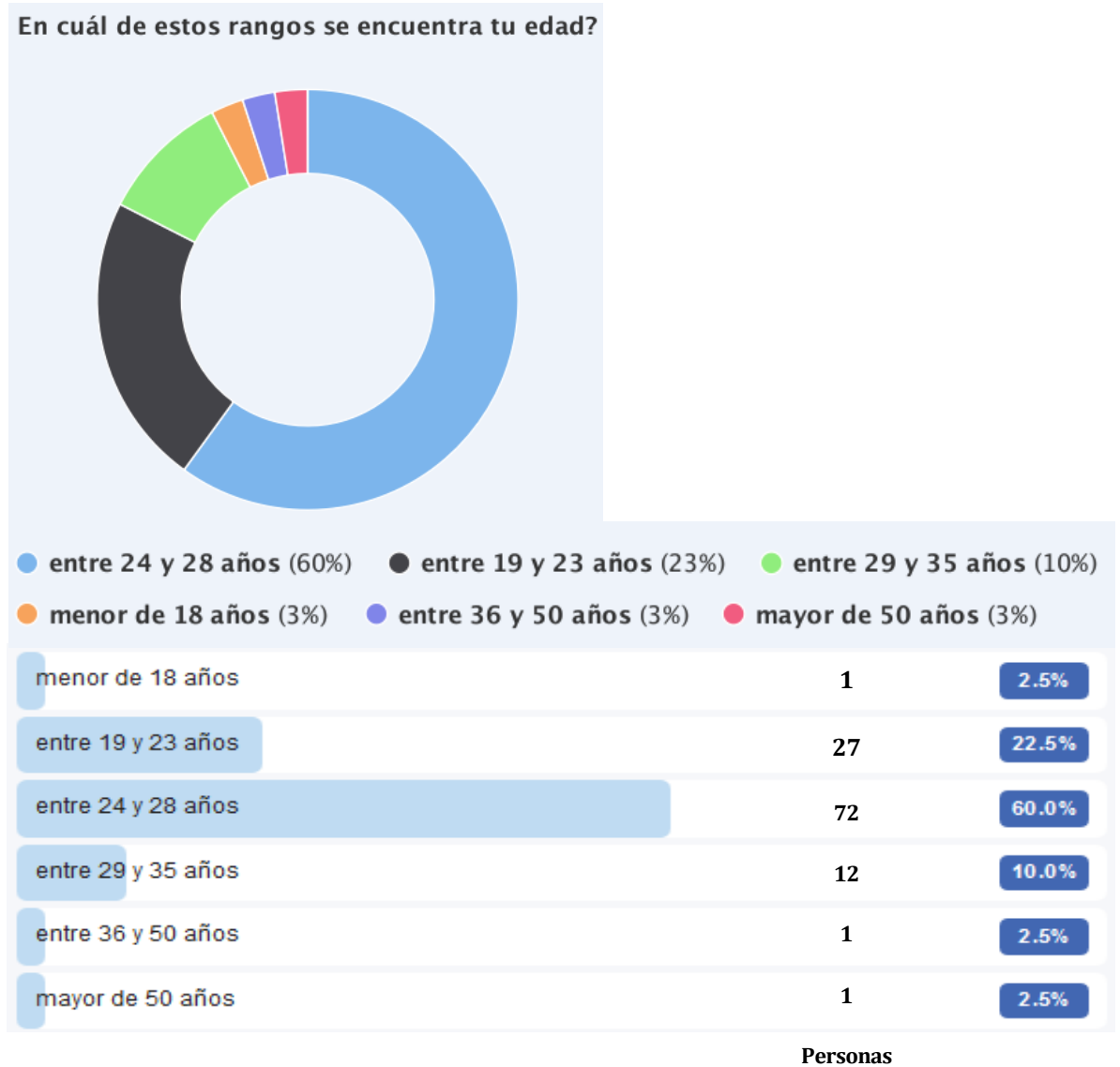
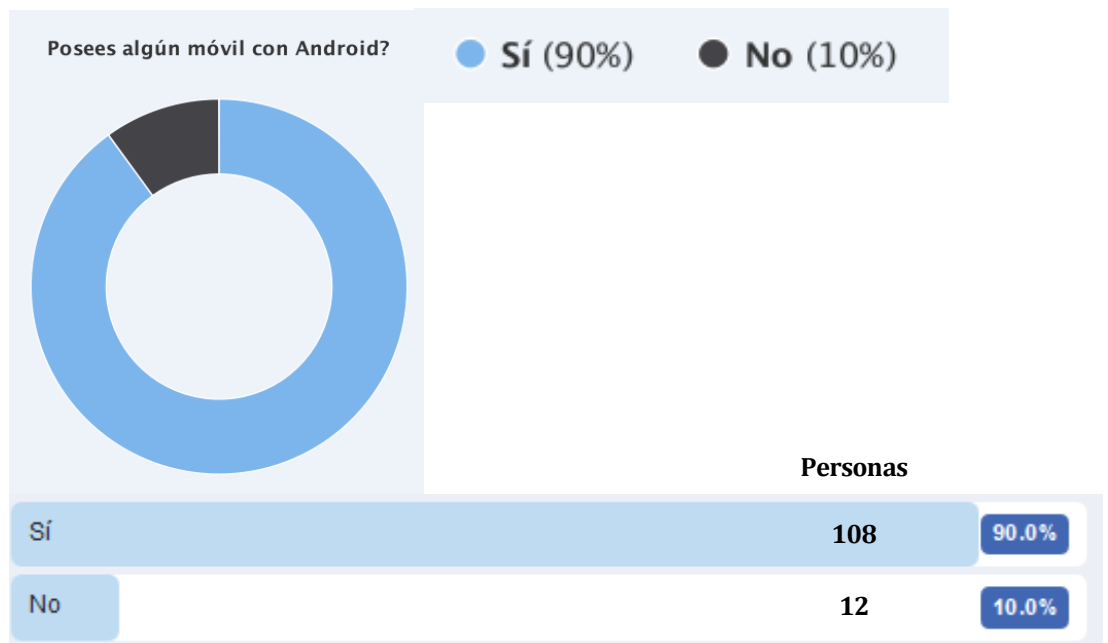


Figura 37: Encuesta pregunta 1 – Análisis potenciales usuarios.
Fuente: Elaboración propia.

La primera pregunta entrega una idea sobre el rango de edad predominante en las personas que visitan el campus, se puede deducir que las personas entre los 24 y 28 años poseen un alto grado de familiaridad con la tecnología.

- Pregunta 2 y 3:

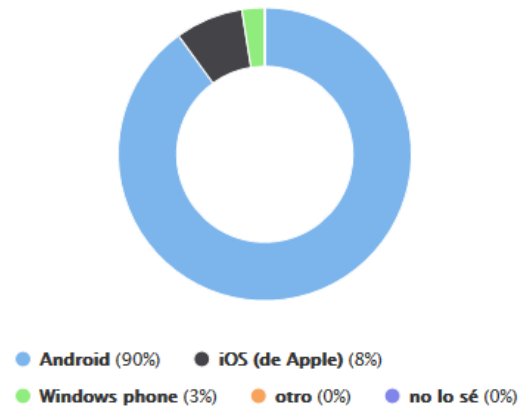


*Figura 38: Encuesta pregunta 2 y 3 (a) – Análisis potenciales usuarios.
Fuente: Elaboración propia.*

Se puede notar que 108 de las 120 personas encuestadas efectivamente poseen un dispositivo con sistema operativo Android como se estimaba en el punto 2.2.2.

De esas 108 personas, el 100% de ellos utiliza su equipo Android como dispositivo móvil principal.

Qué sistema utiliza el equipo que usas más.

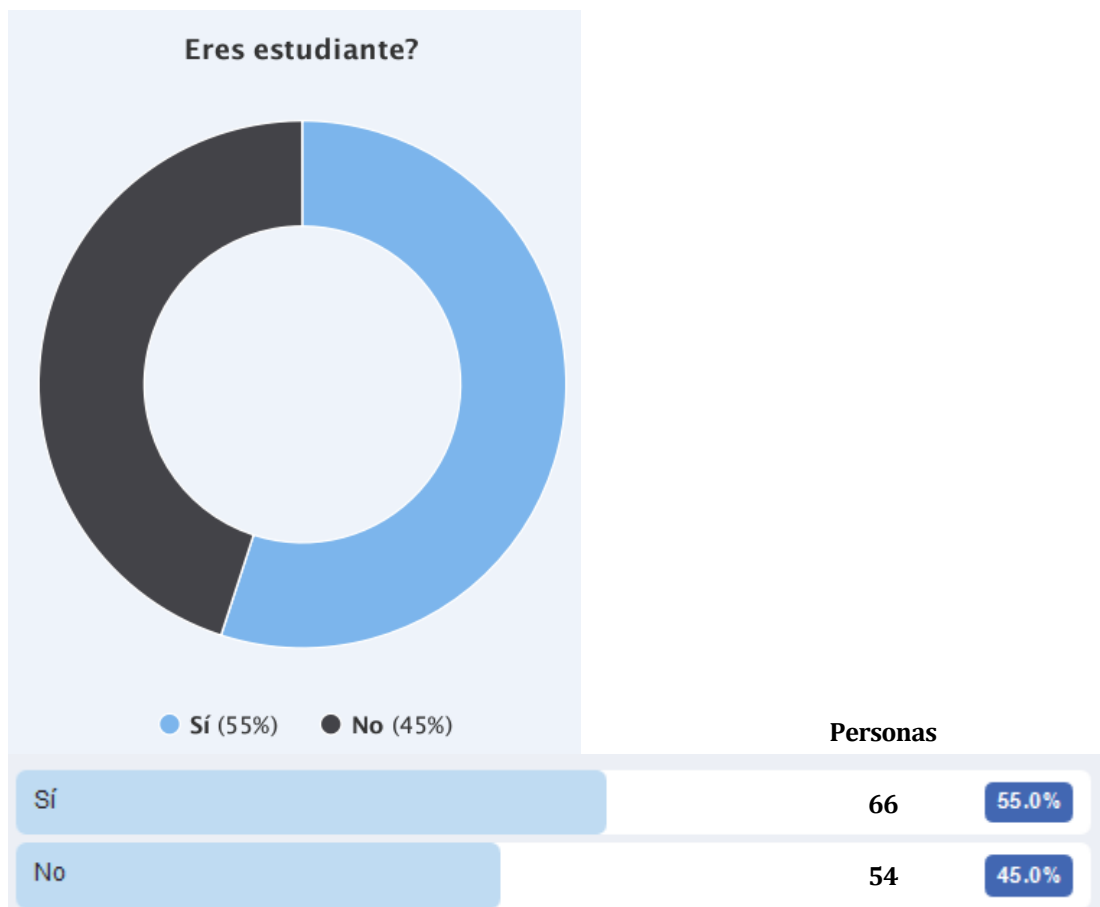


Personas

Android	108	90.0%
iOS (de Apple)	9	7.5%
Windows phone	3	2.5%
otro	0	0%
no lo sé	0	0%

Figura 39: Encuesta pregunta 2 y 3 (b) – Análisis potenciales usuarios.
Fuente: Elaboración propia.

- Pregunta 4:



*Figura 40: Encuesta pregunta 4 – Análisis potenciales usuarios.
Fuente: Elaboración propia.*

Con una encuesta de tipo no probabilístico y debido al variado grupo de personas que visitan la universidad se ha rescatado el resultado anterior, casi la mitad de las personas que contestaron las preguntas no son actualmente estudiantes aunque sí son recurrentes en el área que comprende el proyecto.

- Pregunta 5:



Figura 41: Encuesta pregunta 5 – Análisis potenciales usuarios.
Fuente: Elaboración propia.

En la pregunta 5 queda manifiesta la importancia que las personas le dan a sus móviles en su día a día a diferencia de lo que se veía años atrás, no solo la conectividad a través de las redes sociales influye en este aspecto, sino también la habilidad de los smartphones de facilitar algunas tareas cotidianas.

• Pregunta 6:

Cuál de estas opciones crees que se acomoda más a tu nivel para usar tu smartphone?

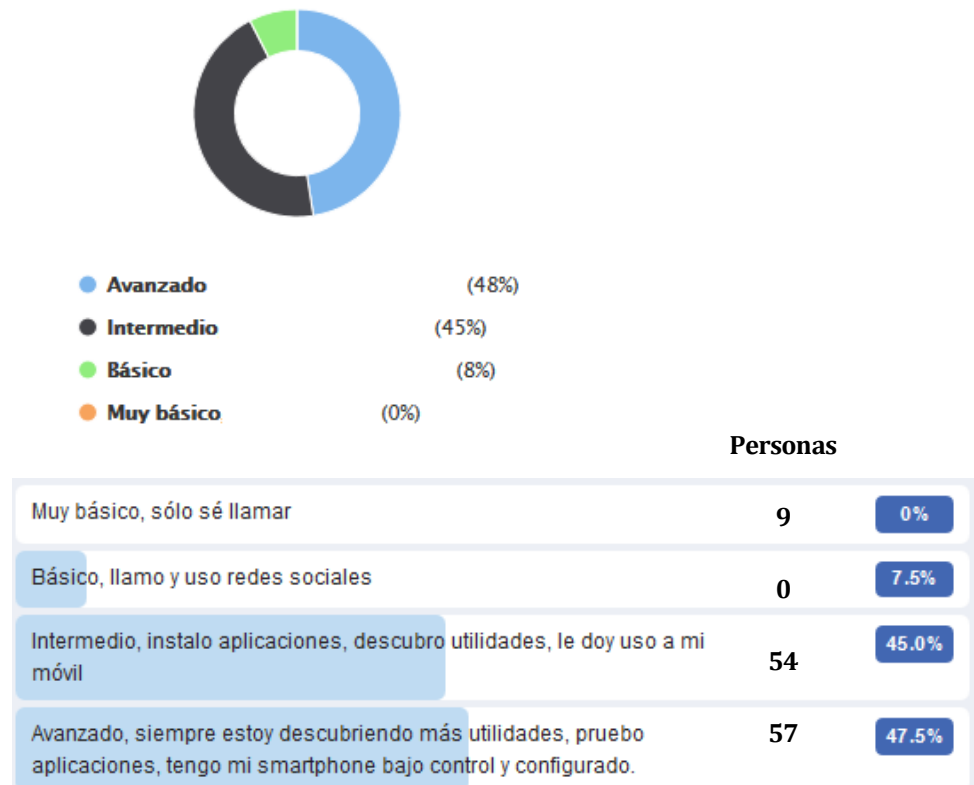


Figura 42: Encuesta pregunta 6 – Análisis potenciales usuarios.
Fuente: Elaboración propia.

De la mano con la pregunta anterior, las personas que poseen actualmente dispositivos móviles se preocupan cada día más de sacarle el máximo de utilidad a sus equipos, ya no solamente se encuentra el típico usuario que llama y nada más, sino que cada vez son más los usuarios que consideran tener conocimientos intermedios o avanzados en el uso de sus dispositivos.

- Pregunta 7 y 8:

Sabes en qué consiste la realidad virtual?



- No lo tengo muy claro, pero algo sé (45%)
- Por supuesto (40%)
- Casi nada (13%)
- No (3%)

Sabes en qué consiste la realidad aumentada?



- Casi nada (33%)
- Por supuesto (25%)
- No (23%)
- No lo tengo muy claro, pero algo sé (20%)

Figura 43: Encuesta pregunta 7 y 8 – Análisis potenciales usuarios.
Fuente: Elaboración propia.

Respecto al conocimiento de la población sobre la realidad virtual y realidad aumentada se da un resultado esperado, para la realidad virtual predomina un noción mediana, entre “algo sé” y “por supuesto que sé”, mientras que para la realidad aumentada predomina el “casi nada” como es de esperarse para un tema que no se ha desarrollado en gran manera actualmente, aunque un 25%(30 de 120) de las personas afirma saber de qué se trata, dato más que alentador.

- Pregunta 9:

Qué tanto interés tendrías en una aplicación que te entregue información del lugar donde te encuentras con sólo apuntar con tu cámara posterior?



Personas

Me interesaría bastante, quiero probar esa app	54	45.0%
Me interesaría, quiero saber más al respecto	51	42.5%
Creo que me interesaría, pero no le daría tanto uso	12	10.0%
No le veo la utilidad	3	2.5%

Figura 44: Encuesta pregunta 9 – Análisis potenciales usuarios.
Fuente: Elaboración propia.

A pesar que sólo el 45% de las personas afirma tener conocimientos o saber algo sobre la realidad aumentada un 87.5% expresa un interés en este tipo de aplicaciones ya que considera que se pueden obtener beneficios favorables de estas, a un 10% podría interesarle al menos para probarla.

ANEXO 3 – ESPECIFICACIÓN DE PRUEBAS

En este anexo se especifican las pruebas establecidas en el punto 8.3.

ID	E.P1	Prueba	Funcionamiento de interfaz en diferentes resoluciones		
Caso de prueba	Características a probar		Salida esperada	Salida obtenida	Éxito / Fracaso
E.P1-4.5	Los elementos de interfaz se adecúan correctamente a las resoluciones de dispositivos con pantalla de 4.5”		Interfaz se sitúa y comporta correctamente	Interfaz se sitúa y comporta correctamente	Éxito
E.P1-5	Los elementos de interfaz se adecúan correctamente a las resoluciones de dispositivos con pantalla de 5”		Interfaz se sitúa y comporta correctamente	Interfaz se sitúa y comporta correctamente	Éxito
E.P1-5.5	Los elementos de interfaz se adecúan correctamente a las resoluciones de dispositivos con pantalla de 5.5”		Interfaz se sitúa y comporta correctamente	Interfaz se sitúa y comporta correctamente	Éxito

Tabla 35: E.P1 – Especificación de pruebas.
Fuente: Elaboración propia.

ID	E.P2	Prueba	Comunicación con GPS		
Caso de prueba	Características a probar		Salida esperada	Salida obtenida	Éxito / Fracaso
E.P2 – Situación A	Abrir el mapa de la aplicación con el GPS activado, luego desactivarlo.		Se abre el mapa normalmente, al desactivar GPS se muestra mensaje GPS no disponible.	Se abre el mapa normalmente, al desactivar GPS se muestra mensaje GPS no disponible.	Éxito
E.P2 – Situación B	Abrir el mapa de la aplicación con el GPS desactivado, luego activarlo.		Se abre el mapa normalmente con mensaje GPS no disponible, al activarlo el mensaje desaparece.	Se abre el mapa normalmente con mensaje GPS no disponible, al activarlo el mensaje desaparece.	Éxito
E.P2 – Situación A Dentro de los límites del mapa	Ejecutar (P2 - A) dentro del área marcada en el mapa.		Se obtiene el resultado de E.P2 – A, pero al estar dentro del área del mapa se puede ver el ícono de ubicación, al desactivar el ícono desaparece y se muestra el	Se obtiene el resultado de E.P2 – A, pero al estar dentro del área del mapa se puede ver el ícono de ubicación, al desactivar el ícono desaparece y se muestra el	Éxito

		mensaje GPS no disponible.	mensaje GPS no disponible.	
E.P2 – Situación A Fuera de los límites del mapa	Ejecutar (P2 - A) fuera del área marcada en el mapa.	Se obtiene el resultado de E.P2 – A	Se obtiene el resultado de E.P2 – A	Éxito
E.P2 – Situación B Dentro de los límites del mapa	Ejecutar (P2 - B) dentro del área marcada en el mapa.	Se obtiene el resultado de E.P2 – B, al activar el GPS el mensaje desaparece y se puede ver el ícono de ubicación.	Se obtiene el resultado de E.P2 – B, al activar el GPS el mensaje desaparece y se puede ver el ícono de ubicación.	Éxito
E.P2 – Situación B Fuera de los límites del mapa	Ejecutar (P2 - B) fuera del área marcada en el mapa.	Se obtiene el resultado de E.P2 – B, al activar el GPS el mensaje desaparece.	Se obtiene el resultado de E.P2 – B, al activar el GPS el mensaje desaparece.	Éxito

Tabla 36: E.P2 – Especificación de pruebas.
Fuente: Elaboración propia.

ID	E.P3	Prueba	Detección de objetivos		
Caso de prueba	Características a probar		Salida esperada	Salida obtenida	Éxito / Fracaso
E.P3- Válido	Apuntar la cámara del dispositivo con la aplicación en funcionamiento hacia un objetivo válido.		La aplicación indica el nombre del lugar enfocado.	La aplicación indica el nombre del lugar enfocado.	Éxito
E.P3- No válido	Apuntar la cámara del dispositivo con la aplicación en funcionamiento hacia un objetivo no válido.		Se visualiza un mensaje de objetivo no encontrado.	Se visualiza un mensaje de objetivo no encontrado.	Éxito

Tabla 37: E.P3 – Especificación de pruebas.
Fuente: Elaboración propia.

ID	E.P4	Prueba	Interfaz correcta en diferentes fases del funcionamiento		
Caso de prueba	Características a probar		Salida esperada	Salida obtenida	Éxito / Fracaso
E.P4-1	La interfaz funciona correctamente en las diferentes etapas de navegación.		La interfaz se comporta correctamente en cada etapa de funcionamiento	La interfaz se comporta correctamente en cada etapa de funcionamiento	Éxito

Tabla 38: E.P4 – Especificación de pruebas.
Fuente: Elaboración propia.

ID	E.P5	Prueba	Navegar por la aplicación utilizando la interfaz		
Caso de prueba	Características a probar		Salida esperada	Salida obtenida	Éxito / Fracaso
E.P5-1	Se puede navegar correctamente por la totalidad de las funciones de la aplicación utilizando la interfaz de navegación.		La interfaz se comporta correctamente en cada etapa de funcionamiento	La interfaz se comporta correctamente en cada etapa de funcionamiento	Éxito

Tabla 39: E.P5 – Especificación de pruebas.
Fuente: Elaboración propia.

ID	E.P6	Prueba	Movimiento ícono de ubicación		
Caso de prueba	Características a probar		Salida esperada	Salida obtenida	Éxito / Fracaso
E.P6-1	El ícono responde a los movimientos del usuario sobre el terreno.		El ícono de ubicación se mueve correctamente.	El ícono de ubicación se mueve correctamente.	Éxito

Tabla 40: E.P6 – Especificación de pruebas.
Fuente: Elaboración propia.

ID	E.P7	Prueba	Funcionamiento a ciertas distancias del objetivo		
Caso de prueba	Características a probar		Salida esperada	Salida obtenida	Éxito / Fracaso
E.P7-1	Los objetivos son reconocidos a distancias correctas.		El objetivo sigue siendo detectado dentro de los rangos de distancia establecidos.	El objetivo sigue siendo detectado dentro de los rangos de distancia establecidos.	Éxito

*Tabla 41: E.P7 – Especificación de pruebas.
Fuente: Elaboración propia.*

ANEXO 4 – CÓDIGO RELEVANTE

En el presente anexo se describe los scripts mencionados en el punto A1.1.8 de este documento.

ConexionInt.cs

```

1  using UnityEngine;
2  using System.Collections;
3  using System.Net;
4  using System.IO;
5  using UnityEngine.UI;
6
7  public class ConexionInt : MonoBehaviour {
8
9      //Objetos utilizados
10     public GameObject panel;
11     public GameObject canvas;
12     public GameObject nonet;
13
14
15     void start () {
16
17         //inicialización
18         //PANEL ES EL ELEMENTO EN EL CANVAS QUE CONTENDRÁ EL MAPA DE GOOGLE MAPS
19         //SE ACTIVA PREVIAMENTE SI EL RESULTADO DE CHECK ES POSITIVO
20         panel.SetActive(Check());
21         nonet.SetActive(false);
22         //Se verifica si panel se encuentra activo o no
23         //Si no se encuentra activo en la escena se activa el RAWIMAGE del canvas
24         //(La imagen proyectada en el canvas) y se activa la bandera NONET (no internet)
25         if(panel.activeSelf==false)
26         {
27             canvas.GetComponent<RawImage>().enabled=true;
28             nonet.SetActive(true);
29         }
30     }
31
32
33     public bool Check(){
34         //en el caso de que el panel se encuentre activo se realiza una petición de conexión
35         //a una página estable (www.google.com) de realizarse exitosamente se devuelve un TRUE
36         //en lo contrario devuelve FALSE
37         try{
38             WebClient client = new WebClient();
39             Stream stream = client.OpenRead("http://www.google.com");
40             stream.Close();
41             return true;
42         }
43
44         catch{
45             return false;
46         }
47     }
48 }

```

Figura 45: ConexionInt.cs – Código Relevante.
Fuente: Captura propia Sublime Text – Programación C# Unity 3D.

Gps1.cs

```

1  using UnityEngine;
2  using UnityEngine.UI;
3  using System.Collections;
4  using System;
5
6
7  public class gps1 : MonoBehaviour {
8
9
10     //YA QUE NO EXISTE FORMA GRATUITA DE ESTABLECER LA UBICACIÓN EN UN MAPA PARA UNITY
11     //SE CREA UNA MATRIZ DE PÍXELES QUE ESTA RELACIONADA CON LAS COORDENADAS DE LOS LÍMITES DEL MAPA
12
13
14     public GameObject avisoGPS;
15     public GameObject cursor;
16     public GameObject Título;
17
18     //Se establecen Las coordenadas Límites del mapa en LONGITUD
19
20     public float A = -73.01594f;
21     public float B = -73.015659f;
22     public float C = -73.015378f;
23     public float D = -73.015097f;
24     public float E = -73.014816f;
25     public float F = -73.014535f;
26     public float G = -73.014254f;
27     public float H = -73.013973f;
28     public float I = -73.013692f;
29     public float J = -73.013411f;
30     public float K = -73.01313f;
31     public float L = -73.012849f;
32     public float M = -73.012568f;
33     public float N = -73.012287f;
34     public float O = -73.012006f;
35     public float P = -73.011725f;
36     public float Q = -73.011444f;
37     public float R = -73.011163f;
38     public float S = -73.010882f;
39     public float T = -73.010601f;
40     public float U = -73.01032f;
41     public float prueba = -36.55191f;
42
43     //Límites de La matriz en píxeles HORIZONTAL
44
45     public int AA = 318;
46     public int BB = 322;
47     public int CC = 327;
48     public int DD = 331;
49     public int EE = 335;
50     public int FF = 340;
51     public int GG = 344;
52     public int HH = 348;
53     public int II = 352;
54     public int JJ = 356;
55     public int KK = 361;
56     public int LL = 365;
57     public int MM = 369;
58     public int NN = 373;
59     public int OO = 378;
60     public int PP = 382;
61     public int QQ = 386;
62     public int RR = 390;
63     public int SS = 395;
64     public int TT = 399;
65     public int UU = 403;
66

```

```

67
68 //Se establecen Las coordenadas Límites del mapa en LATITUD
69
70 public float a = -36.82081f;
71 public float b = -36.821051f;
72 public float c = -36.821292f;
73 public float d = -36.821533f;
74 public float e = -36.821774f;
75 public float f = -36.822015f;
76 public float g = -36.822256f;
77 public float h = -36.822497f;
78 public float i = -36.822738f;
79 public float j = -36.822979f;
80 public float k = -36.82322f;
81
82 //Límites de La matriz en pixeles VERTICAL
83
84 public int aa= 251;
85 public int bb= 246;
86 public int cc= 241;
87 public int dd= 236;
88 public int ee= 232;
89 public int ff= 227;
90 public int gg= 222;
91 public int hh= 217;
92 public int ii= 213;
93 public int jj= 208;
94 public int kk= 203;
95
96 public int CoordX=0;
97 public int CoordY=0;
98
99 float myLat = 0f;
100 float myLong= 0f;
101
102
103 void Start () {
104
105 //elementos en Unity ocultos inicialmente (elemento = false)
106 avisoGPS.SetActive(false);
107 cursor.SetActive(false);
108
109 StartCoroutine (getLocation ());
110 }
111
112
113 void Update () {
114
115 //obtener coordenadas en cada update
116 StartCoroutine (getLocation ());
117 }
118
119 IEnumerator getLocation ()
120 {
121
122 //Código de obtención de coordenadas mediante GPS en android.
123 if (!Input.location.isEnabledByUser){
124 if(Titulo.activeSelf==false)avisoGPS.SetActive(true);
125 cursor.SetActive(false);
126
127 yield break;
128 } else {
129
130 Input.location.Start ();

```



```

131
132     int maxwait = 180;
133     while (Input.location.status == LocationServiceStatus.Initializing && maxwait > 0) {
134         yield return new WaitForSeconds (1);
135         maxwait--;
136     }
137
138
139     if (maxwait < 1) {
140         Debug.Log ("Timed out");
141
142         if(Titulo.activeSelf==false)avisoGPS.SetActive(true);
143         cursor.SetActive(false);
144         yield break;
145     }
146
147
148     if (Input.location.status == LocationServiceStatus.Failed) {
149         yield break;
150         if(Titulo.activeSelf==false)avisoGPS.SetActive(true);
151         cursor.SetActive(false);
152
153     } else {
154
155         //de Llegar a este punto La conexión con el GPS se ha realizado correctamente.
156
157         avisoGPS.SetActive(false);
158
159         //Obtención de Latitud y Longitud.
160         myLat = Input.location.lastData.latitude;
161         myLong = Input.location.lastData.longitude;
162
163         //se genera el cursor en La Latitud y Longitud obtenidas.
164         pintarCursor(myLat, myLong);
165
166
167     }
168
169
170
171     Input.location.Stop ();
172
173 }
174 }
175
176 public void pintarCursor(float myLat, float myLong){
177
178     //Aquí se identifica en qué punto de La matriz se encuentran Las coordenadas encontradas provenientes
179     //de myLat y myLong, cada "punto" en La matriz es representado por cada IF a continuación
180     //cuando se cumpla La condición se pintará La ubicación en el mapa según Los píxeles entregados para
181     //CoordX y CoodY dentro de Los rangos establecidos.
182
183     if(myLong<=-73.01032f && myLong >= -73.01594f){
184         if(myLat >= -36.82322f && myLat <=-36.82081f){
185
186             if(myLong==U){CoordX=UU;}
187             else{
188                 if(myLong>=T){CoordX=TT;}
189                 else{
190                     if(myLong>=S){CoordX=SS;}
191                     else{
192                         if(myLong>=R){CoordX=RR;}
193                         else{
194                             if(myLong>=Q){CoordX=QQ;}
195                             else{

```

```

196     if(myLong>=P){CoordX=PP;}
197     else{
198     if(myLong>=O){CoordX=OO;}
199     else{
200     if(myLong>=N){CoordX=NN;}
201     else{
202     if(myLong>=M){CoordX=MM;}
203     else{
204     if(myLong>=L){CoordX=LL;}
205     else{
206     if(myLong>=K){CoordX=KK;}
207     else{
208     if(myLong>=J){CoordX=JJ;}
209     else{
210     if(myLong>=I){CoordX=II;}
211     else{
212     if(myLong>=H){CoordX=HH;}
213     else{
214     if(myLong>=G){CoordX=GG;}
215     else{
216     if(myLong>=F){CoordX=FF;}
217     else{
218     if(myLong>=E){CoordX=EE;}
219     else{
220     if(myLong>=D){CoordX=DD;}
221     else{
222     if(myLong>=C){CoordX=CC;}
223     else{
224     if(myLong>=B){CoordX=BB;}
225     else{
226     if(myLong>=A){CoordX=AA;}
227
228     }
229     }
230     }
231     }
232     }
233     }
234     }
235     }
236     }
237     }
238     }
239     }
240     }
241     }
242     }
243     }
244     }
245     }
246     }
247     }
248
249
250     if(myLat==a){CoordY=aa;}
251     else{
252         if(myLat>=b){CoordY=bb;}
253         else{
254             if(myLat>=c){CoordY=cc;}
255             else{
256                 if(myLat>=d){CoordY=dd;}
257                 else{
258                     if(myLat>=e){CoordY=ee;}
259                     else{
260                         if(myLat>=f){CoordY=ff;}

```

```

261     else{
262         if(myLat>=g){CoordY=gg;}
263         else{
264             if(myLat>=h){CoordY=hh;}
265             else{
266                 if(myLat>=i){CoordY=ii;}
267                 else{
268                     if(myLat>=j){CoordY=jj;}
269                     else{
270                         if(myLat>=k){CoordY=kk;}
271                     }
272                 }
273             }
274         }
275     }
276 }
277 }
278 }
279 }
280 }
281
282 cursor.transform.position = new Vector3(CoordX,CoordY,-887);
283 cursor.SetActive(true);
284
285 }
286 }
287 }
288 }

```

*Figura 46: Gps1.cs – Código Relevante.
Fuente: Captura propia Sublime Text – Programación C# Unity 3D.*

DatabaseLoadBehaviour.cs

```

1 using System.Collections.Generic;
2 using UnityEngine;
3
4 namespace Vuforia
5 {
6     //Carga la base de datos de Vuforia en Unity
7     public class DatabaseLoadBehaviour : DatabaseLoadAbstractBehaviour
8     {
9         public override void AddSpecificExternalDatasetSearchDirs()
10        {
11            #if UNITY_ANDROID
12            if (Application.platform == RuntimePlatform.Android)
13            {
14                // Obtiene el directorio de almacenamiento externo
15                AndroidJavaClass jclassEnvironment = new AndroidJavaClass("android.os.Environment");
16                AndroidJavaObject jobjectFile = jclassEnvironment.CallStatic<AndroidJavaObject>("getExternalStorageDirectory");
17                string externalStorageDirectory = jobjectFile.Call<string>("getAbsolutePath");
18
19                // Obtiene el nombre del paquete
20                AndroidJavaObject jobjectActivity = new AndroidJavaClass("com.unity3d.player.UnityPlayer").getStatic<AndroidJavaObject>("currentActivity");
21                string packageName = jobjectActivity.Call<string>("getPackageName");
22
23                // Carga los archivos requeridos a la cámara de Vuforia
24                AddExternalDatasetSearchDir(externalStorageDirectory + "/" + packageName + "/files/");
25                AddExternalDatasetSearchDir(externalStorageDirectory + "/" + packageName + "/files/Vuforia/DeviceDatabases/");
26                AddExternalDatasetSearchDir(externalStorageDirectory + "/" + packageName + "/files/QCAR/");
27            }
28        }
29    }
30 }
31

```

Figura 47: DatabaseLoadBehaviour.cs – Código Relevante. Fuente: Captura propia Unity 3D.

DefaultTrackableEventHandler.cs

```

1  using UnityEngine;
2
3  namespace Vuforia
4  {
5
6      public class DefaultTrackableEventHandler : MonoBehaviour,
7          ITrackableEventHandler
8      {
9          #region PRIVATE_MEMBER_VARIABLES
10
11         private TrackableBehaviour mTrackableBehaviour;
12         public GameObject ObjetivoSi, ObjetivoNo, StartCanvas, Mono, Corners, MenuCanvas, IfCanvas;
13
14         #endregion
15
16         #region UNITY_MONOBEHAVIOUR_METHODS
17
18         void Start()
19         {
20             IfCanvas.SetActive(true);
21             ObjetivoSi.SetActive(false);
22             ObjetivoNo.SetActive(false);
23             Mono.SetActive(false);
24             StartCanvas.SetActive(false);
25
26             mTrackableBehaviour = GetComponent<TrackableBehaviour>();
27             if (mTrackableBehaviour)
28             {
29                 mTrackableBehaviour.RegisterTrackableEventHandler(this);
30             }
31         }
32
33         #endregion
34
35         #region PUBLIC_METHODS
36
37         //Funcion Llamada cuando se cambia el estado del target
38         public void OnTrackableStateChanged(
39             TrackableBehaviour.Status previousStatus,
40             TrackableBehaviour.Status newStatus)
41         {
42             if (newStatus == TrackableBehaviour.Status.DETECTED ||
43                 newStatus == TrackableBehaviour.Status.TRACKED ||
44                 newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
45             {
46                 OnTrackingFound();
47             }
48             else
49             {
50                 OnTrackingLost();
51             }
52         }
53
54         #endregion
55
56
57
58         #region PRIVATE_METHODS

```

```

59
60
61 //Funcion Llamada al encontrar un target válido
62 private void OnTrackingFound()
63 {
64
65     //Acciones al encontrar un objetivo válido
66     ObjetivoSi.SetActive(true);
67     ObjetivoNo.SetActive(false);
68
69     if(StartCanvas.activeSelf==true){
70         IfCanvas.SetActive(true);
71     }
72
73     if(Corners.activeSelf==false){
74         StartCanvas.SetActive(true);
75     }
76
77
78
79     Renderer[] rendererComponents = GetComponentsInChildren<Renderer>(true);
80     Collider[] colliderComponents = GetComponentsInChildren<Collider>(true);
81
82
83     foreach (Renderer component in rendererComponents)
84     {
85         component.enabled = true;
86     }
87
88
89     foreach (Collider component in colliderComponents)
90     {
91         component.enabled = true;
92     }
93
94     Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");
95
96
97 }
98
99 //Funcion Llamada al perder un target válido
100 private void OnTrackingLost()
101 {
102
103     //Acciones al perder un objetivo válido
104     ObjetivoSi.SetActive(false);
105     ObjetivoNo.SetActive(true);
106     Mono.SetActive(false);
107     StartCanvas.SetActive(false);
108
109     if(MenuCanvas.activeSelf==true) MenuCanvas.SetActive(false);
110
111     Renderer[] rendererComponents = GetComponentsInChildren<Renderer>(true);
112     Collider[] colliderComponents = GetComponentsInChildren<Collider>(true);
113
114
115     foreach (Renderer component in rendererComponents)
116     {
117         component.enabled = false;
118     }
119
120
121     foreach (Collider component in colliderComponents)
122     {
123         component.enabled = false;
124     }
125
126

```

```
125  
126     Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " lost");  
127  
128  
129  
130     }  
131  
132     #endregion  
133 }  
134 }
```

*Figura 48: DefaultTrackerEventHandler.cs – Código Relevante.
Fuente: Captura propia Sublime Text – Programación C# Unity 3D.*