



**UNIVERSIDAD DEL BÍO-BÍO**

FACULTAD DE CIENCIAS EMPRESARIALES

Departamento de Ciencias de la Computación y Tecnologías de la  
Información

# **ESTUDIO DE HERRAMIENTAS DE REALIDAD AUMENTADA SOBRE LA PLATAFORMA ANDROID**

---

**Memoria para optar al título de Ingeniero Civil en Informática**

AUTOR:

FERRADA VALERIA HECTOR PIERO

Profesor Guía: Gajardo Díaz Luis

Chillán, Marzo de 2014

## **AGRADECIMIENTOS**

### **A mis padres y hermana**

*Por darme la oportunidad de estudiar lo que siempre quise, la confianza puesta en mí al estar lejos de casa y darme siempre la fuerza necesaria para lograr terminar esta etapa en mi vida.*

### **A mi tía**

*Tía Iris, por estos 5 años que me dejaste entrar a tu casa, siendo una segunda madre para mí, soportándome con mis defectos (que son bastante) y mimarme todos los días que estuve en Chillán.*

### **A Daniela Montero**

*Por brindarme tu amistad y además facilitarme tu Smartphone, que me ayudó terminar este proyecto, sin esperar nada a cambio, gracias "Dani".*

### **A mis compañeros de Generación**

*Por su amistad y compartir tantas experiencias conmigo y a pesar de tener varias discrepancias en el camino, siempre se solucionaron y nunca hubo rencores. Gracias a todos.*

### **A mi familia**

*Por darme ánimos y buenas vibras en todo momento.*

### **Al profesor Luis Gajardo**

*Por ayudarme en este proyecto y alcanzar los objetivos previsto.*

**Héctor Ferrada Valeria**

## **RESUMEN**

Actualmente, la plataforma android se encuentra en la mayoría de los Smartphone de nuestro país y siempre están apareciendo aplicaciones novedosas que captan el interés de los usuarios. Además, con la constante evolución de los teléfonos inteligentes, se pueden materializar ideas que años atrás sería impensable realizar, debido a que no existía los recursos para construirlas.

El objetivo de este proyecto es el estudio de herramientas de realidad aumentada sobre plataforma android. Se realiza un análisis comparativo que permite dar a conocer al lector las nuevas características que proporciona la Realidad Aumentada y que actualmente se ha comenzado a utilizar en dispositivos móviles. Es un estudio que además, incluye el desarrollo de casos prácticos para mostrar el funcionamiento de estas herramientas.

Para realizar el estudio, se ha decidido analizar tres herramientas de realidad aumentada: Wikitude, Vuforia, Metaio SDK. Se detallan las características de cada una, su arquitectura y tipos de aplicaciones que se puede realizar con ellas. Además, para desarrollar el estudio comparativo se utilizan instrumentos basados en la escala de Likert, debido a que es usado por la mayoría de los estudios de investigación y es de fácil comprensión.

El estudio determinó que Wikitude es la más completa en comparación a Metaio SDK y Vuforia, debido a que presenta más y/o mejores funcionalidades con respecto a las otras herramientas. Además, el estudio permitió observar todas las nuevas características que brinda la tecnología de AR al mundo de los Smartphones, ya sea Geo localización, reconocimiento de imágenes o la mezcla de estas, poniendo a prueba la imaginación a los programadores que las utilizan.

---

## ÍNDICE GENERAL

---

<b>ÍNDICE GENERAL .....</b>	<b>.....</b>
ÍNDICE DE IMÁGENES.....	7
ÍNDICE DE TABLAS.....	10
<b>CAPÍTULO 1: INTRODUCCIÓN.....</b>	<b>.....</b>
1.1. INTRODUCCIÓN GENERAL.....	12
1.2. ANTECEDENTES DEL PROYECTO.....	13
1.3. OBJETIVOS.....	13
1.4. METODOLOGÍA .....	14
1.5. ALCANCES Y LIMITACIONES .....	14
1.6. DEFINICIONES Y ABREVIATURAS .....	15
<b>CAPÍTULO 2: MARCO TEÓRICO .....</b>	<b>.....</b>
2.1. REALIDAD AUMENTADA .....	18
2.1.1. Introducción .....	18
2.1.2. Cronología de la Realidad Aumentada .....	19
2.1.3. Tecnología empleada actualmente .....	21
2.1.3.1. Hardware:.....	22
2.1.3.2. Software:.....	24
2.1.4. Clasificación de AR.....	25
2.1.4.1. Reconocimiento de objetos mediante marcadores:.....	25
2.1.4.2. Reconocimiento de objetos mediante formas: .....	27
2.1.4.3. Reconocimientos de objetos mediante geo localización: .....	28
2.1.5. Aplicaciones .....	29
2.1.5.1. Proyectos educativos:.....	29
2.1.5.2. Cirugía:.....	30
2.1.5.3. Televisión .....	30
2.1.5.4. Entretenimiento:.....	31
2.1.5.5. Simulación:.....	32
2.1.5.6. Arquitectura:.....	33
2.1.5.7. Navegación: .....	33
2.1.5.8. Aplicaciones Industriales: .....	34
2.1.5.9. Prospección: .....	34
2.1.5.10. Publicidad: .....	35
2.1.5.11. Información: .....	36
2.2. TECNOLOGÍA MÓVIL.....	37

2.2.1.	Introducción .....	37
2.2.2.	Evolución de la Tecnología Móvil.....	37
2.2.2.1.	Primera Generación (1G).....	38
2.2.2.2.	Segunda Generación (2G).....	38
2.2.2.3.	Generación de Transición (2.5 G).....	38
2.2.2.4.	Tercera Generación (3G).....	39
2.2.2.5.	Cuarta Generación (4G).....	39
2.2.3.	Sistemas Operativos para móviles.....	39
2.2.3.1.	Componentes de los Sistemas operativos en Dispositivos Móviles.....	39
2.2.3.2.	Sistemas Operativos en el Mercado.....	40

**CAPÍTULO 3: ESTUDIO COMPARATIVO DE TECNOLOGÍAS DE REALIDAD AUMENTADA.....**

3.1.	INTRODUCCIÓN.....	45
3.2.	PRESENTACIÓN DE LAS HERRAMIENTAS.....	47
3.2.1.	Wikitude.....	47
3.2.1.1.	Características:.....	48
3.2.1.2.	Arquitectura:.....	48
3.2.1.3.	Ciclo de Vida:.....	48
3.2.1.4.	Aplicaciones:.....	49
3.2.2.	Vuforia.....	50
3.2.2.1.	Características:.....	50
3.2.2.2.	Arquitectura:.....	51
3.2.2.3.	Aplicaciones:.....	52
3.2.3.	Metaio SDK.....	54
3.2.3.1.	Características:.....	54
3.2.3.2.	Arquitectura:.....	54
3.2.3.3.	Aplicaciones:.....	55
3.3.	ANÁLISIS COMPARATIVO.....	57
3.3.1.	Definición de parámetros de comparación:.....	57
3.3.2.	Análisis de parámetros y variables de comparación.....	58
3.3.3.	Puntajes alcanzados:.....	70
3.4.	INTERPRETACIÓN DE LOS RESULTADOS OBTENIDOS.....	72

**CAPÍTULO 4: CASOS PRÁCTICOS.....**

4.1.	INTRODUCCIÓN.....	75
4.2.	CASO PRÁCTICO 1: GEO LOCALIZACIÓN.....	76
4.2.1.	Objetivo.....	76
4.2.2.	Descripción.....	77
4.2.3.	Implementación.....	78

4.2.4.	Capturas de la aplicación .....	80
4.2.5.	Resultado.....	82
4.3.	CASO PRÁCTICO 2: RECONOCIMIENTO DE IMAGEN .....	83
4.3.1.	Objetivo .....	83
4.3.2.	Descripción.....	83
4.3.3.	Implementación .....	84
4.3.4.	Capturas de pantallas .....	85
4.3.5.	Resultado.....	85
<b>CONCLUSIONES .....</b>		
CONCLUSIONES GENERALES .....		87
RECOMENDACIONES: .....		89
TRABAJOS FUTUROS.....		90
<b>BIBLIOGRAFÍA.....</b>		<b>91</b>
<b>ANEXO A: CÓDIGO FUENTE DE CASO PRÁCTICO DE GEO LOCALIZACIÓN.....</b>		<b>95</b>
<b>ANEXO B: CÓDIGO FUENTE DE CASO PRÁCTICO DE RECONOCIMIENTO DE IMAGEN .....</b>		<b>104</b>

## ÍNDICE DE IMÁGENES

ILUSTRACIÓN 1: REALIDAD DE MILGRAM-VIRTUALITY CONTINUUM .....	19
ILUSTRACIÓN 2: SENSORAMA.....	19
ILUSTRACIÓN 3: DISPLAY DE CABEZA (HMD).....	20
ILUSTRACIÓN 4: PERSONA PROBANDO EL VIDEOJUEGO “ARQUEKE” .....	20
ILUSTRACIÓN 5: LOGO OFICIAL DE REALIDAD AUMENTADA .....	21
ILUSTRACIÓN 6: GOOGLE GLASS .....	21
ILUSTRACIÓN 7: ACELEROMETROS REPRESENTANDO LA ORIENTACION TRIDIMENSIONAL .....	22
ILUSTRACIÓN 8: ACELERÓMETRO DE UN SMARTPHONE .....	23
ILUSTRACIÓN 9: FUNCIONAMIENTO DEL SISTEMA GPS .....	24
ILUSTRACIÓN 10: DETECCIÓN DE UN OBJETO .....	24
ILUSTRACIÓN 11: MARCADORES DE PLANTILLA Y MATRICES DE DATOS .....	25
ILUSTRACIÓN 12: EJEMPLO DE UN QR.....	26
ILUSTRACIÓN 13: MODELO DE ROTACIONES EN UN RECONOCIMIENTO BASADO EN GEO LOCALIZACIÓN.....	29
ILUSTRACIÓN 14: LIBRO QUE UTILIZA AR.....	30
ILUSTRACIÓN 15: INGENIEROS USM APLICAN REALIDAD AUMENTADA PARA “VER” A TRAVÉS DEL CRÁNEO.....	30
ILUSTRACIÓN 16: TRANSMISIÓN DE UN CANAL DE TELEVISION UNA COMPETENCIA DE NATACIÓN .....	31
ILUSTRACIÓN 17: PERSONA JUGANDO "CAN YOU SEE ME NOW?" .....	32
ILUSTRACIÓN 18: LENTES DESARROLLADO POR AGENCIA DE PROYECTOS DE INVESTIGACIÓN AVANZADOS DE DEFENSA DE LOS ESTADOS UNIDOS (DARPA).....	32
ILUSTRACIÓN 19: VIRTUALIZACIÓN DE UN EDEIFICIO .....	33
ILUSTRACIÓN 20: WIKITUDE DRIVE, UNA APLICACIÓN QUE COMBINA LA NAVEGACIÓN GPS POR CARRETERA, CON MAPAS 2D Y REALIDAD AUMENTADA .....	33
ILUSTRACIÓN 21: PROCESO DE ENSAMBLAJE DE UN AUTOMOVIL.....	34
ILUSTRACIÓN 22: APLICACIÓN EN GEOLOGÍA EN LA RECOPIACIÓN DE DATOS.....	35
ILUSTRACIÓN 23: EJEMPLAR DE LA REVISTA ESQUIRE, CON UN CODIGO QR.....	36
ILUSTRACIÓN 24: APLICACION DE WIKITUDE QUE DA INFORMACIÓN DE PUNTOS DE INTERES EN UNA CIUDAD.....	36
ILUSTRACIÓN 25: LINEA DE TIEMPO DE LAS GENERACIONES DE LOS DISPOSITIVOS MOVILES. ....	38
ILUSTRACIÓN 26: SISTEMAS OPERATIVOS VENDIDOS EN EL MUNDO .....	43
ILUSTRACIÓN 27: LOGO DE WIKITUDE .....	47
ILUSTRACIÓN 28: ARQUITECTURA DE WIKITUDE.....	48
ILUSTRACIÓN 29: CICLO DE VIDA DE WIKITUDE.....	49
ILUSTRACIÓN 30: EJEMPLO DE RECONOCIMIENTO DE IMAGEN CON WIKITUDE.....	49
ILUSTRACIÓN 31: EJEMPLO DE GEO-LOCALIZACIÓN CON WIKITUDE.....	50



ILUSTRACIÓN 32: EJEMPLO DE UN MODELO 3D CON WIKITUDE.....	50
ILUSTRACIÓN 33: LOGO DE VUFORIA .....	50
ILUSTRACIÓN 34: ARQUITECTURA DE VUFORIA.....	52
ILUSTRACIÓN 35: EJEMPLO IMAGEN OBJETIVOS DE VUFORIA .....	52
ILUSTRACIÓN 36: EJEMPLO DE BOTONES VIRTUALES DE VUFORIA.....	53
ILUSTRACIÓN 37: FRAME DE VUFORIA .....	53
ILUSTRACIÓN 38: EJEMPLO DE PALABRA OBJETIVO CON VUFORIA .....	54
ILUSTRACIÓN 39: LOGO DE METAIO SDK.....	54
ILUSTRACIÓN 40: ARQUITECTURA DE METAIO SDK.....	55
ILUSTRACIÓN 41: EJEMPLO DE SEGIMIENTO DE IMÁGENES.....	55
ILUSTRACIÓN 42: EJEMPLO DE AR BASADA EN LA UBICACIÓN .....	56
ILUSTRACIÓN 43: EJEMPLO DE MUEBLES INTERACTIVO.....	56
ILUSTRACIÓN 44: GRÁFICO DE RESULTADO DEL PARÁMETRO - PLATAFORMA .....	59
ILUSTRACIÓN 45: GRÁFICO DE RESULTADO DEL PARÁMETRO - USO DEL GPS.....	61
ILUSTRACIÓN 46: GRÁFICO DE RESULTADO DEL PARÁMETRO - BÚSQUDA VISUAL.....	63
ILUSTRACIÓN 47: GRÁFICO DE RESULTADO DEL PARÁMETRO - CONTENIDO DE LAS API.....	65
ILUSTRACIÓN 48: GRÁFICO DE RESULTADO DEL PARÁMETRO – FRAMEWORK .....	67
ILUSTRACIÓN 49: GRÁFICO DE RESULTADO DEL PARÁMETRO - LICENCIA .....	69
ILUSTRACIÓN 50: GRÁFICO DE PONDERACIONES DE LOS PARÁMETROS DE COMPARACIÓN .....	70
ILUSTRACIÓN 51: MAQUETA DE PANTALLA CON POI EN LA UNIVERSIDAD.....	77
ILUSTRACIÓN 52: MAQUETA DE PANTALLA CUANDO UN POI ES SELECCIONADO.....	77
ILUSTRACIÓN 53: POI SIN MARCAR.....	80
ILUSTRACIÓN 54: POI SELECCIONADO.....	81
ILUSTRACIÓN 55: PUNTERO INDICANDO DONDE ESTA EL POI SELECCIONADO.....	81
ILUSTRACIÓN 56: MAQUETA DE PANTALLA DEL CASO PRÁCTICO DE RECONCIMIENTO DE IMAGEN.....	83
ILUSTRACIÓN 57: DIFERENTES CAPTURAS DE PANTALLAS DE LA APLICACIÓN FUNCIONANDO .....	85
ILUSTRACIÓN 58: CLASE MAINACTIVITY .....	95
ILUSTRACIÓN 59: MÉTODO ONCREATE ( ).....	96
ILUSTRACIÓN 60: MÉTODO ONCREATE ( ) 2 .....	97
ILUSTRACIÓN 61: MÉTODOS ONPAUSE Y ONDESTROY ( ).....	98
ILUSTRACIÓN 62: MÉTODOS ONPOSTCREATE ( ) .....	98
ILUSTRACIÓN 63: HTML QUE MANEJA LOS JAVASCRIPT .....	99
ILUSTRACIÓN 64: JAVASCRIPT EXTRAER LOS DATOS DE JSON 1 .....	99
ILUSTRACIÓN 65: JAVASCRIPT EXTRAER LOS DATOS DE JSON 2 .....	100
ILUSTRACIÓN 66: JAVASCRIPT DEL RADAR .....	100
ILUSTRACIÓN 67: JAVASCRIPT DE JSON.....	101

ILUSTRACIÓN 68: JAVASCRIPT DE LA FORMA DE LOS POI 1.....	101
ILUSTRACIÓN 69: JAVASCRIPT DE LA FORMA DE LOS POI 2.....	102
ILUSTRACIÓN.70: CLASE TUTORIAL1, ENCARGADO DE CONFIGURAR LA AR.....	104
ILUSTRACIÓN 71: ESTRUCTURA GENERAL DE LA CLASE MAINACTIVITY.....	105
ILUSTRACIÓN 72: MÉTODOS DEL CICLO DE VIDA DE LA APLICACIÓN .....	106
ILUSTRACIÓN 73: MÉTODO QUE EXTRAE LOS RECURSOS UTILIZADOS POR LA APLICACIÓN .....	107
ILUSTRACIÓN 74: HTML DE LA APLICACIÓN .....	108
ILUSTRACIÓN 75: CONFIGURACIÓN XML DE SEGUIMIENTO .....	108
ILUSTRACIÓN 76: PARÁMETRO DEL SENSOR: ALINEACIÓN.....	109
ILUSTRACIÓN 77: PARÁMETRO DEL SENSOR: NÚMERO DE OBJETO DETECTADO POR CUADROS.....	109
ILUSTRACIÓN 78: PARÁMETRO DEL SENSOR: MAX DE OBJETOS A SEGUIR.....	109
ILUSTRACIÓN 79: PARÁMETROS DEL SENSOR: UMBRAL.....	110
ILUSTRACIÓN 80: DEFINICIÓN DE UNA "SENSORCOS" .....	110
ILUSTRACIÓN 81: DEFINICIÓN DE CONEXION ENTRE ENTIDADES .....	111
ILUSTRACIÓN 82: DEFINICIÓN DEL TIPO DE FUSOR.....	111
ILUSTRACIÓN 83: PARÁMETRO DEL FUSOR: NÚMERO DE FOTOGRAMAS A SEGUIR.....	111
ILUSTRACIÓN 84: PARÁMETRO DEL FUSOR: TRASLACIÓN ALPHA .....	111
ILUSTRACIÓN 85: PARÁMETRO DEL FUSOR: TRASLACIÓN GAMMA .....	111
ILUSTRACIÓN 86: PARÁMETRO DEL FUSOR: ROTACIÓN ALPHA .....	112
ILUSTRACIÓN 87: PARÁMETRO DEL FUSOR: ROTACIÓN GAMMA .....	112
ILUSTRACIÓN 88: PARÁMETRO DEL FUSOR: SI LA BUSQUEDA SE PIERDE CON EL CAMBIO DE ORENTACIÓN.....	112
ILUSTRACIÓN 89: ESTABLECER CALIBRACIÓN DE LOS SENSORES.....	112
ILUSTRACIÓN 90: CALIBRACIÓN DE LOS SENSORES DE TIPO "MANO-OJO" .....	113
ILUSTRACIÓN 91: CALIBRACIÓN DE LOS SENSORES DE TIPO "RÍGIDA" .....	113

## ÍNDICE DE TABLAS

TABLA 1: INFORMACIÓN GENERAL DE ANDROID .....	41
TABLA 2: INFORMACIÓN GENERAL DE IOS .....	42
TABLA 3: INFORMACIÓN GENERAL BLACKBERRY .....	42
TABLA 4: INFORMACIÓN GENERAL WINDOWS PHONE.....	43
TABLA 5: ESCALA EVALUACIÓN HERRAMIENTAS AR.....	58
TABLA 6: DEFINICIÓN DE VARIABLES DEL PARÁMETROS – PLATAFORMA.....	58
TABLA 7: VALORACIÓN DE VARIABLES DEL PARÁMETRO - PLATAFORMA.....	58
TABLA 8: INTERPRETACIÓN DE RESULTADOS – PLATAFORMA .....	59
TABLA 9: DEFINICIÓN DE VARIABLES DEL PARÁMETRO - USO DEL GPS .....	60
TABLA 10: VALORACIÓN DE VARIABLES DEL PARÁMETRO - USO DEL GPS .....	60
TABLA 11: IMPLEMENTACIÓN DE RESULTADOS - USO DEL GPS .....	60
TABLA 12: DEFINICIÓN DE VARIABLES DEL PARÁMETRO - BÚSQUEDA VISUAL.....	62
TABLA 13: VALORACIÓN DE VARIABLES DEL PARÁMETRO – BÚSQUEDA VISUAL.....	62
TABLA 14: IMPLEMENTACIÓN DE RESULTADOS - BÚSQUEDA VISUAL.....	62
TABLA 15: DEFINICIÓN DE VARIABLES DEL PARÁMETRO - CONTENIDO DE LAS API.....	64
TABLA 16: VALORACIÓN DE VARIABLES DEL PARÁMETRO – CONTENIDO DE LAS API.....	64
TABLA 17: IMPLEMENTACIÓN DE RESULTADOS - CONTENIDO DE LAS API.....	64
TABLA 18: DEFINICIÓN DE VARIABLES DEL PARÁMETRO – FRAMEWORK.....	66
TABLA 19: VALORACIÓN DE VARIABLES DEL PARÁMETRO – FRAMWORK.....	66
TABLA 20: IMPLEMENTACIÓN DE RESULTADOS - FRAMEWORK.....	66
TABLA 21: DEFINICIÓN DE VARIABLES DEL PARÁMETRO – LICENCIA .....	68
TABLA 22: VALORACIÓN DE VARIABLES DEL PARÁMETRO - LICENCIA.....	68
TABLA 23: INTERPRETACIÓN DE RESULTADOS - LICENCIA .....	68
TABLA 24: PUNTAJES TOTALES ALCANZADOS.....	71
TABLA 25: RESUMEN DE CARACTERISTICAS DE LOS SDK ANALIZADOS .....	73

# **CAPÍTULO 1: INTRODUCCIÓN**

## **1.1.Introducción General**

Este proyecto tiene como propósito obtener el título profesional de ingeniero civil en informática, demostrando que el autor tiene las aptitudes y conocimientos necesarios para optar a dicho título a través de la investigación llamada “ESTUDIO DE HERRAMIENTAS DE REALIDAD AUMENTADA SOBRE PLATAFORMA ANDROID”, la cual consiste en comparar diferentes herramientas de desarrollo de este tipo de tecnología en plataforma android para posteriormente implementar prototipos.

La realidad aumentada (AR) es el término que se usa para definir una visión directa o indirecta de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real.

El principal motivo para realizar este proyecto sobre la plataforma android, se debe a que hoy en día se encuentra en la mayoría de los Smartphone de nuestro país y siempre están apareciendo aplicaciones novedosas que captan el interés de los usuarios; el autor como postulante a ingeniero civil en informática tiene la responsabilidad de estudiar estas nuevas tecnologías que ocupan estas aplicaciones y poder implementarlas.

Con la constante evolución de los teléfonos inteligentes, se pueden materializar ideas que años atrás sería impensable realizar, debido que no existían los recursos para construirlas, este es el caso de la Realidad Aumentada.

Además con esta investigación, se puede crear un punto de comienzo para futuros proyectos que deseen realizar aplicaciones android con realidad aumentada y encontrar toda la información necesaria en un solo documento, y así ahorrar tiempo en la recopilación datos.

Por último, al construir prototipos, es una muestra empírica que la tecnología es realizable por cualquier programador que tenga la información de las librerías necesarias para poder obtener un producto.

Este documento consta de cuatro capítulos principales, el primero es el marco teórico del proyecto, el cual se centra en el contexto de la tecnología de realidad aumentada, respondiendo a la pregunta “¿En qué se apoya esta tecnología?”. El segundo capítulo del documento es el estudio comparativo de tecnologías de realidad aumentada, donde se

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Introducción

presentan las herramientas seleccionadas, para luego realizar el análisis comparativo e interpretar los resultados obtenidos durante el desarrollo de esta actividad. El tercer capítulo es llamado “Casos prácticos”, en el que se muestra la documentación de los prototipos creados con las herramientas analizadas para este estudio. Por último, el cuarto capítulo son las conclusiones generales y específicas, después de haber realizado el estudio e implementación de los prototipos, llegando a recomendar una herramienta dependiendo de la situación que se presenta al desarrollador, además de nombrar los trabajos futuros a realizar, terminado esta investigación.

### **1.2. Antecedentes del proyecto**

Actualmente no se han realizados trabajos similares en la universidad del Bío-Bío. Este proyecto aportará nueva información sobre la tecnología AR a estudiantes y/o académicos que les interese crear aplicaciones móviles añadiendo esta tecnología en estudio.

En nuestro país, el diario “Emol” creó una aplicación para plataforma iPhone (de la empresa Apple) con AR, llamada “Emol AR”, la cual consiste en encontrar “puntos de interés” del usuario, como bancos, cajeros automáticos, colegios, bencineras, farmacias, hospitales, cines, restaurantes, centros comerciales, entre otros; a través de la cámara del celular e indicando información adicional de esos “puntos”.

En el mundo, el estandarte de la AR es el Google Glass, que tiene como propósito crear un asistente virtual al usar lentes o gafas de alta tecnología. En pocas palabras: mostrar información disponible para los usuarios de teléfono inteligente sin utilizar las manos, permitiendo también el acceso a Internet mediante órdenes de voz, pero este dispositivo se encuentra en fase final de desarrollo y estará disponible para los usuarios a partir del año 2014.

### **1.3. Objetivos**

#### **a) General:**

Este estudio consiste en analizar diferentes herramientas de desarrollo para la realidad aumentada sobre plataforma android e implementación de un prototipo.

#### **b) Específicos:**

- Investigar las características de las diferentes herramientas de desarrollo.
- Comparar requerimientos y características de las herramientas de desarrollo.

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Introducción

- Implementar prototipos de Realidad Aumentada con la herramienta adecuada.

### 1.4. Metodología

La metodología utilizada en este proyecto es la investigación experimental sin hipótesis previa, la cual se emplea en aquellos casos en donde la indagación tiene por objetivo dar a conocer tecnologías nuevas y que puede ser un importante avance en el área. Constituye un método de investigación muy importante, tanto por su amplitud como por su utilización en el campo tecnológico [3].

Las etapas de la metodología elegida son:

- E1) Delimitar y definir el objeto de la investigación:** Consiste en determinar claramente los objetivos de la investigación y además, decidir el número de herramientas a estudiar.
- E2) Elaborar el diseño experimental:** Ya conocido los objetos de estudios, la precisión deseada y el equipo adecuado, se debe analizar si el estudio va a ser la interpretación de una gráfica, un valor o una relación empírica; esto señalará el procedimiento experimental, es decir ¿cómo medir?, ¿en qué orden?, y ¿qué precauciones tomar al hacerlo?.
- E3) Realizar el experimento:** Consiste en realizar prototipos con la tecnología y observar el comportamiento de ellos según los parámetros establecidos en el punto anterior.
- E4) El análisis o interpretación de resultados:** Ya sean valores, gráficas, tabulaciones, etcétera, debe contestar lo más claramente posible la o las preguntas planteadas por la investigación discriminando entre los modelos y los resultados deben permitir hacer la discriminación en forma tajante y proporcionar los motivos para aceptar uno y rechazar otro.
- E5) Obtener conclusiones.** Ya logrados los resultados del experimento el investigador debe aplicar su criterio científico para recomendar qué modelo ocupar o descartar en las situaciones vividas a lo largo del estudio.

### 1.5. Alcances y Limitaciones

La investigación se centra en describir las características de las herramientas de desarrollo para AR que existen actualmente, además de indicar los requerimientos técnicos necesarios para utilizar las herramientas estudiadas.

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Introducción

Para realizar la comparación entre las herramientas se seleccionan distintos parámetros que fueron escogidos durante el proceso de investigación y se justifica el “¿por qué?” se eligió esa medida.

Al final de la investigación se presentan las conclusiones extraídas del estudio y se sugiere al lector qué herramienta utilizar en las diferentes situaciones que se puedan presentar al momento de implementación de una aplicación con realidad aumentada.

Finalmente, se desarrollan prototipos de realidad aumentada con una de las herramientas estudiadas, de manera de proporcionar una guía de desarrollo para los futuros programadores.

### 1.6. Definiciones y abreviaturas

- **Tecnología:** Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico.
- **Realidad Aumentada (AR):** Término que se usa para definir una visión directa o indirecta de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real [30].
- **Realidad Virtual (VR):** Ciencia basada en el empleo de ordenadores y otros dispositivos, cuyo fin es producir una apariencia de realidad que permita al usuario tener la sensación de estar presente en ella [29].
- **Android:** Sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles [26].
- **Prototipo:** Ejemplar original o primer molde en que se fabrica una figura u otra cosa [17].
- **Kit de desarrollo de software (SDK):** Herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto [28].
- **Ambiente integrado de desarrollo (IDE):** Es un tipo de SDK e incluye un conjunto de herramientas de desarrollo en una aplicación de programación elegante [9].
- **Renderizado:** Es un término usado en jerga informática para referirse al proceso de generar una imagen o vídeo mediante el cálculo de iluminación indirecta (GI) partiendo de un modelo en 3D [31].
- **Punto de interés (POI):** En inglés Points of interest, marcador que se coloca en un punto de geográfico específico.



Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Introducción

- **Time to market (TTM):** Término para el período de tiempo entre las primeras ideas alrededor de un producto y su eventual disponibilidad en los mercados de consumo.
- **Open Graphics Library (OpenGL):** Es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.
- **JavaScript Object Notation (JSON):** Es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

# **CAPÍTULO 2: MARCO TEÓRICO**

## 2.1. Realidad Aumentada

### 2.1.1. Introducción

Realidad aumentada es el término que se usa para definir una visión directa o indirecta de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real. Consiste en un conjunto de dispositivos que añaden información virtual a la información física ya existente, es decir, añadir una parte sintética virtual a lo real. Esta es la principal diferencia con la realidad virtual, puesto que no sustituye la realidad física, sino que unen los datos informáticos y/o imágenes al mundo real.

AR abarca tres principales características [14]:

- **Inserción de imágenes virtuales en los entornos reales:** Integrar información en el ambiente real mediante objetos virtuales, ya sea mediante animaciones o imágenes estáticas sobre el video grabado del entorno mediante la cámara del dispositivo.
- **Reconocimiento de objetos en tercera dimensión:** Sobreponer información de los objetos en el entorno, colocándola de modo precisa aplicando técnicas de reconocimiento de los elementos del ambiente.
- **Interactividad en tiempo real:** Característica que destaca a la AR, debido a que mientras el usuario interactúa con el entorno real puede interactuar con los objetos virtuales al mismo tiempo.

Con la ayuda de dispositivos, la información sobre el mundo real alrededor del usuario es capturada y digitalizada. Esta información sobre el medio ambiente y los objetos, puede ser almacenada y recuperada.

AR también es la incorporación de datos e información digital en un entorno real, por medio del reconocimiento de patrones que se realiza mediante un software, en otras palabras, es una herramienta interactiva que está dando sus primeros pasos alrededor del mundo y alcanzando todas las disciplinas: videojuegos, medios masivos de comunicación, arquitectura, educación e incluso en la medicina, trayendo un mundo digital inimaginable a nuestro entorno real. Su gran diferencia con la realidad virtual, es que ésta nos extrae de nuestro entorno para llevarnos a una realidad paralela.

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

Una definición de Realidad Aumentada fue realizada por Ronald Azuma en 1997 [2], la cual dice que la AR combina elementos reales y virtuales, es interactiva en tiempo real y está registrada en 3D.

Además, Paul Milgram y Fumio Kishino definen en 1994 la realidad de Milgram-Virtuality Continuum (Ilustración 1) como un continuo que abarca desde el entorno real a un entorno virtual puro [13]. En medio hay Realidad Aumentada (más cerca del entorno real) y Virtualidad Aumentada (más cerca del entorno virtual).



ILUSTRACIÓN 1: REALIDAD DE MILGRAM-VIRTUALITY CONTINUUM

**2.1.2. Cronología de la Realidad Aumentada**

A continuación se muestra la evolución de la realidad aumentada, comenzando por la realidad virtual y terminando con lo más actual de aplicaciones de AR, como es “GoogleGlass”

- **1962:** Morton Heilig, un director de fotografía, crea un simulador de moto llamado Sensorama (Ilustración 2) con imágenes, sonido, vibración y olfato.

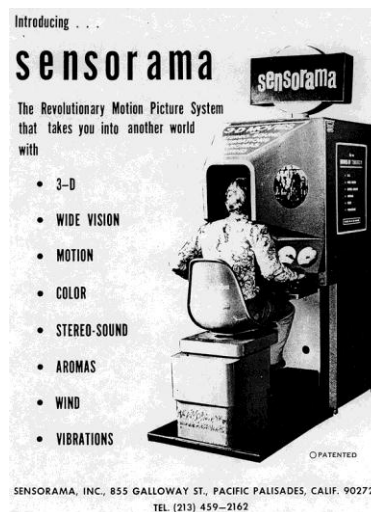


ILUSTRACIÓN 2: SENSORAMA

- **1973:** Ivan Sutherland inventa la display de cabeza, HMD (Ilustración 3), lo que sugiere una ventana a un mundo virtual.

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico



ILUSTRACIÓN 3: DISPLAY DE CABEZA (HMD)

- **1985:** Myron Krueger crea Videoplance que permite a los usuarios interactuar con objetos virtuales por primera vez.
- **1990:** Jaron Lanier acuña el término realidad virtual y crea la primera actividad comercial en torno a los mundos virtuales.
- **1992:** Tom Caudell crea el término Realidad Aumentada.
- **1994:** Steven Feiner, Blair MacIntyre y Doree Seligmann primera utilización importante de un sistema de Realidad Aumentada en un prototipo, KARMA, presentado en la conferencia de la interfaz gráfica. Ampliamente citada en la publicación Communications of the ACM al siguiente año.
- **1999:** Hirokazu Kato desarrolla ARToolKit en el HitLab y se presenta en SIGGRAPH ese año.
- **2000:** Bruce H. Thomas desarrolla ARQuake (Ilustración 4), el primer juego al aire libre con dispositivos móviles de Realidad Aumentada, y se presenta en el International Symposium on Wearable Computers.



ILUSTRACIÓN 4: PERSONA PROBANDO EL VIDEOJUEGO “ARQUEKE”

- **2008:** AR Wikitude Guía sale a la venta el 20 de octubre de 2008 con el teléfono Android G1.

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Marco Teórico

- **2009:** AR Toolkit es portado a Adobe Flash (FLARToolkit) por Saqoosha, con lo que la realidad aumentada llega al navegador Web.
- **2009:** Se crea el logo oficial de la Realidad Aumentada (Ilustración 5) con el fin de estandarizar la identificación de la tecnología aplicada en cualquier soporte o medio por parte del público general. Desarrolladores, fabricantes, anunciantes o investigadores pueden descargar el logo original desde la web oficial.



ILUSTRACIÓN 5: LOGO OFICIAL DE REALIDAD AUMENTADA

- **2012:** Mahe Innovation crea una aplicación móvil de AR capaz de interactuar con libros y juguetes.
- **2012:** Google se lanza al diseño de unas gafas que crearían la primera realidad aumentada comercializada. Bautiza a su proyecto como Project Glass (Ilustración 6).



ILUSTRACIÓN 6: GOOGLE GLASS

### 2.1.3. Tecnología empleada actualmente

Actualmente los dispositivos de AR constan de dos partes, la primera es la física (hardware), la cual se encarga en capturar la información proveniente del medio, para luego ser combinada con la información adicional. La segunda es la intangible (software), para que la información sea añadida, coherentemente del mundo real.

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

**2.1.3.1. Hardware:**

Los dispositivos normalmente constan de sensores necesarios para localizar con precisión la situación y un sistema de display para mostrar al usuario la información virtual que se añade a la real, usando imágenes virtuales que se muestran al usuario mezcladas con la realidad o bien proyectadas directamente en la pantalla.

Los Sistemas de realidad aumentada modernos utilizan una o más de las siguientes tecnologías: sensores ópticos, acelerómetros, GPS, cámaras digitales, giroscopios, brújulas de estado sólido, identificación por radiofrecuencia, etcétera. Los Sistemas de cámaras basadas en AR requieren de un procesador potente y gran cantidad de memoria RAM para procesar imágenes de dichas cámaras. La combinación de todos estos elementos se da a menudo en los Smartphone modernos, que los convierten en una posible plataforma de realidad aumentada.

A continuación se profundiza, cómo funcionan las tecnologías más comunes que tienen los teléfonos inteligentes:

- **Acelerómetros:**

Los acelerómetros corrientes consta de dos partes una base o cubierta unida al objeto que queremos medir la aceleración y una masa, que aun estando unida a la base, puede moverse. Si se mueve la base hacia arriba, la masa se queda atrás estirando el resorte que lo sostiene. Si se mide la distancia que el soporte se estira se puede calcular la fuerza de gravedad. Además, si se tienen tres de estos dispositivos se podrá determinar la orientación de un objeto tridimensional (correspondiente al eje X, Y, Z) [7].

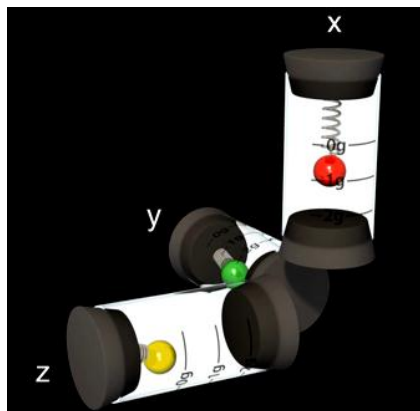


ILUSTRACIÓN 7: ACCELEROMETROS REPRESENTANDO LA ORENTACION TRIDIMENSIONAL

Pero en los Smartphone es más complejo que una simple masa y un resorte, pero tiene los mismos elementos fundamentales. Se creó un acelerómetro hecho de silicio (Ilustración 8), que cuenta con una base que está unida al teléfono y una sección similar a un peine, llamada masa sísmica, que representa a la masa en los corrientes. La flexibilidad del silicio juega el papel del resorte [7].

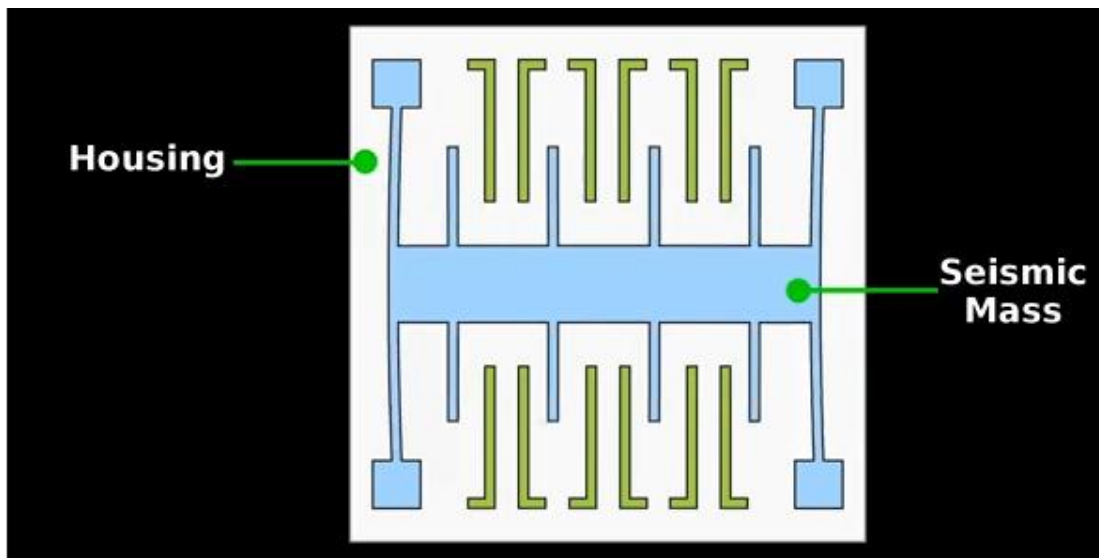


ILUSTRACIÓN 8: ACELERÓMETRO DE UN SMARTPHONE

- **Sistema de posicionamiento global (GPS)**

El sistema GPS se basa en el principio matemático de la triangulación, es decir sabiendo la distancia que separa al receptor de 4 satélites (como mínimo) en diferentes posiciones se puede determinar la posición del dispositivo a través de la intersección de circunferencias cuyo radio es dicha distancia entre el receptor y el satélite (Ilustración 9). Por tanto, para calcular la posición de un punto será necesario que el receptor GPS determine con exactitud la distancia que lo separa de los satélites [5].



Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Marco Teórico

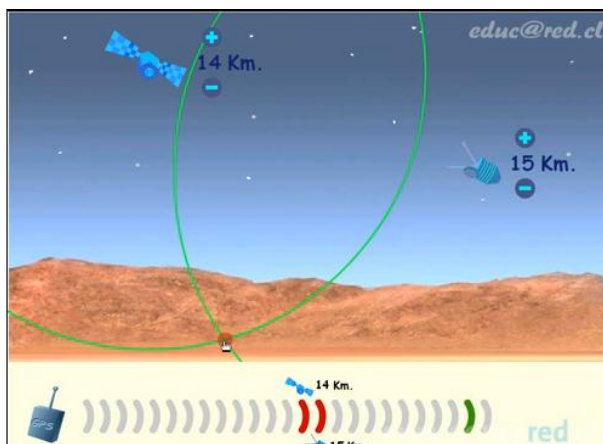


ILUSTRACIÓN 9: FUNCIONAMIENTO DEL SISTEMA GPS

Cada satélite transmite una señal que es recibida por el receptor, éste, por su parte mide el tiempo que las señales tardan a llegar hasta él. Multiplicando el tiempo medido por la velocidad de la señal (la velocidad de la luz), obtenemos la distancia receptor-satélite, (Distancia= Velocidad x Tiempo).

- **Sensores ópticos:**

Los detectores ópticos basan su funcionamiento en la emisión de un haz de luz que es interrumpido o reflejado por el objeto a detectar (Ilustración 10). Para lograr que objetos poco brillantes puedan ser detectados, el haz de luz no se trasmite en una sola dirección, sino que viaja en varias direcciones [8].

Esta configuración presenta la desventaja de tener muy corta distancia de detección, pero es muy útil cuando es difícil acceder ambos lados del objeto.

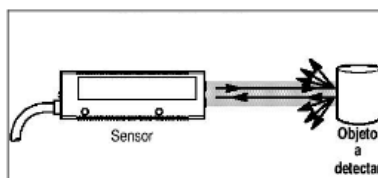


ILUSTRACIÓN 10: DETECCIÓN DE UN OBJETO

**2.1.3.2. Software:**

Para fusiones coherentes entre las imágenes del mundo real e imágenes virtuales en 3D, las imágenes virtuales deben atribuirse a lugares del mundo real. Ese mundo real debe ser

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Marco Teórico

situado, a partir de imágenes de la cámara, en un sistema de coordenadas. Dicho proceso se denomina registro de imágenes. Este proceso usa diferentes métodos de visión por ordenador, en su mayoría relacionados con el seguimiento de vídeo.

### 2.1.4. Clasificación de AR

Los sistemas de AR requieren de un reconocimiento exacto de la orientación y el posicionamiento del usuario para poder ajustar la información virtual en los objetos reales, utilizando técnicas para identificarlos en el entorno. Estas técnicas se clasifican dependiendo de las estrategias y tecnología ocupadas [18]:

#### 2.1.4.1. Reconocimiento de objetos mediante marcadores:

Un marcador visual es una imagen que contiene algún código preestablecido, el cual puede ser decodificado por algún dispositivo para el reconocimiento de objetos, transferencias de información, etcétera [14].

Estos marcadores se almacenan en una base de datos sea local o externa contra la cual se compara con lo obtenido por la cámara con el objeto de tener una coincidencia; cada marcador lleva asociado algún tipo de información para mostrar [18].

Tipos de marcadores son los siguientes:

- **Marcadores de plantilla y matrices de datos:**

Los marcadores de plantilla son aquellos que se diseñan con alguna figura en particular, especialmente diseñada para reconocer objetos en el entorno, pueden facilitar la interpretación de estos por los usuarios. Las matrices de datos son capaces de almacenar códigos binarios de manera bidimensional, comúnmente se emplean matrices de 6x6 capaces de generar 4096 marcadores distintos, también se han llegado a implementar matrices de datos de 144x144 capaces de almacenar hasta 1558 bytes [14].



ILUSTRACIÓN 11: MARCADORES DE PLANTILLA Y MATRICES DE DATOS

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

- **Quick Response Code (QR):**

Existen los códigos de respuesta rápida, los cuales representan un código bidimensional con el cual se puede codificar texto alfanumérico, códigos numéricos e imágenes, la especificación permite codificar tres clases de información: QR's con 7,089 caracteres numéricos, con 4,296 caracteres alfanuméricos y con 2,953 códigos binarios (imágenes), En la Ilustración 12 puede apreciarse un ejemplo de un QR [14].



ILUSTRACIÓN 12: EJEMPLO DE UN QR

Para reconocer la información que se encuentra dentro del marcador se utiliza un algoritmo, el cual consta de cinco fases descritas de la siguiente forma:

- **Umbralización:** En este proceso se eliminan las sombras y los otros colores capturados del entorno, conservando únicamente aquellos colores que permitan identificar al marcador.
- **Detección de marcadores:** Una vez que se ha obtenido una imagen con la umbralización, deberán detectarse todas las figuras que puedan formar un marcador, en este proceso se obtendrán diversas formas sin importar que no formen un rectángulo o la forma establecida del marcador.
- **Definición de la forma:** Una vez obtenidas las figuras candidatas, se someten a una verificación que comprueba que cumplen con la forma de los marcadores, en caso de que los marcadores sean cuadrados, se descartarán todas las figuras que no tengan las propiedades de la forma del marcador.
- **Verificación de patrones:** Una vez obtenidas las figuras de los marcadores, puede identificarse el patrón que este contiene, este proceso permitirá reconocer el marcador y añadir los objetos virtuales correspondientes

- **Estimación de la orientación:** Finalmente puede obtenerse una orientación del dispositivo con respecto al marcador, esto puede hacerse al detectar alguna inclinación en la imagen capturada del marcador

#### 2.1.4.2. Reconocimiento de objetos mediante formas:

En este tipo de sistemas a diferencia del punto anterior, ya no busca códigos QR o marcas determinadas, sino formas conocidas. De igual manera lo capturado por la cámara debe ser comparado con una base de datos para tener una coincidencia de formas y para mostrar la información asociada [18].

Las técnicas de reconocimiento de objetos mediante formas se utilizan en los sistemas inteligentes para facilitar la recolección de los datos y la toma de decisiones, este tipo de técnica se encarga de clasificar objetos, capturados en alguna escena o imagen capturada, en cierto número de categorías o clases.

Muchos sistemas de este tipo se basan en la identificación de formas, bordes, o puntos clave de los objetos en el entorno, donde es necesario aplicar un algoritmo de tratamiento de imágenes por visión artificial. Dicho algoritmo propuesto por [10], la cual se extraen los bordes de una imagen para el reconocimiento de objetos y, posteriormente, para la inserción de información virtual de ese objeto [14].

- **Extracción de la estructura del objeto:** Es necesario que primero se obtenga una imagen de algún objeto mediante la cámara del dispositivo, en este paso también se identificarán los bordes y puntos clave que puedan formar alguna figura que pueda ayudar a reconocer el objeto.
- **Acceso a los datos 3D-CAD:** Una vez obtenidas los bordes, puntos o figuras candidatas a formar un objeto, se accede a los modelos creados en programas de diseño asistido mediante computadora (CAD), estos modelos almacenan las formas de los objetos del entorno que pueden ser identificados, de esta forma ayudarán a comparar las figuras obtenidas en el primer paso para determinar si se trata de algún objeto contextualizado.
- **Detección de figuras auxiliares:** Una vez obtenido el modelo 3D-CAD, se compara este modelo con las figuras obtenidas en la captura de la imagen, de esta forma se obtendrán todas las coincidencias con el modelo, obteniendo una primera forma del objeto.

- **Reducción de polígonos:** Ya que se han obtenido los bordes, puntos clave y figuras que coinciden con el modelo 3D-CAD, aquellos que no hayan coincidido con el modelo serán descartados, reduciendo las figuras capturadas para simplificar la forma del objeto y su reconocimiento.
- **Extracción de bordes dominantes:** Ya que se cuenta únicamente con las figuras que coinciden con el modelo, se definen los bordes y superficies que formarán al objeto.
- **Definición de relaciones:** Una vez definidos los bordes y las superficies capturadas, se procede a formar relaciones entre estos para definir el objeto, esto permite reconocer finalmente al objeto
- **Conversión de datos:** Finalmente, este objeto puede pasarse a un formato de reconocimiento, en donde pueda devolverse una clave de identificación, que le pueda indicar al sistema cómo reaccionar una vez que se ha reconocido el objeto, en algunos casos este objeto puede llegar a almacenarse para enriquecer los modelos 3D-CAD.

#### 2.1.4.3. Reconocimientos de objetos mediante geo localización:

Los sistemas basados en geo localización utilizan el sistema GPS para obtener la posición geográfica del usuario, por medio del compás y el acelerómetro del dispositivo, con esto sabe hacia qué lugar está apuntando la cámara. Las imágenes virtuales que se proyectan están basada en coordenadas de tal manera que si están apuntando hacia ellas se visualizará con información asociada [18].

El reconocimiento en base de los sensores de orientación (GPS, acelerómetro, etcétera), el cual después se complementa con un reconocimiento de formas para lograr una mayor precisión en la correspondencia de la información virtual con los objetos reales.

En Mulloni, 2010 [15] se propuso un método con el cual puede saberse si el dispositivo se encuentra viendo un POI a partir de su orientación, en la Ilustración 13 puede apreciarse el modelo de orientación de un dispositivo con respecto a un POI, como puede verse, existen tres rotaciones R definidas por los acelerómetros de los ejes X, Y y Z. Una rotación del dispositivo con respecto al norte (RDN), la rotación del POI con respecto al norte (RPN) y finalmente una rotación del dispositivo con respecto al POI (RDP), el norte se define con las mediciones del magnetómetro del dispositivo.

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

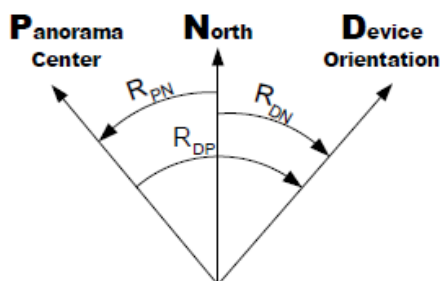


ILUSTRACIÓN 13: MODELO DE ROTACIONES EN UN RECONOCIMIENTO BASADO EN GEO LOCALIZACIÓN

Para que este modelo funcione es necesario establecer una localización del usuario con el entorno, ya que la orientación de los POIs puede variar de acuerdo a la ubicación del usuario en el entorno, una vez establecida la localización del usuario, se toman los datos de las rotaciones RPN, donde se encontrarán referenciados los POIs. Cada vez que el usuario se encuentre en la localización definida se obtiene la rotación del dispositivo RDN, esta rotación obtenida puede compararse con la rotación de los POIs RPN y obtener la rotación RDP, esta última rotación definirá la rotación que existe entre el punto de interés y el campo de visión del dispositivo, cuando la orientación del campo de visión del dispositivo este cerca de un POI, podrá visualizarse la información contextual de este.

### 2.1.5. Aplicaciones

La realidad aumentada ofrece infinidad de nuevas posibilidades de interacción, que hacen que esté presente en muchos y varios ámbitos, como son la arquitectura, el entretenimiento, la educación, el arte, la medicina o las comunidades virtuales, a continuación se explicarán algunas de estas posibilidades.

#### 2.1.5.1. Proyectos educativos:

Actualmente, la mayoría de aplicaciones de realidad aumentada para proyectos educativos se usan en museos, exhibiciones, parques de atracciones temáticos. Estos lugares aprovechan las conexiones inalámbricas para mostrar información sobre objetos o lugares, así como imágenes virtuales como por ejemplo ruinas reconstruidas o paisajes tal y como eran en el pasado, Además de escenarios completos en realidad aumentada, donde se pueden apreciar e interactuar con los diferentes elementos en 3D, como partes del cuerpo.

En los últimos años la Realidad Aumentada está consiguiendo un protagonismo cada vez más importante en diversas áreas de conocimiento, mostrando la versatilidad y

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Marco Teórico

posibilidades que presenta esta nueva tecnología derivada de la Realidad Virtual. La capacidad de insertar objetos virtuales en el espacio real y el desarrollo de interfaces de gran sencillez, la han convertido en una herramienta muy útil para presentar determinados contenidos bajo las premisas de entretenimiento y educación.

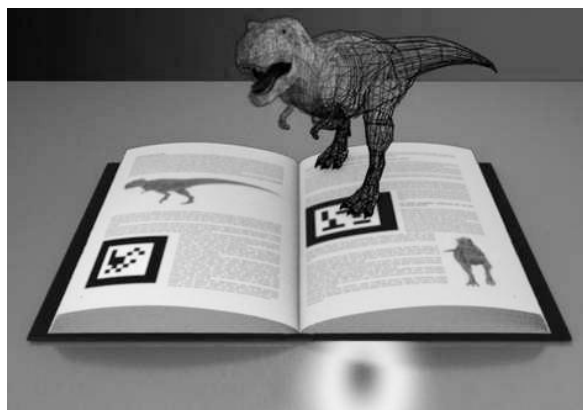


ILUSTRACIÓN 14: LIBRO QUE UTILIZA AR

### 2.1.5.2. Cirugía:

La aplicación de realidad aumentada en operaciones permite al cirujano superponer datos visuales como por ejemplo termografías o la delimitación de los bordes limpios de un tumor, invisibles a simple vista, minimizando el impacto de la cirugía [21].



ILUSTRACIÓN 15: INGENIEROS USM APLICAN REALIDAD AUMENTADA PARA “VER” A TRAVÉS DEL CRÁNEO

### 2.1.5.3. Televisión

La AR se ha vuelto común en la teledifusión de deportes, se utiliza en las transmisiones de fútbol para mostrar el resultado (o un anuncio) en el círculo central o para mostrar las situaciones de fuera de juego. Del mismo modo, en los partidos de hockey sobre hielo se coloreaba en Realidad Aumentada la ubicación y dirección de la pastilla (puck), aunque fue

rechazada por los puristas del hockey. Las transmisiones de natación suelen añadir una línea a través de los carriles para indicar la posición del poseedor del récord actual y compararla con la carrera (Ilustración 16).



ILUSTRACIÓN 16: TRASMISIÓN DE UN CANAL DE TELEVISION UNA COMPETENCIA DE NATACIÓN

#### 2.1.5.4. Entretenimiento:

Teniendo en cuenta que el mercado de los juegos que mueve unos 30.000 millones de dólares al año en los Estados Unidos, es comprensible que se esté apostando mucho por la realidad aumentada en este campo puesto que ésta puede aportar muchas nuevas posibilidades a la manera de jugar. Una de las puestas en escena más representativas de la realidad aumentada es el "Can You See Me Now?" de Blast Theory (Ilustración 17). Es un juego on-line de persecución por las calles donde los jugadores empiezan en localizaciones aleatorias de una ciudad, llevan un ordenador portátil y están conectados a un receptor de GPS. El objetivo del juego es procurar que otro corredor no llegue a menos de 5 metros de ellos, puesto que en este caso se les hace una foto y pierden el juego. La primera edición tuvo lugar en Sheffield pero después se repitió en otras muchas ciudades europeas. Otro de los proyectos con más éxito es el ARQuake Project, donde se puede jugar al videojuego Quake en exteriores, disparando contra monstruos virtuales. A pesar de estas aproximaciones, todavía es difícil obtener beneficios del mercado de los juegos puesto que el hardware es muy costoso y se necesitaría mucho tiempo de uso para amortizarlo.



Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

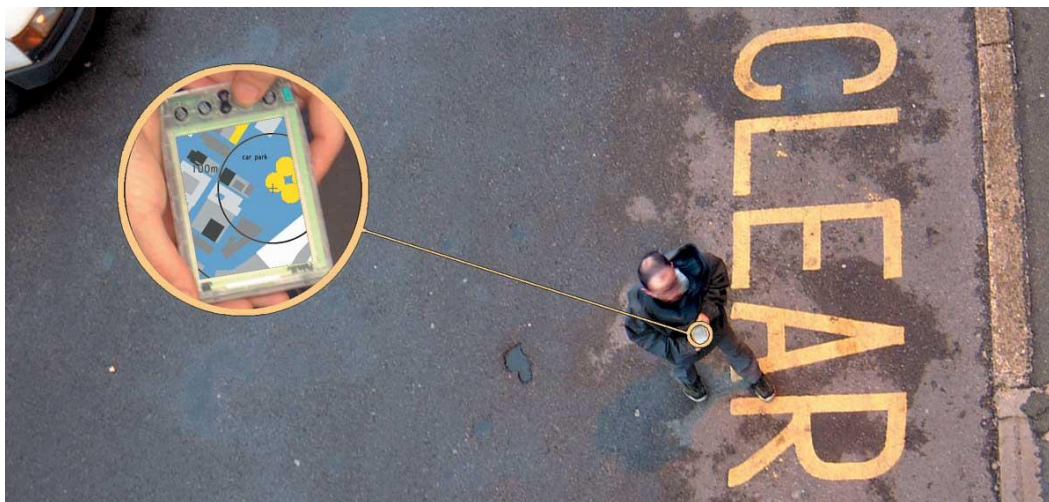


ILUSTRACIÓN 17: PERSONA JUGANDO "CAN YOU SEE ME NOW?"

**2.1.5.5. Simulación:**

Se puede aplicar la realidad aumentada para simular vuelos y trayectos terrestres.

Servicios de emergencias y militares: En caso de emergencia la realidad aumentada puede servir para mostrar instrucciones de evacuación de un lugar. En el campo militar, puede mostrar información de mapas, localización de los enemigos, etcétera

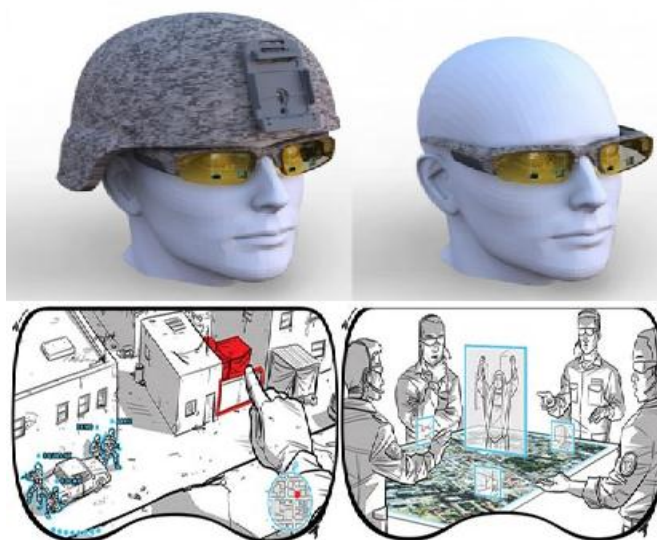


ILUSTRACIÓN 18: LENTES DESARROLLADO POR AGENCIA DE PROYECTOS DE INVESTIGACIÓN AVANZADOS DE DEFENSA DE LOS ESTADOS UNIDOS (DARPA)

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

**2.1.5.6. Arquitectura:**

La realidad aumentada es muy útil a la hora de resucitar virtualmente edificios históricos destruidos, así como proyectos de construcción que todavía están bajo plano.



ILUSTRACIÓN 19: VIRTUALIZACIÓN DE UN EDEFICIO

**2.1.5.7. Navegación:**

La AR puede mejorar la eficacia de los dispositivos de navegación para una variedad de aplicaciones. Por ejemplo, la navegación dentro de un edificio puede ser mejorada con el fin de dar soporte al encargado del mantenimiento de instalaciones industriales. Los parabrisas de los automóviles pueden ser usados como pantallas de visualización para proporcionar indicaciones de navegación e información de tráfico.



ILUSTRACIÓN 20: WIKITUDE DRIVE, UNA APLICACIÓN QUE COMBINA LA NAVEGACIÓN GPS POR CARRETERA, CON MAPAS 2D Y REALIDAD AUMENTADA

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

**2.1.5.8. Aplicaciones Industriales:**

La realidad aumentada puede ser utilizada para comparar los datos digitales de las maquetas físicas con su referente real para encontrar de manera eficiente discrepancias entre las dos fuentes. Además, se pueden emplear para salvaguardar los datos digitales en combinación con prototipos reales existentes, y así ahorrar o reducir al mínimo la construcción de prototipos reales y mejorar la calidad del producto final.



ILUSTRACIÓN 21: PROCESO DE ENSAMBLAJE DE UN AUTOMOVIL.

**2.1.5.9. Prospección:**

En los campos de la hidrología, la ecología y la geología, la AR puede ser utilizada para mostrar un análisis interactivo de las características del terreno. El usuario puede utilizar, modificar y analizar, tres mapas bidimensionales interactivos.

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Marco Teórico

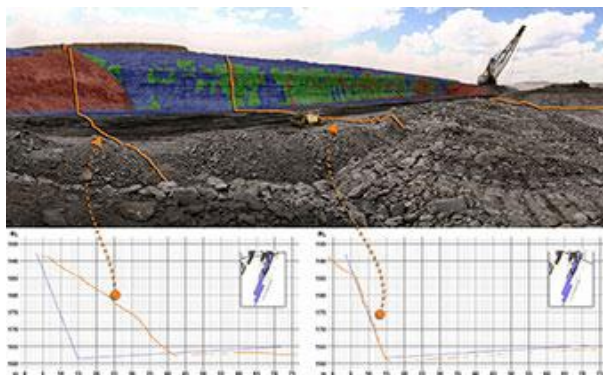


ILUSTRACIÓN 22: APLICACIÓN EN GEOLOGÍA EN LA RECOPIACIÓN DE DATOS

### 2.1.5.10. Publicidad:

Una de las últimas aplicaciones de la realidad aumentada es la publicidad. Hay diferentes campañas que utilizan este recurso para llamar la atención del usuario.

Fiat ha lanzado una campaña en la que cualquier usuario puede crear su propio anuncio de televisión con el Fiat 500 como protagonista a través de la página web, el usuario solo necesita tener una webcam.

La revista Esquire publica en la edición de diciembre del 2009 diferentes códigos QR, que son una variante más potente de los códigos de barras que pueden ser escaneados por una webcam que al reconocerlos nos ofrece información extra sobre el producto. Los códigos QR que incorpora la revista son reconocidos por las webcams de los usuarios y al ser reconocidos activan un video superpuesto a la imagen de la webcam (Ilustración 23). Para poder interpretarlos se necesita un software específico.

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico



ILUSTRACIÓN 23: EJEMPLAR DE LA REVISTA ESQUIRE, CON UN CODIGO QR

**2.1.5.11. Información:**

La empresa austriaca Mobilizy ha desarrollado Wikitude. Al apuntar la cámara del móvil hacia un edificio histórico, el GPS reconoce la localización y muestra información de la Wikipedia sobre el monumento. En Japón, Sekai Camera, de la empresa Tonchidot, añade al mundo real los comentarios de la gente acerca de direcciones, tiendas, restaurantes., entre otros edificios.



ILUSTRACIÓN 24: APLICACION DE WIKITUDE QUE DA INFORMACIÓN DE PUNTOS DE INTERES EN UNA CIUDAD

## 2.2. Tecnología Móvil

### 2.2.1. Introducción

La tecnología móvil consiste en la utilización de medios informáticos, sin la necesidad de estar emplazados, es decir, que pueden ser utilizados desde cualquier parte. Ésta tecnología, se utiliza para la comunicación de datos y voz de área amplia [20]. Además, han pasado por 4 generaciones distintas, con tecnologías diferentes:

- Voz analógica. (1G)
- Voz digital. (2G)
- Voz y datos digitales (Internet, correo electrónico, etcétera). (3G)
- Voz y datos digitales a mayor Velocidad. (4G)

La tecnología móvil se ha desarrollado rápidamente en los últimos años. Desde el comienzo de este milenio, un dispositivo móvil estándar ha pasado de ser un simple teléfono celular a un Smartphone, con sistema GPS, navegador web, mensajería instantánea y en algunos casos llegando a ser una consola de juegos portátil, traspasando fronteras que años atrás eran inimaginables. Aportando esta tecnología a la “no dependencia” de cables para que las personas puedan comunicarse o acceder a Internet en casi cualquier lugar, favoreciendo el teletrabajo y aumentando la productividad de sus empleados.

### 2.2.2. Evolución de la Tecnología Móvil

A medida que avanza la tecnología móvil, los tipos de conexión y la calidad de las mismas, las diferentes generaciones de antenas y tecnologías evolucionan a la vez y mantienen la compatibilidad con los tipos de frecuencias y conexiones anteriores.





Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

ILUSTRACIÓN 25: LINEA DE TIEMPO DE LAS GENERACIONES DE LOS DISPOSITIVOS MOVILES.

### 2.2.2.1. Primera Generación (1G)

Es la que lo empezó todo y engloba todas las tecnologías de comunicación móvil analógicas. Tiene la capacidad de transmitir voz, pero no datos. En la actualidad está en desuso [20].

En 1986, Ericsson modernizó el sistema, llevándolo hasta el nivel NMT 900. Esta nueva versión utilizaba frecuencias superiores (del orden de 900 MHz). Esto permitió dar servicio a un mayor número de usuarios y avanzar en la portabilidad de los terminales.

### 2.2.2.2. Segunda Generación (2G)

Es una tecnología ya digital, cuya funcionalidad es la transmisión de voz y la transmisión de datos a baja velocidad. Aun siendo de baja velocidad ha permitido el éxito absoluto gracias al envío de mensajes cortos. [20]

El desarrollo de esta generación tiene como piedra angular la digitalización de las comunicaciones. Las comunicaciones digitales ofrecen una mejor calidad de voz que las analógicas, además se aumenta el nivel de seguridad y se simplifica la fabricación del Terminal.

### 2.2.2.3. Generación de Transición (2.5 G)

En esta generación se incluyen todas las tecnologías de comunicaciones móviles digitales que permitieron una mayor capacidad de transmisión de datos y que surgieron como paso previo a las tecnologías 3G, por ejemplo, la tecnología GPRS (General Packet Radio System), basada en la transmisión de paquetes de una manera mucho más dinámica e interactiva [20].

Se incluyen nuevos servicios como EMS y MMS:

- **EMS** es el servicio de mensajería mejorado, permite la inclusión de melodías e iconos dentro del mensaje basándose en los sms; un EMS equivale a 3 o 4 sms.
- **MMS** (Sistema de Mensajería Multimedia) Este tipo de mensajes se envían mediante GPRS y permite la inserción de imágenes, sonidos, videos (de no más de 15 segundos de duración) y texto.

#### **2.2.2.4. Tercera Generación (3G)**

La tecnología 3G se encuentra contenida dentro del IMT-2000 de la Unión Internacional de Comunicación (ITU), que puede considerarse como la guía que marca los puntos en común que deben cumplirse para conseguir el objetivo de la itinerancia global, es decir, que un usuario de 3G pueda comunicarse con cualquier otra red 3G del mundo. Los servicios que ofrecen las tecnologías 3G son básicamente: acceso a Internet, servicios de banda ancha, roaming internacional e inter-operatividad. Gracias a ella surgen nuevos servicios y aplicaciones como videoconferencias o el comercio electrónico. Permite velocidades de conexión de hasta 2 Mbps, es decir, dos megas reales de descarga [20].

#### **2.2.2.5. Cuarta Generación (4G)**

La tecnología 4G está basada completamente en el protocolo IP, siendo un sistema que engloba a otros sistemas y una red de redes que se alcanza gracias a la convergencia entre las redes de cables y las inalámbricas. Todo esto en conjunto se ha denominado UMTS (Universal Mobile Telecommunications System). Esta tecnología podrá ser usada por módems inalámbricos, Smartphone, etcétera. La principal diferencia con las anteriores generaciones es la capacidad para proveer velocidades de acceso mayores a 100 MB/s en movimiento y 1 GB/s en reposo manteniendo una calidad de servicio de punta a punta y de alta seguridad con el mínimo coste posible.

### **2.2.3. Sistemas Operativos para móviles**

Un sistema operativo móvil o SO móvil es un sistema que controla un dispositivo móvil, al igual que los computadores que utilizan Windows, Linux o Mac OS, entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

#### **2.2.3.1. Componentes de los Sistemas operativos en Dispositivos Móviles**

Al igual que los sistemas operativos presentes en los componentes informáticos más grandes, un sistema operativo móvil también se encuentra compuesto por varias capas [16]:

- **Kernel o Núcleo**



## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Marco Teórico

Una de las más importantes y esenciales piezas que componen cualquier sistema operativo. Es la capa de software que permite el acceso a los diferentes elementos de hardware que conforman un Smartphone. Encargado de brindar diferentes servicios a las capas superiores como los controladores de hardware, gestión de procesos, sistemas de archivos, además del acceso y administración de la memoria del sistema.

- **Middleware**

Esta capa es el conjunto de módulos que permite que las aplicaciones diseñadas y escritas para tales plataformas puedan ser ejecutadas.

Su funcionamiento es totalmente transparente para el usuario, no debiendo realizar ninguna acción ni configurar alguna para su correcto desenvolvimiento. El Middleware brinda la posibilidad de ejecutar servicios muy importantes para que otras aplicaciones, en capas superiores de la jerarquía, puedan ejecutarse. Entre los servicios que presta esta capa podemos citar los motores de comunicaciones y mensajería, funciones de seguridad, servicios para la gestión de diferentes aspectos del móvil, ofrece servicios claves como el motor de mensajera y comunicaciones, códecs multimedia, intérpretes de páginas Web y servicios WAP, además de soporte para una gran variedad de servicios concernientes al apartado multimedia que es capaz de ejecutar el móvil.

- **Entorno de ejecución de aplicaciones**

Esta capa provee de todos los elementos necesarios para la creación y desarrollo de software a los programadores, es decir contiene elementos que serán de gran ayuda a los mismos, al momento de escribir aplicaciones compatibles con ese sistema operativo. Entre los servicios que los programadores pueden encontrar, se destacan un gestor de aplicaciones y una serie de interfaces programables (APIs).

- **Interfaz de Usuario**

Las interfaces de usuario facilitan la interacción con el usuario y el diseño de la presentación visual de la aplicación. Los servicios que incluye son el de componentes gráficos (botones, pantallas, listas, etcétera.) y el del marco de interacción.

### 2.2.3.2. **Sistemas Operativos en el Mercado**

- **Android**

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

Es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tablets, Google TV y entre otros. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google.

Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.

Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se han sobrepasado las 700.000 aplicaciones (de las cuales, dos tercios son gratuitas) disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android, como pueden ser la App Store de Amazon o la tienda de aplicaciones Samsung Apps de Samsung. Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente.

Última versión	4.4 KitKat
Lenguaje desarrollo	Java, C/C++
Idiomas	Multilenguaje

TABLA 1: INFORMACIÓN GENERAL DE ANDROID

- **Apple iOS**

iOS (anteriormente denominado iPhone OS) es un sistema operativo móvil de Apple. Originalmente desarrollado para el iPhone, siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV. Apple, Inc. no permite la instalación de iOS en hardware de terceros.

iOS se deriva de Mac OS X, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Unix.

iOS cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de "Servicios Principales", la capa de "Medios" y la capa de "Cocoa Touch".

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

Última versión	7.0.2
Lenguaje desarrollo	Objective-C
Idiomas	Multilenguaje

TABLA 2: INFORMACIÓN GENERAL DE IOS

- **Black Berry**

El BlackBerry OS es un sistema operativo móvil desarrollado por RIM para sus dispositivos BlackBerry. El sistema permite multitarea y tiene soporte para diferentes métodos de entrada adoptados por RIM para su uso en computadoras de mano, particularmente la trackwheel, trackball, touchpad y pantallas táctiles [27].

El BlackBerry OS está claramente orientado a su uso profesional y empresarial como gestor de correo electrónico y agenda. Desde la cuarta versión se puede sincronizar el dispositivo con el correo electrónico, el calendario, tareas, notas y contactos de Microsoft Exchange Server además es compatible también con Lotus Notes y Novell GroupWise.

Última versión	10
Lenguaje desarrollo	Java, C/C++
Idiomas	Multilenguaje

TABLA 3: INFORMACIÓN GENERAL BLACKBERRY

- **Windows Mobile/Phone**

Windows Phone es un sistema operativo móvil desarrollado por Microsoft, ofreciendo en su interfaz de usuario varios servicios propios como SkyDrive, Skype y Xbox Live en el sistema operativo [32].

El 29 de octubre de 2012 se lanzó al mercado la última versión del sistema operativo, Windows Phone 8 solo para nuevos dispositivos, debido a un cambio completo en el kernel que lo hace incompatible con dispositivos basados en la versión anterior, fragmentando de esta forma el mercado de aplicaciones disponibles. Esta versión incluye nuevas funciones que

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Marco Teórico

de acuerdo a Microsoft lo harán competitivo con sistemas operativos como iOS de Apple o Android de Google.

Última versión	Windows Phone 8
Lenguaje desarrollo	C#, C/C++
Idiomas	Multilenguaje

TABLA 4: INFORMACIÓN GENERAL WINDOWS PHONE

- **Cuota en el mercado**

Porcentaje de teléfonos inteligentes vendidos según su sistema operativo hasta el segundo trimestre del 2013 en el mundo:



ILUSTRACIÓN 26: SISTEMAS OPERATIVOS VENDIDOS EN EL MUNDO

Del gráfico anterior<sup>1</sup> (Ilustración 26), muestra claramente que en el último año el sistema operativo Android acapara el 79,30 % del mercado, siendo líder en muy pocos años desde su lanzamiento; en el segundo puesto se encuentra el sistema operativo de la empresa Apple.

<sup>1</sup> Fuente: <http://www.diarioabierto.es/145215/google-y-android-no-son-el-nuevo-microsoft-la-clave-esta-en-analizar-las-cuotas-de-mercado>

# **CAPÍTULO 3: ESTUDIO COMPARATIVO DE TECNOLOGÍAS DE REALIDAD AUMENTADA**

### 3.1. Introducción

En el mercado actual de dispositivos móviles, android es líder con millones usuarios en más de 190 países. Cada día, otro millón de interesados comienzan a usarlo por primera vez, buscando aplicaciones, juegos y otros contenidos digitales. Por esta razón, el autor ha decidido centrar en la investigación en este sistema operativo.

La AR es una de las nuevas tecnologías llamadas a revolucionar el desarrollo de aplicaciones en los próximos años. En una línea de tan rápido crecimiento, es normal que aún exista un cierto desconcierto entre los programadores, hasta que aparezcan librerías que aprovechen al máximo todas las posibilidades de los dispositivos móviles.

Actualmente, existen varias herramientas de desarrollo (SDK) para AR actualmente, que ofrecen buenos resultados, algunos de ellas son:

- **Wikitude:** SDK gratuita bastante bien documentada para la creación de aplicaciones de AR con Android.
- **Vuforia:** SDK gratuita de Qualcomm para diseñar aplicaciones que utilicen AR tanto en iOS como en Android. Además, es compatible con el Framework de desarrollo de Unity 3D<sup>2</sup>.
- **Metaio SDK:** SDK de paga crear aplicaciones de AR, pero tiene una versión gratuita con menos funcionalidades. La empresa dueña de esta herramienta se destaca en desarrollar realidad aumentada. El SDK es compatible actualmente en Android y iOS.
- **D 'Fusion Studio:** Plataforma de desarrollo gratuita para la creación de aplicaciones en AR de la empresa Total Immersion<sup>3</sup>.
- **DroidAr:** Framework de desarrollo para Android de aplicaciones de AR, está publicado como código abierto bajo la licencia GPLv3 [18].
- **NyARToolkit:** Librería derivada de ARToolKit que permite la creación de aplicaciones de AR. Corre sobre diversas plataformas virtuales: Java, C#, Actionscript3; y además algunos proyectos derivados para flash, Silverlight, Processing y Android.<sup>4</sup>

---

<sup>2</sup> Unity es un ecosistema de desarrollo de juegos: un potente motor de renderizado, sitio web: <http://spanish.unity3d.com>

<sup>3</sup> Sitio web: <http://www.t-immersion.com/>

<sup>4</sup> Sitio Web: [http://nyatla.jp/nyartoolkit/wp/?page\\_id=729](http://nyatla.jp/nyartoolkit/wp/?page_id=729)

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Estudio Comparativo

- **Layar:** SDK que se actualiza bastante a menudo y con un amplio rango de características para el desarrollo de aplicaciones de AR, basado en servicios web<sup>5</sup>.

En este capítulo se compone de tres ejes principales: presentación de las herramientas, análisis comparativo e interpretación de los resultados del estudio; la que permitirá al lector conocer más en profundidad los SDK que se analizan en la investigación. Además, observar las diferencias y semejanzas entre ellas, guiándole a tomar una futura decisión, sobre cual ocupar al momento de crear sus propias aplicaciones.

Conjuntamente, generalmente una aplicación de AR se puede estructurar en cuatro etapas [19]:

- 1) Captura del entorno real.
- 2) Reconocimiento y tracking del marcador o target.
- 3) Renderizado de los objetos virtuales.
- 4) Visualización o superposición del mundo real y virtual

Pues bien, las etapas 1) y 4), más que del SDK utilizado, dependerán en su mayoría de las características (especialmente hardware) del dispositivo utilizado. Por esto, son consideradas de interés para nuestro estudio únicamente las etapas 2 y 3, que darán lugar a las características más diferenciadoras de los SDK objeto de estudio.

Para la presente investigación se analizarán las herramientas “wikitude”, “Vuforia” y “Metaio SDK”, debido a que presentan documentación adecuadas para realizar el análisis comparativo, como al mismo tiempo, ser libre de pago.

---

<sup>5</sup> Sitio Web: <https://www.layar.com/>

### 3.2. Presentación de las herramientas

Es común que los programadores descarguen o adquieran un SDK y/o un IDE que contenga un conjunto de herramientas de programación al momento de crear sus propias aplicaciones.

Un Software Development Kit (SDK o en español equipo de desarrollo de software) es un conjunto de instrumentos de programación específica de un lenguaje que permite a un programador desarrollar aplicaciones para una plataforma determinada. Un SDK básico incluye un compilador, documentación acerca del lenguaje y la sintaxis, e instrucciones de instalación. Los más sofisticados también incluyen editor, un depurador, un módulo de diseño de interfaz visual, y las API's del lenguajes [9].

Un IDE (ambiente integrado de desarrollo) es un tipo de SDK e incluyen un conjunto de herramientas de desarrollo en un aplicación de programación elegante. Los módulos de aplicación tienen un conjunto uniforme de menús y controles las cuales simplifican el proceso de programación [9].

A continuación, profundizaremos en los SDK's elegidos para el análisis comparativo:

#### 3.2.1. Wikitude

En su sitio web<sup>6</sup>, la empresa se define como “Wikitude (Ilustración 27) es el proveedor de la tecnología de realidad aumentada más importantes del mundo. Nuestra gama completa de la tecnología de la realidad aumentada incluye un motor de geo-basado AR, servicios de reconocimiento y seguimiento de imágenes y modelos 3D y capas de presentación. Esta tecnología central le proporciona todos los componentes necesarios para crear experiencias de AR profundas y atractivas”.



ILUSTRACIÓN 27: LOGO DE WIKITUDE

---

<sup>6</sup> Sitio Web : <http://www.wikitude.com/>



Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

**3.2.1.1. Características:**

Wikitude SDK, es la principal biblioteca que se utiliza para la aplicación de AR. Sobre la base de los lenguajes de programación conocidos: HTML5, JavaScript y CSS3, Wikitude reduce drásticamente el ciclo de vida del time-to-market, y elimina la necesidad de aprender nuevos lenguajes de programación. El avanzado motor de renderizado 3D incluido en la potente tecnología de Wikitude, ofrece a los desarrolladores la capacidad de satisfacer y superar incluso las expectativas de los clientes más exigentes.

**3.2.1.2. Arquitectura:**

Wikitude posee una arquitectura basada en servicio (ver Ilustración 28), la cual genera una “capa” entre el mundo real y el dispositivo; aquella “capa” se denomina ARchitect Engine y está basado en HTML5 y JavaScript, esta se encarga de la interacción y soporte multimedia.

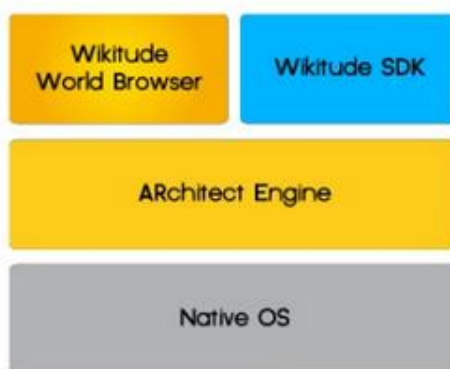


ILUSTRACIÓN 28: ARQUITECTURA DE WIKITUDE

**3.2.1.3. Ciclo de Vida:**

El ciclo de vida de Wikitude viene dado por las siguientes cinco etapas (ver Ilustración 29) [34]:

1. Solicitar
2. Búsqueda de índice
3. Petición / repuesta
4. Fusionar Resultados
5. Respuesta

Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Estudio Comparativo

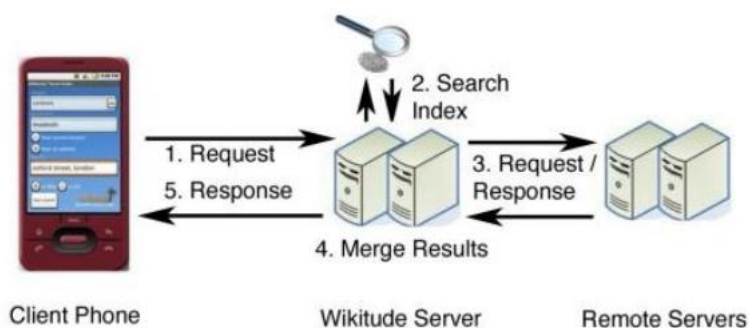


ILUSTRACIÓN 29: CICLO DE VIDA DE WIKITUDE

**3.2.1.4. Aplicaciones:**

Este SDK permite las siguientes aplicaciones:

- **Reconocimiento de imagen y seguimiento:** Wikitude SDK incorpora reconocimiento de imágenes y tecnología de seguimiento. Funciona con un máximo de 1.000 imágenes que pueden ser reconocidos fuera de línea. Los desarrolladores pueden cambiar entre aumentar las imágenes reconocidas y geo-localizar los puntos de interés dentro de la imagen de la cámara en vivo.



ILUSTRACIÓN 30: EJEMPLO DE RECONOCIMIENTO DE IMAGEN CON WIKITUDE

- **Servicios basados en la localización geográfica con los datos:** Wikitude posee características convenientes para trabajar con datos geo-referenciados. El diseño y maquetación de los puntos de interés son completamente adaptables para satisfacer las necesidades del programador.

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Estudio Comparativo



ILUSTRACIÓN 31: EJEMPLO DE GEO-LOCALIZACIÓN CON WIKITUDE

- **Modelos 3D y renderizado:** Wikitude logra cargar y renderizar modelos 3D en la escena de la AR. Se pueden importar modelos 3D de herramientas como Autodesk ® Maya ® 3D o Blender. Cada modelo 3D se puede utilizar en combinación con los objetivos de reconocimiento de imagen o con geo-localizaciones.



ILUSTRACIÓN 32: EJEMPLO DE UN MODELO 3D CON WIKITUDE

### 3.2.2. Vuforia

En su sitio web<sup>7</sup>, se definen como “Vuforia es una plataforma de software para Android y iOS que permite a su aplicación ver imágenes desde una base de datos de objetivos Vuforia en el dispositivo o en la nube. Cuando se reconoce un objetivo, su aplicación genera experiencias de AR”.



ILUSTRACIÓN 33: LOGO DE VUFORIA

#### 3.2.2.1. Características:

Vuforia, su aplicación permite en una gran variedad de objetos como textos, imágenes, objetos simples 3D, marcadores, etcétera. Además, presenta el servicio de reconocimiento

<sup>7</sup> Sitio Web: <https://www.vuforia.com/>

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Estudio Comparativo

de la nube Vuforia, que aprovecha el poder de la nube para darle flexibilidad y escala a aplicaciones con varias imágenes a reconocer. También se pueden crear aplicaciones de botones virtuales, efectos de fondos y reproducción de videos.

### 3.2.2.2. **Arquitectura:**

Una aplicación AR basada en Vuforia se compone de los siguientes componentes básicos [25].

- **Cámara:** El componente de la cámara se asegura que cada cuadro de vista previa se capturó y pasó de manera eficiente al rastreador. El desarrollador sólo tiene que inicializar la cámara para iniciar y detener la captura.
- **Convertidor de imagen:** El convertidor de formato de píxel convierte desde el formato de la cámara (por ejemplo, YUV12) a un formato adecuado para OpenGL ES de renderizado (por ejemplo, RGB565) y para el seguimiento (por ejemplo, la luminancia) internamente.
- **Rastreador:** El componente rastreador contiene los algoritmos de visión por computador que permiten detectar y rastrear objetos del mundo real en los marcos de la cámara de video. Basado en la imagen de la cámara, diferentes algoritmos se encargan de la detección de nuevos objetivos o marcadores y la evaluación de los botones virtuales. Los resultados se almacenan en un objeto de estado que es utilizado por el procesador de video de fondo y se puede acceder desde el código de aplicación. El seguidor puede cargar varios conjuntos de datos al mismo tiempo y activarlos.
- **Código de la aplicación:** Los desarrolladores de aplicaciones deben inicializar todos los componentes anteriores y realizar tres pasos clave en el código de aplicación. Para cada trama procesada, el objeto de estado se actualiza y las aplicaciones de método procesamiento es llamado. El desarrollador de la aplicación debe:
  - Consultar el objeto de estado para los objetivos recientemente detectados, marcadores o estados actualizados de estos elementos.
  - Actualización de la lógica de la aplicación con los nuevos datos de entrada.
  - Procesamiento la superposición de gráficos aumentada.

Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Estudio Comparativo

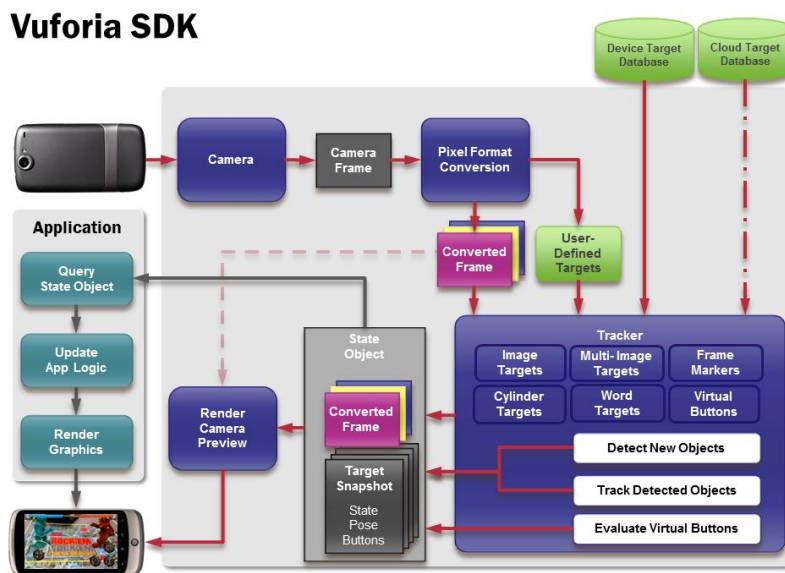


ILUSTRACIÓN 34: ARQUITECTURA DE VUFORIA

3.2.2.3. Aplicaciones:

Este SDK permite las siguientes aplicaciones:

- **Imágenes Objetivos:** Como su nombre indica, las “imagen objetivos” son imágenes que Vuforia puede detectar y realizar un seguimiento. Vuforia utiliza algoritmos sofisticados para detectar y rastrear las características que se encuentran de forma natural en la propia imagen. El SDK reconoce la imagen objetivo mediante la comparación de estas características naturales contra una base de datos de recursos conocidos. Una vez que se detecta la imagen de destino, hará un seguimiento de la imagen siempre esté al menos parcialmente en el campo de visión de la cámara.



ILUSTRACIÓN 35: EJEMPLO IMAGEN OBJETIVOS DE VUFORIA

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

- **Botones virtuales:** son donde el desarrollador define regiones rectangulares en objetivos de imagen que desencadenan un evento cuando se toca o se ocluye en la vista de cámara.

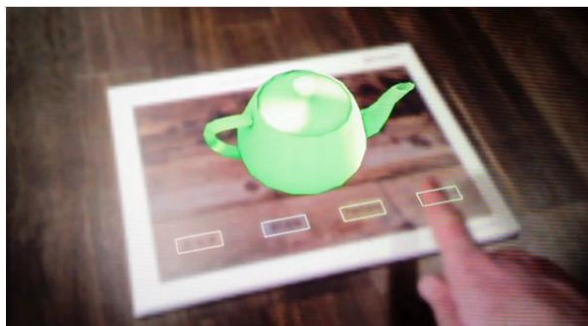


ILUSTRACIÓN 36: EJEMPLO DE BOTONES VIRTUALES DE VUFORIA

- **Frame Markers:** es un marcador rectangular especial con un código incrustado en su frontera. La región interior puede ser llenada con cualquier tipo de imagen de manera que el usuario puede identificar el marcador visualmente. Muchos de ellos se pueden detectar simultáneamente y realiza un seguimiento de la vista de la cámara sin pérdida de rendimiento.

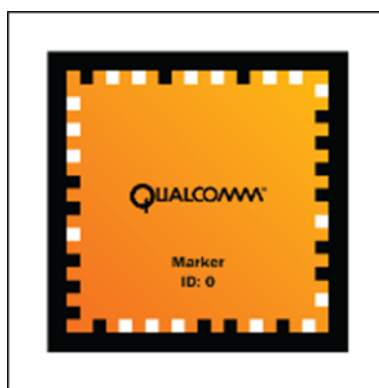


ILUSTRACIÓN 37: FRAME DE VUFORIA

- **Palabra objetivo:** representan elementos textuales que el motor Vuforia puede reconocer y rastrear, se caracteriza por los siguientes rasgos [23]:
  - Una palabra puede tener una longitud de 2 a 24 caracteres.
  - Una palabra puede contener hypens (palabra compuesta y cerrada, por ejemplo, "la madre-en-ley").
  - Una palabra puede contener un espacio (por ejemplo, "hola mundo").

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

- Una palabra no puede contener números.

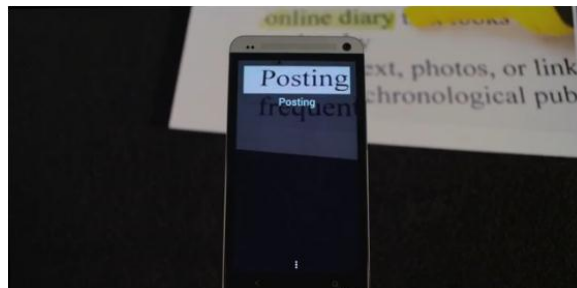


ILUSTRACIÓN 38: EJEMPLO DE PALABRA OBJETIVO CON VUFORIA

### 3.2.3. Metaio SDK

En su sitio web<sup>8</sup>, se definen como “Metaio, galardonado SDK libre permite a las aplicaciones de Realidad Aumentada un nuevo soporte a dispositivos portátiles, una mejor visualización, la velocidad y estabilidad de seguimiento. Metaio SDK permite una implementación flexible y concesión de licencias para su proyecto independiente o basado en la nube”.



ILUSTRACIÓN 39: LOGO DE METAIO SDK

#### 3.2.3.1. Características:

La plataforma Metaio SDK, presenta un potente motor de renderizado 3D que mejora la visualización y velocidad de los modelos utilizados en las aplicaciones. También presenta reconocimiento de figuras, ya sea imágenes (2D) u objetos (3D), que puede dar soluciones de seguimiento en las aplicaciones de AR [11].

#### 3.2.3.2. Arquitectura:

SDK Metaio está implementado de forma modular, de forma que divide la actividad de AR en tres componentes: Tracking, Captura y Renderizado (Ilustración 40). En la versión

<sup>8</sup> Sitio Web: <http://www.metaio.com/>

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

gratuita la etapa de tracking está oculta bajo una aplicación estándar configurable mediante un archivo de configuración. XML. [19]

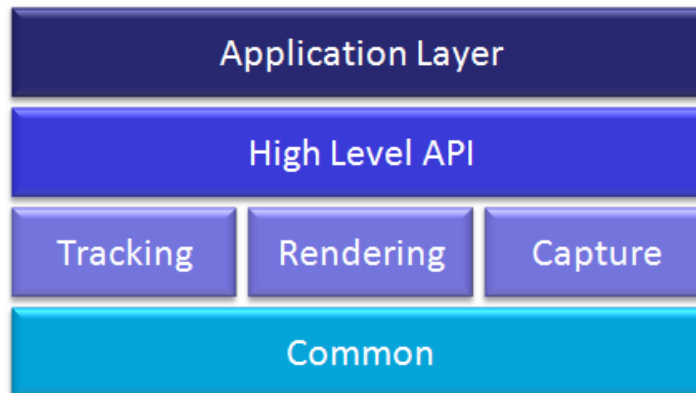


ILUSTRACIÓN 40: ARQUITECTURA DE METAIO SDK

### 3.2.3.3. Aplicaciones:

Este SDK permite las siguientes aplicaciones:

- **Seguimiento:** Detectar y rastrear las características que se encuentran en una imagen que una vez que se detecta, hará un seguimiento siempre que esté al menos parcialmente en el campo de visión de la cámara.



ILUSTRACIÓN 41: EJEMPLO DE SEGUIMIENTO DE IMÁGENES



Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

- **AR basada en la ubicación:** Características convenientes para trabajar con datos georeferenciados. El diseño y maquetación de los puntos de interés son completamente adaptables para satisfacer las necesidades del programador.



ILUSTRACIÓN 42: EJEMPLO DE AR BASADA EN LA UBICACIÓN

- **Muebles interactivo:** Una aplicación móvil básica que está diseñado para arrastrar, escalar y rotar modelos 3D y los situarlo en una imagen personalizada.



ILUSTRACIÓN 43: EJEMPLO DE MUEBLES INTERACTIVO

### 3.3. Análisis Comparativo

#### 3.3.1. Definición de parámetros de comparación:

La gran mayoría de las herramientas de AR presentan características similares al momento de crear aplicaciones. Actualmente, muchos programadores han comparado estos instrumentos en distintos foros de internet diferenciando un SDK con otro. En este capítulo se recopilan los parámetros más nombrados de aquellos foros, y otros, se han propuesto a partir del estudio personal de las herramientas mencionadas en el punto anterior. [1], [18], [6]

- P1. Plataforma:** Este parámetro se refiere a la variedad de Sistemas Operativos sobre los cuales la herramienta de AR opera.
- P2. Uso del Sistema de Posicionamiento Global (GPS):** Si la herramienta ocupa el GPS del dispositivo y determina el lugar donde se encuentra, para añadir información adicional dependiendo del lugar.
- P3. Búsqueda Visual (Visual Search):** Se reseña a comprobar si la herramienta puede reconocer una imagen, procesarla y añadir nueva información, ya sea una figura o texto adicional.
- P4. Contenido de las API:** Comprobar si las herramientas presentan API y sirven de apoyo al programador.
- P5. Framework:** Determinar si la herramienta presenta programas anexos para facilitar la construcción de aplicaciones de AR.
- P6. Licencia:** Este parámetro se refiere a si para ocupar la herramienta se debe pagar una licencia o si presenta una versión de prueba o es totalmente gratis para ocuparla.

Este estudio se basa en la escala de Likert, utilizado prácticamente por todos los estudios de investigación. La escala de Likert es de nivel ordinal <sup>9</sup> y se caracteriza por ubicar una serie de frases seleccionadas en una escala con grados de excelente/pésimo [4].

Se utiliza una tabla de escalas para evaluar el cumplimiento de cada parámetro en las herramientas, ver Tabla 5:

---

<sup>9</sup> También llamada escala de orden jerárquico, con ella se establecen posiciones relativas de los objetos o fenómenos en estudio, respecto a alguna característica de interés, sin que se reflejen distancias entre ellos.

Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Estudio Comparativo

<b>Criterios</b>	<b>Valor</b>
Pésimo	0
Malo	1
Regular	2
Bueno	3
Excelente	4

TABLA 5: ESCALA EVALUACIÓN HERRAMIENTAS AR

### 3.3.2. Análisis de parámetros y variables de comparación

La forma de evaluar y calificar las tres herramientas seleccionadas, “Wikitude”, “Vuforia” y “Metaio SDK”, se realiza basándose en el estudio de cada una y considerando los 7 parámetros anteriormente mencionados. A través de estos, se realizará la selección de la más adecuada en cada caso.

#### P1. Plataforma:

<b>Variables</b>	<b>Concepto</b>
P1.1 La herramienta crea aplicaciones para Android.	Crear aplicaciones para que funcionen en la plataforma de Android.
P1.2 La herramienta crea aplicaciones para iOS.	Crear aplicaciones para que funcionen en la plataforma de iOS de Apple.
P1.3 La herramienta crea aplicaciones para BlackBerry.	Crear aplicaciones para que funcionen en la plataforma de BlackBerry.
P1.4 La herramienta crea aplicaciones para Windows Phone.	Crear aplicaciones para que funcionen en la plataforma de Windows Phone.

TABLA 6: DEFINICIÓN DE VARIABLES DEL PARÁMETROS – PLATAFORMA

<b>Parámetros y variables de comparación</b>	<b>WIKITUDE</b>	<b>VUFORIA</b>	<b>METAIO SDK</b>
<b>Plataforma</b>			
La herramienta crea aplicaciones para Android.	4	4	4
La herramienta crea aplicaciones para iOS.	4	4	4
La herramienta crea aplicaciones para BlackBerry.	2	0	0
La herramienta crea aplicaciones para Windows Phone.	0	0	0

TABLA 7: VALORACIÓN DE VARIABLES DEL PARÁMETRO - PLATAFORMA

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

Resultados

Herramientas	Resultados	Porcentaje
<b>Ideal</b>	<b>4+4+4+4=16</b>	<b>100%</b>
Wikitude	4+4+2+0=10	62,5%
Vuforia	4+4+0+0=8	50%
Metaio SDK	4+4+0+0=8	50%

TABLA 8: INTERPRETACIÓN DE RESULTADOS – PLATAFORMA

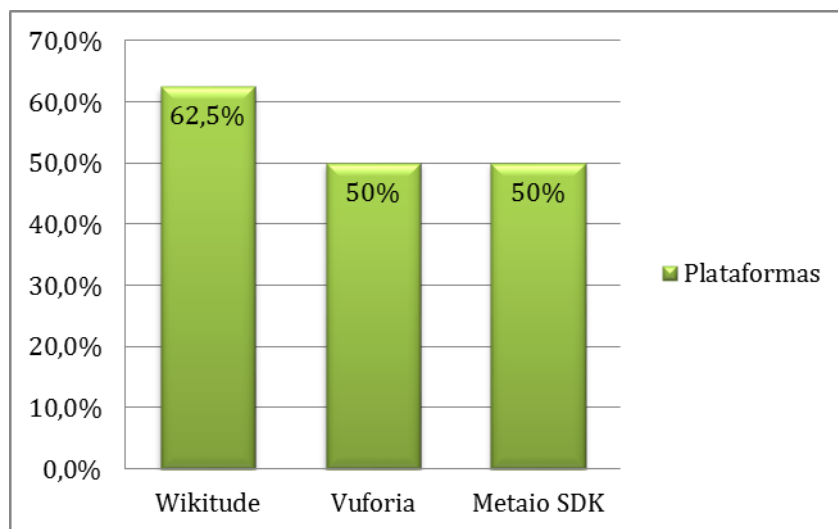


ILUSTRACIÓN 44: GRÁFICO DE RESULTADO DEL PARÁMETRO - PLATAFORMA

Como se puede observar en la Tabla 8 el ideal para el autor es que la herramienta se pueda ejecutar en cuatro sistemas operativos (Android, iOS, BlackBerry, Windows Phone), pero como se observa en la Ilustración 44, ninguno cumple el ideal del 100%.

La primera herramienta Wikitude, cumple un 62,5% de lo ideal, en otras palabras corre en tres de cuatro sistemas operativos por lo cual no tiene todo el puntaje. Además, solo funciona en la versión 10 del sistema operativo BlackBerry, la funcionalidad no está garantizada para otros dispositivos. [33]

Las otras dos Herramientas Vuforia y Metaio SDK, obtienen el mismo porcentaje, con un 50% de lo ideal, reafirmando que Android e iOS son los sistemas operativos líderes del mercado y es donde mayormente se desarrollan nuevas aplicaciones para los Smartphone.

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

**P2. Uso del Sistema de Posicionamiento Global (GPS)**

<b>Variables</b>	<b>Concepto</b>
P2.1 Utiliza tecnología GPS	La herramienta ocupa el sistema de posicionamiento global (GPS)
P2.2 Utilización de etiquetas	La herramienta al ocupar GPS muestra imágenes.
P2.3 Actualización de puntos geográficos	La herramienta al ocupar GPS actualiza los puntos geográficos donde aparece la AR.
P2.4 Recuperación de datos	La herramienta al ocupar GPS, obtiene los datos del POI de varias formas.

TABLA 9: DEFINICIÓN DE VARIABLES DEL PARÁMETRO - USO DEL GPS

<b>Parámetros y variables de comparación</b>	<b>WIKITUDE</b>	<b>VUFORIA</b>	<b>METAIO SDK</b>
<b>Uso del Sistema de Posicionamiento Global (GPS)</b>			
Utiliza tecnología GPS	4	0	4
Utilización de etiquetas	4	0	4
Actualización de puntos geográficos	4	0	3
Recuperación de datos	4	0	4

TABLA 10: VALORACIÓN DE VARIABLES DEL PARÁMETRO - USO DEL GPS

Resultados

<b>Herramientas</b>	<b>Resultados</b>	<b>Porcentaje</b>
<b>Ideal</b>	<b>4+4+4+4=16</b>	<b>100%</b>
Wikitude	4+4+4+4=16	100%
Vuforia	0+0+0+0=0	0%
Metaio SDK	4+4+3+4=15	93,75%

TABLA 11: IMPLEMENTACIÓN DE RESULTADOS - USO DEL GPS

Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Estudio Comparativo

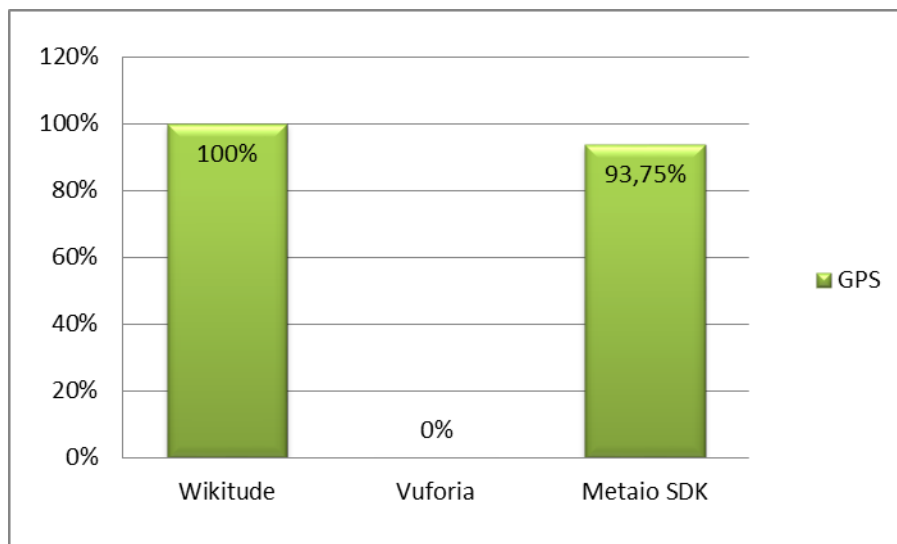


ILUSTRACIÓN 45: GRÁFICO DE RESULTADO DEL PARÁMETRO - USO DEL GPS

Como se observa en la Tabla 11 el ideal de las herramientas es utilizar el sistema de posicionamiento global y además, obtener, modificar o simplemente ver la información del medio gracias a este.

Wikitude obtiene el 100%, debido a que su principal especialidad es crear aplicaciones para sistemas GPS utilizando realidad aumentada. Además, se puede agregar POI y proporcionar información al usuario. Conjuntamente es posible ingresar POI de manera remota y/o desde la misma aplicación.

Por el contrario, Vuforia no utiliza GPS entre sus características, debido a que tiene otro enfoque, un ejemplo de esto es, la posibilidad de crear botones virtuales.

Por último, Metaio SDK (al igual que la primera herramienta) utiliza GPS, pero a diferencia con este, la actualización de los datos geográficos se hace por medio de un servicio web o modificación de la BD de la aplicación.

**P3. Búsqueda Visual (Visual Search):**

Variables	Concepto
P3.1 Puede reconocer una imagen determinada	La herramienta puede buscar una imagen establecida en el medio.
P3.2 Puede generar una imagen en dos dimensiones	La herramienta puede crear una imagen en 2D, cuando reconoce la imagen determinada como objetivo

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

P3.3 Puede generar una imagen en tres dimensiones	La herramienta puede crear una imagen en 3D, cuando reconoce la imagen determinada como objetivo
P3.4 Puede generar varias imágenes a la vez en dos dimensiones	La herramienta puede crear dos o más imágenes en 2D, cuando reconoce la imagen determinada como objetivo
P3.5 Puede generar varias imágenes a la vez en tres dimensiones	La herramienta puede crear dos o más imágenes en 3D, cuando reconoce la imagen determinada como objetivo

TABLA 12: DEFINICIÓN DE VARIABLES DEL PARÁMETRO - BÚSQUEDA VISUAL

Parámetros y variables de comparación	WIKITUDE	VUFORIA	METAIO SDK
<b>Búsqueda Visual</b>			
Puede reconocer una imagen determinada	4	4	4
Puede generar una imagen en dos dimensiones	4	4	4
Puede generar una imagen en tres dimensiones	4	4	4
Puede generar varias imágenes a la vez en dos dimensiones	4	4	4
Puede generar varias imágenes a la vez en tres dimensiones	4	4	4

TABLA 13: VALORACIÓN DE VARIABLES DEL PARÁMETRO – BÚSQUEDA VISUAL

Resultados

Herramientas	Resultados	Porcentaje
<b>Ideal</b>	<b>4+4+4+4+4=20</b>	<b>100%</b>
Wikitudo	4+4+4+4+4=20	100%
Vuforia	4+4+4+4+4=20	100%
Metaio SDK	4+4+4+4+4=20	100%

TABLA 14: IMPLEMENTACIÓN DE RESULTADOS - BÚSQUEDA VISUAL

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

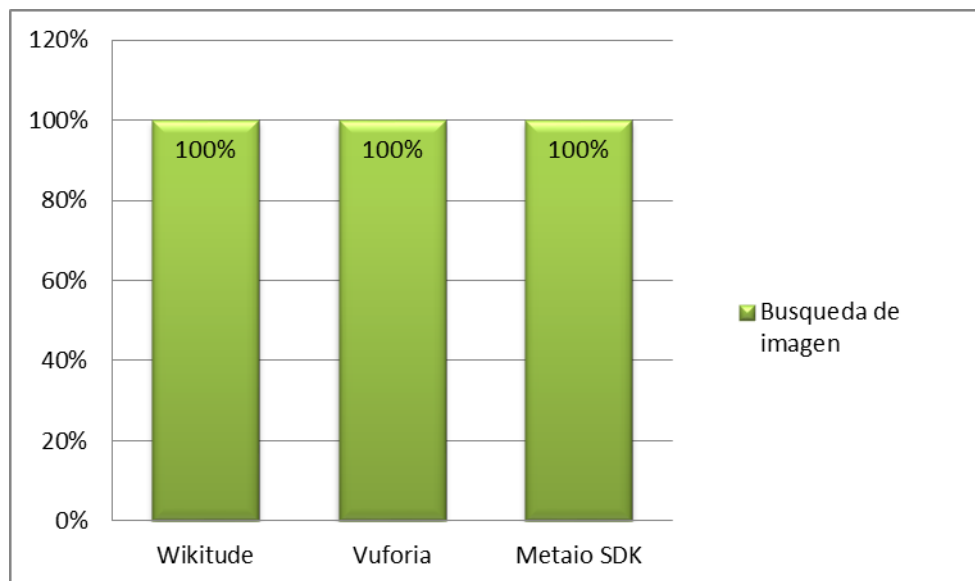


ILUSTRACIÓN 46: GRÁFICO DE RESULTADO DEL PARÁMETRO - BÚSQUEDA VISUAL

A criterio del autor, este parámetro es el mínimo que debe tener una herramienta de realidad aumentada, porque reconocer una imagen es primordial para unir el mundo real con el virtual, por lo tanto, lo anterior se ratifica con los resultados obtenidos.

Si se observa el gráfico, las tres herramientas obtienen el puntaje ideal, debido a que todas ellas generan imágenes, de distintas formas, en dos dimensiones y tres dimensiones; y que dependiendo de la licencia de uso puede generar dos o más simultáneamente.

Para establecer diferencias entre las herramientas, se observa la forma de generar las imágenes a reconocer, denominadas “target”. Wikitude utiliza una herramienta en línea llamada “target manager”<sup>10</sup> (necesita un registro previo). Metaio SDK, utiliza imágenes en formato PNG. Por último, Vuforia también se debe registrar en su página de desarrolladores para generar los target de las aplicaciones de AR, pero en vez de generar imágenes con formato especial, se construye una base de datos que contendrá dichas iconografías<sup>11</sup>.

<sup>10</sup> Sitio Web: <http://developer.wikitude.com/tools/target-manager/?level=0>

<sup>11</sup> Sitio Web: <https://developer.vuforia.com/target-manager>



Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Estudio Comparativo

**P4. Contenido de las API**

<b>Variables</b>	<b>Concepto</b>
P4.1 Documentación para instalación	La documentación incluye todos los pasos necesarios para la instalación de la herramienta y la descripción de cada componente en ella
P4.2 Documentación para el desarrollo de aplicaciones	La documentación incluye toda la información necesaria para el desarrollo de aplicaciones en la herramienta.
P4.3 Documentación en internet	Documentación se encuentra en el sitio oficial de la herramienta.

TABLA 15: DEFINICIÓN DE VARIABLES DEL PARÁMETRO - CONTENIDO DE LAS API

<b>Parámetros y variables de comparación</b>	<b>WIKITUDE</b>	<b>VUFORIA</b>	<b>METAIO SDK</b>
<b>Contenido de las API</b>			
Documentación para instalación	3	2	3
Documentación para el desarrollo de aplicaciones	3	3	3
Documentación en internet	4	4	4

TABLA 16: VALORACIÓN DE VARIABLES DEL PARÁMETRO – CONTENIDO DE LAS API

Resultados

<b>Herramientas</b>	<b>Resultados</b>	<b>Porcentaje</b>
<b>Ideal</b>	<b>4+4+4=12</b>	<b>100%</b>
Wikitude	3+3+4=10	83,3%
Vuforia	2+3+4=9	75%
Metaio SDK	3+3+4=10	83,3%

TABLA 17: IMPLEMENTACIÓN DE RESULTADOS - CONTENIDO DE LAS API

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

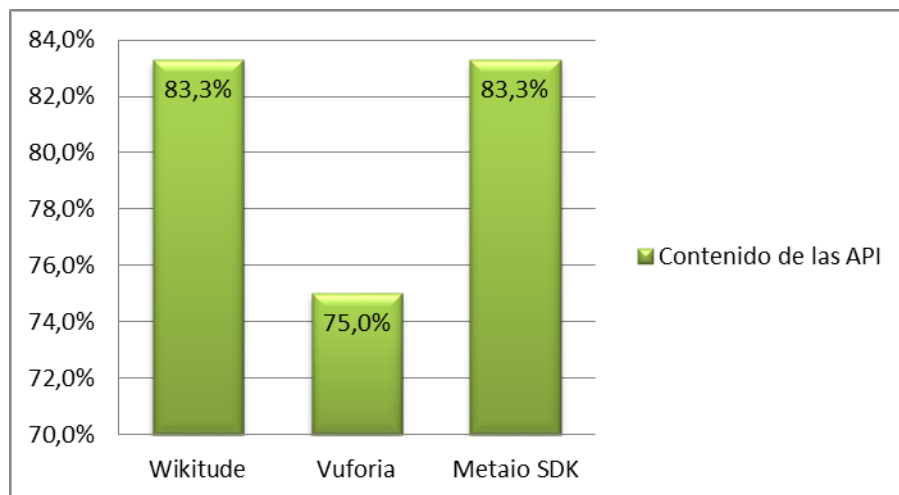


ILUSTRACIÓN 47: GRÁFICO DE RESULTADO DEL PARÁMETRO - CONTENIDO DE LAS API

Como se puede observar en la Tabla 17 el ideal para esta medida es de doce puntos, la cual contempla la documentación para la instalación de la herramienta, también la documentación para construir aplicaciones y si estas se encuentran en la web.

Wikitude obtuvo un 83,3%, debido a que presenta la documentación para instalar la herramienta, pero no es tan detallada, al punto que puede llevar al error, por eso el puntaje obtenido, igualmente al momento de crear aplicaciones. Sus ejemplos son claros, pero según el autor, la empresa desarrolladora asume pasos que un programador novato puede no saber. Por último, la herramienta también presenta una documentación en su página oficial.

La razón de que Vuforia obtenga un 75% en esta medición es debido a que la documentación para instalar la herramienta no es tan explícita a diferencia de las otras, pero en los demás puntos de estudio es muy similar a la herramienta anterior.

Finalmente, Metaio SDK logró un 83,3%, similar a Wikitude, con pasos no tan detallados para su instalación, ejemplos que demuestran cómo funciona el SDK y documentación en línea.

Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Estudio Comparativo

**P5. Framework**

<b>Variables</b>	<b>Concepto</b>
P5.1 Programa propio para generar imágenes.	La herramienta presenta una programa en línea o anexo, para crear los objetivos de la aplicación
P5.2 Programa propio para generar modelos 3D	La herramienta presenta una programa en línea o anexo, para crear los objetos que se visualizará en la aplicación
P5.3 De simple utilización	Los programas sean rápido de aprender a utilizarlos, independiente de los conocimientos en el área del desarrollador.
P5.4 Compatibilidad con otros Softwares de modelado	Comprobar que archivos de programas externos de modelado funcionen en una aplicación creada con la herramienta

TABLA 18: DEFINICIÓN DE VARIABLES DEL PARÁMETRO – FRAMEWORK

<b>Parámetros y variables de comparación</b>	<b>WIKITUDE</b>	<b>VUFORIA</b>	<b>METAIO SDK</b>
<b>Framework</b>			
Programa propio para generar imágenes.	2	2	2
Programa propio para generar modelos 3D	1	0	3
De simple utilización	3	2	3
Compatibilidad con otros utilidades de modelado	2	2	2

TABLA 19: VALORACIÓN DE VARIABLES DEL PARÁMETRO – FRAMWORK

Resultados

<b>Herramientas</b>	<b>Resultados</b>	<b>Porcentaje</b>
<b>Ideal</b>	<b>4+4+4+4=16</b>	<b>100%</b>
Wikitude	2+1+3+2=8	50%
Vuforia	2+0+2+2=6	37,5%
Metaio SDK	2+3+3+2=10	62,5%

TABLA 20: IMPLEMENTACIÓN DE RESULTADOS - FRAMEWORK

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

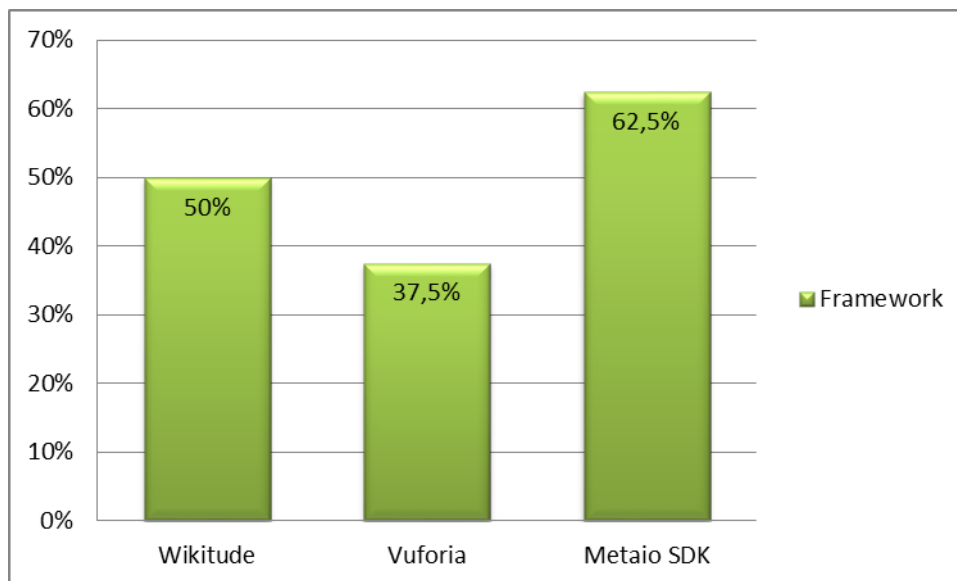


ILUSTRACIÓN 48: GRÁFICO DE RESULTADO DEL PARÁMETRO – FRAMEWORK

Como se puede observar en la Tabla 20, el ideal de la herramienta es de dieciséis puntos. Ninguna de las herramientas lo alcanzó, a pesar que todas presentan utilidades anexas para crear aplicaciones AR.

Wikitude obtuvo un 50% en la evaluación del parámetro Framework. La herramienta llamada “target manager”, incluida en el paquete, permite la utilización de “drag and drop”<sup>12</sup> facilitando el proceso de crear targets, pero si no se tiene una buena conexión a internet puede tomar más tiempo de lo recomendado. Conjuntamente, Wikitude tiene una herramienta para crear modelos en tres dimensiones, llamado “3D encoder”<sup>13</sup>, herramienta que se puede descargar y se debe contar con un computador con versión de OpenGL igual o superior a 3.2. Por otro lado, Wikitude al ser compatible con software de modelados 3D puede facilitar el trabajo con persona externa con conocimiento en diseño gráfico.

Vuforia obtuvo un 37,5% debido a que no contienen una utilidad de modelado en tres dimensiones, por lo cual se debe recurrir a programas externos. Además, su herramienta para crear target no es tan intuitiva como las herramientas contrincantes y es por eso su valoración.

<sup>12</sup> Sistema de programación de arrastre y soltar

<sup>13</sup> <http://developer.wikitude.com/tools/3d-encoder>

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

Por último, Metaio SDK es similar a Wikitude, pero la ventaja que obtiene es que presenta una herramienta de modelado 3D más estable que el primer SDK nombrado y también puede soportar modelos creados por otros softwares.

**P6. Licencia**

<b>Variables</b>	<b>Concepto</b>
P6.1 Versión libre	La Herramienta presenta una versión gratis, donde se tiene acceso a la mayoría de las funcionalidades sin costo.
P6.2 Versión de prueba	La herramienta presenta una versión de prueba, donde se tiene acceso a la mayoría de las funcionalidades por un corto tiempo y luego se debe pagar para obtenerlas nuevamente.
P6.3 Versión de paga	Se debe pagar para obtener funcionalidades que las otras versiones de la herramienta no presentan.

TABLA 21: DEFINICIÓN DE VARIABLES DEL PARÁMETRO – LICENCIA

<b>Parámetros y variables de comparación</b>	<b>WIKITUDE</b>	<b>VUFORIA</b>	<b>METAIO SDK</b>
<b>Licencia</b>			
Versión Libre	0	4	4
Versión de prueba	4	0	0
Versión de paga	4	0	4

TABLA 22: VALORACIÓN DE VARIABLES DEL PARÁMETRO - LICENCIA

Resultados

<b>Herramientas</b>	<b>Resultados</b>	<b>Porcentaje</b>
<b>Ideal</b>	<b>4+4+4=12</b>	<b>100%</b>
Wikitude	0+4+4=8	66,6%
Vuforia	4+0+0=4	33,3%
Metaio SDK	4+0+4=8	66,6%

TABLA 23: INTERPRETACIÓN DE RESULTADOS - LICENCIA

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

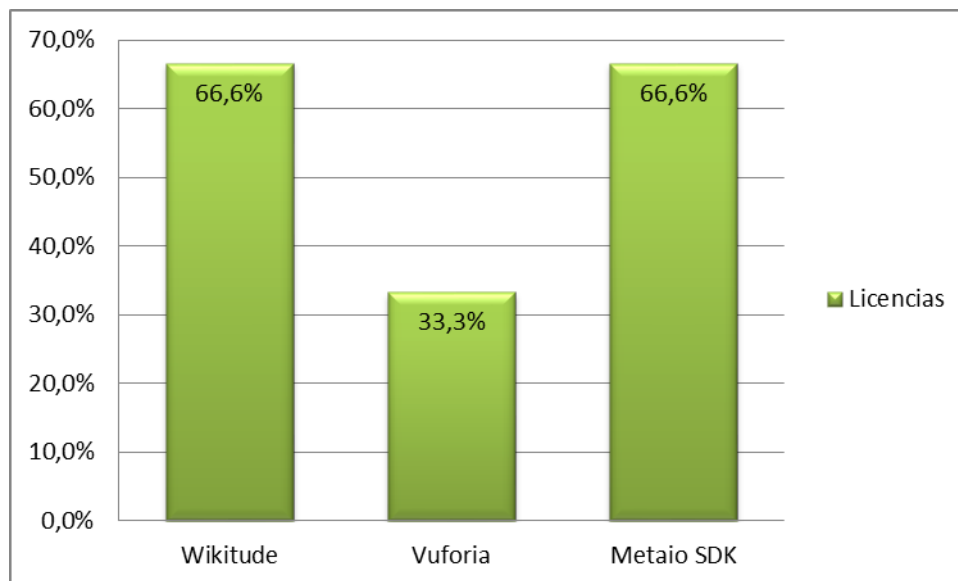


ILUSTRACIÓN 49: GRÁFICO DE RESULTADO DEL PARÁMETRO - LICENCIA

Como se observa en la Tabla 23 se consideran tres tipos de licencia, las cuales dependiendo de las funcionalidades y poder adquisitivos del programador pueden resultar en un factor importante a considerar. Como muestra el gráfico de la Ilustración 49, dos de tres herramientas presentan el mismo porcentaje, con un 66.6%, pero este se debe complementar con la Tabla 22.

Wikitude no presenta una versión libre de pago, pero sí una versión de prueba que tiene todas las funcionalidades de la herramienta, la diferencia que presenta con la que se debe pagar, es que la primera tiene una marca de agua indicando que es la versión trial. El valor de la versión profesional va desde €99 a €1499, unos \$69.314 a \$1.049.507 pesos chilenos.

Vuforia es el único de la muestra que es 100% libre y siendo la primera opción para empezar a trabajar con AR, pero esa gratuidad puede tener un punto en contra, ya que esta herramienta no trabaja con GPS, como se observó en puntos anteriores.

Metaio SDK a diferencia de Wikitude, tiene una versión libre que tiene las mayorías de las funcionalidades de Metaio, pero para tener acceso a todas ellas se deben pagar €2,950.00, unos \$2.065.408 pesos chilenos.

### 3.3.3. Puntajes alcanzados:

Teniendo conocimiento sobre todos los parámetros de evaluación, es necesario mencionar que no todos tienen la misma importancia. Es por esto que cada parámetro tiene una ponderación diferente para calcular el puntaje final. Se ha designado con una de las mayores ponderaciones a los parámetros “Plataforma” y “Licencia” con un 25% cada uno, debido a que una herramienta que permite crear aplicaciones para varios OS y además tiene distintos tipos de licencia, es muy bien valorado por los desarrolladores al momento de la elección de una de estas. En un peldaño más abajo se encuentra “Contenido de API” con un 20%, ya que es una parte fundamental para cualquier programador tener esta documentación al momento de crear de aplicaciones de AR. Luego se consideran con un 12,5% las técnicas de AR, ya sea el “Uso de GPS” o la “Búsqueda Visual”. Finalmente, se tiene el parámetro “Framework” con el restante 5%, que a razón del autor, debe tener la más baja ponderación, debido a que los esfuerzos que se dedican a construir la lógica de una aplicación son más importantes que lo visual, incluso se puede delegar lo visual a un diseñador gráfico profesional.

En la Ilustración 50 ilustra las ponderaciones que se le asignaron a cada parámetro:

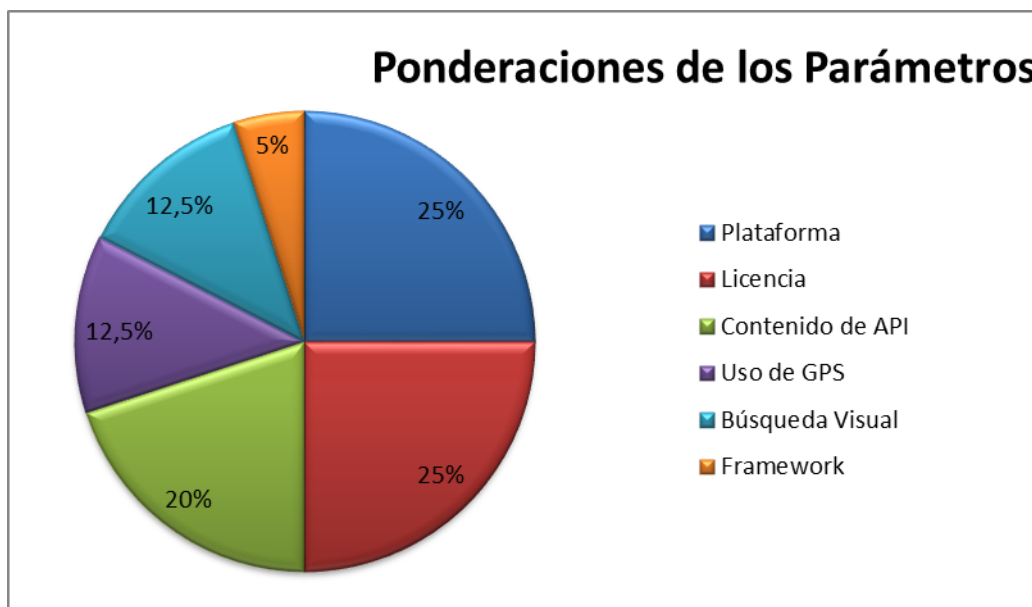


ILUSTRACIÓN 50: GRÁFICO DE PONDERACIONES DE LOS PARÁMETROS DE COMPARACIÓN

La Tabla 24 muestra los resultados de todos los parámetros de evaluación anteriores.

Parámetro	Ponderación	Variable	Wikitude	Puntaje Obtenido	Vuforia	Puntaje Obtenido	Metaio SDK	Puntaje Obtenido
P1	25%	1.1	4	1	4	1	4	1
		1.2	4	1	4	1	4	1
		1.3	2	0,5	0	0	0	0
		1.4	0	0	0	0	0	0
<b>Sub-total</b>				<b>2,5</b>		<b>2</b>		<b>2</b>
P2	12,5%	2.1	4	0,5	0	0	4	0,5
		2.2	4	0,5	0	0	4	0,5
		2.3	4	0,5	0	0	3	0,375
		2.4	4	0,5	0	0	4	0,5
<b>Sub-total</b>				<b>2</b>		<b>0</b>		<b>1,875</b>
P3	12,5%	3.1	4	0,5	4	0,5	4	0,5
		3.2	4	0,5	4	0,5	4	0,5
		3.3	4	0,5	4	0,5	4	0,5
		3.4	4	0,5	4	0,5	4	0,5
		3.5	4	0,5	4	0,5	4	0,5
<b>Sub-total</b>				<b>2,5</b>		<b>2,5</b>		<b>2,5</b>
P4	20%	4.1	3	0,6	2	0,4	3	0,6
		4.2	3	0,6	3	0,6	3	0,6
		4.3	4	0,8	4	0,8	4	0,8
<b>Sub-total</b>				<b>2</b>		<b>1,8</b>		<b>2</b>
P5	5%	5.1	2	0,1	2	0,1	2	0,1
		5.2	1	0,05	0	0	3	0,15
		5.3	3	0,15	2	0,1	3	0,15
		5.4	2	0,1	2	0,1	2	0,1
<b>Sub-total</b>				<b>0,4</b>		<b>0,3</b>		<b>0,5</b>
P6	25%	6.1	0	0	4	1	4	1
		6.2	4	1	0	0	0	0
		6.3	4	1	0	0	4	1
<b>Sub-total</b>				<b>2</b>		<b>1</b>		<b>2</b>
<b>Total</b>				<b>11,4</b>		<b>7,6</b>		<b>10,875</b>

TABLA 24: PUNTAJES TOTALES ALCANZADOS

Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android - Estudio Comparativo



### **3.4. Interpretación de los resultados obtenidos**

Luego de realizar las comparaciones entre las herramientas en estudio, se puede observar que existe una pequeña diferencia entre la mejor evaluada versus la peor, dicha discrepancia es de 3,8 puntos la cual se debe casi en su totalidad a que la herramienta Vuforia no trabaja con uso GPS, perdiendo terreno con respecto a los otros dos SDK, pero si se observa en los demás parámetros se encuentran a la par de sus contrincantes. Entre el primero (Wikitude) y el segundo (Metaio SDK) la brecha se acorta más aún con un 0,525, demostrando que estas herramientas tienen más similitudes que diferencias y que independiente de la elección, el desarrollador podrá crear una aplicación de AR cualquiera sea su orientación.

Luego de realizar el estudio, la herramienta que destaca es Wikitude, siendo la primera opción a recomendar para construir una aplicación con AR. Con una licencia trial, se puede crear aplicaciones tanto con utilización de GPS y/o con reconocimiento de imágenes. De simple instalación y bastante documentación.

En segundo lugar, pero no menos valorado, se encuentra Metaio SDK, que al compartir muchas similitudes con la herramienta anterior, igual es recomendable para construcción de aplicaciones de AR, debido a que presenta una licencia libre con bastantes características de la versión de paga, y al igual que Wikitude se puede trabajar con GPS y reconocimiento de imágenes.

En un tercer lugar terminó Vuforia, su principal desventaja es no soportar el GPS, perdiendo terreno frente a los desarrolladores que crean aplicaciones cuyo funcionamiento depende del lugar en que se encuentra el dispositivo. Pero no todo es malo, si desea trabajar con imágenes en dos dimensiones es bastante competente con respecto a los otros dos instrumentos estudiados.

A continuación, se presenta la Tabla 25 con los principales puntos expuestos durante el estudio:

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Estudio Comparativo

	<b>Wikitude</b>	<b>Vuforia</b>	<b>Metaio SDK</b>
<b>Plataforma</b>	Android / iOS /BlackBerry	Android/iOS	Android/iOS
<b>GPS</b>	✓	✗	✓
<b>Búsqueda Visual</b>	✓	✓	✓
<b>API</b>	✓	✓	✓
<b>Framework</b>	Target manager /3D Encoder	Target manager	Creator
<b>Licencia</b>	Libre con restricciones	Libre	Libre con restricciones

TABLA 25: RESUMEN DE CARACTERISTICAS DE LOS SDK ANALIZADOS

# **CAPÍTULO 4: CASOS PRÁCTICOS**

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Casos Prácticos

### **4.1.Introducción**

Una vez realizado el estudio comparativo de las tecnologías de realidad aumentada del capítulo 3, es el momento de exponer las implementaciones de pequeñas aplicaciones android utilizando 2 de las 3 herramientas ya mencionadas en esta memoria (Wikitude, Metaio SDK).

Con cada herramienta escogida, se crearon ejemplos para mostrar las principales características de la AR, como reconocimiento de imagen, crear puntos de interés y posicionarlos en un sistema GPS y así también establecer el modo de trabajo de estas tecnologías.

## **4.2. Caso Práctico 1: Geo localización**

### **4.2.1. Objetivo**

El objetivo general de este caso práctico es crear una aplicación android, la cual contiene los puntos geográficos de los edificios y salas del campus “Fernando May” de la universidad del Bío-bío ubicado en la ciudad de Chillán. Gracias a la AR, el usuario puede saber dónde está y cómo llegar a dicho lugar, seleccionando uno de esos Puntos de Interés (POI). En otras palabras el ejemplo es un “mini-GPS” de la universidad del Bío-Bío.

La aplicación está conformada por los POI de los edificios y aulas del campus, los cuales representan de modo visual las coordenadas geográficas del lugar (Ilustración 51). Cada punto de interés debe ser un icono para que se fácil de visualizar por el usuario.

Además, estos puntos son mostrados por la cámara del Smartphone, ya que es este componente quien mezcla el mundo real con el virtual, capturando imágenes a su alrededor y añadiéndole información extra, como por ejemplo la descripción de los edificios.

También en esta aplicación, se puede seleccionar cada POI para obtener información adicional a la que muestra el icono antes mencionado o simplemente destacar en relación a los otros puntos que muestra la aplicación (Ilustración 52).

Finalmente, se puede hacer un seguimiento a un punto determinado, es decir que aparece un indicador mostrando donde está dicho POI en la cámara, para que el usuario pueda seguirlo y encontrarlo, como hacen los GPS que se conocen actualmente.

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Casos Prácticos

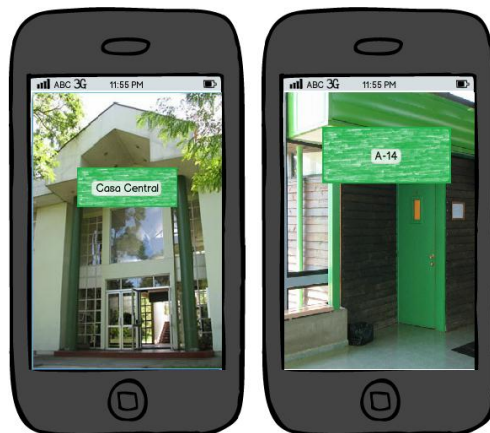


ILUSTRACIÓN 51: MAQUETA DE PANTALLA CON POI EN LA UNIVERSIDAD



ILUSTRACIÓN 52: MAQUETA DE PANTALLA CUANDO UN POI ES SELECCIONADO

### 4.2.2. Descripción

Para la implementación de este caso práctico, se utilizó la herramienta Wikitude debido a que obtuvo el mayor puntaje en este ítem del estudio realizado anteriormente.

Para crear la AR se utiliza tecnología JavaScript (JS) inserta en de un archivo HTML, Este archivo es quien se encarga de la parte visual de la AR, donde se deben crear los puntos de interés, el indicador y todos los recursos visuales utilizados en la aplicación, utilizando las librerías del SDK. Este archivo es referenciado por la actividad que contiene el “architectView”, quien es el que notifica acerca de los eventos del ciclo de vida del programa android.

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Casos Prácticos

El `architectView` crea una cámara de superficie y se ocupa de los eventos del sensor. Se implementa en JavaScript y se envasa en la carpeta `assets` de la aplicación (como en este proyecto) o en un servidor propio. `ArchitectView` es un HTML / JavaScript y para llamar a estos métodos de Wikitude se debe escribir `AR.nombrecomponente` (por ejemplo `AR.GeoObject`).

Es muy importante notificar al `ArchitectView` sobre los eventos del ciclo de vida de la actividad implementando los métodos `onCreate()`, `onPostCreate()`, `onPause()`, `onDestroy()`. La mejor práctica es definir una variable para el `architectView` en la actividad e inicializarlo con `setContentview()` en el método `onCreate()`, para luego acceder a través de la variable.

En este caso los datos de la posición global son estáticos, por lo tanto, el contenido puede ser almacenado dentro de la aplicación. Donde se crea un archivo JavaScript (por ejemplo `myjsonData.js`) donde se define una variable accesible globalmente.

Adicionalmente, se utiliza una clase auxiliar, llamada `ProveedorUbicacion.class` la cual verifica y notifica si está activado el GPS y la conexión a la red, y obtiene la ubicación actual del dispositivo.

### 4.2.3. Implementación

A continuación se presenta los pasos básicos para construir la aplicación android con AR:

1. Al momento de crear el proyecto android en el IDE eclipse, se debe agregar la librería de Wikitude para poder crear la experiencia AR.
2. Se debe crear la estructura básica de toda aplicación android, con sus métodos del ciclo de vida, como `onCreate()`, `onDestroy()`, etcétera.
3. Luego se agregan variables para manejar y observar estados de la aplicación. Algunas de las variables importante son: “`ArchitectView`” (notifica acerca de los eventos del ciclo de vida del programa), “`SensorAccuracyListener`” (quien se encarga de ver el estado del sensor del teléfono), “`LocationListener`” (Se utiliza para recibir notificaciones cuando la ubicación ha cambiado) y “`Location`” (quien guarda la última localización conocida por la aplicación).
4. Posteriormente se implementa el método `onCreate()`, inicializando la variable `ArchitectView` e enlazarla con el componente correspondiente en el lado visual de la

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Casos Prácticos

- aplicación (XML de la actividad), además de ingresar la serial de uso que nos proporciona la herramienta (en el caso de la versión de prueba, puede ser cualquier cadena de String).
5. Conjuntamente al paso anterior se debe inicializar la variable encargada del sensor del programa, `SensorAccuracyListener`.
  6. También en el método `onCreate()`, se debe programar el comportamiento cuando la posición del Smartphone es actualizado o cambiado, en otras palabras inicializar la variable `LocationListener`.
  7. El métodos `onPostCreate()`, su principal función es cargar el archivo HTML con los JavaScript (JS) de la AR que más adelante se detallarán.
  8. Por último implementar con los métodos restantes del ciclo de vida de la aplicación, como `onPause()`, `onDestroy()`, entre otros.
  9. Una vez que se tiene la estructura básica de la aplicación android, se debe trabajar en los JS que crea la información. Se crea un archivo HTML que sirve de “manejador de Script”, invocando a todos los que sean necesarios para crear la AR. Este HTML se recomienda colocarlo en una carpeta de recursos o “assets”.
  10. En este caso práctico, presenta un JS que se encarga de extraer e imprimir los datos de los POI mostrados en la cámara del teléfono. Además, se encarga de cargar los recursos visuales que se muestran en pantalla, ya sean imágenes para los POI e indicador con el método “`AR.ImageResource`” de la API de Wikitude. También debe estar en la misma o en una sub carpeta de “assets” con todos los recursos a utilizar.
  11. Como se mencionó en el punto anterior, existe un JS con los datos de los POI, este solo contiene una variable JSON, con todos los datos que se desea mostrar en la aplicación, como en este caso, los nombres y direcciones geográficas de los edificios y salas del campus Fernando May.
  12. De igual forma, se puede programar la forma de los POI, y darle comportamiento cuando son seleccionado, como por ejemplo, cambie de color o tamaño. Utilizando la API de Wikitude llamada “`AR.Label`”.
  13. Por último, se puede agregar componentes extras a nuestra AR, como por ejemplo un radar, donde también es un JS que es llamado por el HTML mencionado anteriormente y se utiliza las API de Wikitude llamada “`AR.radar`”.



## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Casos Prácticos

### 4.2.4. Capturas de la aplicación



ILUSTRACIÓN 53: POI SIN MARCAR

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Casos Prácticos



ILUSTRACIÓN 54: POI SELECCIONADO



ILUSTRACIÓN 55: PUNTERO INDICANDO DONDE ESTA EL POI SELECCIONADO

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Casos Prácticos

### **4.2.5. Resultado**

Luego de la implementación del caso práctico el autor saca como conclusión, que si es posible crear una aplicación de AR de posicionamiento global, gracia a la herramienta Wikitude. Su sistema de HTML/JavaScript hace que la curva de aprendizaje sea rápida, con solo aprender algunos elementos de la librería de la herramienta ya se puede familiarizar con su metodología de trabajo. Además su lado lógico (clases Java) es muy similar a las de otras aplicaciones android, sobrescribiendo los métodos de la clase activity.

Como se puede observar en las capturas de la aplicación, el programa cumple con los objetivos mencionados al comienzo de esta sección de observar los POI y hacer seguimiento a uno de ellos, que incluso este mismo caso puede ser una parte de una aplicación más grande, que su público objetivo sea docentes y alumnos que no conocen el campus Fernando May o ingresar puntos geográficos de otros campus de la universidad del Bío-Bío.

## 4.3. Caso Práctico 2: Reconocimiento de Imagen

### 4.3.1. Objetivo

El objetivo general de este caso práctico es crear una aplicación android, la cual reconozca el logo de la institución y que en la pantalla del Smartphone aparezca un modelo en tercera dimensión. Gracias a la AR el usuario podrá ver de manera lúdica, otra forma de utilizar la característica de reconocer imágenes.



ILUSTRACIÓN 56: MAQUETA DE PANTALLA DEL CASO PRÁCTICO DE RECONCIMIENTO DE IMAGEN

### 4.3.2. Descripción

Para la implementación de este caso práctico, se utilizó la herramienta Metaio SDK debido a que obtuvo un puntaje aceptable en este ítem del estudio realizado anteriormente.

El fin del caso práctico es ver un modelo en 3D en la imagen del logo de la universidad. A partir de aquí, vamos a llamar a esta imagen "referencia de seguimiento".

La aplicación se implementa en una clase llamada Tutorial1. Esta clase es una subclase de la MetaioSDKViewActivity, que implementa los métodos necesarios para crear la experiencia de AR. Esta clase se puede extender y sobrescribir los métodos necesarios para la lógica de la aplicación. Por ejemplo en este caso *loadContent()*.

Hay algunos pasos simples que hay que hacer para mostrar el modelo en la referencia de seguimiento:

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Casos Prácticos

- **Asignar archivo de referencia de seguimiento al SDK:** Para cargar la configuración de seguimiento, simplemente se ejecuta el método booleano *setTrackingConfiguration(trackingDataFile)*, donde el parámetro “metaioSDK” es una instancia de la clase SDK y es él, quien debe llamar dicho método. *TrackingDataFile* es un camino para archivo XML de configuración de seguimiento utilizado en la escena. En este caso se utiliza la configuración de seguimiento con una imagen del logo de la universidad como patrón de referencia.
- **Cargar el modelo a la escena:** Para cargar el modelo se crea un geometría 3D con el método *IGeometry createGeometry(metaioModel)*, donde *IGeometry* es una clase dentro de SDK que trata los objetos en la escena y *metaioModel* es la ruta de acceso para el archivo que contiene el modelo 3D.
- **Establecer la escala y la rotación para el modelo (si es necesario):** Cuando el modelo se ha cargado correctamente se puede escalar usando *setScale()*. De manera similar se puede mover y rotar el modelo, simplemente llamando a los métodos *setTranslation()* y *setRotation()*.

### 4.3.3. Implementación

A continuación se presenta los pasos básicos para construir la aplicación android con AR:

1. Antes de comenzar a crear un proyecto con la herramienta Metaio SDK, se debe cargar su librería en el IDE eclipse, que es un proyecto android.
2. Se crea una clase que extiende de “ARViewActivity” perteneciente a la librería de Metaio SDK y su principal función es cargar y configurar el contenido de toda la aplicación de AR, como el archivo de configuración XML de seguimiento y el modelo 3D a renderizar.
3. Por su parte, la Actividad principal (mainActivity) de la aplicación se encarga de asignar los recursos y cargar el HTML que que invoca a la clase que extiende de “ARViewActivity”.
4. La configuración XML de seguimiento de este caso es de un rastreador sin marcadores, y se encarga de establecer el target de la aplicación y el modelo a renderizar, con parámetros que los desarrolladores de la herramienta recomiendan dejarlas por defectos si no es un experto en el tema.

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Casos Prácticos

### 4.3.4. Capturas de pantallas



ILUSTRACIÓN 57: DIFERENTES CAPTURAS DE PANTALLAS DE LA APLICACIÓN FUNCIONANDO

### 4.3.5. Resultado

Luego de la implementación del caso práctico el autor saca como conclusión, que si es posible crear una aplicación de AR con reconocimiento de imagen, gracias a la herramienta Metaio SDK. Con su clase principal java (donde se crea la AR), implementando elementos de la librería de la herramienta se puede comprender rápidamente como se debe trabajar.

Como se puede observar en las capturas de la aplicación, el programa cumple con el objetivo mencionado al comienzo de este caso práctico, ya que reconoce la imagen del logo de la universidad del Bío-bío, en diferentes partes (Ilustración 57), dando paso al modelo 3D sobre él.

# **CAPÍTULO: CONCLUSIONES**

## CONCLUSIONES

### Conclusiones Generales

- C1.** Mediante el estudio comparativo que se realizó a las herramientas de AR, se determinó que Wikitude es la más completa en comparación a Metaio SDK y Vuforia, debido a que presenta más y/o mejores funcionalidades con respecto a las otras herramientas.
- C2.** Con el estudio comparativo realizado, se permitió observar todas las nuevas características que brinda la tecnología de AR al mundo de los Smartphones, ya sea Geo localización, reconocimiento de imágenes o la mezcla de estas, poniendo a prueba la imaginación a los programadores que las utilizan.
- C3.** La característica de Geo localización es a la fecha, la que más sorprende a los usuarios, ya sea por lo novedoso de utilizar el Smartphone como un “ojo extra” o por la forma de entregar información de los POI. Esta característica debería estar presente en las aplicaciones de AR futuras.
- C4.** La característica de reconocimiento de imágenes, es la funcionalidad básica que debe tener una herramienta de AR, reconociendo marcas (código QR) o una imagen en particular y así obtener información adicional de nuestro alrededor.
- C5.** Esta tecnología está en un constante cambio y mejoramiento, a pesar que se comenzó a trabajar con ella hace varias décadas atrás, solo en estos últimos años ha tenido un verdadero auge entre los desarrolladores, debido a que en el pasado no existía el hardware adecuado para esta tecnología y actualmente tenemos unos verdaderos computadores de bolsillos capaz de soportarlas.
- C6.** La curva de aprendizaje de las distintas herramienta, depende del nivel de conocimiento de los lenguajes involucrados en la construcción de AR, los principales lenguajes son Java (enfocado en la programación en android), JavaScript y HTML.



Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Conclusiones

- C7.** El sistema operativo Android es el más apropiado para desarrollar aplicaciones si se quiere llegar a un gran número de usuarios, ya que abarca el 79.3% del mercado de dispositivos móviles en el mundo.
  
- C8.** A modo personal, fue una tarea difícil realizar todo el trabajo, debido a que cursar el último período académico y la falta de un Smartphone adecuado para realizar las pruebas de los prototipos, retrasó el proyecto, pero gracias a una reorganización y el apoyo del profesor guía y otras personas, se logró cumplir los objetivos de esta memoria desarrollando nuevas habilidades y utilizar las herramientas estudiadas.

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Conclusiones

### **Recomendaciones:**

- Para realizar aplicaciones android de reconocimiento de imágenes, se recomienda utilizar Metaio SDK, debido a que tiene un nivel de reconocimiento más amplio y general, utilizando el formato de imágenes conocidas como PNG.
- Para realizar aplicaciones android de posicionamiento global, se recomienda utilizar Wikitude, debido a que ofrece variadas formas de trabajar la obtención de puntos de interés (POI), como por ejemplo con un archivo JSON o un servicio Web (WebService). Además, la programación de la AR se realiza mediante tecnología JavaScript, separándolo de la estructura básica de cualquier aplicación y facilitando el encapsulamiento en el caso que éste se encuentre en una aplicación más grande.
- Para trabajar con Metaio SDK se recomienda tener conocimientos sobre elaboración de modelos 3D (con extensión MD2, FBX o OBJ) o en su defecto tener contactos con diseñadores gráficos que faciliten los modelos para incluirlas en la aplicación.
- Se recomienda utilizar un dispositivo móvil con un procesador dual-core de 1GHz o superior y una cámara de 4 o más Mega pixeles, ya que permiten un renderizado más rápido y una mejor captura de la escena para el procesado.

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Conclusiones

### **Trabajos Futuros**

Es importante destacar que este proyecto no tiene como objetivo el desarrollo de una aplicación, sino, el estudio comparativo entre las herramientas de AR, Sin embargo, se pueden tomar las ideas de los casos prácticos y unirlos a una aplicación más grande, como por ejemplo, una aplicación para estudiantes recién ingresados a la Universidad del Bío-Bío, con un mapa de los campus correspondiente con AR, un agenda, horario y fichero virtual, entre otras funcionalidades.

También siguiendo la línea de la memoria, una buena idea sería realizar un estudio de las herramientas de realidad aumentada en otra plataforma, por ejemplo iOS, debido a que es el segundo OS más vendido en el mercado y sus equipos tienen características distintas a la mayoría de los Smartphone, además se puede obtener otro punto de comparación para elegir la mejor herramienta de AR.

## BIBLIOGRAFÍA

1. *Augmented Reality SDK Comparison*. (2013). Retrieved Noviembre 7, 2013, from <http://socialcompare.com>: <http://socialcompare.com/es/comparison/augmented-reality-sdks>
2. Azuma, R. T. (1997). Teleoperators and Virtual Environments. In R. T. Azuma, *A survey of augmented reality* (pp. 355-385.).
3. Cegarra Sánchez, J. (2011). La investigación experimental. In J. Cegarra Sánchez, *Metodología de la investigación científica y tecnológica* (pp. 85-88). Ediciones Díaz de Santos.
4. *Escala de Likert*. (n.d.). Retrieved Diciembre 2, 2013, from [ict.edu](http://www.ict.edu): [http://www.ict.edu.mx/acervo\\_bibliotecologia\\_escalas\\_Escala%20de%20Likert.pdf](http://www.ict.edu.mx/acervo_bibliotecologia_escalas_Escala%20de%20Likert.pdf)
5. García Álvarez, J. A. (2012, Abril). *Así funciona el GPS*. Retrieved Octubre 24, 2013, from [asifunciona.com](http://www.asifunciona.com): [http://www.asifunciona.com/electronica/af\\_gps/af\\_gps\\_10.htm](http://www.asifunciona.com/electronica/af_gps/af_gps_10.htm)
6. Ghumatkar, N. (2013, Junio 17). *Top 5 Augmented Reality SDK*. Retrieved Diciembre 10, 2013, from [techgyaan.com](http://www.techgyaan.com): <http://www.techgyaan.com/top-5-augmented-reality-sdk>
7. Hammarck, B. (2012, Mayo 22). *How a Smartphone Knows Up from Down (accelerometer)*. Retrieved Octubre 22, 2013, from Youtube: <http://www.youtube.com/watch?v=KZVgKu6v808>
8. Indriago Alvarado, P. M. (2004). *Sensores de posición descripción, selección y uso*. Retrieved Octubre 27, 2013, from <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r34716.PDF>
9. Jamrich Parsons, J. (2008). Herramientas de programación. In J. Jamrich Parsons, *Conceptos de Computación: Nuevas Perspectivas* (pp. 664-665). Cengage Learning Editores.
10. Matyszczok, C. (2003). *Development of a Real Time Image Based Object Recognition Method for Mobile AR-Devices*.
11. Metaio . (2013). *The NEW Metaio SDK*. Retrieved Noviembre 15, 2013, from [www.metaio.com](http://www.metaio.com): <http://www.metaio.com/products/sdk/>
12. Metaio. (2013). *Getting Started with SDK*. Retrieved Noviembre 15, 2013, from [dev.metaio.com](http://dev.metaio.com): <http://dev.metaio.com/sdk>

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Bibliografía

13. Milgram, P., & Kishino, F. (1994). Proceedings of Telem manipulator and Telepresence Technologies. In P. Milgram, & F. Kishino, *Augmented Reality: A class of displays on the reality-virtuality continuum* (p. 2351).
14. Morán Leal, R. A. (2012). Realidad Aumentada. In R. A. Morán Leal, *Servicios de Recomendación Contextual para Instituciones* (pp. 18-23). Cuernavaca, Morelos, México.
15. Mulloni, A. (2010). *North-centred Orientation Tracking on Mobile Phones*.
16. Pedrozo Petrazzini., G. O. (2012). *Monografía: Sistemas Operativos en Móviles*. Retrieved Septiembre 24, 2013, from [http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/Sistemas\\_Operativos\\_en\\_Dispositivos\\_Moviles.pdf](http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/Sistemas_Operativos_en_Dispositivos_Moviles.pdf)
17. RAE. (2013). *Prototipo*. Retrieved Agosto 26, 2013, from RAE: <http://lema.rae.es/drae/srv/search?id=FdZrDwnWtDXX2M1aTCwz>
18. Salazar Alvarez, I. A. (2013). *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA INFORMACIÓN TURÍSTICA BASADO EN REALIDAD AUMENTADA*. Lima.
19. Serrano Mamolar, A. (2012). Estudio de las Herramientas. In A. Serrano Mamolar, *Herramientas de desarrollo libres para aplicaciones de Realidad para aplicaciones de Realidad comparativo entre ellas* (pp. 38-39). Valencia.
20. Tanenbaum, A. S. (2003). EL SISTEMA TELEFÓNICO MÓVIL. In A. S. Tanenbaum, *Redes de computadoras* (pp. 152-166). Pearson Educación.
21. Univesidad Tecnica Federico Santa Maria. (2009, Septiembre 4). *Ingenieros USM aplican realidad aumentada para "ver" a través del cráneo*. Retrieved Octubre 21, 2013, from Dirección General de comunicaciones: <http://www.dgc.usm.cl/2009/09/04/ingenieros-usm-aplican-realidad-aumentada-para-%E2%80%9Cver%E2%80%9D-a-traves-del-craneo>
22. Vicente García Díaz. (2012). *Wikitude. Servicios web*. Retrieved Noviembre 12, 2013, from slideshare.net/: <http://www.slideshare.net/vicegd/wikitude-servicios-web>
23. Vuforia. (2013). *Resources: Text - Word Targets*. Retrieved Noviembre 13, 2013, from developer.vuforia.com: <https://developer.vuforia.com/resources/dev-guide/word-targets>
24. Vuforia. (2013). *Resources: Text Recognition*. Retrieved Noviembre 13, 2013, from developer.vuforia.com: <https://developer.vuforia.com/resources/sample-apps/text-recognition>

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Bibliografía

25. Vuforia. (2013). *Resources: Vuforia SDK Architecture*. Retrieved Noviembre 18, 2013, from vuforia: <https://developer.vuforia.com/resources/dev-guide/vuforia-ar-architecture>
26. Wikipedia. (2013). *Android*. Retrieved Agosto 26, 2013, from Wikipedia: <http://es.wikipedia.org/wiki/Android#Etimolog.C3.ADa>
27. Wikipedia. (2013). *BlackBerry*. Retrieved Septiembre 24, 2013, from Wikipedia: <http://es.wikipedia.org/wiki/BlackBerry>
28. Wikipedia. (2013). *Kit de desarrollo de software*. Retrieved Agosto 26, 2013, from Wikipedia: [http://es.wikipedia.org/wiki/Kit\\_de\\_desarrollo\\_de\\_software](http://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software)
29. Wikipedia. (2013). *Realidad Virtual*. Retrieved Agosto 26, 2013, from Wikipedia : [http://es.wikipedia.org/wiki/Realidad\\_virtual](http://es.wikipedia.org/wiki/Realidad_virtual)
30. Wikipedia. (2013). *Realidada Aumentada*. Retrieved agosto 2013, 26, from Wikipedia: [http://es.wikipedia.org/wiki/Realidad\\_aumentada](http://es.wikipedia.org/wiki/Realidad_aumentada)
31. Wikipedia. (2013). *Renderización*. Retrieved Nobiembre 7, 2013, from Wikipedia: <http://es.wikipedia.org/wiki/Renderizaci%C3%B3n>
32. Wikipedia. (2013). *Windows Phone*. Retrieved Septiembre 24, 2013, from Wikipedia: [http://es.wikipedia.org/wiki/Windows\\_Phone](http://es.wikipedia.org/wiki/Windows_Phone)
33. Wikitude. (n.d.). Retrieved Noviembre 8, 2013, from Wikitude SDK for BlackBerry 10: <http://developer.wikitude.com/documentation/blackberry10>
34. Wikitude. (2012, Agosto). *Wikitude SDK 1.1 - webinar*. Retrieved Noviembre 12, 2013, from slideshare.net: <http://de.slideshare.net/wikitude/wikitude-sdk-11-webinar>

**ANEXO A:  
CÓDIGO FUENTE DE CASO  
PRÁCTICO DE GEO  
LOCALIZACIÓN**

## Anexo A: código fuente de caso práctico de geo localización

A continuación se presenta el código utilizado para construir la aplicación android, donde en primera instancia se observará la lógica del programa para luego seguir con los JavaScript:

```

1 package com.example.geolocalizacionubb;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 public class MainActivity extends Activity {
19     // Variables Básicas
20     protected ArchitectView architectView;
21     protected SensorAccuracyChangeListener sensorAccuracyListener;
22     protected Location lastKnownLocation;
23     protected ProveedorUbicacion locationProvider;
24     protected LocationListener locationListener;
25
26
27     protected void onCreate(Bundle savedInstanceState) {
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94     protected void onResume() {
95
96
97
98
99
100
101
102
103
104
105     protected void onPause() {
106
107
108
109
110
111
112
113
114
115
116     protected void onStop() {
117
118
119
120
121     protected void onDestroy() {
122
123
124
125
126
127
128
129
130
131
132
133     public void onLowMemory() {
134
135
136
137
138
139
140
141     protected void onCreate(Bundle savedInstanceState) {
142
143
144
145
146
147
148
149
150
151
152
153
154
155     public boolean onCreateOptionsMenu(Menu menu) {
156
157
158
159
160
161 }
162

```

ILUSTRACIÓN 58: CLASE MAINACTIVITY



Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo A

```
26 @Override
27 protected void onCreate(Bundle savedInstanceState) {
28     super.onCreate(savedInstanceState);
29     setContentView(R.layout.activity_main);
30     this.architectView = (ArchitectView) this
31         .findViewById(R.id.architectView);
32     final ArchitectConfig config = new ArchitectConfig("key1");
33     this.architectView.onCreate(config);
34
35     this.sensorAccuracyListener = new SensorAccuracyChangeListener() {
36         @Override
37         public void onCompassAccuracyChanged(int accuracy) {
38             /* UNRELIABLE = 0, LOW = 1, MEDIUM = 2, HIGH = 3 */
39             if (accuracy < SensorManager.SENSOR_STATUS_ACCURACY_MEDIUM
40                 && MainActivity.this != null
41                 && !MainActivity.this.isFinishing()) {
42                 Toast.makeText(MainActivity.this, "Precisión cambió",
43                     Toast.LENGTH_LONG).show();
44             }
45         }
46     };
```

ILUSTRACIÓN 59: MÉTODO ONCREATE ( )

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo A

```

49     this.locationListener = new LocationListener() {
50         @Override
51         public void onStatusChanged(String provider, int status,
52             Bundle extras) {
53         }
54         @Override
55         public void onProviderEnabled(String provider) {
56         }
57         @Override
58         public void onProviderDisabled(String provider) {
59         }
60         @Override
61         public void onLocationChanged(final Location location) {
62             if (location != null) {
63                 MainActivity.this.lastKnownLocaton = location;
64                 if (MainActivity.this.architectView != null) {
65                     if (location.hasAltitude()) {
66                         MainActivity.this.architectView.setLocation(
67                             location.getLatitude(),
68                             location.getLongitude(),
69                             location.getAltitude(),
70                             location.hasAccuracy() ? location
71                                 .getAccuracy() : 1000);
72                     } else {
73                         MainActivity.this.architectView.setLocation(
74                             location.getLatitude(),
75                             location.getLongitude(),
76                             location.hasAccuracy() ? location
77                                 .getAccuracy() : 1000);
78                     }
79                 }
80             }
81         }
82     };
83     this.architectView
84         .registerSensorAccuracyChangeListener(this.sensorAccuracyListener);
85     this.locationProvider = new ProveedorUbicacion(this,
86         this.locationListener);
87 }

```

ILUSTRACIÓN 60: MÉTODO ONCREATE ( ) 2

## Estudio de Herramientas de Realidad Aumentada Sobre Plataforma Android – Anexo A

- Métodos `onPostCreate ()`, `onPause`, `OnDestroy ()` y otros de la clase `MainActivity`:

```

100 @Override
101 protected void onPause() {
102     super.onPause();
103     if (this.architectView != null) {
104         this.architectView.onPause();
105     }
106     if (this.locationProvider != null) {
107         this.locationProvider.onPause();
108     }
109 }
110
111 @Override
112 protected void onStop() {
113     super.onStop();
114 }
115
116 @Override
117 protected void onDestroy() {
118     super.onDestroy();
119     if (this.architectView != null) {
120         if (this.sensorAccuracyListener != null) {
121             this.architectView
122                 .unregisterSensorAccuracyChangeListener(this.sensorAccuracyListener);
123         }
124         this.architectView.onDestroy();
125     }
126 }
127
128 @Override
129 public void onLowMemory() {
130     super.onLowMemory();
131     if (this.architectView != null) {
132         this.architectView.onLowMemory();
133     }
134 }

```

ILUSTRACIÓN 61: MÉTODOS ONPAUSE Y ONDESTROY ()

```

136 @Override
137 protected void onCreate(Bundle savedInstanceState) {
138     super.onCreate(savedInstanceState);
139     if (this.architectView != null) {
140         this.architectView.onCreate();
141     }
142     try {
143         this.architectView.load("index.html");
144     } catch (IOException e1) {
145         Toast.makeText(MainActivity.this, "INDEX NO ENCONTRADO !! ",
146             Toast.LENGTH_LONG).show();
147     }
148 }
149
150 @Override
151 public boolean onCreateOptionsMenu(Menu menu) {
152     // Inflate the menu; this adds items to the action bar if it is present.
153     getMenuInflater().inflate(R.menu.main, menu);
154     return true;
155 }
156
157 }

```

ILUSTRACIÓN 62: MÉTODOS ONPOSTCREATE ()

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo A

- HTML:

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
2 <html>
3   <head>
4     <!-- meta-información básica -->
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6     <meta http-equiv="Content-Type" content="application/json; charset=utf-8">
7     <meta name="viewport" content="target-densitydpi=device-dpi, width = device-width,
8       user-scalable = 0, minimum-scale=1.0, maximum-scale=1.0" />
9     <title>Geo Localización UBB</title>
10    <link rel="stylesheet" href="css/default.css" id="tooltipStyle">
11    <script src="architect://architect.js"></script>
12    <script src="js/myJsonData.js"></script>
13    <script src="js/markerR.js"></script>
14    <script src="js/radar.js"></script>
15  </head>
16
17  <body>
18    <!-- elemento de estado que muestra el progreso de aplicacion-->
19    <div id="statusElement" class="status">
20      Iniciando ...
21    </div>
22    <div id="radarContainer"></div>
23    <!-- principal código JS de esta escena -->
24    <script src="js/poiRadar.js"></script>
25  </body>
26 </html>

```

ILUSTRACIÓN 63: HTML QUE MANEJA LOS JAVASCRIPT

- PoiRadar JavaScript:

```

1 var World = {
2   markerDrawable_idle: new AR.ImageResource("img/label.png"),
3   markerDrawable_selected: new AR.ImageResource("img/label2.png"),
4   markerDrawable_directionIndicator: new AR.ImageResource("assets/indi.png"),
5   markerList: [],
6   // llamado para inyectar nuevos datos POI
7   loadPoisFromJsonData: function loadPoisFromJsonDataFn(poiData) {
8     document.getElementById("statusElement").innerHTML = 'Cargando objetos JSON';
9     for (var i = 0; i < poiData.length; i++) {
10      var singlePoi = {
11        "lat": parseFloat(poiData[i].lat),
12        "lon": parseFloat(poiData[i].lon),
13        "title": poiData[i].name,
14        "alt": parseFloat(poiData[i].alt),
15        "description": poiData[i].description
16      };
17      World.markerList.push(new Marker(singlePoi));
18    }
19    document.getElementById("statusElement").innerHTML = 'Objetos JSON cargados';
20    PoiRadar.show();
21  },

```

ILUSTRACIÓN 64: JAVASCRIPT EXTRAE LOS DATOS DE JSON 1

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo A

```

22  /* última ubicación conocida del usuario, accesible a través
23  de userLocation.latitude, userLocation.longitude, userLocation.altitude*/
24  userLocation: null,
25  // actualizaciones de ubicación
26  locationChanged: function locationChangedFn(lat, lon,alt) {
27      World.userLocation = {
28          'latitude': lat,
29          'longitude':lon,
30          'altitude': alt
31      };
32      if (!World.alreadyRequestedData) {
33          World.requestDataFromLocal(lat, lon,alt);
34          World.alreadyRequestedData = true;
35      }
36  },
37  // datos de petición de POI
38  requestDataFromLocal: function requestDataFromLocalFn(lat, lon,alt) {
39      World.loadPoisFromJsonData(JsonData);
40  }
41  };
42  AR.context.onLocationChanged = World.locationChanged;

```

ILUSTRACIÓN 65: JAVASCRIPT EXTRAER LOS DATOS DE JSON 2

- Radar JavaScript:

```

1  var PoiRadar = {
2  show: function showFn() {
3      AR.radar.enabled = true;
4  },
5  hide: function hideFn() {
6      AR.radar.enabled = false;
7  },
8  init: function initFn() {
9      AR.radar.container = document.getElementById("radarContainer");
10     AR.radar.background = new AR.ImageResource("assets/radar.png");
11     AR.radar.northIndicator.image = new AR.ImageResource("assets/radar_north.png");
12     AR.radar.positionX = 0.04;
13     AR.radar.positionY = 0.06;
14     AR.radar.width = 0.3;
15     AR.radar.centerX = 0.5;
16     AR.radar.centerY = 0.5;
17     AR.radar.radius = 0.3;
18     AR.radar.northIndicator.radius = 0.0;
19     AR.radar.enabled = true;
20     AR.radar.onClick = PoiRadar.clickedRadar;
21  },
22  clickedRadar: function clickedRadarFn() {
23     alert("Radar Clicked");
24  },
25  setMaxDistance: function setMaxDistanceFn(maxDistanceMeters) {
26     AR.radar.maxDistance = maxDistanceMeters;
27  }
28  };
29  PoiRadar.init();

```

ILUSTRACIÓN 66: JAVASCRIPT DEL RADAR

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo A

- Json JavaScript:

```

1  var jsonData =
2  [
3      {
4          "id": "1",
5          "longitude": "-36.617603",
6          "latitude": "-72.087071",
7          "description": "De Prueba",
8          "altitude": "100.0",
9          "name": "POI"
10     }
11 ];

```

ILUSTRACIÓN 67: JAVASCRIPT DE JSON

- Marker JavaScript:

```

1  var kMarker_AnimationDuration_ChangeDrawable = 500;
2  var kMarker_AnimationDuration_Resize = 1000;
3  function Marker(poiData) {
4      this.poiData = poiData;
5      this.isSelected = false;
6      this.animationGroup_idle = null;
7      this.animationGroup_selected = null;
8      var markerLocation = new AR.GeoLocation(poiData.latitude, poiData.longitude, poiData.altitude);
9      this.markerDrawable_idle = new AR.ImageDrawable(World.markerDrawable_idle, 2.5, {
10         zIndex: 0,
11         opacity: 1.0,
12         onClick: Marker.prototype.getOnClickTrigger(this)
13     });
14     this.markerDrawable_selected = new AR.ImageDrawable(World.markerDrawable_selected, 2.5, {
15         zIndex: 0,
16         opacity: 0.0,
17         onClick: null
18     });
19     this.titleLabel = new AR.Label(poiData.title.trunc(10), 1, {
20         zIndex: 1,
21         offsetY: 0.55,
22         style: {
23             textColor: '#FFFFFF',
24             fontStyle: AR.CONST.FONT_STYLE.BOLD
25         }
26     });
27     this.descriptionLabel = new AR.Label(poiData.description.trunc(15), 0.8, {
28         zIndex: 1,
29         offsetY: -0.55,
30         style: {
31             textColor: '#FFFFFF'
32         }
33     });

```

ILUSTRACIÓN 68: JAVASCRIPT DE LA FORMA DE LOS POI 1

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo A

```

34   this.directionIndicatorDrawable = new AR.ImageDrawable(World.markerDrawable_directionIndicator, 0.5, {
35       enabled: false
36   });
37   this.radarCircle = new AR.Circle(0.03, {
38       horizontalAnchor: AR.CONST.HORIZONTAL_ANCHOR.CENTER,
39       opacity: 0.8,
40       style: {
41           fillColor: "#FFFFFF"
42       }
43   });
44   this.radardrawables = [];
45   this.radardrawables.push(this.radarCircle);
46   // Changed:
47   var markerObject = new AR.GeoObject(markerLocation, {
48       drawables: {
49           cam: [this.markerDrawable_idle, this.markerDrawable_selected, this.titleLabel, this.descriptionLabel],
50           indicator: this.directionIndicatorDrawable,
51           radar: this.radardrawables
52       }
53   });
54
55   return this;
56 }

```

ILUSTRACIÓN 69: JAVASCRIPT DE LA FORMA DE LOS POI 2

**ANEXO B:  
CÓDIGO FUENTE DE CASO  
PRÁCTICO DE  
RECONOCIMIENTO DE  
IMAGEN**



## Anexo B: código fuente de caso práctico de reconocimiento de imagen

A continuación se presentará el código utilizado para construir la aplicación android, donde en primera instancia se observará la lógica del programa para luego seguir con los HTML:

- Clase Tutorial 1:

```

14 public class Tutorial1 extends ARViewActivity {
15     private IGeometry mModel;
16
17     protected int getGUILayout() {
18
19
20
21
22
23     public void onClick(View v) {
24
25
26
27     @Override
28     protected void loadContents() {
29         try {
30             // Getting a file path for tracking configuration XML file
31             String trackingConfigFile = AssetsManager
32                 .getAssetPath("Tutorial1/Assets1/TrackingData_MarkerlessFast.xml");
33
34             // Assigning tracking configuration
35             boolean result = metaioSDK
36                 .setTrackingConfiguration(trackingConfigFile);
37             MetaioDebug.Log("Tracking data loaded: " + result);
38
39             // Getting a file path for a 3D geometry
40             String metaioModel = AssetsManager
41                 .getAssetPath("Tutorial1/Assets1/metaioman.md2");
42             if (metaioModel != null) {
43                 // Loading 3D geometry
44                 mModel = metaioSDK.createGeometry(metaioModel);
45                 if (mModel != null) {
46                     // Set geometry properties
47                     mModel.setScale(new Vector3d(4.0f, 4.0f, 4.0f));
48
49                 } else
50                     MetaioDebug.Log(Log.ERROR, "Error loading geometry: "
51                         + metaioModel);
52             }
53         } catch (Exception e) {
54             e.printStackTrace();
55         }
56     }
57

```

ILUSTRACIÓN.70: CLASE TUTORIAL1, ENCARGADO DE CONFIGURAR LA AR

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo B

## • Clase MainActivity

```
1 package com.metaio.Example;
2
3+ import java.io.IOException;
20
21 @SuppressWarnings("SetJavaScriptEnabled")
22 public class MainActivity extends Activity {
23     WebView mWebView;
24     AssetsExtractor mTask;
25     View mProgress;
26     boolean mLaunchingTutorial;
27
29+     protected void onCreate(Bundle savedInstanceState) {}
46
48+     protected void onResume() {}
53
55+     protected void onPause() {}
59
61+     public void onBackPressed() {}
68
69+     private class AssetsExtractor extends AsyncTask<Integer, Integer, Boolean> {}
109
110+     class WebViewHandler extends WebViewClient {}
172 }
```

ILUSTRACIÓN 71: ESTRUCTURA GENERAL DE LA CLASE MAINACTIVITY

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo B

```

43 @Override
44 protected void onCreate(Bundle savedInstanceState) {
45     super.onCreate(savedInstanceState);
46     setContentView(R.layout.webview);
47     // Habilitar mensajes de registro SDK metaio basado en configuración de
48     // generación
49     MetaioDebug.enableLogging(BuildConfig.DEBUG);
50     mProgress = findViewById(R.id.progress);
51     mWebView = (WebView) findViewById(R.id.webview);
52     // extrae todos los assets
53     mTask = new AssetsExtractor();
54     mTask.execute(0);
55 }
56 @Override
57 protected void onResume() {
58     super.onResume();
59     mWebView.resumeTimers();
60     mLaunchingTutorial = false;
61 }
62 @Override
63 protected void onPause() {
64     super.onPause();
65     mWebView.pauseTimers();
66 }
67 @Override
68 public void onBackPressed() {
69     // if web view can go back, go back
70     if (mWebView.canGoBack())
71         mWebView.goBack();
72     else
73         super.onBackPressed();
74 }

```

ILUSTRACIÓN 72: MÉTODOS DEL CICLO DE VIDA DE LA APLICACIÓN

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo B

```

84-  /**
85   * Esta tarea extrae todos los assets a una ubicación externa o interna a
86   * Hacerlos accesibles a metaio SDK
87   */
88-  private class AssetsExtractor extends AsyncTask<Integer, Integer, Boolean> {
89-      @Override
90      protected void onPreExecute() {
91          mProgress.setVisibility(View.VISIBLE);
92      }
93
94-      @Override
95      protected Boolean doInBackground(Integer... params) {
96          try {
97              AssetsManager.extractAllAssets(getApplicationContext(), true);
98          } catch (IOException e) {
99              MetaioDebug.printStackTrace(Log.ERROR, e);
100             return false;
101         }
102
103         return true;
104     }
105-     @Override
106     protected void onPostExecute(Boolean result) {
107         mProgress.setVisibility(View.GONE);
108
109         if (result) {
110             WebSettings settings = mWebView.getSettings();
111             settings.setCacheMode(WebSettings.LOAD_NO_CACHE);
112             settings.setJavaScriptEnabled(true);
113             mWebView.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
114             mWebView.setWebViewClient(new WebViewHandler());
115             mWebView.loadUrl("file:///android_asset/Menu/tutorial1.html");
116             mWebView.setVisibility(View.VISIBLE);
117         } else {
118             MetaioDebug.Log(Log.ERROR,
119                 "Error extracting assets, closing the application...");
120             finish();
121         }
122     }
123 }

```

ILUSTRACIÓN 73: MÉTODO QUE EXTRAER LOS RECURSOS UTILIZADOS POR LA APLICACIÓN

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo B

• HTML:

```

2  <html>
3  <head>
4      <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
5      <title>Prototipo</title>
6      <link rel="stylesheet" type="text/css" href="style.css"/>
7  </head>
8  <body>
9      <div class="nav-header">
10         <a href="index.html" class="button back-button" onclick="history.go(-1);return false;">
11             
12             Back
13         </a>
14         <h1>Buscando Logo UBB</h1>
15         <div style="clear: both"></div>
16     </div>
17
18     <div class="content">
19         <div class="tutorial-heading">
20             <div></div>
21             <h1>Logo UBB</h1>
22         </div>
23
24         <div class="start-buttons">
25             <a href="metaioSDKExample:///startTutorialID=1" class="button">
26                 A Buscar !
27             </a>
28         </div>
29     </div>
30 </body>
31 </html>

```

ILUSTRACIÓN 74: HTML DE LA APLICACIÓN

• XML de configuración de seguimiento:

```

1  <?xml version="1.0"?>
2  <!-- Configuración de seguimiento de muestras para el uso de la "rápida" rastreador sin marcadores.
3  El seguimiento que hace la "localización" y luego "seguimiento entre tramas".
4  Este tipo de sensor de seguimiento de los objetos planos (imágenes impresas, carteles,
5  etc), haciendo coincidir sus intensidades. La variante "rápida" tiene la
6  siguientes propiedades:
7  - Funciona a una tasa de fotogramas más altas.
8  - Es capaz de realizar un seguimiento de múltiples objetos planos simultáneamente.
9  - Puede ser sensible a las oclusiones, specularities y los cambios de luz.
10 A menos que se indique lo contrario, todos los valores utilizados en esta configuración son
11 los valores por defecto.
12 -->
13
14 <TrackingData>
15     <Sensors>
16         <!-- Utilice la variante "rápida" de la "FeatureBasedSensorSource".
17             la características de este rastreador se explican anteriormente. -->
18         <Sensor Type="FeatureBasedSensorSource" Subtype="Fast">
19             <!-- Asigne una ID para este sensor -->
20             <SensorID>FeatureTracking1</SensorID>

```

ILUSTRACIÓN 75: CONFIGURACIÓN XML DE SEGUIMIENTO

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo B

```

22     <!-- Parámetros que se aplican al sensor -->
23     <Parameters>
24
25         <!-- Los siguientes tipos de descriptores de función están disponibles:
26             " regular", " vertical ", " gravedad" , " rectificar " .
27             - La función de tipo de descriptor "regular" es la más
28             característica general tipo de descriptor y se utiliza como
29             por defecto si no se especifica la etiqueta .
30             - La " vertical " tipo de descriptor función asume que
31             la cámara no se gira con respecto a la óptica
32             eje , es decir, se pone al revés , durante el seguimiento
33             proceso .
34             - La "gravedad " type descriptor función sólo puede ser
35             utilizado con dispositivos con sensores inerciales que
36             medidas de la gravedad. Se utiliza para la localización de estática
37             objetos que proporcionan ( cerca de ) las superficies verticales ,
38             por ejemplo edificios o carteles en una pared . La orientación
39             de las características entonces será alineado con la gravedad.
40             - El " rectificado " type descriptor función sólo puede ser
41             utilizado con dispositivos con sensores inerciales que
42             medidas de la gravedad. Se utiliza para objetos planos en un
43             superficie horizontal , por ejemplo, una revista sobre una mesa.
44             Esto mejorará el resultado de la localización de los
45             planar objetos bajo los ángulos de cámara empinadas a costa
46             de una tasa de fotogramas más baja durante la localización .
47             Este parámetro es sólo para uso experto. En general, se
48             se recomienda dejar el valor sin cambios. -->
49     <FeatureDescriptorAlignment>regular</FeatureDescriptorAlignment>

```

ILUSTRACIÓN 76: PARÁMETRO DEL SENSOR: ALINEACIÓN

```

50     <!-- Una restricción en el número de objetos planos de referencia
51         ser localizados por cuadro. Localización tarda más
52         de seguimiento de intertrama, y si el sistema intenta
53         localizar demasiados objetos al mismo tiempo, podría
54         provocar un framerate inferior. El valor por defecto es 5
55         y se utiliza si no se especifica la etiqueta.
56         Otro nombre que puede ser utilizado para este parámetro es
57         <MultipleReferenceImagesFast>. Este nombre es sin embargo
58         obsoleto y no debe utilizarse más.
59         Este parámetro es sólo para uso experto. En general, se
60         se recomienda dejar el valor sin cambios. -->
61     <MaxObjectsToDetectPerFrame>5</MaxObjectsToDetectPerFrame>

```

ILUSTRACIÓN 77: PARÁMETRO DEL SENSOR: NÚMERO DE OBJETO DETECTADO POR CUADROS

```

63     <!-- El número máximo de objetos que deben ser seguidos en
64         paralelas. El seguimiento de muchos objetos en paralelo es muy
65         caro y podría conducir a una velocidad de fotogramas más baja. Tan pronto
66         como se alcanza el número máximo de objetos rastreados,
67         el sistema ya no tratar de localizar nuevos objetos.
68         El valor por defecto es 1 y se utiliza si la etiqueta
69         no se especifica.
70         Otro nombre que puede ser utilizado para este parámetro es
71         <MaxNumCosesForInit>. Sin embargo, este nombre está en desuso
72         y no debe ser utilizado más.
73         Este parámetro es sólo para uso experto. En general, se
74         se recomienda dejar el valor sin cambios. -->
75     <MaxObjectsToTrackInParallel>1</MaxObjectsToTrackInParallel>

```

ILUSTRACIÓN 78: PARÁMETRO DEL SENSOR: MAX DE OBJETOS A SEGUIR

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo B

```

77         <!-- Similitud umbral predeterminado para especificar si
78             seguimiento de la plantilla se ha realizado correctamente o no. la
79             medida de la calidad de rastreo se define entre -1 y 1,
80             donde 1 es el mejor valor posible. Si el seguimiento
81             calidad es informado de que por debajo del umbral, la
82             persecuidor tratará la trama correspondiente como perdida.
83             El valor por defecto es 0.7 y se utiliza si el
84             no se especifica la etiqueta. Esta configuración se puede anular
85             por cada "COS" si se define allí.
86             Este parámetro es sólo para uso experto. En general, se
87             se recomienda dejar el valor sin cambios. -->
88         <SimilarityThreshold>0.7</SimilarityThreshold>
89     </Parameters>

```

ILUSTRACIÓN 79: PARÁMETROS DEL SENSOR: UMBRAL

```

91     <!-- Definir un "SensorCOS" para este sensor. Esto es esencialmente una
92         sistema asociado una imagen de plantilla que está con coordenadas
93         para ser rastreados. -->
94     <SensorCOS>
95
96         <!-- Un ID de que esta clase de servicio se asocia con. -->
97         <SensorCosID>Patch1</SensorCosID>
98
99         <!-- Los parámetros que se deben aplicar a esta "SensorCOS".
100             -->
101         <Parameters>
102
103             <!-- Nombre del archivo de imagen de referencia. Si está disponible, el ancho y
104                 altura de la plantilla de rastreo en milímetros puede
105                 especificar por la adición de atributos "WidthMM" y / o
106                 "HeightMM". Si no se especifican estos valores en
107                 todo, la anchura y la altura de la imagen de referencia en
108                 se utilizarán píxeles. Si tan sólo la anchura o altura del piso son
109                 especificado, se calculará la cantidad faltante
110                 de tal manera que los tamaños de acuerdo con la relación de aspecto de la
111                 imagen.
112                 Estos parámetros no son necesarios y tienen por objeto
113                 experto utilizar solamente. -->
114             <ReferenceImage>target.png</ReferenceImage>
115             <!-- <ReferenceImage WidthMM="100" HeightMM="100">metaioman.png</ReferenceImage> -->
116
117             <!-- Otra similitud umbral se puede especificar aquí.
118                 Se anulará el umbral de similitud por defecto
119                 especificado para el sensor.
120                 Este parámetro es sólo para uso experto. en general
121                 se recomienda dejar el valor sin cambios. -->
122             <SimilarityThreshold>0.7</SimilarityThreshold>
123
124         </Parameters>
125     </SensorCOS>
126 </Sensor>
127 </Sensors>

```

ILUSTRACIÓN 80: DEFINICIÓN DE UNA "SENSORCOS"



Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo B

```

142 <!-- Las conexiones entre las entidades SensorCOS y COS se definen aquí.
143 Mientras que el SensorCOS representa la pose del objeto seguido
144 con respecto al sensor, las COS es la pose que se utilizará cuando
145 aumentar objetos. El COS se calcula a partir de los SensorCOS por
146 realizar los pasos de procesamiento adicionales:
147 - Un fusor se puede utilizar para suavizar el movimiento, y también para predecir el movimiento
148 en el caso de lecturas de los sensores que faltan.
149 - Una transformación rígida se puede aplicar. El modelo para ser ampliado
150 puede ser desplazado y girado contra un SensorCOS.
151 - Una calibración mano-ojo se puede aplicar.
152 -->
153 <Connections>
154 <COS>
155
156 <!-- Un nombre descriptivo para esta clase de servicio -->
157 <Name>MarkerlessCOS1</Name>

```

ILUSTRACIÓN 81: DEFINICIÓN DE CONEXION ENTRE ENTIDADES

```

159 <!-- ¿Qué tipo de fusor de usar. Aquí, utilizamos el
160 "SmoothingFuser", que se aplica He suavizado con el fin de predecir
161 el movimiento y el ruido mango.-->
162 <Fuser Type="SmoothingFuser">

```

ILUSTRACIÓN 82: DEFINICIÓN DEL TIPO DE FUSOR

```

163 <Parameters>
164
165 <!-- Número de fotogramas en los que seguirá el tracker
166 la predicción de la pose cuando el seguimiento entre tramas
167 falla. Después de que el número especificado de tramas, la
168 rastreador se detiene la predicción.
169 Este parámetro es sólo para uso experto. en general
170 se recomienda dejar el valor sin cambios. -->
171 <KeepPoseForNumberOfFrames>2</KeepPoseForNumberOfFrames>

```

ILUSTRACIÓN 83: PARÁMETRO DEL FUSOR: NÚMERO DE FOTOGRAMAS A SEGUIR

```

182 <!-- Datos (posición) Factor de suavizado de doble
183 suavizado exponencial de la traducción. Este valor
184 debe ser alto si se espera que las medidas que
185 precisa y bajo otra forma. Un valor alto asigna un
186 peso superior a una nueva medición. Típicamente, la
187 precisión de las estimaciones de traducción es bastante
188 alta, por lo que estableció un factor de suavización a 0,8. la
189 valor predeterminado es 0,5.
190 Este parámetro es sólo para uso experto. en general
191 se recomienda dejar el valor sin cambios. -->
192 <AlphaTranslation>0.8</AlphaTranslation>

```

ILUSTRACIÓN 84: PARÁMETRO DEL FUSOR: TRASLACIÓN ALPHA

```

194 <!-- Tendencia (velocidad) Factor de suavizado de doble
195 suavizado exponencial de la traducción. Este valor
196 debe ser alto si se espera que las medidas que
197 precisa y bajo otra forma. Con la misma
198 razonamiento que el anterior, se establece el factor de suavizado para
199 0.8. El valor predeterminado es 0,5.
200 Este parámetro es sólo para uso experto. en general
201 se recomienda dejar el valor sin cambios. -->
202 <GammaTranslation>0.8</GammaTranslation>

```

ILUSTRACIÓN 85: PARÁMETRO DEL FUSOR: TRASLACIÓN GAMMA



Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo B

```

204         <!-- Datos (posición) Factor de suavizado de doble
205             suavizado exponencial de rotación. rotación
206             mediciones por lo general no son tan precisos como
207             mediciones de traducción, por lo que utilizan un valor de 0,5.
208             Este parámetro es sólo para uso experto. en general
209             se recomienda dejar el valor sin cambios -->
210         <AlphaRotation>0.5</AlphaRotation>
    
```

ILUSTRACIÓN 86: PARÁMETRO DEL FUSOR: ROTACIÓN ALPHA

```

212         <!-- Tendencia (velocidad) Factor de suavizado de doble
213             suavizado exponencial de rotación. Con la misma
214             razonamiento que para AlphaRotation anterior, se establece este
215             valor a 0,5.
216             Este parámetro es sólo para uso experto. en general
217             se recomienda dejar el valor sin cambios. -->
218         <GammaRotation>0.5</GammaRotation>
    
```

ILUSTRACIÓN 87: PARÁMETRO DEL FUSOR: ROTACIÓN GAMMA

```

220         <!-- Si un sensor de orientación está disponible, el sistema
221             pueden tratar de mantener la actualización de la postura sobre la base de que
222             mediciones del sensor de orientación. Si esto
223             por hacer, esta opción se debe establecer en true. la
224             valor predeterminado es false.
225             Este parámetro es sólo para uso experto. en general
226             se recomienda dejar el valor sin cambios. -->
227         <ContinueLostTrackingWithOrientationSensor>false</ContinueLostTrackingWithOrientationSensor>
228     </Parameters>
229 </Fuser>
    
```

ILUSTRACIÓN 88: PARÁMETRO DEL FUSOR: SI LA BUSQUEDA SE PIERDE CON EL CAMBIO DE ORENTACIÓN

```

231     <SensorSource>
232         <!-- El sensor y SensorCOS que queremos utilizar. nota
233             que estas identificaciones son los mismos que hemos utilizado en el
234             Definición Sensor arriba. -->
235         <SensorID>FeatureTracking1</SensorID>
236         <SensorCosID>Patch1</SensorCosID>
    
```

ILUSTRACIÓN 89: ESTABLECER CALIBRACIÓN DE LOS SENSORES

Estudio de Herramientas de Realidad Aumentada Sobre  
Plataforma Android – Anexo B

```

238      <!-- Una calibración mano-ojo permite especificar la relación
239      plantean entre dos sensores. En el caso simple de tener
240      un sensor basado en cámaras, por lo general no se utiliza. lo
241      permite mover los COS "como si" la cámara se movió,
242      y por lo tanto es inversa a la COSOffset rígido
243      transformación que se especifica a continuación. -->
244      <HandEyeCalibration>
245      <!-- El vector de traslación 3D. -->
246      <TranslationOffset>
247      <X>0</X>
248      <Y>0</Y>
249      <Z>0</Z>
250      </TranslationOffset>
251      <!-- Las rotaciones se especifican mediante cuaterniones unitarios, donde
252      se especifica la "Y", "Z" partes imaginarias "X"
253      primero, y luego la parte real "W". -->
254      <RotationOffset>
255      <X>0</X>
256      <Y>0</Y>
257      <Z>0</Z>
258      <W>1</W>
259      </RotationOffset>
260      </HandEyeCalibration>

```

ILUSTRACIÓN 90: CALIBRACIÓN DE LOS SENSORES DE TIPO "MANO-OJO"

```

262      <!-- El COSOffset especifica una transformación rígida que
263      se aplica a los SensorCOS. Esto hace que sea posible
264      mover el modelo aumentado. Se especifica la misma
265      manera que el ojo-mano de calibración. -->
266      <COSOffset>
267      <TranslationOffset>
268      <X>0</X>
269      <Y>0</Y>
270      <Z>0</Z>
271      </TranslationOffset>
272      <RotationOffset>
273      <X>0</X>
274      <Y>0</Y>
275      <Z>0</Z>
276      <W>1</W>
277      </RotationOffset>
278      </COSOffset>
279      </SensorSource>
280      </COS>
281      </Connections>
282      </TrackingData>

```

ILUSTRACIÓN 91: CALIBRACIÓN DE LOS SENSORES DE TIPO "RÍGIDA"