

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ciencias Computacionales

TESIS DE MAESTRÍA EN CIENCIAS

**Metodología para Crear Patrones Multivista para Aplicaciones de
Realidad Aumentada**

presentada por
Ing. Irving Ulises Herrera Molina

como requisito para la obtención del grado de
Maestro en Ciencias de la Computación

Director de tesis
Dr. Juan Gabriel González Serna

Codirector de tesis
Dr. Raúl Pinto Elías

Cuernavaca, Morelos a 30 de septiembre del 2016
OFICIO No. DCC/197/2016

Asunto: Aceptación de documento de tesis

C. DR. GERARDO V. GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **Ing. Irving Ulises Herrera Molina**, con número de control M14CE060, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado "**Metodología para crear patrones multivista para aplicaciones de realidad aumentada**" y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS



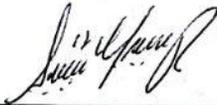
Dr. Juan Gabriel González Serna
Doctor en Ciencias de la
Computación
7820329

CO-DIRECTOR DE TESIS



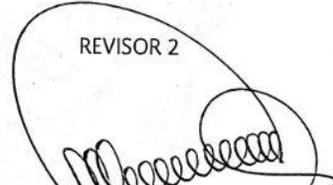
Dr. Raúl Rincón Elías
Doctor en Ciencias en la Especialidad
de Ingeniería Eléctrica
3890453

REVISOR 1



Dra. Azucena Montes Rendón
Doctora en Ciencias
4001014

REVISOR 2



Dr. Máximo López Sánchez
Doctor en Ciencias de la
Computación
7498547

C.p. M.T.I. María Elena Gómez Torres - Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

NACS/lmz



Cuernavaca, Mor., 13 de octubre de 2016
OFICIO No. SAC/301/2016

Asunto: Autorización de impresión de tesis

**ING. IRVING ULISES HERRERA MOLINA
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
DE LA COMPUTACIÓN
PRESENTE**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "**Metodología para crear patrones multivista para aplicaciones de realidad aumentada**", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE
"CONOCIMIENTO Y TECNOLOGÍA AL SERVICIO DE MÉXICO"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. M.T.I. María Elena Gómez Torres.- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr



Dedicatoria

A mi madre
Sra. Nery Molina Estrada como una mínima retribución por
todo lo que ha dado por mí y por formarme como la persona
que soy.

A mi novia
Vicky por estar en esta etapa en mi vida, por su cariño, amor y
apoyo.

Agradecimientos

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por haberme permitido pertenecer a su comunidad estudiantil y realizar así mis estudios de maestría.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico que me permitió el desarrollo de esta tesis.

A mi director de tesis, Dr. Juan Gabriel González Serna, por haberme formado durante todo el tiempo del posgrado, ayudándome a desarrollarme como persona y como profesional, así como a mi codirector Dr. Raúl Pinto Elías por sus observaciones y consejos.

A mis revisores la Dra. Azucena Montes Rendón y el Dr. Máximo López Sánchez gracias por sus observaciones e interés por hacer de este el mejor trabajo posible.

A los profesores que conocí como alumno de CENIDET, por sus enseñanzas y, sobre todo, por su consideración de la ética en la investigación y de la responsabilidad social. Por los días en los que no pude evitar pensar "hoy he aprendido algo".

A mis padres quienes me estuvieron brindando con su apoyo moral, su cariño y sus sabios consejos. Por hacer de mí una mejor persona.

A mi madre que siempre estuvo al pendiente a pesar de la distancia, siempre estuvo presente en mi mente.

A mi novia Vicky que fue lo mejor que me dejó esta etapa de la maestría, por su apoyo, consejos, por todos los momentos buenos y malos que pasamos juntos, por ser una parte fundamental en este trabajo de tesis.

A mis profesores en general, por las regañadas que me llegaron a dar, que a pesar de todo esto me brindaron su confianza y enseñanza profesional.

A todos aquellos que me brindaron su amistad en este tiempo.

Resumen

Actualmente la realidad aumentada es uno de los tópicos de investigación que está cobrando mayor importancia en diversos campos de investigación debido a su versatilidad, esta tecnología se está introduciendo en nuevas áreas de aplicación como son el entrenamiento en procesos industriales, guías de museos, en el ámbito educativo, entre otras muchas áreas de aplicaciones. La capacidad de mezclar objetos virtuales en un ambiente real, así como el desarrollo de interfaces que permiten hacer esto de una manera más sencilla son las características que la convierten en una tecnología de gran utilidad.

En este trabajo de investigación se presenta una metodología desarrollada para la caracterización de objetos multi-vista, para cada vista se desagregará cada uno de sus componentes individuales, generando los patrones necesarios para identificarlos, posteriormente a los componentes individuales desagregados se le asociará información de ayuda que será representada mediante realidad aumentada.

Se desarrolló una aplicación móvil de información aumentada con la cual se pudo comprobar que, los objetos caracterizados con la metodología propuesta, se pueden identificar los componentes que conforman cada vista del objeto analizado, y representar mediante realidad aumentada la información asociada a estos componentes.

Finalmente, las contribuciones de este trabajo de investigación a las ciencias computacionales, se enfocan en las áreas de interacción Humano-Computadora, realidad aumentada, sistemas de información contextual y sistemas de información aumentada.

Abstract

Currently the augmented reality is one of the research areas is becoming increasingly important in many areas of research because of its versatility, this technology is being introduced into new application areas such as training in industrial processes, museum guides, in the educational level, among many other application areas. The ability to mix virtual objects in a real environment, and the development of interfaces that allow do this in a more simple way are the characteristics that make it a very useful technology.

In this research was developed a methodology for the characterization of multi-view objects, for each view we need disaggregate each one of individual components, generating the patterns necessary to identify them, then to the individual components separated will associate it help information that will be represented by augmented reality

Was developed a mobile application of augmented information that it can be check that the characterized objects with the proposed methodology can be identified individually the components that compose each view the scanned object, and represent through augmented reality the information associated with these components.

Finally, the contribution of this research to computer science, is focused on the areas of Human-Computer Interaction, Augmented Reality, Contextual Information Systems and Augmented Information Systems.

Índice

Lista de figuras.....	iv
Lista de tablas.....	vii
1. Introducción	1
1.1 Introducción	2
1.2 Antecedentes	3
1.2.1 Servicios de recomendación contextual para instituciones de educación superior mediante realidad aumentada utilizando Smartphones y ontologías organizacionales	3
1.2.2 Sistema de información aumentada para navegación indoor-outdoor basado en modelos semánticos organizacionales	4
1.2.3 Framework de realidad aumentada para la extracción y clasificación de información visual de contexto	4
1.3 Descripción del problema.....	5
1.4 Objetivos del proyecto	6
1.4.1 Objetivos específicos.....	6
1.5 Justificación.....	6
2. Marco Teórico.....	10
2.1 Sistemas de recomendación conscientes del contexto.....	9
2.2 Visión artificial.....	9
2.3 Realidad aumentada	10
2.3.1 Gestión de realidad aumentada en computadoras convencionales.....	10
2.3.2 Gestión de realidad aumentada en equipo portátil	10
2.3.3 Gestión de realidad aumentada con equipos específicos.....	11
2.4 Reconocimiento de patrones	12
3 Estado del Arte	16
3.1 EMOVIS - An efficient mobile visual search system for landmark recognition (Dawei & Mooi Choo , 2013).....	15
3.1.1 Arquitectura.....	16
3.1.2 Pre procesamiento de la imagen.....	16
3.1.3 Punto de interés en la imagen.....	16
3.2 Edge-SIFT: Discriminative Binary Descriptor for Scalable Partial-Duplicate Mobile Search (Shiliang Zhang, 2013).....	18

3.2.1	Extracción EDGE-SIFT	18
3.2.2	Detección de puntos de interés	19
3.2.3	Extracción y normalización de la imagen	19
3.3	MemoryEfficient image databases for mobile visual search (Girod & Chen, 2014)	20
3.3.1	Esquema de funcionamiento	21
3.3.2	Características de una base de datos compacta	22
3.4	Interactive Multimodal Visual Search on Mobile Device (Houqiang , Yang , Tao , Jingdong , & Shipeng, 2013)	24
3.4.1	Funcionamiento del sistema	24
3.4.2	Entidad de discurso	25
3.4.3	Formulación de consultas visuales	25
3.4.4	Indexación y recuperación.....	26
3.5	Car Recognition from frontal images in mobile environment (Tanacs & Varjas, 2013)	26
3.5.1	Extracción de características desde la región de interés (ROI)	27
3.5.2	Reconocimiento semiautomático y totalmente automático	27
3.6	Tecnologías wareables	29
3.6.1	Smart Helmet.....	29
3.6.2	HoloLens	30
3.6.3	Atheer AiR Platform	30
3.6.4	(ODG) Elegante Gafas de Osterhout Design Group	31
3.6.5	Moverio BT-200.....	32
4	Metodología de Solución	42
4.1	Metodología	37
4.2	Fase 1. Extracción de patrones.....	38
4.3	Fase 2. Prototipo de información aumentada	40
4.3.2	Proceso de renderizado.....	41
5	Diseño e Implementación.....	43
5.1	Diseño e implementación de la fase 1	44
5.1.1	Creación de la base de datos.....	48
5.2	Diseño e implementación de la fase 2	66
5.2.1	SDK de Qualcomm Vuforia.....	66
5.2.2	Unity.....	66

5.2.3	Yii framework	68
5.2.4	SQLite	70
6	Pruebas y Resultados.....	76
6.1	Pruebas Heurísticas	77
6.1.1	Técnica de evaluación Heurística.....	77
6.2	Plan de pruebas de la metodología	79
6.2.1	Objetivo.....	79
6.2.2	Alcance de las pruebas	79
6.2.3	Limitaciones de las pruebas	79
6.2.	Plan de pruebas de la aplicación Android	106
6.2.1.	Plan de pruebas.....	106
6.2.2.	Enfoque	106
6.2.3.	Características a probar	106
6.2.4.	Características excluidas	106
6.2.5.	Elementos de prueba	107
6.2.6.	Requerimientos para realizar las pruebas	107
6.2.7.	Criterio éxito/fracaso de los casos de prueba	107
6.2.8.	Responsabilidades	108
6.2.9.	Reporte de pruebas	108
6.3.	Casos de prueba SmartTV	108
6.4.	Casos de prueba multifuncional HP	111
6.5.	Casos de prueba Bocina	114
7.	Conclusiones y Trabajos futuros	118
7.1	Conclusiones	119
7.2	Trabajos futuros	120
	Referencias.....	121
A.	Anexos.....	125

Lista de figuras

Figura 1.1 Interfaz de usuario interactiva con realidad aumentada (Leal, 2012).	3
Figura 1.2 Sistema de navegación indoor-outdoor por Fuente especificada no válida.	4
Figura 1.3 Prototipo del trabajo de investigación de (Torre, 2015).	4
Figura 2.1 Similitud de los dispositivos móviles (Javier Fombona Cadavieco, 2012).	11
Figura 2.2 Pasos fundamentales en el procesamiento digital de imágenes (Gonzalez Rafael C., 2001).	12
Figura 3.1 Arquitectura del software EMOVIS (Dawei & Mooi Choo , 2013).	15
Figura 3.2 Algoritmo MaxMin (Dawei & Mooi Choo , 2013).	17
Figura 3.3 Algoritmo MaxUnion (Dawei & Mooi Choo , 2013).	17
Figura 3.4 Framework propuesto de extracción Edge-SIFT (Shiliang Zhang, 2013).	18
Figura 3.5 (a) Extracción de una región de la imagen. (b) ejemplo de normalización orientación y normalización de la escala (Shiliang Zhang, 2013).	19
Figura 3.6 Líneas rojas: emparejados 1024 bits descriptores iniciales Edge-SIFT. Las líneas azules: acertaron SIFT descriptores entre imágenes visualmente similares (Shiliang Zhang, 2013).	20
Figura 3.7 El sistema reconoce al aire libre (Girod & Chen, 2014).	21
Figura 3.8 Esquema de funcionamiento del sistema (Girod & Chen, 2014).	21
Figura 3.9 Comparación de THC, ICC, REVV, y scFv en la base de datos MPEG con características SIFT en términos de (a) la precisión en el rango 1, (b) significa precisión media, y (c) RAM de uso de una base de datos de 10,000 imágenes del conjunto de datos (Girod & Chen, 2014).	23
Figura 3.10 Aplicación Android en Smartphone. (a) Distribución de reconocimiento latencias de 400 consultas diferentes. (b) El porcentaje de tiempo dedicado a la función de extracción, la búsqueda de base de datos, y la verificación geométrica (Girod & Chen, 2014).	23
Figura 3.11 La arquitectura del sistema propuesto de búsqueda visual multimodal. Hay cinco componentes principales: (1) El reconocimiento de voz, (2) La extracción de entidades, (3) La generación de imágenes a modo de ejemplo, (4) La composición visual de consultas de ejemplar interactivo, y (5) Basada en ejemplos de imágenes búsqueda consciente del contexto (Girod & Chen, 2014).	24
Figura 3.12 Un ejemplo de procesamiento de una consulta visual con múltiples ejemplares (Girod & Chen, 2014).	25
Figura 3.13 Frontal original (izquierda una parte de la matrícula es oscurecida por el anonimato), la compensación de la rotación, número de placa de identificación y la extracción de retorno de la inversión (en el centro), y el retorno de la inversión extraído (derecha) (Tanacs & Varjas, 2013).	27
Figura 3.14 Los pasos de la segmentación semiautomática. Crecimiento de región, orificio de llenado, post-procesamiento morfológico y el contorno de la región detectada superpuesta en la parte superior de la imagen original (Tanacs & Varjas, 2013).	28
Figura 3.15 La detección de la línea que va a través de la placa de la matrícula. Umbral para la magnitud del bode de la imagen y su proyección (izquierda) y la línea (en rojo) para los candidatos de puntos de semilla (derecha) (Tanacs & Varjas, 2013).	28
Figura 3.16 Smart Helmet (casco inteligente) (DAQRI Smart Helmet, 2014)	29

Figura 3.17 Gafas HoloLens (HoloLens, 2015).....	30
Figura 3.18 Gafas Atheer AiR (labs, 2015).....	31
Figura 3.19 Gafas de ODG (ODG, 2015).	32
Figura 3.20 Gafas Moverio BT-200 (EPSON, 2015).....	33
Figura 4.1 Metodología para la detección de objetos tridimensionales.	37
Figura 4.2 Fase de extracción de patrones.	38
Figura 4.3 Vistas de un objeto.....	39
Figura 4.4 Sistema de vistas europeo (Bartolome, 2015).....	39
Figura 4.5 Sistema de vistas americano (Bartolome, 2015).....	39
Figura 4.6 Fase del prototipo de información aumentada.	40
Figura 4.7 Proceso de renderizado en Realidad Aumentada (Parmar, 2015).....	42
Figura 5.1 Segmentación de la vista que contienen elementos de contexto.....	45
Figura 5.2 Calificaciones del SDK Vuforia.	46
Figura 5.3 Características en una imagen.....	47
Figura 5.4 Detalles de una imagen.	48
Figura 5.5 Pantalla del sitio web de Vuforia.	48
Figura 5.6 Iniciar sección en el sitio web de Vuforia.....	49
Figura 5.7 Pantalla donde se crean y visualizan las bases de datos.	49
Figura 5.8 Ventana para especificar el nombre y tipo de base de datos que se desea crear.....	50
Figura 5.9 Ventana para ver y crear unos Target.	50
Figura 5.10 Ventana para especificar propiedades del nuevo Target.....	51
Figura 5.11 Vista frontal de la lavadora.	52
Figura 5.12 Elementos desagregados de la imagen del panel de control.	52
Figura 5.13 Lista de los Target en la base de datos.....	54
Figura 5.14 Patrones identificados por el framework Vuforia.	54
Figura 5.15 Descarga de la base de datos de Vuforia.	55
Figura 5.16 Archivos descargados de Vuforia.	55
Figura 5.17 Aplicación de escritorio para la asociación	56
Figura 5.18 Menú de opciones.	56
Figura 5.19 Información acerca de la aplicación.	57
Figura 5.20 Selección del archivo xml.....	57
Figura 5.21 Nombre del archivo xml cargado.....	58
Figura 5.22 Búsqueda y selección del archivo dat.	58
Figura 5.23 Nombre del archivo dat cargado.....	59
Figura 5.24 Botones para agregar y eliminar la imagen descriptiva.	59
Figura 5.25 Selección de la imagen representativa.	60
Figura 5.26 Vista previa de la imagen representativa.	60
Figura 5.27 Botones para agregar y eliminar las imágenes de información contextual.	61
Figura 5.28 Selección de la imagen de información contextual.....	61
Figura 5.29 Vista previa de las imágenes de información contextual.....	62

Figura 5.30 Botón para la generación de la tabla.	62
Figura 5.31 Tabla generada para realizar la asociación de elementos.	63
Figura 5.32 Lista con los nombres de las imágenes de información contextual.	63
Figura 5.33 Campos de texto.....	64
Figura 5.34 Guardar paquete de asociación.	64
Figura 5.35 Directorio del paquete generado con la aplicación de asociación.....	65
Figura 5.36 Pantalla del servidor FTP.....	65
Figura 5.37 Herramientas utilizadas.....	66
Figura 5.38 Logo Vuforia.....	66
Figura 5.39 a) Botón de descripción, b) Botón información detallada.	67
Figura 5.40 Elemento virtual para la representación de etiquetas.....	67
Figura 5.41 Elemento virtual de información detallada.	67
Figura 5.42 Proceso para agregar paquetes de información a la aplicación.....	68
Figura 5.43 Diagrama entidad relación SQLite.	71
Figura 5.44 Pantalla de inicio de la aplicación móvil.	72
Figura 5.45 Pantalla de ayuda.	72
Figura 5.46 Pantalla de descarga de nuevos paquetes.	73
Figura 5.47 Descarga de un nuevo paquete.....	73
Figura 5.48 Pantalla de configuración del prototipo.	74
Figura 5.49 a) Botón de descripción, b) Botón información detallada.	74
Figura 5.50 Botones virtuales al momento de reconocer elementos.	75
Figura 5.51 Presentación de información contextual.	75
Figura 6.1 Tabulación de las heurísticas encontradas en la metodología.....	78
Figura 6.2 Tabulación de las heurísticas encontradas en la aplicación móvil.....	79
Figura A.1 Estructura del Metaio Mobile SDK.	127
Figura A.2 Arquitectura del SDK Vuforia.....	128
Figura A.3 Arquitectura del SDK Wikitude.....	131
Figura A.4 Arquitectura del SDK D'Fusion.....	132
Figura A.5 Arquitectura ARToolkit SDK.....	133
Figura A.6 Arquitectura de ARMedia SDK.....	135

Lista de tablas

Tabla 3.1 Tabla comparativa del estado del arte.	34
Tabla 5.1 Presentación de Target con su nombre correspondiente.	53
Tabla 6.1 Heurísticas encontradas en la evaluación de la metodología.	77
Tabla 6.2 Heurísticas encontradas en la evaluación de la aplicación web.	78
Tabla 6.3 Caso de prueba pantalla LG.	80
Tabla 6.4 Caso de prueba multifuncional HP.....	87
Tabla 6.5 Caso de prueba bocina Sony.	96
Tabla 6.6 Caso de prueba descarga del paquete SamrtTV.	108
Tabla 6.7 Caso de prueba identificación de elementos SamrtTV.	109
Tabla 6.8 Caso de uso información aumentada SmartTV.	110
Tabla 6.9 Caso de prueba descarga de paquete multifuncional HP.	111
Tabla 6.10 Caso de uso identificación de elementos multifuncional HP.	112
Tabla 6.11 Caso de usos elementos virtuales multifuncional HP.	113
Tabla 6.12 Caso de prueba descargar paquete bocina.	114
Tabla 6.13 Caso de prueba identificación de elementos de la bocina	115
Tabla 6.14 Caso de uso elementos virtuales de la bocina.	116

Capítulo 1

Introducción

En este capítulo se describen los antecedentes del proyecto y el contexto del trabajo de tesis desarrollado, el problema a resolver, el objetivo y los alcances mismos.

1.1 Introducción

La tecnología con el paso del tiempo cada vez es más accesible para toda la sociedad, los teléfonos móviles son uno de los avances tecnológicos que más impacto ha tenido en las últimas décadas debido a la gran gama de características que presentan.

Actualmente los teléfonos inteligentes y tabletas ofrecen gran capacidad de cómputo y almacenamiento, lo cual les permite tener una gran variedad de aplicaciones móviles con una gran funcionalidad (Acuna, Mena, & Torres, 2013). Estos dispositivos cuentan con una gran gama de sensores y de tecnologías como giroscopio, magnetómetro, acelerómetro, cámara, acceso a Internet por medio de WIFI o la red 3G, etc.

A través del desarrollo de los dispositivos multisensoriales se han incorporado dos conceptos importantes: visión artificial y realidad aumentada. La *visión artificial* es un subcampo de la inteligencia artificial, el propósito de la visión artificial es desarrollar algoritmos que permitan a una computadora interpretar una escena o las características de una imagen; en cuanto a la *realidad aumentada* es una tecnología que complementa la percepción e interacción con el mundo real y permite al usuario estar en un entorno real aumentado con información adicional generada por la computadora (X. Basogain, 2007). Los sistemas de información son herramientas potentes que ayudan a reducir la sobrecarga de información a la que se enfrenta en los procesos de búsqueda (Kioskea, 2014), con la evolución de los dispositivos multisensoriales este tipo de sistemas pueden ser utilizados por estos dispositivos.

En este proyecto de tesis se diseñó e implementó un gestor de objetos en 3D mediante repositorios de patrones para su aplicación en sistemas de información aumentada a través de dispositivos multisensoriales, se aplicaron técnicas de visión artificial, realidad aumentada y se utilizaron algoritmos de reconocimiento de patrones desde un repositorio de imágenes que definen el contexto visual capturado por el dispositivo multisensorial.

1.2 Antecedentes

En esta sección se describen los trabajos desarrollados en el laboratorio de Sistemas Distribuidos del Centro nacional de Investigación y Desarrollo Tecnológico (CENIDET) relacionados con este trabajo de tesis.

1.2.1 Servicios de recomendación contextual para instituciones de educación superior mediante realidad aumentada utilizando Smartphones y ontologías organizacionales (Leal, 2012)

En esta tesis se realizó una investigación de las técnicas de realidad aumentada implementadas en dispositivos móviles y su uso para la creación de interfaces de usuario. El objetivo de este trabajo de investigación fue la creación de una interfaz de usuario interactiva que presenta las recomendaciones realizadas al usuario de manera gráfica y dinámica, así como también la capacidad de adaptarse a las características que presente el dispositivo. En la Figura 1.1 se puede observar el entorno gráfico de la aplicación que se realizó en este trabajo de investigación.

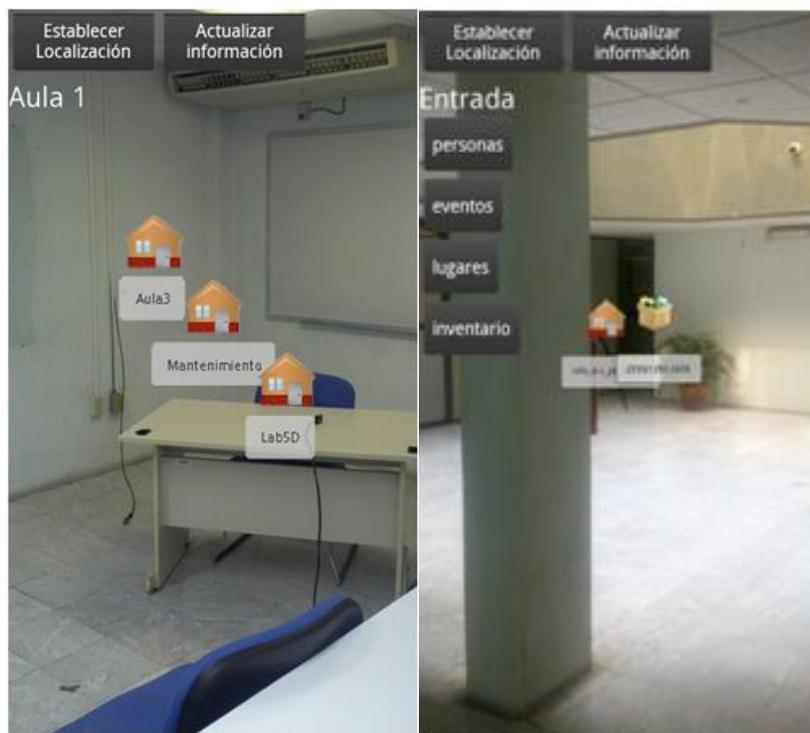


Figura 1.1 Interfaz de usuario interactiva con realidad aumentada (Leal, 2012).

1.2.2 Sistema de información aumentada para navegación indoor-outdoor basado en modelos semánticos organizacionales (Ramírez, 2014)

En este trabajo de investigación se desarrolló un sistema de cómputo consciente del contexto para navegación indoor-outdoor, ver Figura 1.2, basado en modelos semánticos organizacionales sobre dispositivos móviles multisensoriales, este sistema proporciona información de contexto enriquecido para guiar al usuario dentro de instalaciones (in-door) desde un punto de origen hasta un punto destino mostrando información contextual de su entorno teniendo a su alcance recomendaciones de su interés.



Figura 1.2 Sistema de navegación indoor-outdoor por Fuente especificada no válida.

1.2.3 Framework de realidad aumentada para la extracción y clasificación de información visual de contexto (Torre, 2015)

En este trabajo de investigación se desarrolló un framework de realidad aumentada para la implementación de aplicaciones en el sistema operativo Android conscientes del contexto aplicando técnicas de visión artificial y de realidad aumentada utilizando algoritmos para la clasificación de información del contexto visual, el framework tiene la capacidad de realizar reconocimiento de rostros, imágenes, texto y contorno de los objetos, en la Figura 1.3 se muestra el funcionamiento del prototipo del framework.



Figura 1.3 Prototipo del trabajo de investigación de (Torre, 2015).

1.3 Descripción del problema

Un Sistema de Recomendación Consciente del Contexto (SRCC), es un sistema de cómputo capaz de reaccionar de manera autónoma a cambios en el contexto del usuario, utilizando diferentes tecnologías como sensores o sistemas de información, además de dispositivos smartphone para obtener conocimiento de su entorno (Fuentes, 2011).

Tomando en cuenta que los dispositivos móviles cuentan con capacidades multisensorial, una de sus capacidades sensoriales es la cámara, que lo capacita para soportar visión artificial. Gracias a esta capacidad sensorial, se tiene la posibilidad de observar el mundo real a través del dispositivo. Con el uso de esta tecnología se entra a un paradigma denominado realidad aumentada (RA) (Ramos, 2014). La RA es una herramienta eficaz para utilizar y aprovechar el potencial de la información generada por las computadoras (Javier Fombona Cadavieco, 2012).

En el trabajo de investigación (Tanacs & Varjas, 2013) en el estado del arte, se utiliza información visual a través de la cámara del dispositivo, con la imagen de la placa de un automóvil se obtiene información y se presenta al usuario. En los trabajos del estado del arte la comparación de las imágenes se realiza a través de una búsqueda en donde la imagen que se busca es tomada por el dispositivo multisensorial, pero no se realiza un análisis de elementos dentro de la imagen tomada, por lo tanto se pierden detalles de información visual (botones, conectores, etiquetas, códigos de barra) que se puede aprovechar como entradas de información contextual.

Los trabajos relacionados con esta investigación presentan soluciones a problemas muy específicos, los cuales incluyen el procesamiento de imágenes multivista mediante un escaneo activo y posteriormente la generación del patrón de la imagen, la cual, en la mayoría de los casos es procesada en la nube mediante algoritmos propietarios del desarrollador a diferencia de los trabajos antes mencionados, el trabajo propuesto procesa de manera individual cada una de las vistas que conforman el objeto de tal forma que se identifican de manera individual los elementos visuales contenidos en cada una de ellas.

Por lo anterior, el problema que se pretende solucionar en esta tesis está relacionado con el procesamiento de imágenes de objetos multivista, lo cual implica en primera instancia, la identificación individual de cada una de las vistas (escaneo y procesamiento), generación del patrón de la imagen y posteriormente, procesar cada una de las n vistas para subclasificar cada uno de los componentes individuales que se pretende identificar mediante técnicas de visión artificial.

La problemática que se identificó en esta investigación está relacionada con el procesamiento de objetos multivista, para generar un patrón de imagen que incluye múltiples vistas, de acuerdo a la perspectiva desde la cual se enfoque el objeto, una vez identificadas y clasificadas cada una de las vistas, es necesario procesar de manera individual cada una de ellas para identificar, mediante procesos semi-automáticos, los componentes visuales que el usuario requiere identificar de manera individual, hasta el momento este procesamiento de objetos multivista no se realiza a este nivel de detalle, por lo que con este proyecto de investigación se implementaron algoritmos y procesos que proporcionan un primer acercamiento para el reconocimientos de los componentes visuales de objetos multivista.

1.4 Objetivos del proyecto

Desarrollar una metodología para el procesamiento de objetos multivista que permita obtener patrones visuales de cada una de las vistas que lo conforman y la identificación semiautomática de los componentes individuales de cada vista, a cada vista y componente de la misma se le asociarán descriptores de texto, los cuales se mostrarán en la interfaz del prototipo mediante técnicas de realidad aumentada utilizando un dispositivo móvil multisensorial.

1.4.1 Objetivos específicos

- Diseñar y evaluar el modelo de detección de objetos.
- Obtener patrones de las vistas del objeto.
- Detectar objetos en las vistas.
- Crear el repositorio de patrones del objeto.
- Crear el repositorio de información contextual.
- Implementación de realidad aumentada para la visualización de información contextual.

1.5 Justificación

La información proporciona diversos elementos visuales que pueden ser aprovechados en un sistema sensible al contexto, proporcionando al usuario información adicional que puede ser de utilidad, los sistemas de recomendación en su mayoría omiten algunos elementos visuales los cuales se pueden aprovechar para proveer al usuario de mayor información visual del contexto.

En los antecedentes mencionados se puede observar que (Torre, 2015) está trabajando en un sistema para la detección de imágenes, objetos, rostros y texto, para que a través de estos

elementos se proporcione información contextual utilizando tecnologías de visión artificial y de realidad aumentada.

En el estado del arte se detectaron sistemas de reconocimiento de imágenes que proporcionan información, en el caso del trabajo de (Tanacs & Varjas, 2013) se reconoce la marca y modelo de un automóvil a partir de la imagen frontal, esto con la ayuda de algoritmos de visión artificial, y sistemas de información contextual que realizan consultas de imágenes en la nube (Dawei & Mooi Choo , 2013) y localmente desde un dispositivo multisensorial (Girod & Chen, 2014) para proporcionar información de contexto al usuario.

Capítulo 2

Marco Teórico

Este proyecto de investigación involucra diversas áreas de investigación, como son: sistemas de cómputo conscientes del contexto, visión artificial, realidad aumentada y reconocimiento de patrones. En esta sección se detallan temas y conceptos específicos relacionados con áreas y tecnologías importantes en el desarrollo de este trabajo de tesis.

2.1 Sistemas de recomendación conscientes del contexto

Los sistemas de recomendación en (Abbar, Bouzeghoub, & Lopez, 2009) son herramientas eficientes que superan el problema de la sobrecarga de información, proporcionando a los usuarios el contenido más relevante. Esto se hace generalmente a través de información de preferencias/calificaciones de los usuarios, adquirida de los historiales de sus respectivas sesiones. Además de estas preferencias, también se toma en cuenta el contexto de la interacción del usuario, lo cual mejora la relevancia del proceso de recomendación.

Emplea técnicas que ayudan al usuario a seleccionar elementos de una gran cantidad de opciones. Para hacer esto, el perfil de usuario es contrastado con las características de los elementos.

Un sistema de recomendación: toma decisiones basado en información real, requiere de un método de filtrado de información, se aplica en algún ámbito determinado (Ej: películas, música, restaurantes, email).

2.2 Visión artificial

La visión es el más avanzado de nuestros sentidos, por lo que no es sorprendente que las imágenes jueguen el papel más importante en la percepción humana. Sin embargo, a diferencia de los seres humanos, que están limitados a la banda visual del espectro electromagnético (EM), las máquinas empleadas para la formación de imágenes cubren casi todo el espectro EM, que va desde los rayos gamma hasta las ondas de radio. Pueden operar en imágenes generadas por fuentes que los seres humanos no están acostumbrados a asociar con imágenes. Estos incluyen el ultrasonido, microscopía electrónica e imágenes (Gonzalez Rafael C., 2001).

Hay campos como la visión por computadora cuyo objetivo final es utilizar las computadoras para emular la visión humana, incluyendo el aprendizaje y ser capaz de hacer inferencias y tomar acciones basadas en entradas visuales. Esta zona en sí es una rama de la inteligencia artificial (IA), cuyo objetivo es emular la inteligencia humana. El área de análisis de imágenes (también

llamada la comprensión de la imagen) está entre el procesamiento de imágenes y visión por computador (Gonzalez Rafael C., 2001).

2.3 Realidad aumentada

La realidad aumentada (RA) es una variación de los entornos virtuales (EV) o realidad virtual como es más comúnmente llamado. Los EV sumergen completamente al usuario dentro de un entorno simulado. Mientras el usuario se encuentra inmerso no puede ver el mundo real que le rodea. Por el contrario, la RA permite al usuario ver el mundo real, con objetos virtuales superpuestos o compuestos con el mundo real, la RA complementa la realidad, en lugar de sustituirla por completo (Azuma, 1997).

Hay tres formas de presentar la tecnología de realidad aumentada, con la computadora convencional, con dispositivos portátiles miniaturizados y con equipos específicos de realidad aumentada (Javier Fombona Cadavieco, Realidad aumentada, una evolución de las aplicaciones de los dispositivos móviles - Artículo ISSN: 1133-8482 Píxel-Bit. Revista de Medios y Educación, 2012).

2.3.1 Gestión de realidad aumentada en computadoras convencionales

La cámara conectada al equipo de cómputo digitaliza la imagen captada, a la que se le incorporan capas con otros datos, imágenes fijas o en movimiento, textos y/o sonidos. Esta información se añade a partir de un software que selecciona documentos específicos de una base de datos propia o del acceso a los datos de Internet. Una pantalla presenta la síntesis, muestra de forma sincronizada tanto las imágenes reales captadas, como el resto de datos superpuestos sincronizados en tamaño, posición y en tiempo real. Se pretende que los elementos virtuales estén coordinados con precisión con los objetos reales y su posición, puesto que un pequeño error de orientación provoca un desajuste perceptible.

2.3.2 Gestión de realidad aumentada en equipo portátil

Los dispositivos móviles avanzados son pequeñas mini computadoras que incorporan cámaras de captura de imágenes. Con ello se puede reproducir el efecto de realidad aumentada y presentar en la pantalla del dispositivo, el resultado de realidad aumentada.

2.3.3 Gestión de realidad aumentada con equipos específicos

En ocasiones hay dispositivos específicos de captura y visión que se integran en unas gafas especiales que permiten al usuario ver la realidad a través de la lente y mostrar otra información gráfica.

En el caso de los equipos portátiles se puede añadir información de los sistemas de posicionamiento geográfico GPS, necesarios para poder localizar con precisión la situación del usuario. La realidad aumentada es capaz de mostrar al usuario una representación realista del entorno que se ha añadido virtualmente, por ello es importante determinar la orientación y posición exacta del usuario, en la Figura 2.1 se muestra la similitud de los dispositivos móviles.

Esta localización se hace difícil en espacios interiores o en zonas afectadas por campos magnéticos (Javier Fombona Cadavieco, Realidad aumentada, una evolución de las aplicaciones de los dispositivos móviles - Artículo ISSN: 1133-8482 Píxel-Bit. Revista de Medios y Educación, 2012).

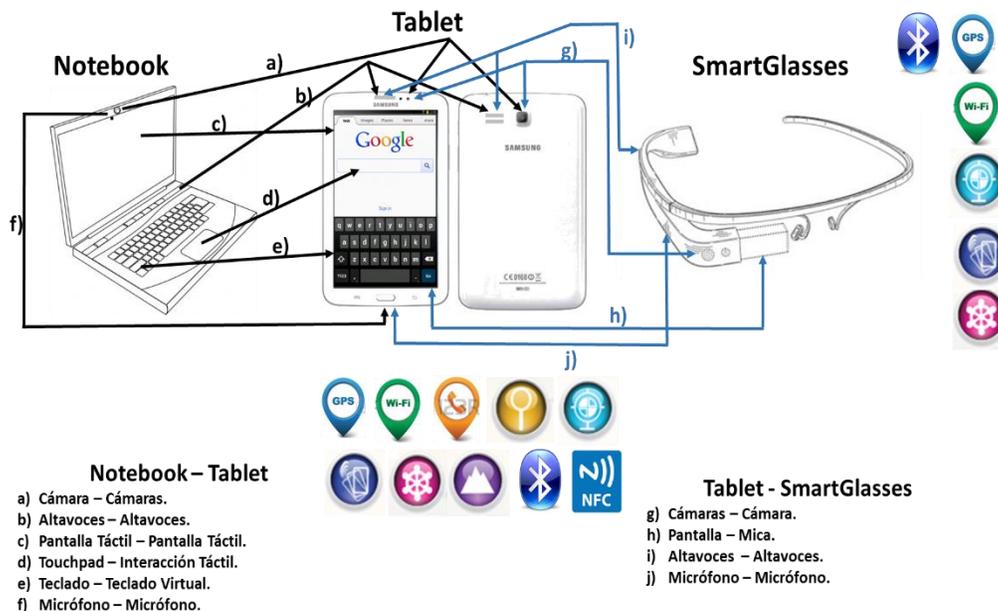


Figura 2.1 Similitud de los dispositivos móviles (Javier Fombona Cadavieco, 2012).

2.4 Reconocimiento de patrones

De manera intuitiva, se puede definir el reconocimiento de patrones como la forma de interpretar al mundo, por ejemplo, un bebe aprende a reconocer su entorno distinguiendo patrones visuales (formas, colores, etc.), auditivos (sonidos, música, etc.), sensitivos (calor, frio, etc.), entre otros. Conforme un niño crece va refinando su capacidad de reconocimiento de patrones, de manera que puede distinguir una melodía de otra, diferentes variaciones de temperatura o diferentes tonalidades de un mismo color. Posteriormente aprende a hablar, a reconocer letras, a formar y entender palabras, etc. En fin, se puede decir que el reconocimiento de patrones es el mecanismo con el que se cuenta para distinguir unas cosas de otras, relacionar cosas semejantes, formar grupos de cosas, describir objetos, tomar y explicar decisiones, etc. (Carrasco Ochoa Jesús Ariel, 2011).

En un sentido más formal, se puede definir al reconocimiento de patrones como *“la ciencia que se ocupa de los procesos sobre ingeniería, computación y matemáticas relacionados con objetos físicos y/o abstractos, con el propósito de extraer información que permita establecer propiedades de o entre conjuntos de dichos objetos, los cuales nos permitan interpretar el mundo que nos rodea”* (Ruiz-Shulcloper J., 1999).

Para poder reconocer los patrones se siguen los siguientes procesos, ver Figura 2.2:



Figura 2.2 Pasos fundamentales en el procesamiento digital de imágenes (Gonzalez Rafael C., 2001).

El punto esencial del reconocimiento de patrones es la clasificación: se quiere clasificar una señal dependiendo de sus características. Las señales, características y clases pueden ser de cualquiera forma, por ejemplo se puede clasificar imágenes digitales de letras en las clases “A” a la “Z” dependiendo de sus píxeles o se puede clasificar ruidos de cantos de los pájaros en clases de órdenes aviares dependiendo de las frecuencias (Calot, 2008).

Capítulo 3

Estado del Arte

Es importante conocer las técnicas más recientes para adquirir el conocimiento necesario, en este capítulo se describirán los trabajos que tienen objetivos similares a los del proyecto.

3.1 EMOVIS - An efficient mobile visual search system for landmark recognition (Dawei & Mooi Choo , 2013)

En este trabajo se presenta un sistema móvil eficiente de búsqueda visual que permite a los usuarios de dispositivos móviles recuperar información relevante mediante consultas basadas en imágenes, el sistema desarrollado recibe el nombre de EMOVIS. Para mejorar la eficiencia computacional, se diseñaron dos nuevos algoritmos para extraer los puntos clave más sobresalientes de un punto de referencia a partir de un conjunto de imágenes de entrenamiento. Todos los descriptores extraídos que son sobresalientes son utilizados para construir un árbol de palabras para acelerar el proceso de reconocimiento de la imagen.

Para mejorar aún más la precisión y reducir tanto la transmisión correspondiente y el costo de la energía, también se diseñó un esquema de pre-procesamiento de imagen. Se elige la arquitectura cliente-servidor, ya que el proceso de comparación de imágenes intensiva debe llevarse a cabo en un servidor remoto de alta gama en lugar de localmente en los dispositivos móviles debido a sus recursos limitados de la batería y de computación, en la Figura 3.1 se presenta la arquitectura del sistema.

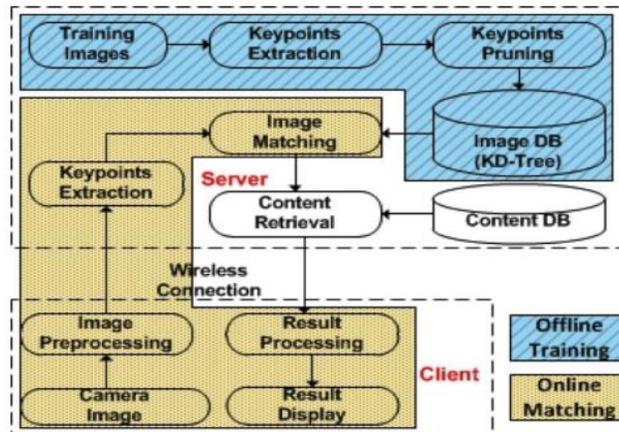


Figura 3.1 Arquitectura del software EMOVIS (Dawei & Mooi Choo , 2013).

3.1.1 Arquitectura

Una vez que se identifica un punto de referencia coincidente, el servidor puede emplear su identificador de objeto único, que es un punto de referencia, para extraer el contenido de la base de datos. La información relacionada con ese punto de referencia, por ejemplo, la historia de ese edificio, se almacena en la base de datos. La descripción del contenido puede ser enviada al usuario mediante una consulta.

3.1.2 Pre procesamiento de la imagen

El pre-procesamiento de imagen implica dos tareas. La primera tarea es cambiar el tamaño de la imagen con la calidad adecuada. La segunda tarea es detectar y eliminar partes irrelevantes de una imagen de consulta, cualquier interferencia potencial de estas partes irrelevantes para el proceso de comparación pueda ser reducida.

3.1.3 Punto de interés en la imagen

En este trabajo, se diseñaron dos esquemas para identificar puntos clave destacados de un conjunto de imágenes de entrenamiento para un punto de referencia. Para explicar mejor los esquemas, hay que tener en cuenta dos definiciones importantes:

Definición 1: Conjunto unido de puntos claves (Union Keypoint Set) de una imagen, es la unión de todos los conjuntos de puntos claves (Keypoint) obtenido apartir de comparacion de imagen con otras imágenes de entrenamiento.

Definición 2: Conjunto minimo de puntos claves (Minimal Keypoint Set) de una imagen es el conjunto más pequeño, punto significativo producido apartir de comparacion de imagen con otras imágenes de entrenamiento.

El método en primer lugar, coincide con los puntos clave extraídos de una imagen con los extraídos de otra imagen de entrenamiento. Los puntos clave coincidentes de la imagen de entrenamiento kps (keypoints) son el conjunto de la unión existente de puntos clave coincidentes encontrados hasta ahora y los contadores correspondientes se actualizan y se almacenan.

Algorithm 2 MaxMin-Based Salient Keypoint Identification

Input: A set of training images $S_{Img} = \{Img_1, Img_2, \dots, Img_n\}$
Output: *Master Salient Keypoints MSK*

```

1: master  $\leftarrow$  null
2: maxCount  $\leftarrow$  0
3: for each image  $Img_i \in S$  do
4:   minCount  $\leftarrow$   $\infty$ 
5:   UKS  $\leftarrow$   $\emptyset$                                      # Union Keypoint Set
6:   for each image  $Img_j \in S$  do
7:     if  $Img_j \neq Img_i$  then
8:       kps  $\leftarrow$  MatchKeypoint( $Img_j, Img_i$ )
9:       UKS  $\leftarrow$  Union(UKS, kps)
10:      if  $len(kps) < minCount$  then
11:        minCount  $\leftarrow$   $len(kps)$ 
12:      end if
13:    end if
14:  end for
15:  if  $minCount > maxCount$  then
16:    maxCount  $\leftarrow$  minCount
17:    MSK  $\leftarrow$  UKS
18:  end if
19: end for
20: return MSK

```

Figura 3.2 Algoritmo MaxMin (Dawei & Mooi Choo , 2013).

Este método determina para cada imagen el conjunto mínimo de puntos clave, entonces el método elige la mejor imagen que produzca el mayor conjunto mínimo de puntos claves. El conjunto unido de puntos clave asociado con esta imagen se utiliza como puntos de interés particular para el objeto.

Algorithm 1 MaxUnion-Based Salient Keypoint Identification (MaxUnion-SKI) Method

Input: A set of training images $S_{Img} = \{Img_1, Img_2, \dots, Img_K\}$
Output: *Master Salient Keypoints MSK*

```

1: master  $\leftarrow$  null
2: maxCount  $\leftarrow$  0
3: for each image  $Img_i \in S$  do
4:   UKS  $\leftarrow$   $\emptyset$                                      # Union Keypoint Set
5:   for each image  $Img_j \in S_{img}$  do
6:     if  $Img_i \neq Img_j$  then
7:       kps  $\leftarrow$  MatchKeypoint( $Img_i, Img_j$ )
8:       UKS  $\leftarrow$  Union(UKS, kps)
9:     end if
10:  end for
11:  if  $len(UKS) > maxCount$  then
12:    maxCount  $\leftarrow$   $len(UKS)$ 
13:    MSK  $\leftarrow$  UKS
14:  end if
15: end for
16: return MSK

```

Figura 3.3 Algoritmo MaxUnion (Dawei & Mooi Choo , 2013).

3.2 Edge-SIFT: Discriminative Binary Descriptor for Scalable Partial-Duplicate Mobile Search (Shiliang Zhang, 2013)

El objetivo de este trabajo de investigación es proponer una solución para la búsqueda a gran escala de imágenes en dispositivos móviles.

Se pretende extraer un nuevo descriptor local binario llamado Edge-SIFT a partir de los mapas de borde binario extraído de las imágenes, Edge-SIFT permite una búsqueda de imágenes precisa y eficiente y es ideal para el cálculo de escenarios sensibles tales como la búsqueda de imágenes móviles.

Los experimentos en un conjunto de datos a gran escala realizado con Edge-SIFT muestran la precisión de recuperación superior a ORB y es superior a SIFT en los aspectos de la precisión de recuperación, la eficiencia, la compacidad y el costo de transmisión

3.2.1 Extracción EDGE-SIFT

Como se ilustra en la Figura 3.4, primero se extraen regiones de la imagen que rodean los puntos de interés. Luego, basándose en las escalas y orientaciones de los puntos de interés, se normalizan las regiones de la imagen en escala y orientación fija para mapas de borde. Entonces se descompone el borde del mapa en diferentes mapas sub-borde, de acuerdo con las direcciones de los bordes. Para hacer Edge-SIFT más robusto a errores de registro causados por localización inexacta del punto de interés, transformaciones afines, etc., se amplían los bordes en cada mapa sub-borde para hacer bordes en otras regiones que puede ser considerada para el cálculo de la similitud.

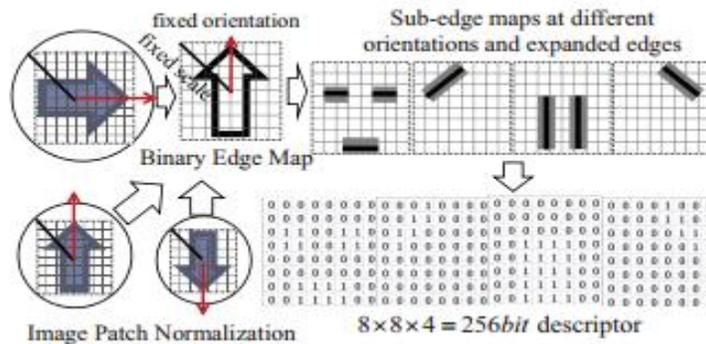


Figura 3.4 Framework propuesto de extracción Edge-SIFT (Shiliang Zhang, 2013).

3.2.2 Detección de puntos de interés

Se utilizó el enfoque de SIFT para la detección del punto de interés, este enfoque consiste en tres pasos.

- 1) Detección extrema Escala-espacio: se calcula una serie de imágenes de diferencia de Gauss con diferentes escalas
- 2) Localización de puntos de interés: ubicaciones de los puntos de interés más precisos se asignan. Mientras tanto, se descartan los puntos inestables.
- 3) Asignación Orientación: orientaciones dominantes de puntos de interés se calculan con un resumen de los gradientes de píxeles en sus regiones de imagen correspondientes y se seleccionan las direcciones más dominantes.

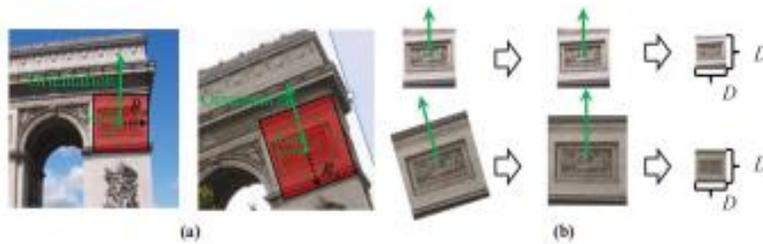


Figura 3.5 (a) Extracción de una región de la imagen. (b) ejemplo de normalización orientación y normalización de la escala (Shiliang Zhang, 2013).

3.2.3 Extracción y normalización de la imagen

Sobre la base de los puntos de interés detectados, que primero extraen regiones de imagen alrededor de los puntos de interés, es decir, como se ilustra en la Figura 3.5(a).

Para lograr la escala y la invariancia de rotación, se normalizaron los parches de imágenes extraídas en la orientación y escala fijas. Como se ilustra en la Figura 3.5(b), la orientación se normaliza mediante la rotación de los parches de imágenes para asegurarse de que sus orientaciones dominantes sean idénticos.

Edge-SIFT registra la ubicación y orientación de los píxeles del borde, por lo tanto, conserva pistas espaciales e impone restricción estricta en función de adaptación. La eficacia de la Edge-SIFT inicial se ilustra en la Figura 3.6. Se puede observar claramente que, la mayoría de los descriptores EdgeSIFT iniciales entre las imágenes se emparejan correctamente.

Algunos pares falsos coincidentes Edge-SIFT se producen debido a que sus regiones de imagen correspondientes contienen pistas de borde similares. También es obvio que el número de descriptores SIFT coincidentes es generalmente mayor que el número de descriptores coincidentes iniciales Edge-SIFT, que ilustra intuitivamente el poder discriminativo de Edge-SIFT.



Figura 3.6 Líneas rojas: emparejados 1024 bits descriptores iniciales Edge-SIFT. Las líneas azules: acertaron SIFT descriptores entre imágenes visualmente similares (Shiliang Zhang, 2013).

3.3 Memory Efficient image databases for mobile visual search (Girod & Chen, 2014)

En esta investigación se tiene como objetivo mejorar la experiencia del usuario en los sistemas de búsqueda visual móvil, aumentando la transmisión de datos a través de la red y que la congestión del servidor no sea un impedimento para el funcionamiento del sistema. Para abordar este problema se plantea una base de datos de firmas de imágenes almacenadas localmente en su totalidad en un dispositivo móvil.

Se presentan cuatro métodos desarrollados recientemente para la construcción de una base de datos compacta de características basadas en imágenes locales y comparar sus actuaciones de recuperación:

- Tree histogram coding (THC)
- Inverted index coding (CII)
- Residual enhanced visual vector (REVV)
- Scalable compressed fisher vector (SCFV)

3.3.1 Esquema de funcionamiento

La Figura 3.7 muestra un esquema de recuperación y el aumento de imágenes en el dispositivo. En primer lugar, el movimiento de los objetos entre los marcos del visor se analiza para detectar períodos de bajo movimiento, cuando es más probable que estén interesados en el contenido del visor del usuario. A continuación, las características locales se extraen de los cuadros seleccionados.



Figura 3.7 El sistema reconoce al aire libre (Girod & Chen, 2014).

El esquema en la Figura 3.8 muestra las etapas del proceso de recuperación de la imagen utilizando las características locales de la imagen. La comparación de las características locales directamente sería demasiado caro y lento computacionalmente, por lo que se utiliza una característica global, está por lo general se genera a partir de muchas características locales extraídas para cada imagen.

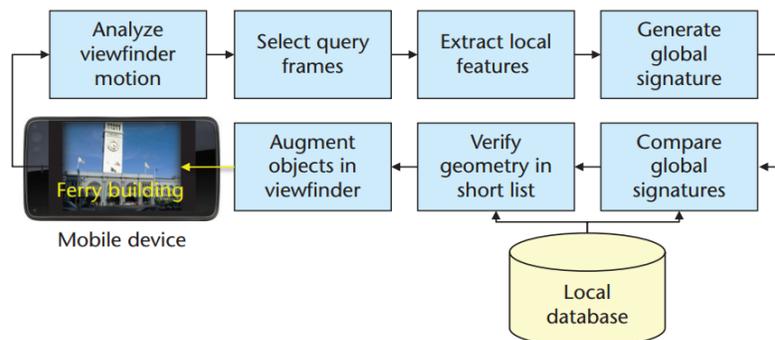


Figura 3.8 Esquema de funcionamiento del sistema (Girod & Chen, 2014).

Una característica global bien diseñada puede resumir las estadísticas más importantes de las características locales y tiene una representación compacta de las características globales de la imagen para permitir comparaciones rápidas. Mediante la comparación de la característica global de las consultas en todas las características globales de la base de datos, se produce una lista ordenada de candidatos en la base de datos. Las características locales de los principales candidatos de la lista son evaluadas mediante consistencia geométrica con respecto a la imagen consultada.

Por último, los aumentos relevantes se dibujan para los objetos reconocidos en el visor sobre la base de la información recuperada. Las posiciones de los objetos reconocidos se actualizan continuamente empleando el mismo algoritmo de análisis de movimiento que se utilizó anteriormente para detectar intervalos de movimiento.

3.3.2 Características de una base de datos compacta

Una representación de base de datos compacta proporcionará las siguientes capacidades principales:

- Más imágenes pueden ser indexadas en la base de datos, lo que aumenta el rango de búsqueda de imágenes.
- Múltiples bases de datos se pueden almacenar en la memoria RAM, permitiendo que los sistemas MVS puedan buscar múltiples clases de objetos.
- Cada base de datos se puede cargar de forma rápida en la memoria RAM de la tarjeta de almacenamiento.
- Una base de datos de memoria eficiente es más rápida para buscar porque se requieren menos accesos de memoria para completar cada consulta.
- Cada uno de los cuatro métodos que se discutieron tiene los intentos de construir bases de datos visuales compactas.

Se comparó el rendimiento de recuperación de THC, CII, REVV y SCFV en un conjunto de datos MPEG que contiene 8313 imágenes de la consulta de cinco clases (gráficos, cuadros, cuadros de video, puntos de referencia y objetos comunes) como se puede ver en la Figura 3.9.

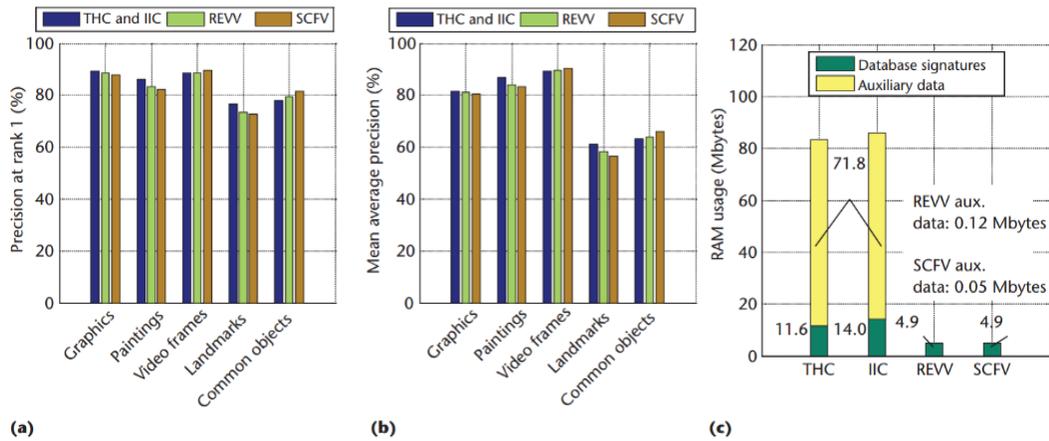


Figura 3.9 Comparación de THC, ICC, REVV, y scFv en la base de datos MPEG con características SIFT en términos de (a) la precisión en el rango 1, (b) significa precisión media, y (c) RAM de uso de una base de datos de 10,000 imágenes del conjunto de datos (Girod & Chen, 2014).

En la Figura 3.10 se muestran los resultados de las pruebas realizadas con el método REVV en un smartphone Samsung Galaxy S3, que cuenta con un procesador de 1.4 GHz y 1 GB de RAM con una base de datos de 10,000 imágenes fue indexada en el dispositivo, la primera gráfica se evaluó la distribución de reconocimiento de 400 consultas diferentes, en la segunda gráfica el porcentaje de tiempo dedicado a la extracción de características, la búsqueda de base de datos y la verificación geométrica.

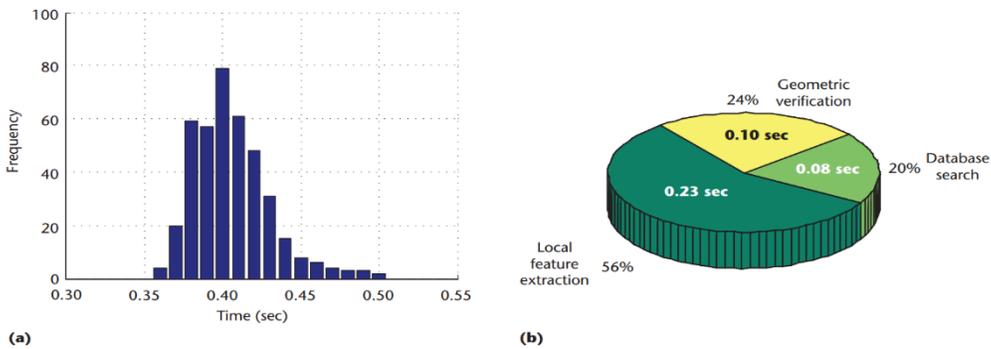


Figura 3.10 Aplicación Android en Smartphone. (a) Distribución de reconocimiento latencias de 400 consultas diferentes. (b) El porcentaje de tiempo dedicado a la función de extracción, la búsqueda de base de datos, y la verificación geométrica (Girod & Chen, 2014).

3.4 Interactive Multimodal Visual Search on Mobile Device (Houqiang , Yang , Tao , Jingdong , & Shipeng, 2013)

En este trabajo se describe un nuevo sistema de búsqueda de imágenes interactivo multimodal en los dispositivos móviles. Está diseñado para los usuarios que ya tienen imágenes en sus mentes, pero no tienen descripciones o nombres precisos para hacer frente a ellos.

Se describe utilizando el habla y se compone una consulta visual usando las imágenes de ejemplos, el usuario puede encontrar fácilmente las imágenes deseadas a través de algunas interacciones multimodales naturales con su dispositivo móvil en la Figura 3.11 Se describe la arquitectura del sistema.

3.4.1 Funcionamiento del sistema

El procedimiento de búsqueda de nuestro sistema propuesto consta de las siguientes fases:

- El usuario habla una oración natural para describir las imágenes de ejemplo.
- El discurso se reconoce y se descompone en palabra (s) que se puede representar por imágenes ejemplares.
- El usuario selecciona el ejemplar preferido (s) y compone un collage esquemático como una imagen compuesta.
- La imagen compuesta se utiliza como una consulta visual para buscar imágenes similares.
- Si es posible, más información como la ubicación de GPS e imagen de descripciones se proporcionan para el usuario.

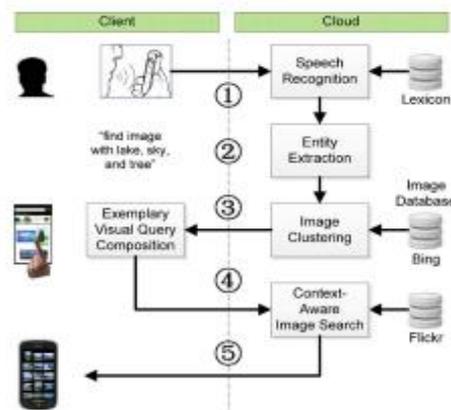


Figura 3.11 La arquitectura del sistema propuesto de búsqueda visual multimodal. Hay cinco componentes principales: (1) El reconocimiento de voz, (2) La extracción de entidades, (3) La generación de imágenes a modo de ejemplo, (4) La composición visual de consultas de ejemplar interactivo, y (5) Basada en ejemplos de imágenes búsqueda consciente del contexto (Girod & Chen, 2014).

3.4.2 Entidad de discurso

La extracción de entidades de habla puede ser dividida en dos pasos:

- 1) Reconocimiento de voz
- 2) Extracción de la clave de entidad.

Un modelo de Markov (HMM) y motor basado en N-gramas es capaz de manejar ambas frases naturales y fragmentos de frases. Sin embargo, se prefiere el uso de frases naturales como "encontrar una imagen con la torre de hierro en la hierba" en lugar de frases separadas como "torre de hierro, la hierba".

Por lo general, el texto reconocido se utiliza directamente como una consulta para buscar imágenes. Sin embargo, por una parte, en una descripción hay muchas palabras no informativas e inútiles en el texto, y, por otro lado, la pura coincidencia de texto ignora el contenido real de las imágenes.

3.4.3 Formulación de consultas visuales

Debido a que el usuario no tiene imágenes ejemplares adecuadas en el teléfono, el sistema primero sugiere automáticamente algunos ejemplos visuales de acuerdo a las palabras clave extraídos de la voz del usuario.

El usuario luego hace un collage con imágenes, usando algunas imágenes ejemplares favorables como piezas de bloques de construcción, la Figura 3.12 demuestra los pocos elegidos de imágenes ejemplares.

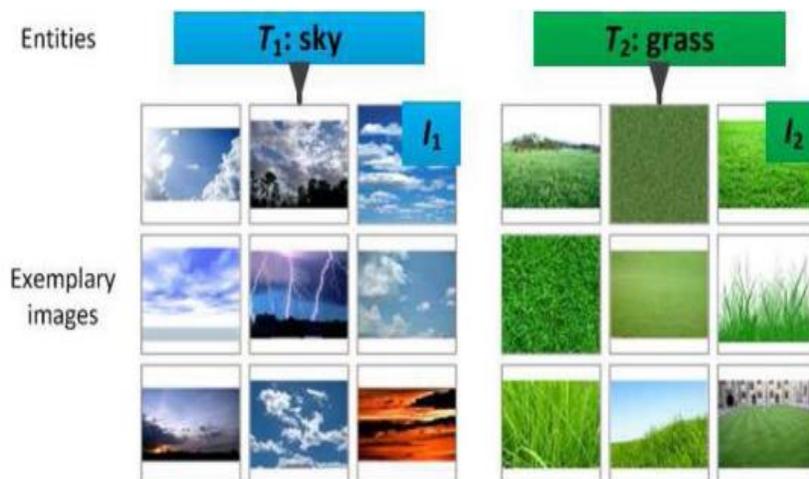


Figura 3.12 Un ejemplo de procesamiento de una consulta visual con múltiples ejemplares (Girod & Chen, 2014).

Una base de datos ejemplar debe estar lista para proporcionar imágenes de ejemplares para el usuario de acuerdo con sus palabras clave. Dada una palabra clave, un conjunto de imágenes se descarga desde Internet mediante un motor de búsqueda de imágenes.

3.4.4 Indexación y recuperación

Millones de imágenes en escala son indexados por las palabras visuales extraídas en un índice revertido. Cada imagen se representa como una serie de códigos visuales, y luego se construye una tabla de asignación de códigos visuales a las imágenes. Los mismos códigos visuales son seguidos por una lista de entradas. Cada entrada mantiene la información de una pieza única de una imagen que incluye ID imagen, ubicación del punto central y peso.

Dado que la cantidad de imágenes son similares, pero con diferentes semánticas visuales es difícil de controlar por este motivo primero se utilizan las palabras claves para marcar todas las imágenes que son relevantes para cualquier palabra clave de la recuperación basada en texto. Muchas imágenes de la web, como imágenes de Flickr tienen información de texto como etiquetas, títulos y descripciones.

3.5 Car Recognition from frontal images in mobile environment (Tanacs & Varjas, 2013)

El objetivo en este trabajo de investigación es el reconocimiento de la marca y modelo de coches a partir de imágenes frontales.

Se proponen dos métodos, semiautomático y completamente automáticos, la clasificación se basa en Square Mapped Gradient (SMG), es capaz de dar resultados casi perfectos cuando no hay una clase de valores atípicos, que disminuye a 92% y 87% en el caso de los métodos semiautomáticos y completamente automáticos.

Se utiliza el algoritmo región de interés (ROI) para el reconocimiento, se descartan partes de los datos de imagen. Dado que la información de color no es relevante para el reconocimiento, se pueden utilizar las imágenes en escala de grises. Si tres puntos de las esquinas de la placa de la matrícula están disponibles, se puede calcular su anchura, altura, punto central y su ángulo respecto a la línea horizontal, la Figura 3.13 presenta las principales etapas de detección ROI.



Figura 3.13 Frontal original (izquierda una parte de la matrícula es oscurecida por el anonimato), la compensación de la rotación, número de placa de identificación y la extracción de retorno de la inversión (en el centro), y el retorno de la inversión extraído (derecha) (Tanacs & Varjas, 2013).

3.5.1 Extracción de características desde la región de interés (ROI)

El contenido de información de la imagen ROI no es adecuado para el reconocimiento directamente. El tamaño ROI es diferente para cada imagen, contiene información que no es específica del modelo. Por extracción de características estos contenidos de información están normalizados a fin de obtener vectores comparables para coches diferentes.

La tarea de la clasificación es responder a qué marca y modelo del vehículo corresponde un vector de características en particular. El clasificador calcula la distancia entre un vector de característica de entrada y los que están en el conjunto de entrenamiento que se utiliza para el reconocimiento.

Lo más rápido es el clasificador vecino más cercano (nearest neighbor) que selecciona la etiqueta del vector de características más cerca del conjunto de entrenamiento. También se puede considerar los tres vectores más cercanos y tomar un voto de la mayoría de ellos.

3.5.2 Reconocimiento semiautomático y totalmente automático

La evaluación de las capacidades de reconocimiento del método clasificador debe impedir los efectos de la degradación en los errores de segmentación. Es por ello que se utilizan los datos de punto de la esquina del número de la placa seleccionados manualmente para tal fin. En la práctica, sin embargo, que no es aplicable. Para este tipo de problemas se proponen dos enfoques.

El semi-automático requiere la identificación de un punto dentro del número de la placa, mientras que uno totalmente automático funciona sin interacción del usuario, el método totalmente automático determina un punto dentro del área del número de placa; el semi-

automático detecta los puntos de la esquina del número de la placa de un punto de semilla dada, el reconocimiento se basa en ROI determinada por estos puntos de esquina.

El método semi-automático se basa en una técnica de cultivo de región sencilla. Desde un punto de semilla, un punto vecino se añade a la región si su valor de intensidad está dentro de un intervalo dado y su distancia desde el punto semilla está por debajo de un umbral. El intervalo de intensidad se determina con base en el entorno del punto de semilla por lo que puede funcionar en casos en el que estén claros o más oscuros los números de la placa de origen. Los agujeros se rellenan dentro de la región y las operaciones morfológicas son aplicadas a refinar la forma de la región Figura 3.14.

El intervalo de intensidad se determina con base en el entorno del punto de semilla por lo que puede funcionar en casos en el que este claro o más oscuro los números de la placa origen. Los agujeros se rellenan dentro de la región y las operaciones morfológicas son aplicadas a refinar la forma de la región Figura 3.14.



Figura 3.14 Los pasos de la segmentación semiautomática. Crecimiento de región, orificio de llenado, post-procesamiento morfológico y el contorno de la región detectada superpuesta en la parte superior de la imagen original (Tanacs & Varjas, 2013).

El método totalmente automático intenta determinar un punto dentro de la placa de la matrícula y llamar al método semi-automático. Dado que las imágenes deben tomarse siguiendo un protocolo se puede restringir la región de búsqueda a una ventana en el tercio inferior de la imagen y a la mitad de su anchura en el centro. Se asume que la línea que pertenece al valor de proyección máxima pasa a través de la placa Figura 3.15.



Figura 3.15 La detección de la línea que va a través de la placa de la matrícula. Umbral para la magnitud del bode de la imagen y su proyección (izquierda) y la línea (en rojo) para los candidatos de puntos de semilla (derecha) (Tanacs & Varjas, 2013).

3.6 Tecnologías wareables

3.6.1 Smart Helmet

El casco inteligente, desarrollado por DAQRI, es un verdadero intento de integrar la tecnología de realidad aumentada en el sector de la construcción. El casco inteligente es similar a un casco de construcción, pero en realidad tiene una pantalla de alta definición de manos libres que está justo detrás de la visera protectora. Esta pantalla se utiliza para crear una realidad aumentada en tiempo real, en contenido 4D para el usuario.

El casco inteligente también tiene una cámara de profundidad en 3D de alta resolución, así como cámaras de navegación de 360 grados que soportan la grabación de vídeo de alta definición, la cartografía 3D, y la captura alfanumérica. Permite a los usuarios crear listas de verificación como cuando están haciendo una inspección, y se puede comparar modelos en 3D del sitio real al plan para ver si el proyecto se ajusta al calendario. Los usuarios también pueden emitir instrucciones y tareas de inmediato para reducir los errores y aumentar la eficiencia (DAQRI Smart Helmet, 2014).



Figura 3.16 Smart Helmet (casco inteligente) (DAQRI Smart Helmet, 2014) .

Aplicaciones

- Reconocimiento de contornos
- Reconocimiento y seguimiento de objetos
- Interfaces virtuales
- Cascos para la construcción
- Cascos para motocicletas

3.6.2 HoloLens

Los HoloLens, desarrollado por Microsoft, son similares a Google Glass, ya que son dos auriculares que ofrecen experiencias de realidad aumentada, es una computadora holográfica que a través de los hologramas permite tomar decisiones con más confianza, trabajar con mayor eficacia, y llevar ideas a la vida ante sus ojos, utiliza movimientos, la visión y al voz para interactuar con el contenido y al información de la manera más natural posible.

Permiten la presentación en vivo de los elementos del mundo real físico que se incorporan con elementos virtuales de tal manera que se perciben a coexistir en un entorno compartido. Windows HoloLens ofrece un entorno operativo de realidad aumentada en el que cualquier Windows App universal pueda correr. Además, las API holográficas, que forman parte de la Plataforma Universal de Windows y siendo compatible con Windows 10 se puede implementar características de realidad mixta fácilmente en cualquier sistema Windows App universal, para una amplia gama de dispositivos de Windows (HoloLens, 2015).



Figura 3.17 Gafas HoloLens (HoloLens, 2015).

Aplicaciones

- Simulaciones
- Ayuda en tiempo real
- Reconocimiento y seguimiento de objetos
- Creación y ajustes de objetos 3D
- Creación de planos arquitectónicos

3.6.3 Atheer AiR Platform

Atheer es el pionero en realidad aumentada interactiva (AIR™), que combina el poder de la realidad aumentada en 3D con interacción natural para desbloquear la productividad humana. La interactividad es la piedra angular de un flujo de trabajo móvil eficaz. La plataforma Atheer

AiR faculta a los trabajadores para interactuar con el mundo digital con la misma facilidad que lo hace en el mundo físico, poniendo interacción natural, basada en gestos en el centro de la experiencia informática.

A través de gafas inteligentes de inmersión se obtiene una gran área de visualización donde los trabajadores pueden ver e interactuar con su información digital sin tener que sostener un dispositivo físico en sus manos. Permite personalizar la visualización de imágenes para reducir la fatiga visual, permitiendo una experiencia realmente envolvente y entorno visual agradable.

El reconocimiento de gestos proporciona a los trabajadores una forma intuitiva y precisa para interactuar con la información, sin tocar físicamente nada, mantiene las manos libres para completar la tarea en cuestión (labs, 2015).



Figura 3.18 Gafas Atheer AiR (labs, 2015).

Aplicaciones

- Servicio de campo y construcción
- Cuidado de la salud
- Sistema de información de almacenes
- Reconocimiento de objetos para brindar información
- Sector salud

3.6.4 (ODG) Elegante Gafas de Osterhout Design Group

Son gafas inteligentes de ODG tienen una pantalla 3D, así como cámara y sensor de geolocalización. El dispositivo incorpora una gran cantidad de tecnología, ODG espera que sea percibido como un producto de nueva generación. Las gafas pesan sólo cinco onzas y se ven como unas gafas de sol.

Se puede apreciar el entorno a través de los lentes transparentes, pero también se pueden ver las imágenes en la superficie de la parte interior de las gafas como si estuviera viendo una pantalla de alta definición.

Smartglasses de ODG tienen sensores de posición, por lo que las gafas son capaces de saber dónde se encuentra el usuario, dónde se realiza la búsqueda, y cómo se están moviendo. Esto permite que la realidad aumentada sea mucho más precisa. El dispositivo tiene un procesador quad-core Qualcomm Snapdragon, conexión con redes 802.11ac inalámbrica y navegación a través del sistema de posicionamiento global (GPS), el ruso Glonass, y las redes de satélites de Magallanes. Cuenta con 128 gigabytes de almacenamiento, tomas de audio magnético, dos micrófonos digitales, una cámara de 5 megapíxeles y armazón reforzado para entornos difíciles (ODG, 2015).



Figura 3.19 Gafas de ODG (ODG, 2015).

Aplicaciones

- Simulación
- Manuales de usuarios
- Diseños arquitectónicos
- Videoconferencias
- Recibir y enviar contenido multimedia

3.6.5 Moverio BT-200

Moverio BT-200 son lentes compactos inteligentes e innovadores desarrollados por Epson que poseen capacidades de realidad aumentada. Esta es la segunda versión de gafas inteligentes de Epson, y añade nuevas características haciéndolas más ligeras y más compactas. Cuentan con una cámara frontal y una unidad de control con teclado táctil que se ejecuta a través de software

Android 4.0. Estas gafas inteligentes también incorporan capacidades de seguimiento de movimiento que permite una mayor interactividad en las solicitudes de los usuarios. Estas gafas permiten a los usuarios interactuar con la realidad aumentada de una manera más inmersiva y pueden ayudar a aumentar la eficiencia y ofrecer una mayor transparencia a los usuarios.

Las gafas Moverio BT-200 prometen un avance en el mercado de estas gafas, pero siguen enfocadas a un mercado más profesional que el general. La calidad y la resolución no pueden competir con la de las pantallas convencionales, y la comodidad y dimensiones aún tienen que mejorar como para que el usuario considere comprarlas debido a su precio (EPSON, 2015).



Figura 3.20 Gafas Moverio BT-200 (EPSON, 2015)..

Aplicaciones

- Simuladores
- Juegos
- Manuales
- Sistemas de información

Los trabajos anteriores son similares al proyecto que se realizó, complementándose uno al otro, a continuación, se presenta una tabla de características y de comparación.

Tabla 3.1 Tabla comparativa del estado del arte.

Nombre del trabajo	Año	Objetivo	Técnicas y/o métodos	Tipo de reconocimiento
EMOVIS. Un sistema de búsqueda visual móvil para el reconocimiento de Landmark	2013	Permitir a los usuarios móviles recuperar información relevante mediante consultas basadas en imágenes que se lleve a cabo con alta precisión.	-Consulta basada en imágenes -Algoritmos Maxunion y Maxmin -Algoritmo SURF -Eliminar regiones irrelevantes de la imagen de consulta	Reconoce imágenes de paisajes
Discriminativo binario para descriptores escalables en búsqueda móvil	2013	Tener un método de búsqueda de imágenes precisa y eficiente e ideal para el cálculo de escenarios sensibles tales como la búsqueda de imágenes en móviles.	-Algoritmo SIFT, Edge-SIFT -Descriptores basados en histogramas -Método Bolsa de palabras -Algoritmos de indexación	Reconocimiento de imágenes en dispositivos móviles
Memoria eficiente en base de datos de imágenes para búsqueda en móvil	2014	Realizar consultas de imágenes, sin tener el problema de la transmisión lenta a través de una red inalámbrica o la congestión de un servidor ocupado, que no afecte la experiencia del usuario.	-Base de datos local -Métodos para la construcción de base de datos THC, CII, REVV -Descriptores basados en histogramas	Reconocimiento de imágenes desde un dispositivo móvil sin la ayuda de un servidor
Búsqueda interactiva multimodal en dispositivos móviles	2013	Diseñar una aplicación de búsqueda visual eficiente con la ayuda de imágenes en el teléfono móvil y búsqueda de escenas.	-Reconocimiento de voz -Algoritmo SIFT -Algoritmo K-means -Modelos de Markov	Búsqueda de imágenes en internet

Nombre del trabajo	Año	Objetivo	Técnicas y/o métodos	Tipo de reconocimiento
Reconocimiento de coches con imágenes frontales en ambiente móvil	2013	Realizar reconocimiento de los coches, marca y modelo a partir de imágenes frontales.	<ul style="list-style-type: none"> -Regiones de interés RIO -Algoritmo del vecino más cercano -Histogramas 	Reconocimiento de placas de automóviles
Gestor de Repositorio de Patrones de Imágenes 3D para Aplicaciones de Información Aumentada	2015	Desarrollar una metodología para el procesamiento de objetos tridimensionales que permita obtener patrones visuales de cada una de las vistas que lo conforman y la identificación semiautomática de los componentes individuales de cada vista, a cada vista y componente de la misma se le asociarán descriptores de texto.	<ul style="list-style-type: none"> - Sistema de vistas americano - Puntos de interés - Tracking visual - Información contextual - Realidad aumentada - Base de datos local 	Reconocimiento de objetos, reconocimiento de múltiples objetos en una imagen.

Capítulo 4

Metodología de Solución

En este capítulo se presenta la metodología de solución implementada para este proyecto de investigación, se presenta la arquitectura general de la aplicación móvil y la solución desarrollada para permitir la carga dinámica de diferentes objetos caracterizados en la metodología descrita en el capítulo 5.

4.1 Metodología

Para la realización de este trabajo de investigación fue necesario diseñar una metodología la cual consta de dos fases para realizar la caracterización y detección de objetos multivista mediante dispositivos multisensoriales. El diagrama de las fases se puede observar en la Figura 4.1.

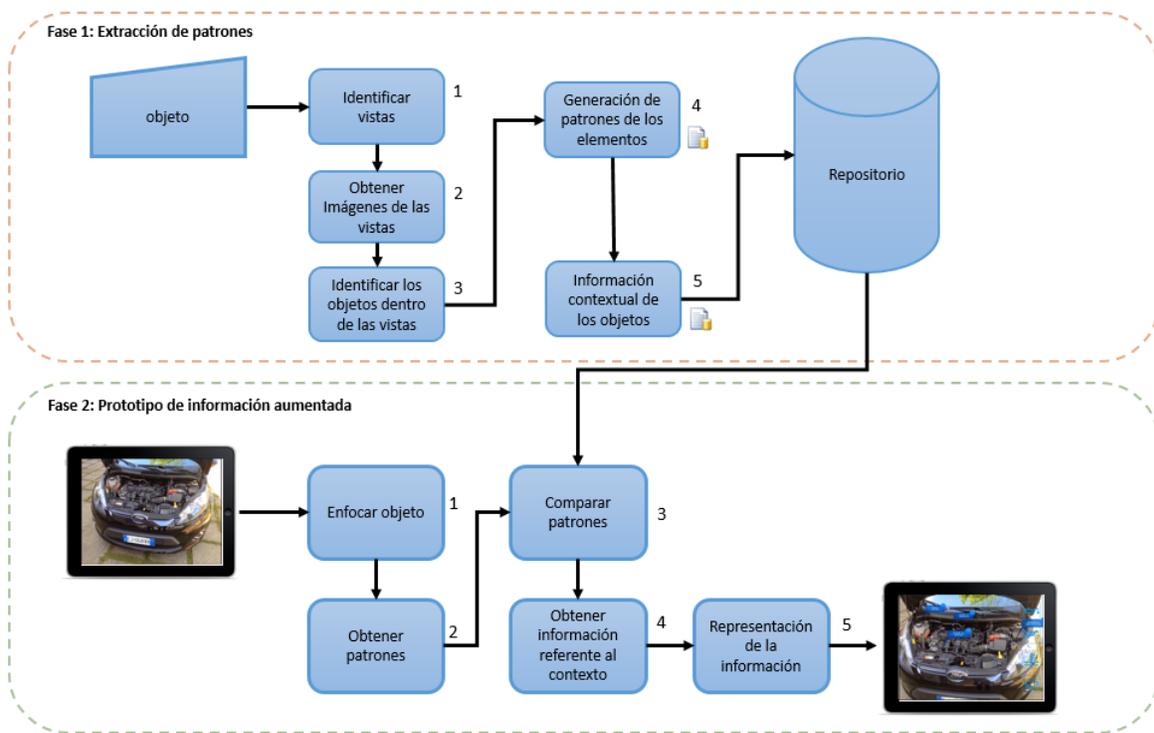


Figura 4.1 Metodología para la detección de objetos tridimensionales.

4.2 Fase 1. Extracción de patrones

Esta fase consiste en la recopilación de información e imágenes de todas las vistas del objeto que se requiere caracterizar, por tanto, recibe como entrada las imágenes capturadas de dicho objeto, ver Figura 4.2. Los procesos a seguir para realizar la recopilación de características son los siguientes:

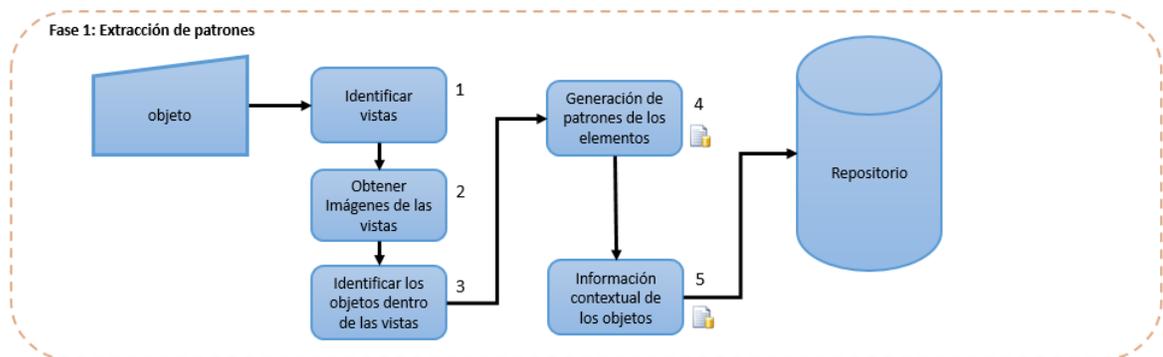


Figura 4.2 Fase de extracción de patrones.

1. **Identificar vistas:** Para este proceso se requiere seleccionar cada de una de las vistas que se desea caracterizar para tomar imágenes de cada una de éstas desde diferentes perspectivas.
2. **Obtener imágenes de las vistas:** De las imágenes obtenidas en el proceso uno (identificar vistas) se seleccionan las vistas del objeto que contienen suficiente información para poder extraer los patrones necesarios para identificar los elementos del objeto, existen dos métodos para la proyección de vistas a continuación se explican los sistemas de obtención de vistas.

Sistema obtención de vistas

Se denominan vistas principales de un objeto, a las proyecciones ortogonales del mismo sobre 6 planos, dispuestos en forma de cubo (Bartolome, 2015). Otra definición es la siguiente: las proyecciones ortogonales de un objeto, según las distintas direcciones desde donde se mire.

Denominación de las vistas

Un observador puede estar situado de acuerdo a las seis direcciones indicadas por las flechas, como se muestra en la Figura 4.3, para obtener las seis vistas posibles de un objeto. Estas vistas reciben las siguientes denominaciones:

Vista A: Vista anterior, de frente o alzado

- Vista B: Vista superior o planta
- Vista C: Vista lateral derecha o simplemente derecha
- Vista D: Vista lateral izquierda o simplemente izquierda
- Vista E: Vista inferior
- Vista F: Vista posterior

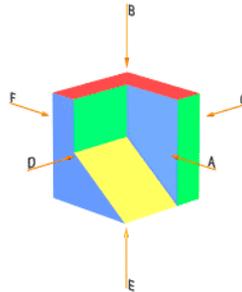


Figura 4.3 Vistas de un objeto

Posiciones relativas de las vistas

Para la disposición de las diferentes vistas sobre el papel, se pueden utilizar dos variantes de proyección ortogonal:

- El método de proyección del primer diedro, también denominado europeo, ver Figura 4.4.



Figura 4.4 Sistema de vistas europeo (Bartolome, 2015).

- El método de proyección del tercer diedro, también denominado americano, ver Figura 4.5.



Figura 4.5 Sistema de vistas americano (Bartolome, 2015).

En ambos métodos, el objeto se supone dispuesto dentro de una caja cúbica, sobre cuyas seis caras interiores, se realizan las correspondientes proyecciones ortogonales del mismo. La diferencia estriba en que, mientras en el sistema Europeo, el objeto se encuentra entre el observador y el plano de proyección, en el sistema Americano, es el plano de proyección el que se encuentra entre el observador y el objeto (Bartolome, 2015).

Debido las características que poseen ambos sistemas y a las necesidades requeridas en esta investigación, se decidió trabajar con el estándar americano. Una vez realizadas las seis proyecciones ortogonales sobre las caras del cubo, y manteniendo fija, la cara de la proyección del alzado (A), se procede a obtener el desarrollo de la caja como se observa en la Figura 4.5.

3. **Identificar los elementos dentro de las vistas:** Se identifican los objetos que contiene cada vista, los objetos deben ser los que proporcionen información contextual relevante para el usuario y contengan suficiente información para poder extraer los patrones.
4. **Generación de patrones de las vistas:** Se extraen los patrones necesarios para la detección de cada una de las vistas seleccionadas del objeto a identificar. Dentro de cada vista se seleccionaron los elementos a los cuales se extraerán los patrones para poder identificarlos.
5. **Información contextual de los objetos:** Se registra información contextual del objeto procesado, es una descripción en texto que será presentada una vez que el objeto es identificado mediante realidad aumentada.

4.3 Fase 2. Prototipo de información aumentada

En esta segunda fase se desarrolló una aplicación móvil en la cual se implementó la fase 1, en esta fase existen cinco procesos los cuales requieren como entrada al objeto que se quiere identificar por el usuario, ver Figura 4.6, una vez que el objeto es identificado se presentará información de contexto al usuario mediante realidad aumentada.

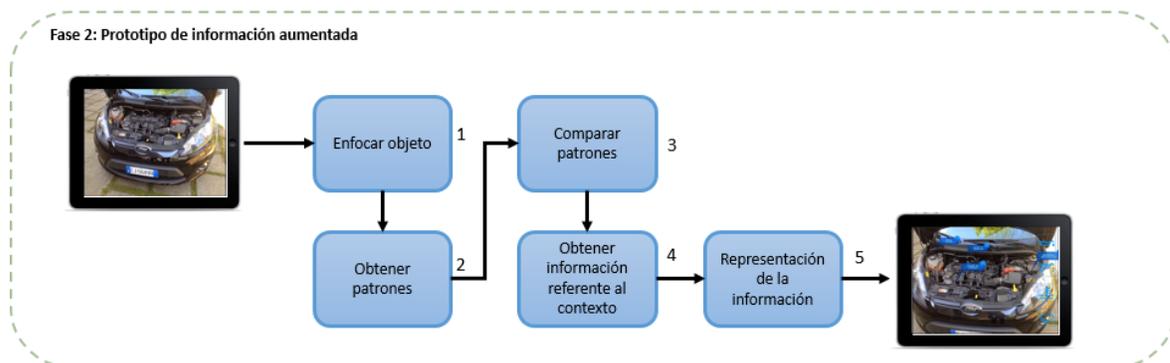


Figura 4.6 Fase del prototipo de información aumentada.

A continuación, se describe cada uno de los procesos:

1. **Enfocar objeto:** Se enfocará el objeto que se requiere identificar, el proceso para enfocar el objeto se realizará mediante la cámara del dispositivo multisensorial.
2. **Obtener patrones:** Se obtendrán los patrones el objeto que será identificado.
3. **Compara patrones:** Se realizará la comparación de los patrones, obtenidos del objeto enfocado con los patrones del repositorio obtenidos en la fase 1, para la identificación del objeto.
4. **Obtener información referente al contexto:** Cuando el objeto es identificado en el repositorio de la fase 1, se obtiene la información de contexto referente a este objeto.
5. **Representación de la información:** En este proceso se representará la información de contexto referente al objeto obtenida en el proceso cuatro.

4.3.2 Proceso de renderizado

El renderizado es el aspecto de mayor complejidad técnica de producción en 3D, pero en realidad se puede entender con bastante facilidad en el contexto de una analogía: Al igual que un fotógrafo la película debe desarrollar e imprimir sus fotos antes de que puedan ser mostradas, los gráficos profesionales de un ordenador tienen la misma necesidad (Slick, 2016).

El término de renderizado se refiere a los cálculos realizados por un paquete de software 3D (motor de render) para traducir la escena desde una aproximación matemática de una imagen 2D finalizado (Slick, 2016).

Tipos de presentación.

Hay dos tipos principales de representación, su principal diferencia es la velocidad a la que se calculan y se finalizaron las imágenes.

- **Real-Time Rendering:** Real-Time Rendering se utilizó habitualmente en juegos interactivos y gráficos, en los que las imágenes deben ser calculadas a partir de la información 3D a un ritmo increíblemente rápido.
- **Fuera de línea o la representación previa:** Es utilizado en situaciones donde la velocidad es un problema menor, con cálculos realizados generalmente con CPUs multi-núcleo en lugar de hardware de gráficos dedicado.

Técnicas de representación:

Hay tres principales técnicas computacionales utilizadas para renderizado. Cada uno tiene su propio conjunto de ventajas y desventajas, por lo que las tres opciones son viables en ciertas situaciones.

- Línea de exploración (o rasterización): Se utiliza cuando la velocidad es una necesidad, lo que hace que sea la técnica de elección para la representación en tiempo real y gráficos interactivos. La técnica de la línea de exploración se utiliza junto con la iluminación precalculada la cual puede alcanzar una velocidad de 60 fotogramas por segundo.
- El trazado de rayos: En el trazado de rayos, para cada pixel en la escena, uno (o más) rayos (s) de la luz se traza desde la cámara al objeto 3D más cercano. El rayo de luz se hace pasar luego a través de un número determinado de "rebote", que puede incluir la reflexión o refracción en función de los materiales en la escena 3D. El color de cada píxel se calcula mediante algoritmos basados en la interacción del rayo de luz con los objetos en su camino trazado. El trazado de rayos es capaz de mayor realismo fotográfico de la línea de exploración, pero es exponencialmente más lento.
- Radiosidad: A diferencia de trazado de rayos, radiosidad se calcula independiente de la cámara, y está orientado a la superficie en lugar de pixel por pixel. Radiosidad se caracteriza típicamente por las sombras suaves, graduadas y pérdida de color.

Proceso de renderizado en realidad aumentada

En la Figura 4.7 se generaliza el proceso de ejecución de la aplicación de realidad aumentada, el proceso se inicia con la entrada de la cámara y termina con la prestación de objetos virtuales en la pantalla. Realidad Aumentada busca continuamente perseguir el objeto (tracking), mostrará el objeto virtual en la pantalla cada vez que identifique el objeto (Parmar, 2015).

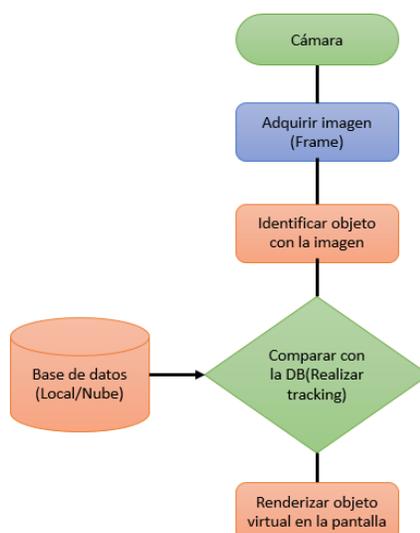


Figura 4.7 Proceso de renderizado en Realidad Aumentada (Parmar, 2015).

Capítulo 5

Diseño e Implementación

En este capítulo se describe la etapa de diseño e implementación, se explica cada uno de los procesos que se deben seguir para la implementación de la metodología, además se realiza una descripción detallada de cada fase, así como las técnicas, tecnologías y herramientas necesarias para la implementación de la metodología propuesta.

5.1 Diseño e implementación de la fase 1

A continuación se describen los pasos que son necesarios en cada uno de los procesos para realizar la Fase 1 de la metodología propuesta.

Antes de que el usuario decida seguir la metodología es necesario que seleccione un objeto de su interés (autos, electrodomésticos, equipos de cómputo, etc.), que sirve como entrada para realizar lo descrito en cada uno de los procesos:

Proceso 1: Identificar vistas

Una vez seleccionado el objeto que se requiere caracterizar, es necesario seleccionar cada una de las vistas del objeto. Se recomienda seleccionar las vistas que proporcionen mayor información de contexto al usuario, por ejemplo: panel de control, botones, conectores, pantallas, etc. Para caracterizar las vistas del objeto se aplica el modelo americano (Bartolome, 2015).

Proceso 2. Obtener imágenes de las vistas

Para realizar este proceso se deben llevar a cabo las siguientes acciones:

- Capturar una serie de fotos de las vistas seleccionadas en el proceso 1.
- Seleccionar las mejores imágenes capturadas, nítidas y con detalles.
- Cambiar el tamaño de las imágenes para que puedan ser procesadas correctamente, el tamaño que deben tener cada una de las fotos es de 711 píxeles de ancho y 400 píxeles de alto. Para cambiar el tamaño de las imágenes el usuario puede elegir el software que le facilite más la tarea, algunos de los que podría utilizar son: Paint, Photoscape, Adobe Photoshop, GIMP, entre otros.

Proceso 3. Identificar los elementos dentro de las vistas

Para este proceso es necesario identificar y seleccionar, en cada una de las vistas, los elementos que complementen la información de contexto al usuario, por ejemplo: botones, pantallas, etiquetas, conexiones, etc., una vez que se seleccionan los elementos, se repiten los procesos 1 y 2 para cada uno de los objetos seleccionados, se capturan las fotos correspondientes de cada elemento detectado dentro de la(s) vista(s), como se muestra en la Figura 5.1.



Figura 5.1 Segmentación de la vista que contienen elementos de contexto

Proceso 4. Generación de patrones de las vistas

La generación de patrones para el reconocimiento de cada una de las vistas se realiza mediante el framework de Qualcomm Vuforia (Vuforia, 2016), a continuación se explican algunas consideraciones que hay que tomar en cuenta para la generación de patrones y reconocimiento de las vistas necesarias para el buen funcionamiento del framework.

Propiedades de Qualcomm Vuforia

El SDK Vuforia utiliza algoritmos para detectar y rastrear las características que se encuentran de forma natural en imágenes. El SDK Vuforia reconoce la imagen objetivo comparando las características naturales contra una base de datos que contiene las características del recurso conocido. Una vez que se detecta esta imagen, el SDK hará un seguimiento de la imagen que se encuentre en el campo de visión de la cámara.

Las *imágenes objetivo* son representaciones de objetos del mundo real que pueden ser detectadas y rastreadas.

Los objetivos incluyen diferentes tipos de objetos, tales como:

- **Objetivos de imagen**, por ejemplo, fotos, tableros de juego, páginas de revistas, portadas de libros, envases de productos, carteles, tarjetas de felicitación.
- **Objetivos del cilindro**, es decir, fotos, imágenes, logotipos u otras imágenes aplicadas sobre la superficie de los objetos cilíndricos y cónicos, tales como: latas, vasos, botellas y cestas.
- **Texto (palabras objetivo)** que representan los elementos textuales tales como palabras simples o compuestos, por ejemplo: las palabras impresas en libros, periódicos, revistas u otros medios de comunicación.
- **Objetivos definidos por el usuario son como los objetivos de imagen**, por ejemplo: fotos, portadas de libros, carteles; sin embargo, éstas permiten aumentar las imágenes en tiempo de ejecución.

- **Objetivos de reconocimiento en la nube**, por ejemplo: los objetivos de imagen que se recuperan de la nube.
- **Multi-objetivos**, por ejemplo: el embalaje del producto o productos que son de formas cuadradas o rectangulares; éstos permiten aumentar objetos 3D simples.
- **Marcadores de marco** que son un tipo particular de cuadros de imagen en 2D con un patrón visual especial y pueden ser utilizados como piezas en un juego de mesa.
- **Botones virtuales** que permiten definir botones como regiones rectangulares en una imagen objetivo.

Atributos de una Imagen Objetivo

La imagen objetivo debe contar con los siguientes atributos, para mejorar la eficiencia de la detección de la misma por medio del dispositivo:

- Rica en detalles, por ejemplo, la escena de una calle, un grupo de personas, *collages* y mezclas de elementos, escenarios deportivos, etc.
- Buen contraste, es decir, las regiones brillantes y oscuras que estén bien iluminadas.
- No hay patrones repetitivos, es decir, un campo de hierba, la fachada de la casa moderna con ventanas idénticas, un tablero de ajedrez.

Debe ser de 8 o 24-bits en formato PNG o JPG; menor de 2 MB de tamaño; JPEG deben ser RGB o escala de grises.

Características Naturales y Evaluación

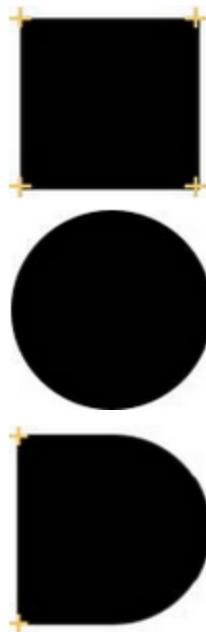
El framework de Vuforia asigna calificaciones a las imágenes que se procesan, por ejemplo: una buena calificación define qué tan bien una imagen puede ser detectada y rastreada con el SDK Vuforia. Esta calificación se muestra en el *Administrador de Target* del sitio Web, ver Figura 5.2, donde se procesan las imágenes capturadas en las tareas 1 y 3.

Target Name	Type	Rating	Status	Date Modified
 obj01_elemento03_vista04	Single Image	★★★★★	Active	Apr 20, 2016 11:54
 obj01_elemento03_vista03	Single Image	★★★★★	Active	Apr 20, 2016 11:54
 obj01_elemento03_vista02	Single Image	★★★★★	Active	Apr 20, 2016 11:53
 obj01_elemento03_vista01	Single Image	★★★★★	Active	Apr 20, 2016 11:53
 obj01_elemento02_vista04	Single Image	★★★★★	Active	Apr 20, 2016 11:52
 obj01_elemento02_vista03	Single Image	★★★★★	Active	Apr 20, 2016 11:51
 obj01_elemento02_vista02	Single Image	★★★★★	Active	Apr 20, 2016 11:51
 obj01_elemento02_vista01	Single Image	★★★★★	Active	Apr 20, 2016 11:50
 obj01_elemento01_vista04	Single Image	★★★★★	Active	Apr 20, 2016 11:36

Figura 5.2 Calificaciones del SDK Vuforia.

La calificación asignada por el SDK Vuforia puede variar de 0 a 5 para cualquier imagen procesada. Cuanto mayor sea la calificación de la imagen, más efectiva es la detección y seguimiento. Una calificación de cero indica que a la imagen no se le realiza un seguimiento en todo el sistema de realidad aumentada, mientras que un número de estrellas de 5 indica que una imagen es fácil de rastrear por el sistema de realidad aumentada.

El analizador de imágenes representa las características en cruces amarillas como se puede ver en la Figura 5.3, al aumentar el número de estos detalles en una imagen, se comprueba que los detalles crean un patrón que no se repite ver Figura 5.4.



Un cuadrado tiene cuatro funciones para cada uno de sus rincones.

Un círculo no contiene características ya que no contiene detalles nítidos o cincelados.

Este objeto contiene sólo dos características de cada esquina aguda.

Figura 5.3 Características en una imagen.

Nota: De acuerdo con la definición de una función las esquinas suaves y los *bordes orgánicos* no están marcados como características.

Una de las limitantes que presenta el SDK de Vuforia es no poder detectar imágenes que contienen contornos redondeados ya que estos no representan características suficientes para poder ser reconocidas, es importante el poder evitar en lo posible imágenes con este tipo de característica.



Figura 5.4 Detalles de una imagen.

Teniendo en cuenta las características y propiedades para la identificación de las vistas con el framework Vuforia, el siguiente proceso es la creación de la base de datos para subir las imágenes de las vistas seleccionadas (Vuforia, 2016).

5.1.1 Creación de la base de datos

Para la creación de la base de datos desde el portal de Vuforia se accede desde la siguiente liga: <https://developer.vuforia.com/license-manager> en la Figura 5.5 se muestra una pantalla del sitio web Vuforia.

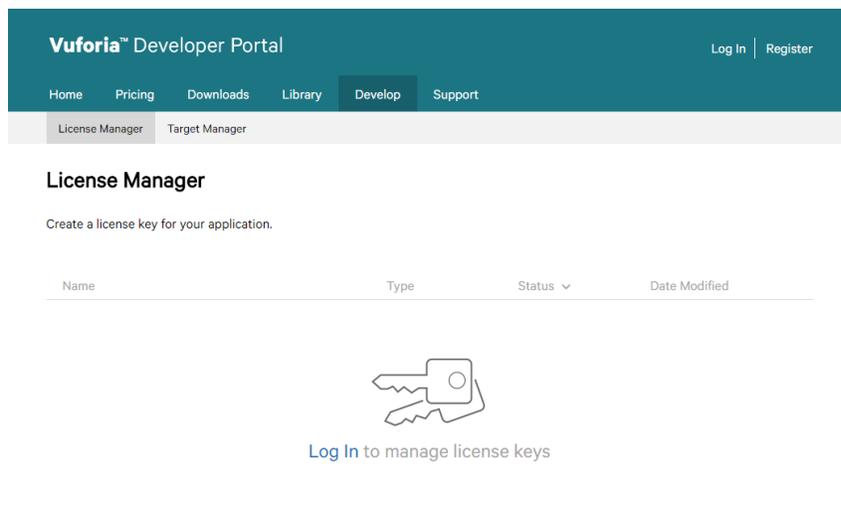


Figura 5.5 Pantalla del sitio web de Vuforia.

Se debe iniciar sesión para poder crear la base de datos, se inicia sesión en el portal con la siguiente credencial, correo: pruebascentid@gmail.com contraseña: [Basix123](#), ver Figura 5.6.

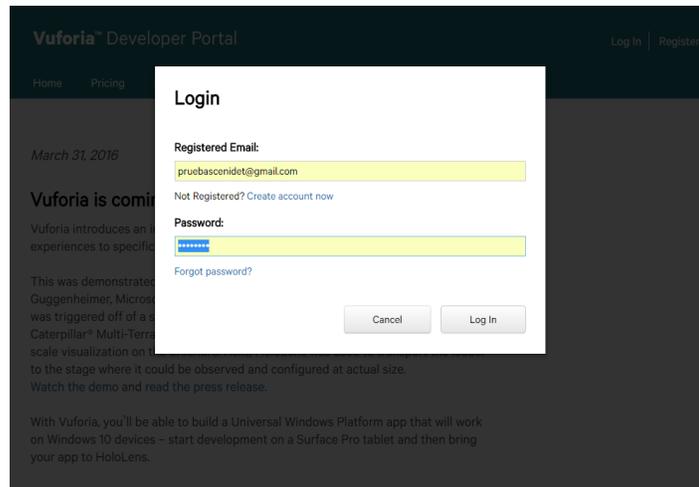


Figura 5.6 Iniciar sección en el sitio web de Vuforia.

Para crear la base de datos se siguen los siguientes pasos:

- 1.- Acceder a la siguiente pestaña *Develop/Target Manager* y seleccionar el botón “*Add Database*” ver Figura 5.7.

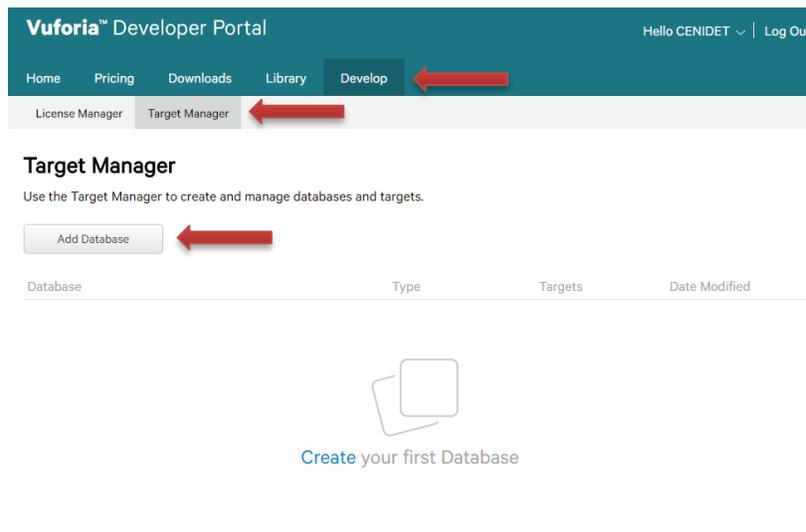


Figura 5.7 Pantalla donde se crean y visualizan las bases de datos.

- 2.- Especificar el nombre de la base de datos, el cual deberá seguir la siguiente nomenclatura y se dejará la opción “*Device*” que se encuentra por default, ver Figura 5.8:

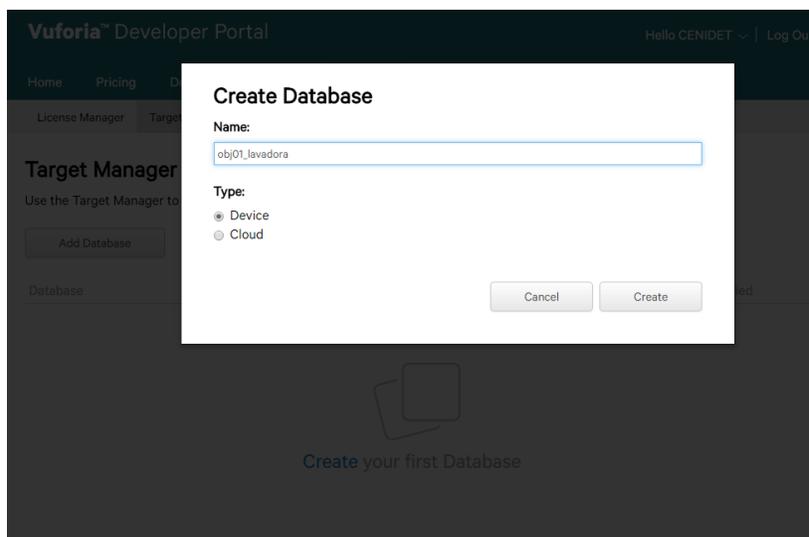


Figura 5.8 Ventana para especificar el nombre y tipo de base de datos que se desea crear.

Para definir una nomenclatura, el nombre debe iniciar con el prefijo “**obj**”, seguido de un número consecutivo 01, 02... (el número depende de los registros previos de bases de datos que existan registradas en la cuenta del usuario), después del número sigue un guion bajo “_” junto con el nombre que el usuario decida asignar a la base de datos, es recomendable que el nombre asignado a la base de datos se relacione con el objeto que se está caracterizando. Por ejemplo, para una base de datos que contendrá la información de una lavadora, el nombre que llevará quedaría de la siguiente forma:

obj01_lavadora

Una vez creada la base de datos se inicia la carga de imágenes objetivo que serán utilizadas para proporcionar información contextual en la aplicación de realidad aumentada, para poder agregar las imágenes se debe dar clic en el botón “*Add Target*” ver Figura 5.8.

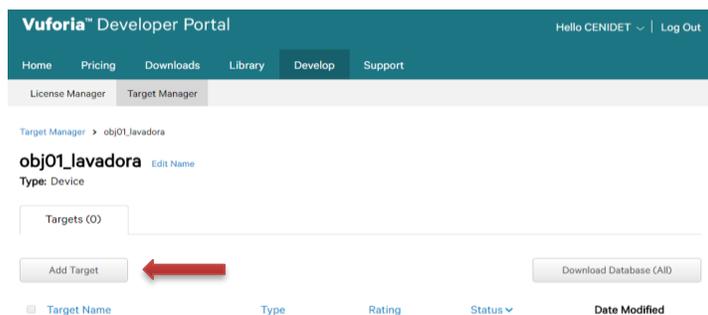


Figura 5.9 Ventana para ver y crear unos Target.

Después de dar clic en el botón aparece una ventana para añadir el Target (imagen), ver Figura 5.10

- En la opción “*Type*” se deja la opción que viene por default “*Single Image*”
- Seleccionar la ubicación donde se encuentra el archivo, en la computadora local.
- En la opción “*Width*” se asigna el valor: 711 ya que con esta medida la imagen no oculta los elementos virtuales que se superponen en la imagen.
- Por último, en la opción “*Name*” se deben tener en cuenta las consideraciones que se detallan en la siguiente sección.

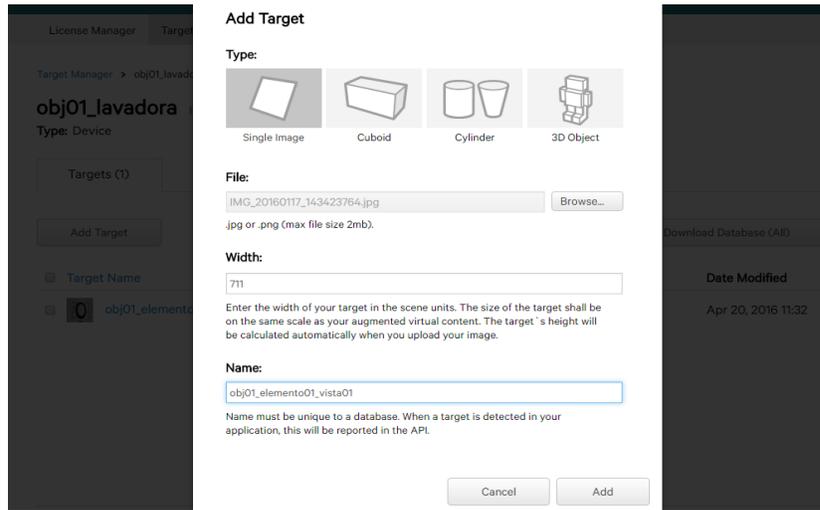


Figura 5.10 Ventana para especificar propiedades del nuevo Target.

Nombre de los target

La nomenclatura del nombre del target que será cargado en la base de datos deber hacer referencia al objeto (la base de datos en la cual se puede cargar información), al elemento dentro de la vista principal y a la vista del objeto identificado dentro de ese elemento. Por ejemplo, para la vista 01 de la lavadora (vista superior, según el estándar americano), se identificaron 3 elementos que pueden proporcionar información contextual (marca, control de lavado y control de ciclos) en cada uno de estos elementos están incluidos botones que proporcionan información y que serán cargados a la base de datos.

En la Figura 5.11 se puede observar la vista frontal del obj01 (Base de datos lavadora). Cabe aclarar que esta imagen no se va a guardar ya que por sí sola no brinda ningún tipo de

información, lo relevante de esta imagen son los elementos que se han identificado y que se han desagregado.



Figura 5.11 Vista frontal de la lavadora.

En la figura 5.12 se identifican tres elementos Figura 5.12a, perilla de control, Figura 5.12b, logo de la marca y Figura 5.12c panel de control, que pueden proporcionar información contextual al usuario.



(a) elemento01.



(b) elemento02.



(c) elemento03.

Figura 5.12 Elementos desagregados de la imagen del panel de control.

A las tres imágenes de la Figura 5.12 se les aplican los pasos 1 y 2 de la metodología, es decir, para cada uno de estos se aplica el estándar americano y se obtienen imágenes de las mismas. Por ejemplo, para el *elemento01* se obtuvieron 4 vistas.

Para cargar las imágenes en Qualcomm Vuforia es necesario asignar un nombre que esté relacionado con la información que va a proporcionar y que es necesario para que posteriormente se pueda interactuar con la aplicación Android “SistemaAR”. Por lo tanto, se debe utilizar la siguiente nomenclatura:

El nombre de la imagen debe iniciar con el prefijo “obj”, seguido de un número consecutivo 01, 02,.. (haciendo referencia a la base de datos que se cargó con anterioridad, para este ejemplo sería obj01) seguido del símbolo especial “_”, se continúa con la palabra “elemento” seguida del número consecutivo 01,02, ..., *n* dependiendo del número de elementos que se identificaron. Después debe llevar el símbolo “_” y por último, la palabra “vista” seguido del número de vista a la que se haga referencia. Siguiendo estas indicaciones un nombre aceptable podría ser: *obj01_elemento01_vista01*

A continuación, en la tabla 5.1 se puede observar el nombre con el cual serán cargadas en el framework.

Tabla 5.1 Presentación de Target con su nombre correspondiente.

Nombre	Imagen
obj01_elemento01_vista01	
obj01_elemento01_vista02	
obj01_elemento01_vista03	
obj01_elemento01_vista04	

Una vez terminado el proceso de carga de imágenes, el SDK Vuforia muestra una lista de los targets agregados y la calificación asignada a cada una de las imágenes, en este punto es recomendable que las imágenes registradas cumplan con un mínimo de calificación de cuatro para garantizar que la aplicación Android las identifique y les dé seguimiento, la lista se muestra en la Figura 5.13.

obj01_lavadora [Edit Name](#)
Type: Device

Targets (12)

[Add Target](#) [Download Database \(All\)](#)

Target Name	Type	Rating	Status	Date Modified
 obj01_elemento01_vista01	Single Image	★★★★☆	Active	Apr 20, 2016 11:32
 obj01_elemento01_vista02	Single Image	★★★★☆	Active	Apr 20, 2016 11:34
 obj01_elemento01_vista03	Single Image	★★★★☆	Active	Apr 20, 2016 11:35
 obj01_elemento01_vista04	Single Image	★★★★☆	Active	Apr 20, 2016 11:36
 obj01_elemento02_vista01	Single Image	★★★★☆	Active	Apr 20, 2016 11:50
 obj01_elemento02_vista02	Single Image	★★★★☆	Active	Apr 20, 2016 11:51
 obj01_elemento02_vista03	Single Image	★★★★☆	Active	Apr 20, 2016 11:51
 obj01_elemento02_vista04	Single Image	★★★★☆	Active	Apr 20, 2016 11:52
 obj01_elemento03_vista01	Single Image	★★★★☆	Active	Apr 20, 2016 11:53
 obj01_elemento03_vista02	Single Image	★★★☆☆	Active	Apr 20, 2016 11:53
 obj01_elemento03_vista03	Single Image	★★★★☆	Active	Apr 20, 2016 11:54
 obj01_elemento03_vista04	Single Image	★★★☆☆	Active	Apr 20, 2016 11:54

Figura 5.13 Lista de los Target en la base de datos.

En la siguiente Figura se puede apreciar los patrones (puntos amarillos) que vuforia genera para la identificación de los elementos.

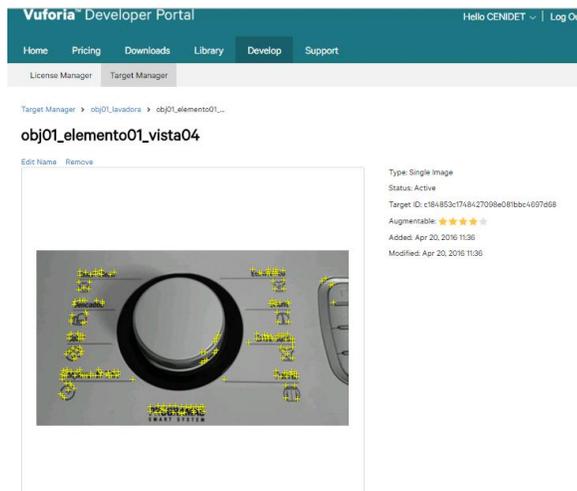


Figura 5.14 Patrones identificados por el framework Vuforia.

Al terminar de cargar todos los targets es necesario descargar los archivos que contienen los patrones de imágenes que se procesaron con el SDK de Vuforia, estos archivos los utiliza la aplicación Android para que el reconocimiento de las vistas del objeto que se caracterizó sea exitoso. La descarga se realiza como se puede ver en la Figura 5.15, Seleccionando la opción “Download DataBase” y posteriormente se selecciona la opción “SDK (Eclipse, Ant, Xcode, etc)”. Se descarga un archivo .zip, al descomprimirlo se obtienen dos archivos uno con extensión .bat y otro con extensión .xml como se muestra en la Figura 5.16.

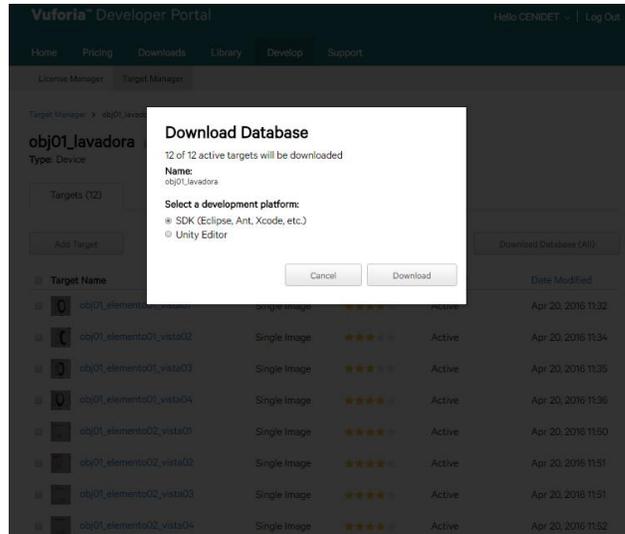


Figura 5.15 Descarga de la base de datos de Vuforia.

Nombre	Fecha de modifica...	Tipo	Tamaño
 obj01_lavadora.dat	20/04/16 16:56	KMP - MPEG Mov...	315 KB
 obj01_lavadora.xml	20/04/16 16:56	Archivo XML	2 KB

Figura 5.16 Archivos descargados de Vuforia.

Proceso 5. Información contextual de los objetos

El último proceso de la Fase 1 de la metodología consiste en asociar los elementos de información contextual con los elementos del objeto definido anteriormente, este proceso se hace mediante una aplicación de escritorio.

La aplicación “Asociación” es una aplicación que como su nombre lo indica permite al usuario asociar las vistas de cada uno de los elementos con la información contextual (imágenes que

muestran el funcionamiento del elemento detectado y los nombres de dichos elementos) que se podrán visualizar posteriormente. En la Figura 5.17 se muestra la pantalla principal de la aplicación.

Para obtener la aplicación el usuario deberá ingresar al siguiente link para descargar el archivo: <http://synapciainfo.com/ServicioCenidet/index.php?r=Servicios/Asociacion>

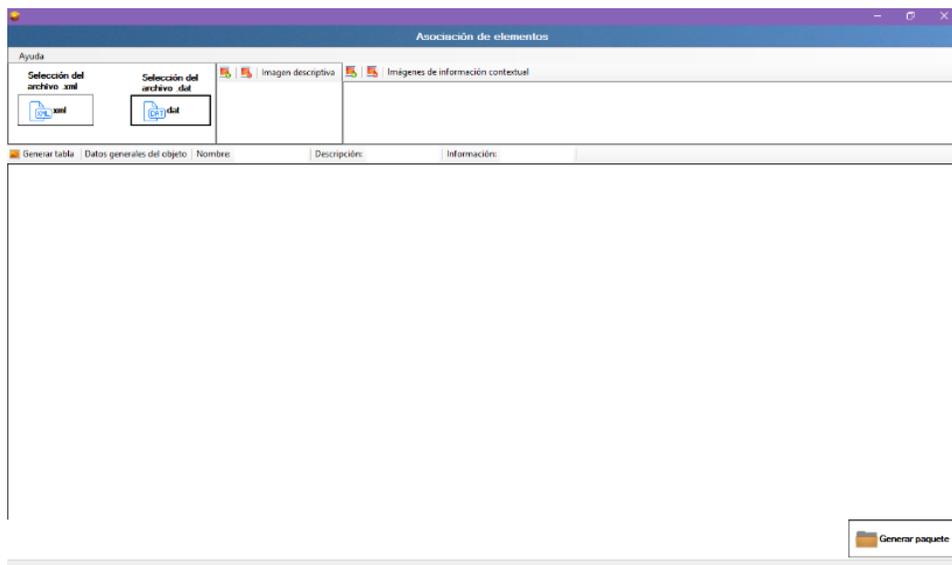


Figura 5.17 Aplicación de escritorio para la asociación de elementos con la información contextual.

La pantalla principal cuenta con un menú de ayuda que consta de dos opciones, ver Figura 5.18.

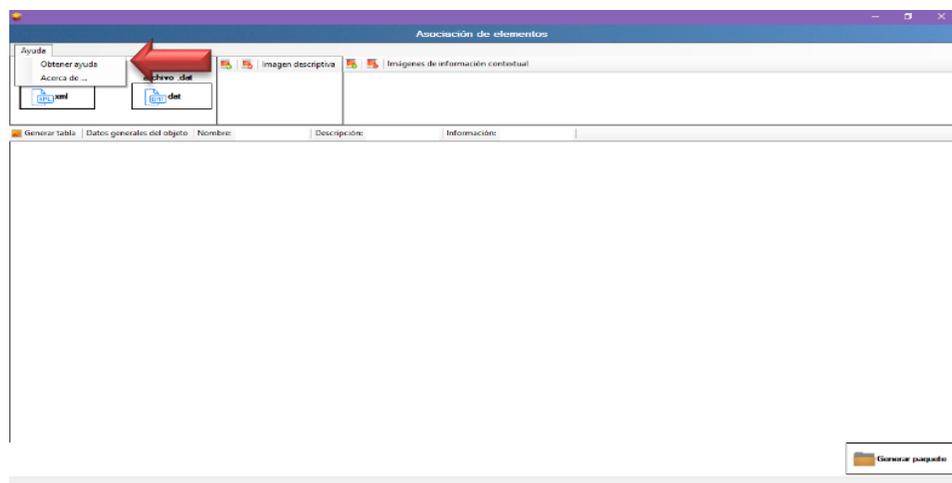


Figura 5.18 Menú de opciones.

En la opción “Obtener ayuda” se puede consultar el manual de usuario de la aplicación.

En la opción “Acerca de...” se muestra una pantalla con información de la aplicación, esa pantalla se puede ver en la Figura 5.19.



Figura 5.19 Información acerca de la aplicación.

A continuación, se describen los pasos a seguir para realizar la asociación del objeto “lavadora”:

1. Cargar el archivo .xml que se descargó del portal de Vuforia.

- 1.1. Para cargar el archivo .xml el usuario debe dar clic al botón “xml” el cual abre una ventana en donde se debe buscar y seleccionar el archivo, ver figura 5.20.

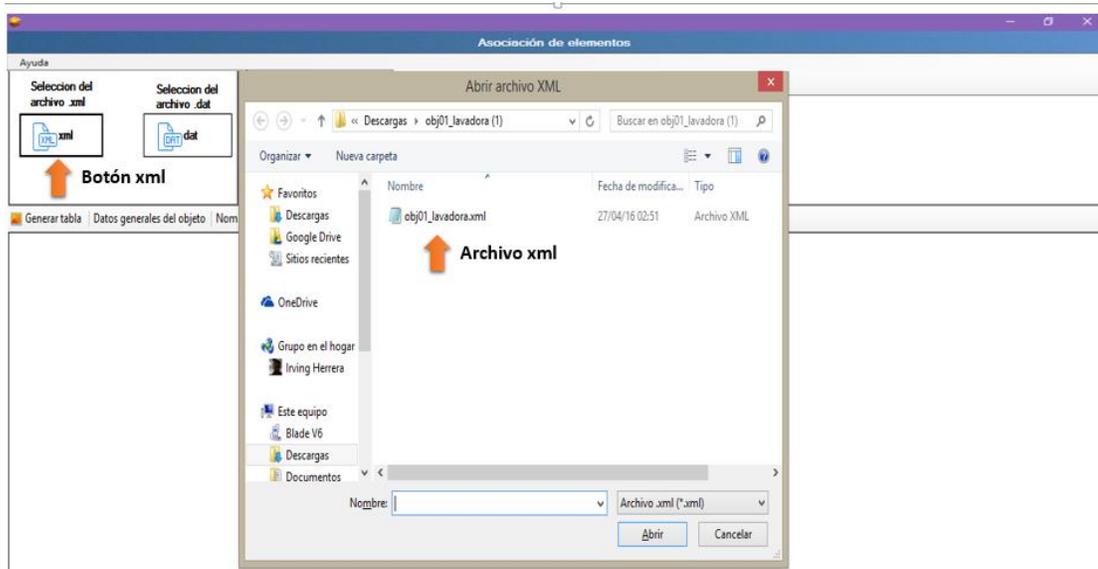


Figura 5.20 Selección del archivo xml.

1.2. Una vez que se ha cargado el archivo aparecerá el nombre debajo del botón “*xml*” como se puede ver en la Figura 5.21.

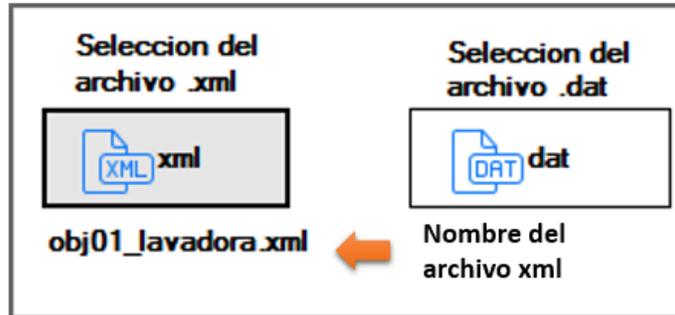


Figura 5.21 Nombre del archivo xml cargado.

2. Cargar el archivo .dat que se descargó del portal de Vuforia.

2.1. Para cargar el archivo .xml se realiza dando clic al botón “*dat*” el cual abre una ventana en donde se debe buscar y seleccionar el archivo ver Figura 5.22.

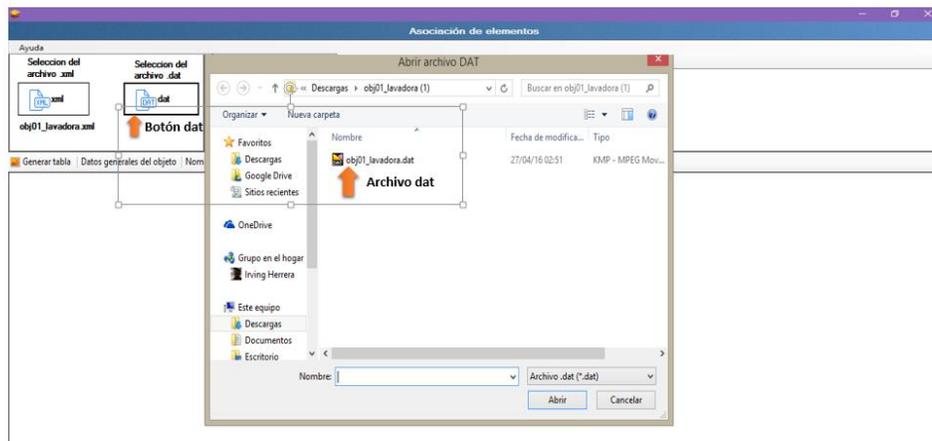


Figura 5.22 Búsqueda y selección del archivo dat.

2.2. Una vez que se ha cargado el archivo aparecerá el nombre debajo del botón “dat” como se puede ver en la Figura 5.23.



Figura 5.23 Nombre del archivo dat cargado.

3. Imagen representativa del objeto multivista caracterizado, en este caso el objeto “lavadora”. El usuario podrá elegir la imagen que crea conveniente, no es necesario que el usuario siga alguna nomenclatura para establecer el nombre de la imagen ni tampoco que establezca el tamaño de la misma. Únicamente debe tomar en consideración que la imagen cargada debe tener la extensión .jpg.

3.1. Se puede cargar y eliminar la imagen seleccionada con los botones que se ven en la figura siguiente.

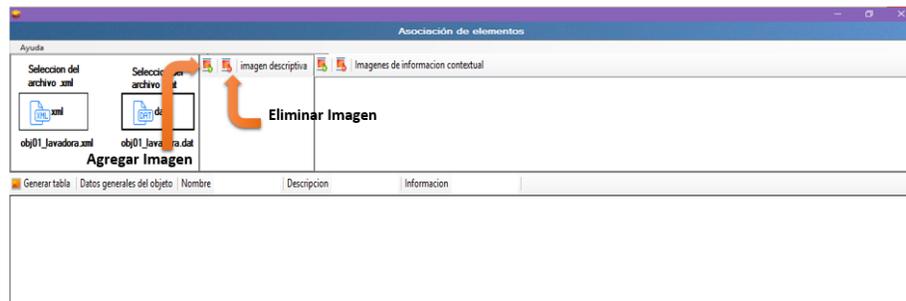


Figura 5.24 Botones para agregar y eliminar la imagen descriptiva.

3.2. Para cargar la imagen debe dar clic en el botón “Agregar imagen” el cual abre una ventana para la búsqueda de la imagen como se puede ver en la siguiente figura.

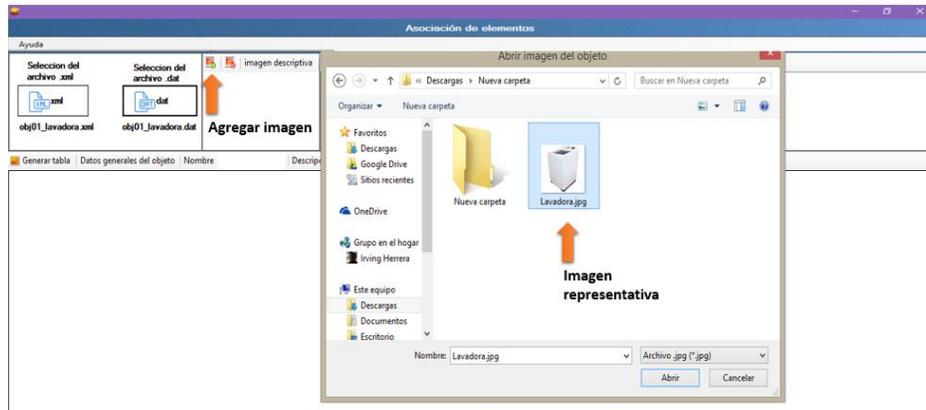


Figura 5.25 Selección de la imagen representativa.

3.3. Una vez que se carga la imagen a la aplicación se puede ver una vista previa de la imagen seleccionada ver Figura 5.26.

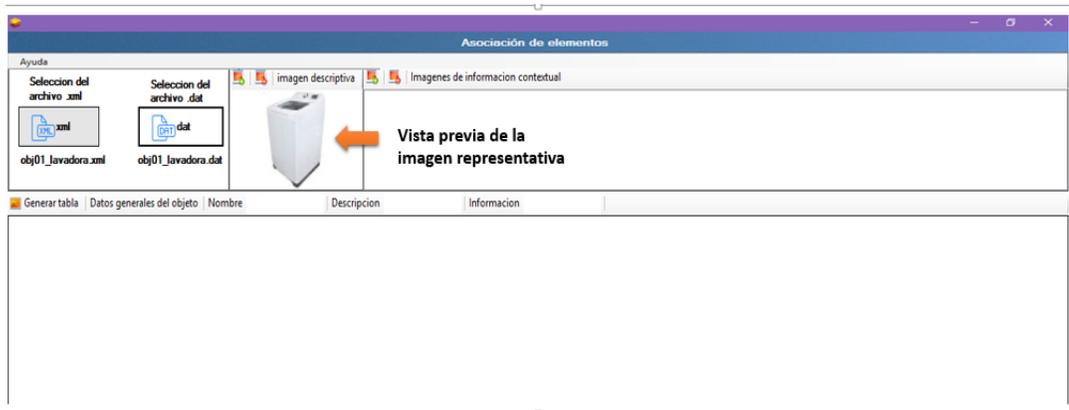


Figura 5.26 Vista previa de la imagen representativa.

4. Cargar imágenes que van a proporcionar información contextual al usuario final, las imágenes deben ser en formato .jpg, se deberán elegir las imágenes que se consideren convenientes, se cargarán tantas imágenes como elementos se detectaron del objeto (ej. en el caso de la lavadora se detectaron tres elementos, se deberá subir una imagen de información contextual por cada elemento). Para estas imágenes que proporcionarán información contextual no se requiere que el usuario defina un tamaño como lo hizo para imágenes de las vistas de los elementos.

Las imágenes deben tener un nombre antes de cargarlas, el nombre de las imágenes debe estar relacionado con los elementos detectados y deberá seguir la siguiente nomenclatura.

Deberá iniciar con el prefijo “obj”, seguido de un número consecutivo 01,02,.. (haciendo referencia a la base de datos que ya se cargó con anterioridad, en este caso sería obj01) y el símbolo especial “_”, continua con la palabra “elemento” seguida del número consecutivo 01,02, ...,n dependiendo del número de elementos que se hayan identificado. Después deberá llevar el símbolo “_” y por último, el nombre de la base de datos de Vuforia. Siguiendo estas indicaciones un nombre aceptable podría ser: *obj01_elemento01_obj01_lavadora.jpg*

4.1. Se pueden cargar y eliminar las imágenes de información contextual con los botones que se muestran en la Figura 5.27.

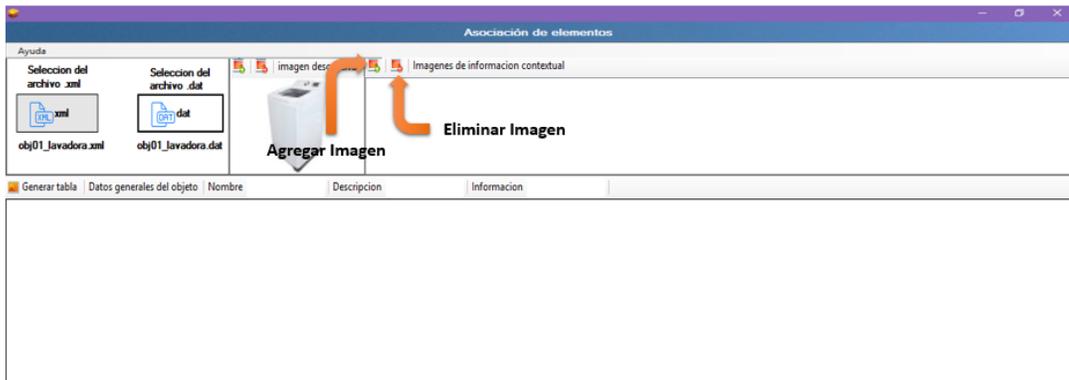


Figura 5.27 Botones para agregar y eliminar las imágenes de información contextual.

4.2. Para cargar la imagen se da clic en el botón “Agregar imagen” el cual abre una ventana para la búsqueda de la imagen, se puede observar que la imagen ya tiene un nombre que ha sido definido mediante la nomenclatura antes mencionada como se puede ver en la Figura 5.28.

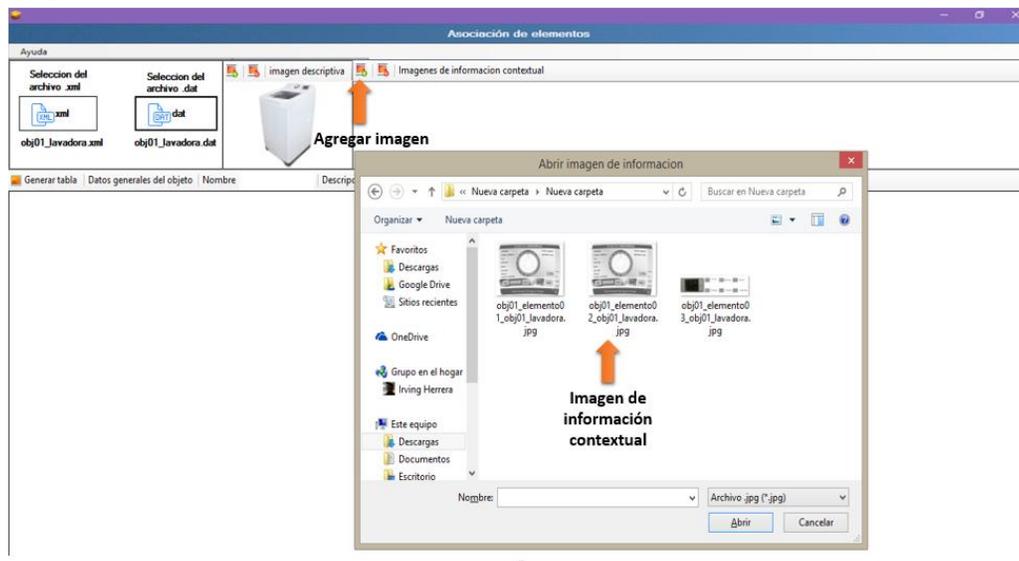


Figura 5.28 Selección de la imagen de información contextual.

4.3. Una vez que se cargaron todas las imágenes, se pueden visualizar tal como se muestra en la figura 5.29.

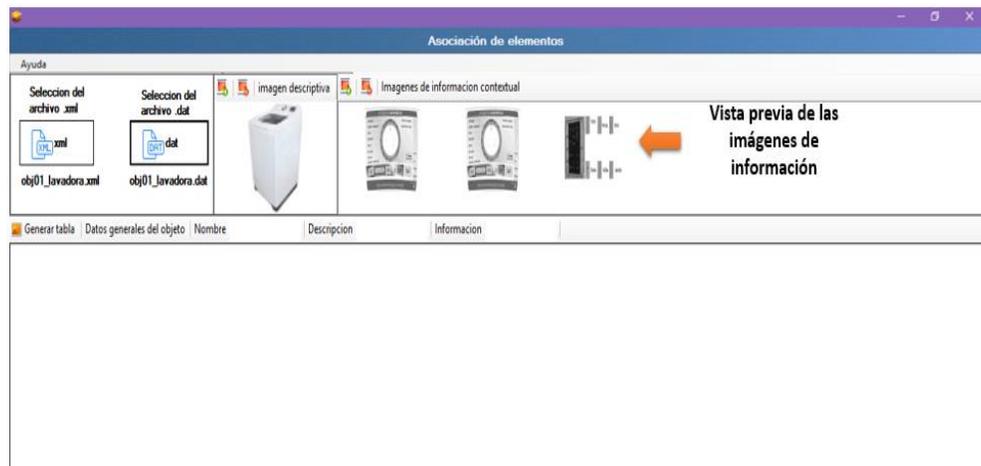


Figura 5.29 Vista previa de las imágenes de información contextual.

5. Generar tabla para la asociación de información contextual.

5.1. Para la generación de la tabla se da clic al botón “*Generar tabla*” como se muestra en la siguiente figura.

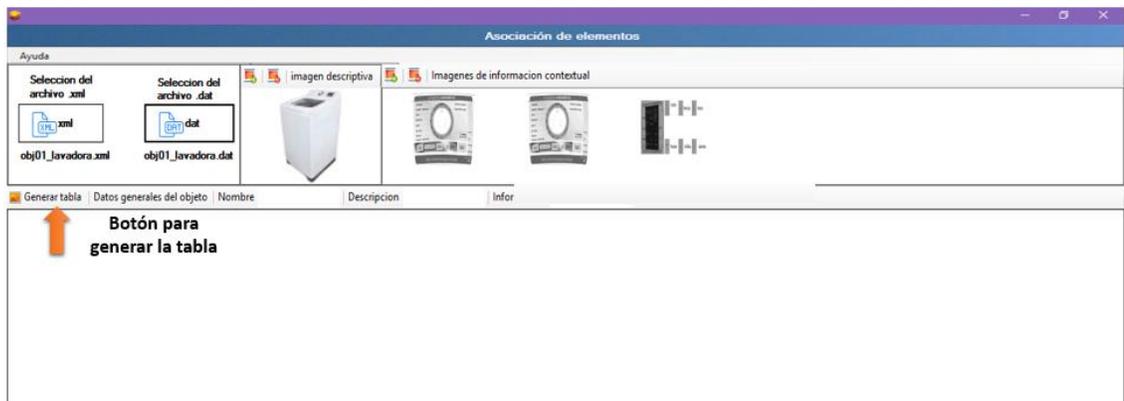


Figura 5.30 Botón para la generación de la tabla.

Cuando el usuario presiona el botón “Generar tabla”, se genera una tabla de forma automática que contiene tres columnas, las cuales se muestran en la Figura 5.31:

- Nombre: Es el nombre de las imágenes que se subieron al portal web de Vuforia.
- Descripción: Es una breve descripción del elemento, el usuario llena manualmente esta columna bajo su propio criterio.
- Información: En esta columna se muestra una lista con los nombres de las imágenes que se cargaron en el punto 4 como se puede ver en la Figura 5.32.

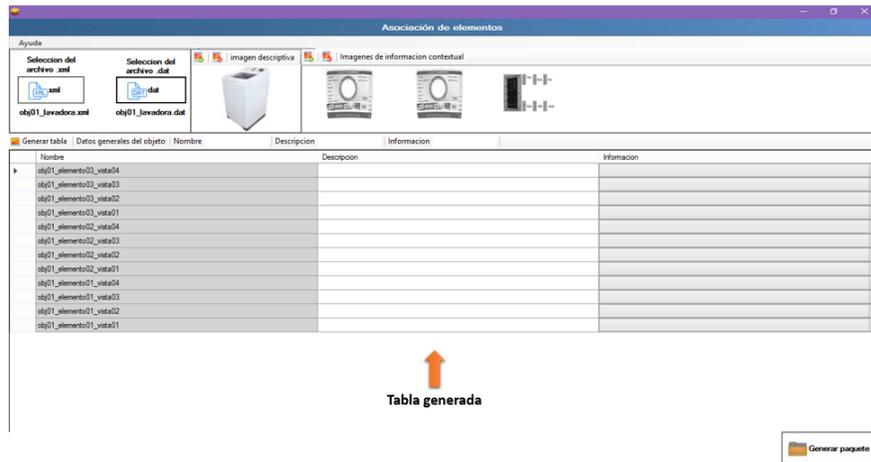


Figura 5.31 Tabla generada para realizar la asociación de elementos.



Figura 5.32 Lista con los nombres de las imágenes de información contextual.

5.2. De igual manera se llenan los siguientes campos de texto los cuales se describen a continuación y que hacen referencia al objeto de interés, ver en la Figura 5.33.

- Nombre: En esta caja de texto el usuario proporciona el nombre del objeto que se analizó, en este caso el objeto “lavadora”.
- Descripción: En esta caja de texto el usuario proporcionar una descripción breve sobre el objeto analizado, en este caso el objeto “lavadora”.
- Información: En esta caja de texto el usuario proporciona información detallada sobre el objeto que se analizó, en este caso el objeto “lavadora”

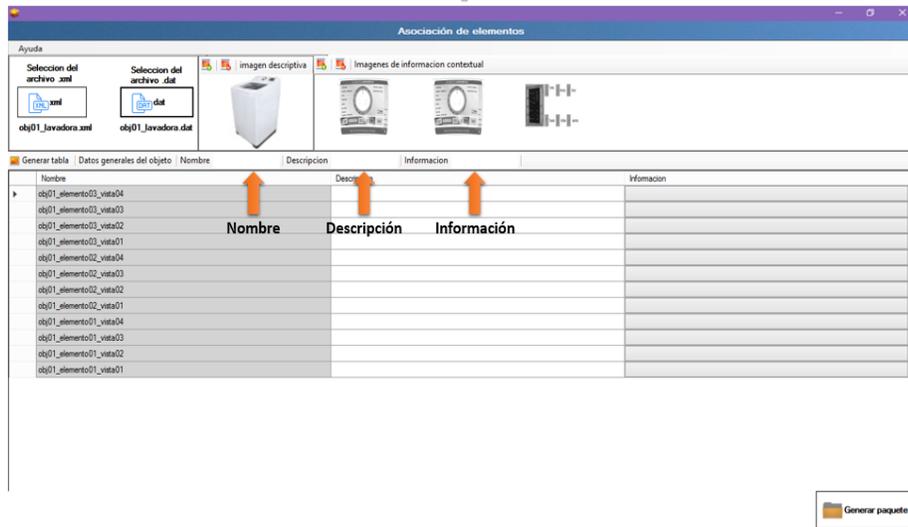


Figura 5.33 Campos de texto.

6. Generación del directorio que contiene los archivos de asociación.

6.1. Para generar el directorio del paquete se da clic en el botón “Generar paquete”, se abre una ventana para seleccionar la ruta donde se guardará el directorio de asociación como se ve en la siguiente figura.



Figura 5.34 Guardar paquete de asociación.

El directorio que se generó está estructurado de la siguiente manera, la carpeta principal tiene el nombre de “*paquete*”, dentro de ella se encuentran las siguientes carpetas, ver Figura 5.35.

Nombre	Fecha de modifica...	Tipo	Tamaño
asociacion	03/05/16 00:47	Carpeta de archivos	
dat	03/05/16 00:47	Carpeta de archivos	
imagen	03/05/16 00:47	Carpeta de archivos	
informacion	03/05/16 00:47	Carpeta de archivos	
xml	03/05/16 00:47	Carpeta de archivos	

Figura 5.35 Directorio del paquete generado con la aplicación de asociación.

7. Subir paquete generado al servidor

Una vez que se realizó la asociación con la aplicación de escritorio y se obtiene el directorio con sus respectivos archivos se subirán al servidor FTP, las credenciales para subir los archivos al servidor son:

Dominio: synapciainfo.com

Usuario de FTP: synapci2

Password del FTP: eGd4iW1g64

Para poder realizar la transferencia de archivos es necesario utilizar un cliente FTP, para poder dar continuidad al ejemplo que se está proporcionando, se utilizó el cliente multiplataforma *FileZilla*.

Una vez que ha ingresado al servidor FTP se debe ingresar a la ruta del proyecto:

/public_html/ServicioCenidet

En la siguiente ruta se encuentran las mismas carpetas que se generaron con la aplicación de escritorio, se deben subir los archivos de cada carpeta del directorio generado a las carpetas correspondientes del servidor FTP ver Figura 5.36:

/public_html/ServicioCenidet/Descargas/



Figura 5.36 Pantalla del servidor FTP.

5.2 Diseño e implementación de la fase 2

En esta segunda fase se desarrolló un prototipo en Android en el cual se implementó la fase 1, para la realización del primer prototipo se utilizaron las siguientes herramientas que tienen soporte para el sistema operativo móvil Android ver Figura 5.37.

- *SDK de Qualcomm Vuforia*
- *Unity*
- *SQLite*



Figura 5.37 Herramientas utilizadas.

5.2.1 SDK de Qualcomm Vuforia

El SDK de Vuforia, ver figura 5.38, se utilizó para implementar la funcionalidad de realidad aumentada, este SDK utiliza tecnología de visión por computadora para reconocer y rastrear imágenes planas y objetos 3D en tiempo real, cuenta con un sistema web para el almacenamiento de las imágenes y objetos que se requiere reconocer (Vuforia, 2016).



Figura 5.38 Logo Vuforia

5.2.2 Unity

Para el entorno de desarrollo y creación de los elementos virtuales de información se utilizó la herramienta Unity. Unity es un motor de videojuego multiplataforma creado por *Unity Technologies*, esta herramienta es muy apropiada ya que soporta Android, iOS, entre otros sistemas operativos móviles lo que lo hace funcional y multiplataforma además que Vuforia cuenta con un *plugin* para combinar estas herramientas simplificando el trabajo de hacer uso de

elementos virtuales (Unity, 2016). Para la presentación de la información contextual y la interacción de elementos virtuales, se implementaron dos botones que se pueden ver en la Figura 5.39; la Figura 5.39a representa un botón que da una pequeña descripción del objeto, la Figura 5.39b representa un botón que presenta información detallada sobre el objeto.

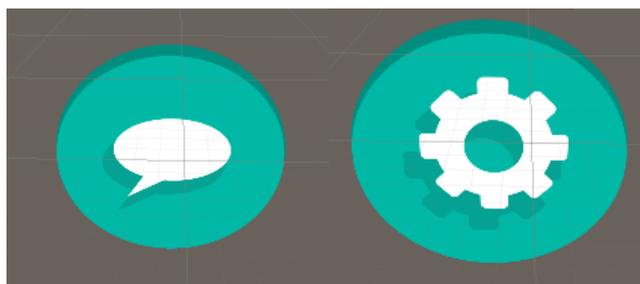


Figura 5.39 a) Botón de descripción, b) Botón información detallada.

Para representación información contextual del elemento se diseñó un elemento virtual, como se muestra en la Figura 5.40 que representa una etiqueta sobre el elemento reconocido.



Figura 5.40 Elemento virtual para la representación de etiquetas.

En la Figura 5.41 se puede ver el elemento virtual que representa información detallada sobre el objeto, es una ventana, la cual proporciona un desplazamiento horizontal para mover la información disponible del elemento.



Figura 5.41 Elemento virtual de información detallada.

5.2.3 Yii framework

Yii es un framework PHP basado en componentes de alto rendimiento para desarrollar aplicaciones Web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo (Framework, 2015), esta herramienta se utilizó para el desarrollo de servicios web para comunicar la aplicación con el servidor que contiene la información de los objetos caracterizados.

Se desarrolló una técnica para lograr dotar la aplicación de nueva información representando así una aplicación dinámica y que no sea necesario rehacer la aplicación para que pueda detectar elementos de un nuevo objeto caracterizado, con lo cual se implementaron servicios web para realizar la descarga de los paquetes de los nuevos objetos caracterizados y poder así añadirle nuevas características a la aplicación permitiendo detectar los elementos de un nuevo objeto, en la Figura 5.42 se puede observar de manera gráfica el proceso para agregar nueva información a la aplicación.

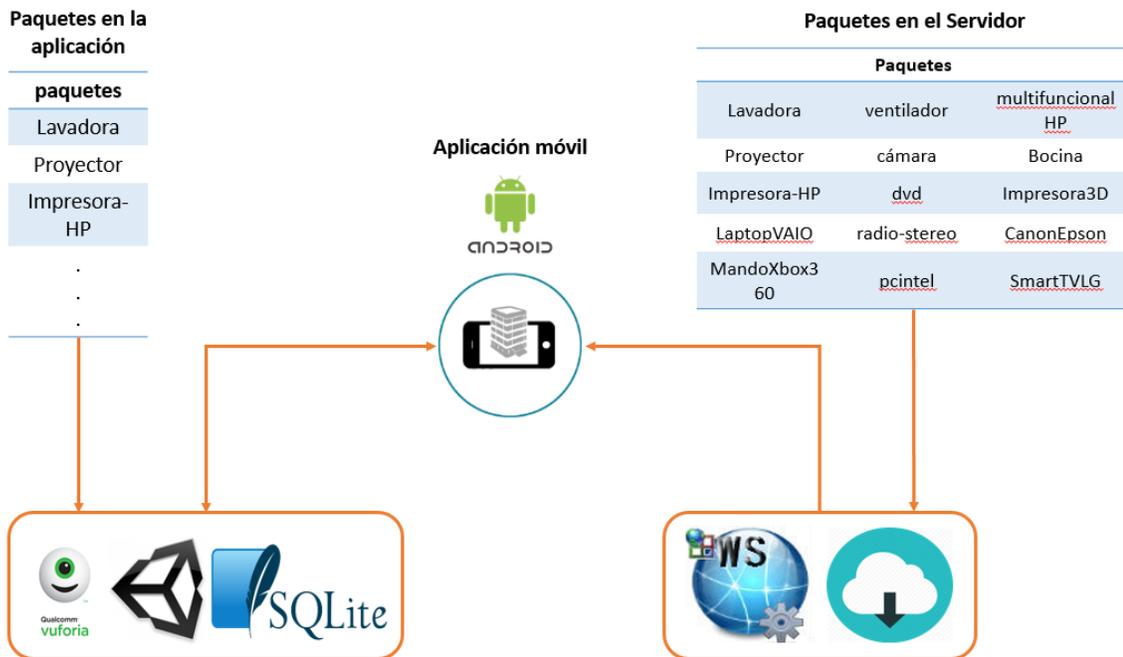


Figura 5.42 Proceso para agregar paquetes de información a la aplicación.

La aplicación inicialmente tiene cargado el SDK de Vuforia, una base de datos en SQLite, los botones virtuales que se pueden ver en la Figura 5.39 así como los elementos virtuales que proporcionan información que se pueden ver en las imágenes 5.40 y 5.41 y las pantallas de internación con el usuario, la funcionalidad para detectar los objetos caracterizados en la fase 1 de esta metodología se agrega dinámicamente descargando un paquete que contiene la información generada.

El paquete descargado contiene la siguiente información:

- Los archivos descargados de Vuforia (archivos con extensión .xml y .dat), que contienen la información para poder detectar el nuevo objeto.
- Las imágenes de información contextual.
- Un archivo de texto plano en formato JSON, que contiene la asociación de los descriptores de texto y las imágenes de información contextual con la información generada en el portal de Vuforia como se explicó en la fase 1 de esta metodología.

Para agregar la nueva información que se descargó a la aplicación, internamente en la aplicación de realizan los siguientes pasos.

- 1.- Se guardan en la memoria del teléfono las imágenes de información contextual, exactamente en la carpeta de instalación de la aplicación.
- 2.- Se guardan los archivos .xml y .dat y posteriormente se cargan a la memoria de la aplicación la información de los dos archivos.
- 3.- Se descompone el JSON para guardarlo en la base de datos de SQLite, el JSON está formado de la siguiente manera:

```

"ElementoAR":{
  "id_elemento":"0",
  "elemento_padre":"Proyector",
  "informacion":"obj02_elemento02.jpg",
  "etiqueta":"Panel de control",
  "nombre":"obj02_elemento01_vista02"},
    
```

Donde:

ElementoAR: Es un elemento correspondiente a una de las vistas subidas al portal de Vuforia, habrá tantos elementos como imágenes se subieron al portal de Vuforia.

id_elemento: Representa el identificador del elemento AR.

elemento_padre: Se le asigna el nombre que corresponde al objeto caracterizado.

Información: Contiene el nombre de la imagen que se representará mediante elementos virtuales y que proporcionará información al usuario.

etiqueta: Corresponde al descriptor de texto que representa una descripción breve sobre el elemento.

nombre: Contiene el nombre de la vista del objeto caracterizado.

Proceso para cargar nuevos patrones

Vuforia proporciona tres clases principales para realizar la carga de nuevos patrones en tiempo de ejecución, las clases son las siguientes:

- TrackerManager
- ObjectTracker
- DataSet

Anteriormente se mencionó que se descarga un paquete que contiene esta información, la información de los nuevos patrones está incluida específicamente en los archivos que proporciona el portal de Vuforia que son el archivo .xml y .dat. estos archivos son guardados en la carpeta de instalación de la aplicación.

Para realizar la carga de los nuevos patrones se sigue los siguientes pasos:

- 1.- Comprobar que los archivos existen en la ubicación de almacenamiento dado:
`DataSet.Exists(string path, DataSet.StorageType storageType);`
- 2.- Solicitar una instancia de ObjectTracker TrackerManager
`TrackerManager:TrackerManager.Instance.GetTracker<ObjectTracker>();`
- 3.- Crear una nueva instancia de conjunto de datos
`ObjectTracker.CreateDataSet();`
- 4.- Cargar un archivo de conjunto de datos a partir de su lugar de almacenamiento:
`DataSet.Load (string ruta, DataSet.StorageType StorageType);`
- 5.- Utilizar la instancia objectTracker para activar el conjunto de datos:
`ObjectTracker.ActivateDataSet(DataSet dataset);`

5.2.4 SQLite

Esta herramienta se utiliza para asociar los elementos que detecta Vuforia con su información contextual. SQLite es una biblioteca en proceso de implementación autónoma, sin servidor, cero configuraciones, transaccional, motor de base de datos SQL. A diferencia de la mayoría de las otras bases de datos, SQLite no tiene un servidor independiente. SQLite lee y escribe directamente a archivos de disco ordinarios (SQLite, 2015).

En la Figura 5.43 se puede ver el diagrama entidad relación para la asociación de los objetos con su respectiva información contextual.

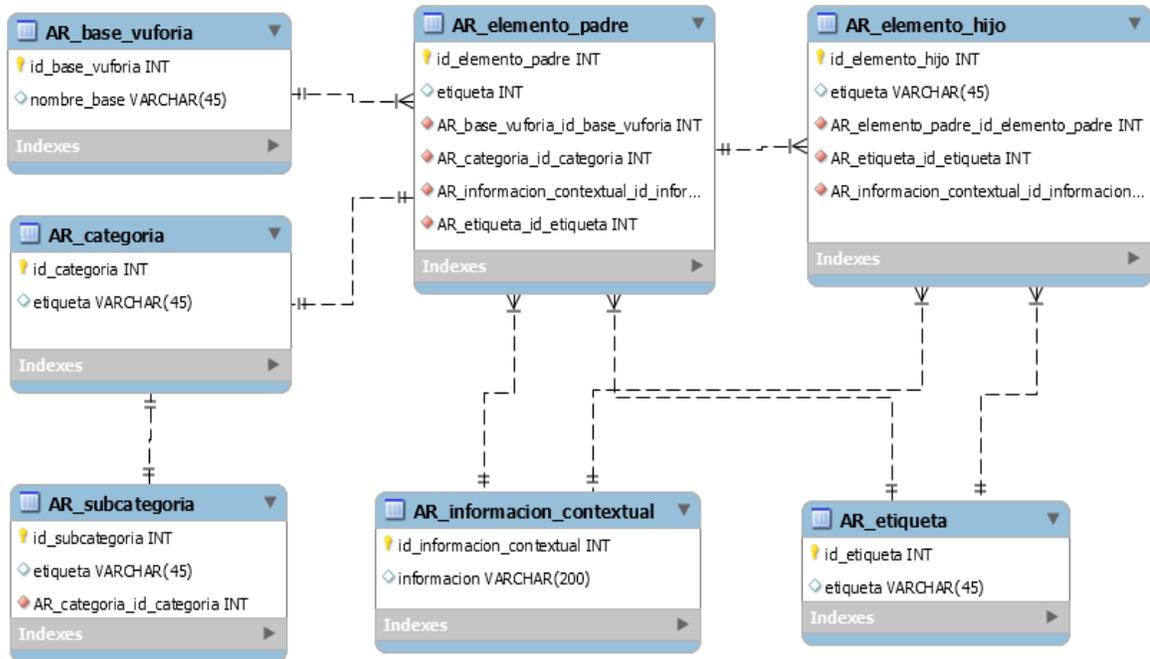


Figura 5.43 Diagrama entidad relación SQLite.

La base de datos está compuesta por siete tablas las cuales son:

- **AR_categoria**: En esta tabla se guardan las categorías de los objetos, se añadió para que en un futuro se pueda realizar filtros por categoría para no saturar la búsqueda de información en el dispositivo móvil.
- **AR_subcategoria**: Esta tabla está asociada a la tabla AR_categoria con el objetivo de realizar filtros de búsqueda.
- **AR_elemento_padre**: En esta tabla se guarda la referencia de los objetos principales que se requiere conocer.
- **AR_elemento_hijo**: Con esta tabla se relaciona los elementos de los objetos principales.
- **AR_etiqueta**: Se guardar las etiquetas de todos los objetos almacenados.
- **AR_informacion_contextual**: En esta tabla se guarda la información contextual asociada a cada objeto almacenado.

Para obtener la aplicación móvil el usuario deberá ingresar al siguiente link para descargar el archivo:

<http://synapciainfo.com/ServicioCenidet/index.php?r=Servicios/APK>

Una vez que se ha instalado, el usuario visualizará la siguiente pantalla mostrada en la Figura 5.44.



Figura 5.44 Pantalla de inicio de la aplicación móvil.

La pantalla contiene un mensaje de bienvenida y dos botones, el botón “Para más información” que al presionarlo abre el navegador del dispositivo para mostrar el manual de usuario de la aplicación, y el botón “Continuar” que al presionarlo activará la cámara trasera de su dispositivo multisensorial aparecerá una pantalla de ayuda, esta pantalla únicamente aparecerá la primera vez que se inició la aplicación como se puede ver en la siguiente figura.

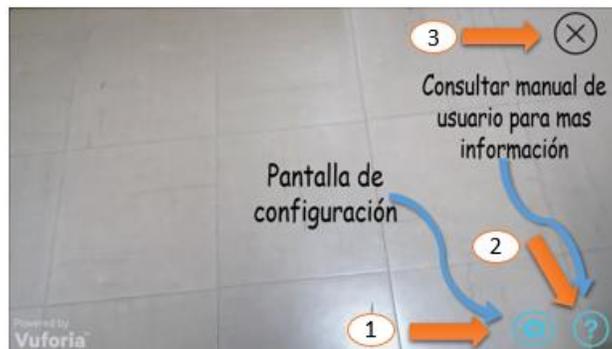


Figura 5.45 Pantalla de ayuda.

1. Botón para ver las configuraciones.
2. Botón para consultar el manual de usuario.
3. Botón para cerrar la pantalla de ayuda.

En la siguiente figura se puede ver la pantalla de configuración que aparece al tocar el botón que se muestra en la Figura 5.47:



Figura 5.47 Vista de la cámara del dispositivo móvil.

Para descargar el paquete del objeto, para este ejemplo, “lavadora” que se ha creado en la Fase 1 de la metodología, el usuario debe dar clic en el botón “Agregar paquete”, aparece la pantalla mostrada en la Figura 5.46 en la cual se visualizan los paquetes que el usuario ha creado con anterioridad y para tener la opción de realizar la descarga:

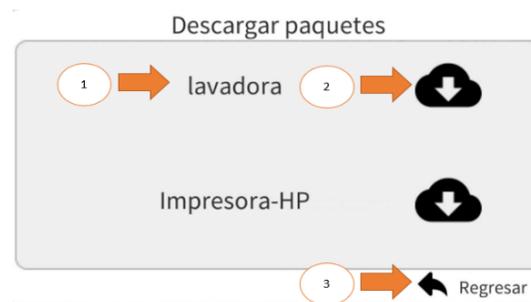


Figura 5.46 Pantalla de descarga de nuevos paquetes.

1. Nombre del paquete que se va a descargar
2. Botón para iniciar la descarga
3. Botón para regresar a la pantalla de configuración

Cuando el usuario presiona el botón para iniciar la descarga se muestra la pantalla de la figura 5.47:



Figura 5.47 Descarga de un nuevo paquete.

En la Figura 5.48 se puede ver la pantalla de configuración de la aplicación donde se muestra el resultado de la descarga. A continuación, se explica cada uno de los elementos que conforman esta pantalla.

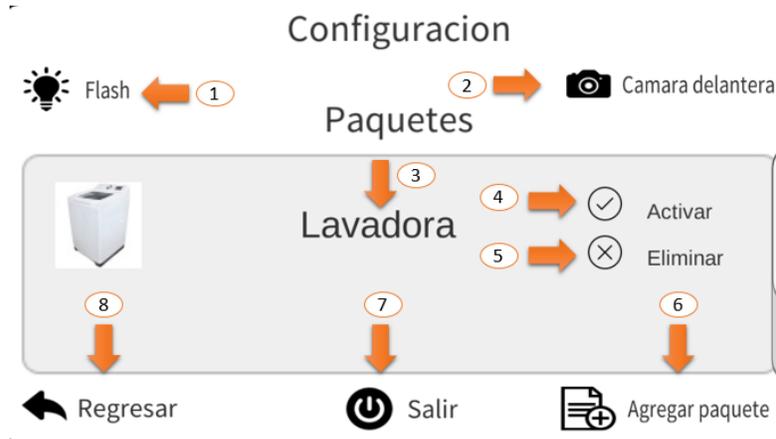


Figura 5.48 Pantalla de configuración del prototipo.

1. Se puede activar la lámpara del flash del dispositivo Android en caso de que no se tenga iluminación adecuada.
2. Se puede cambiar el visor de la cámara a delantera o trasera
3. Indicador del proceso de carga de la aplicación el cual representa el objeto que puede ser detectado.
4. Botón para activar o desactivar la detección del objeto
5. Botón para eliminar el objeto de la aplicación
6. Abre una pantalla para descargar más paquetes dentro de la aplicación para detectar otros objetos.
7. Salir de la aplicación
8. Regresa a la pantalla del visor de la cámara

Una vez que se ha descargado el paquete, el usuario debe seleccionar un objeto de la lista que se presenta en la aplicación Android, para iniciar el proceso de reconocimiento con RA es necesario enfocar el objeto para iniciar el proceso de reconocimiento de patrones, cuando se identifica un patrón se presentan dos botones virtuales sobre la imagen, como se muestra en la Figura 5.49, el botón de la izquierda representa una descripción breve del elemento y el botón de la derecha representa información más detallada sobre el elemento.

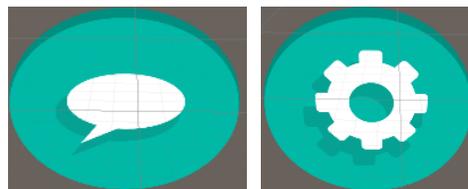


Figura 5.49 a) Botón de descripción, b) Botón información detallada.

En las figuras 5.50 y 5.51 se puede ver imágenes reales del reconocimiento de elementos que realiza la aplicación móvil, así como de la información contextual que está proporcionando al usuario



Figura 5.50 Botones virtuales al momento de reconocer elementos.



Figura 5.51 Presentación de información contextual.

Capítulo 6

Pruebas y Resultados

6.1 Pruebas Heurísticas

Para evaluar la metodología y el prototipo Android se realizaron pruebas heurísticas y de evaluación de la Experiencia del Usuario (UX) mediante la metodología desarrollada por (Llanes, 2014), estas pruebas permitieron la evaluación de la funcionalidad y de UX. Esta evaluación permitió identificar y corregir errores de diseño y funcionales, tanto en la metodología como en el prototipo Android.

6.1.1 Técnica de evaluación Heurística

En (Jakob Nielsen, 1990) se describe la evaluación Heurística como un método informal para realizar análisis de usabilidad, donde una serie de evaluadores visualizan una serie de interfaces diseñadas y se les pide comentar acerca de ellas.

La evaluación Heurística se basa en la observación y revisión de los aspectos buenos y malos encontrados en la interfaz y funcionamiento de un software o aplicación Web/móvil, estos comentarios son acordes a la existencia de las 10 reglas de usabilidad diseñadas por Jakob Nielsen en (Nielsen, 1995), estos aspectos encontrados deben ser reportados dentro de un formato conocido como Usability Aspect Report (UAR).

En relación a lo anterior se presenta la siguiente tabla que resume las heurísticas encontradas en la metodología:

Tabla 6.1 Heurísticas encontradas en la evaluación de la metodología.

	Aspecto	Heurística
1	Confusión de etiquetas	Consistencia y estándares
2	Especificación al crear la base de datos	Visibilidad del estado del sistema
3	Selección de imágenes contextuales del objeto	Prevención de errores
4	Subir paquete generado al servidor	Prevención de errores, flexibilidad y eficiencia de uso
5	Ruta del proyecto	Prevención de errores
6	Opciones adicionales en la interfaz de escritorio	Visibilidad del estado del sistema
7	Método para identificar vistas	Consistencia y estándares
8	Obtener imágenes de las vistas	Reconocimiento antes que recuerdo
9	Aplicación Asociación	Visibilidad del estado del sistema
10	Botones no indican cuál es su función	Reconocimiento antes que recuerdo
11	Cambiar de lugar un botón	Reconocimiento antes que recuerdo
12	No se especifica la función del botón generar paquete	Visibilidad del estado del sistema

En la siguiente imagen se presenta una gráfica con la tabulación de las heurísticas y los aspectos encontrados en la evaluación de la metodología.



Figura 6.1 Tabulación de las heurísticas encontradas en la metodología.

En la siguiente tabla se encuentran las heurísticas de la evaluación aplicada a la aplicación móvil.

Tabla 6.2 Heurísticas encontradas en la evaluación de la aplicación web.

	Aspecto	Heurística
1	Falta de scroll en la lista	Visibilidad del estado del sistema
2	La aplicación no cuenta con ayuda visible	Ayuda y documentación
3	Pantalla de inicio de la aplicación	Visibilidad del estado del sistema
4	Forma de salir de la aplicación	Visibilidad del estado del sistema
5	Descarga del paquete del objeto	Visibilidad del estado del sistema
6	Funcionalidad de los botones virtuales	Consistencia y estándares
7	Inconsistencia de términos	Consistencia y estándares
8	No hay botón de configuración	Visibilidad del estado del sistema

En la siguiente imagen se muestra la tabulación de las heurísticas que se encontraron en la evaluación de la aplicación móvil.

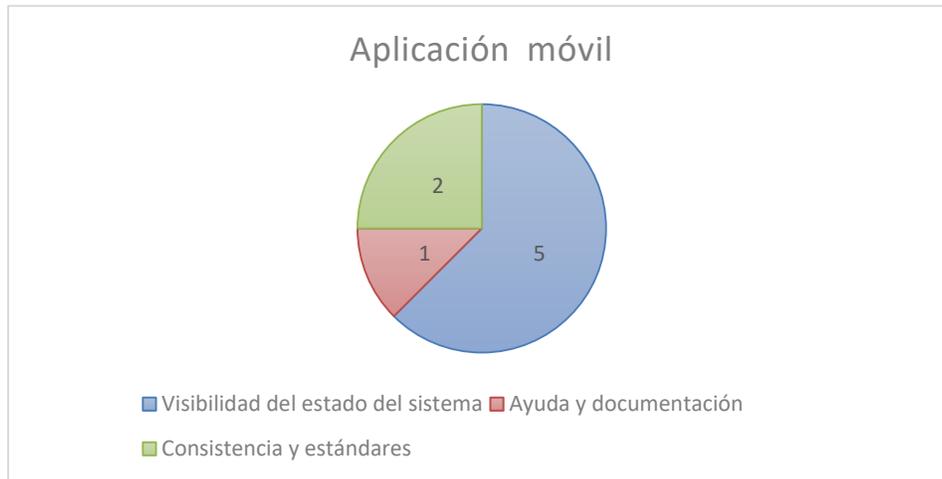


Figura 6.2 Tabulación de las heurísticas encontradas en la aplicación móvil.

6.2 Plan de pruebas de la metodología

Esta sección describe el proceso a seguir para verificar que la metodología desarrollada en este trabajo de tesis se aplique correctamente, y que a través de la evaluación centrada en el usuario, desarrollada por (Llanes, 2014), para que los usuarios finales identifiquen posibles pasos omitidos que dificulten la aplicación de la metodología y que permitan posteriormente realizar las correcciones correspondientes.

6.2.1 Objetivo

Verificar que se puede implementar la fase 1 de la metodología propuesta con los pasos definidos en la descripción de la metodología.

6.2.2 Alcance de las pruebas

Evaluar los siguientes procesos de la fase 1.

- Identificar vistas
- Obtener imágenes de las vistas
- Identificar los elementos dentro de las vistas
- Generación de patrones de las vistas
- Información contextual de los objetos

6.2.3 Limitaciones de las pruebas

- Únicamente se evaluará la fase 1 de la metodología desarrollada
- Las personas quienes realicen las pruebas deben tener un perfil computacional técnico

Tabla 6.3 Caso de prueba pantalla LG.

Caso de prueba: 1	
Descripción	Comprobar que la fase 1 de la metodología propuesta puede ser implementada.
Secuencia	<ol style="list-style-type: none"> 1. Selección de objeto de interés 2. Identificar vistas 3. Obtener Imágenes de las vistas 4. Identificar elementos dentro de las vistas 5. Generación de patrones de las vistas 6. Asociación de la información contextual de los elementos
Resultado	<ol style="list-style-type: none"> 1. Objeto seleccionado: Pantalla LG 2. Se identificaron 2 vistas: frontal y trasera 3. A continuación se muestran las imágenes de las vistas identificadas. <div style="display: flex; justify-content: space-around; align-items: center;">   </div> 4. Se identificaron los siguientes elementos dentro de las vistas: <p>Se identificaron tres elementos, por cada elemento se obtuvieron tres vistas, para definir el nombre de las vistas de cada elemento se utilizó la siguiente nomenclatura:</p> <p>Deberá iniciar con el prefijo “<i>obj</i>”, seguido de un número consecutivo 01,02,.. (haciendo referencia a la base de datos que ya se cargó con anterioridad, para este caso sería obj07) y el símbolo especial “_”, continua con la palabra “<i>elemento</i>” seguida del número consecutivo 01,02, ..., n dependiendo del número de elementos que se identificaron. Después deberá llevar el símbolo “_” y por último la palabra “<i>vista</i>” seguido del número de vista a la que se haga referencia.</p> <p>En las siguientes tablas se pueden ver los elementos identificados, así como las vistas de cada elemento y el nombre de cada vista.</p>

Elemento 1 con tres vistas		
Vista 1	Vista 2	Vista 3
		
obj07_elemento01_vista01	obj07_elemento01_vista02	obj07_elemento01_vista03

Elemento 2 con tres vistas		
Vista 1	Vista 2	Vista 3
		
obj07_elemento02_vista01	obj07_elemento02_vista02	obj07_elemento02_vista03

Elemento 3 con tres vistas		
Vista 1	Vista 2	Vista 3
		
obj07_elemento03_vista01	obj07_elemento03_vista02	obj07_elemento03_vista03

5. En este paso se creó la base de datos y se subieron las vistas de cada elemento detectado al portal web de Vuforia.

Para definir el nombre de la base de datos se utilizó la siguiente nomenclatura:

Deberá iniciar con las letras “**obj**”, seguido de un número consecutivo 01,02,.. (Dependiendo del número de bases de datos que existan), después deberá llevar el símbolo “_” junto con el nombre que el usuario quiera dar a la base de datos, es recomendable que el nombre dado a la base de datos tenga relación con el objeto analizado.

Siguiendo la nomenclatura el nombre de la base de datos quedó de la siguiente manera.

obj07_SmartTVLG

En la siguiente imagen se puede observar las vistas cargadas en el portal web Vuforia.

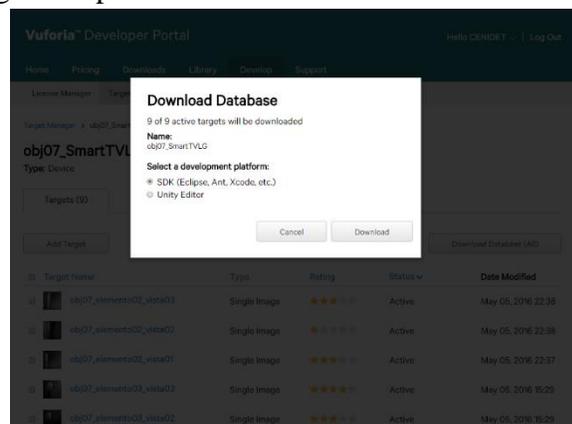
obj07_SmartTVLG Edit Name
Type: Device

Targets (9)

Add Target Download Database (AID)

Target Name	Type	Rating	Status	Date Modified
obj07_elemento01_vista01	Single Image	★★★★☆	Active	May 05, 2016 15:25
obj07_elemento01_vista02	Single Image	★★★★☆	Active	May 05, 2016 15:26
obj07_elemento01_vista03	Single Image	★★★★☆	Active	May 05, 2016 15:26
obj07_elemento02_vista01	Single Image	★★★★☆	Active	May 05, 2016 15:26
obj07_elemento02_vista02	Single Image	★★★★☆	Active	May 05, 2016 15:26
obj07_elemento02_vista03	Single Image	★★★★☆	Active	May 05, 2016 15:27
obj07_elemento03_vista01	Single Image	★★★★☆	Active	May 05, 2016 15:28
obj07_elemento03_vista02	Single Image	★★★★☆	Active	May 05, 2016 15:29
obj07_elemento03_vista03	Single Image	★★★★☆	Active	May 05, 2016 15:29

Una vez que se cargaron todas las vistas de los elementos en el portal web de Vuforia se descarga la base de datos. Se obtiene un archivo .rar que al descomprimirlo contiene los archivos .xml y .dat que se utilizarán en el siguiente paso.



6. Se muestra la pantalla de la aplicación de escritorio en la cual se llevó a cabo la asociación de las imágenes.

Para realizar esta actividad se debe efectuar los siguientes pasos

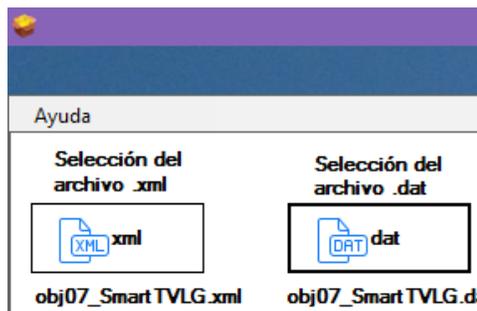
8. Cargar el archivo .xml que se descargó del portal de Vuforia.

Para cargar el archivo .xml se da clic al botón “xml” el cual abre una ventana en donde se debe buscar y seleccionar el archivo. Una vez que se ha cargado el archivo aparecerá el nombre debajo del botón “xml” como se observa en la siguiente imagen.



9. Cargar el archivo .dat que se descargó del portal de Vuforia.

Para cargar el archivo .dat se da clic al botón “*dat*” el cual abre una ventana en donde se debe buscar y seleccionar el archivo. Una vez que se ha cargado el archivo aparecerá el nombre debajo del botón “*dat*” como se observa en la siguiente imagen.



10. Imagen representativa del objeto analizado, en este caso el objeto “Pantalla LG”.

Para cargar la imagen se da clic en el botón “Agregar imagen” el cual abre una ventana para la búsqueda de la imagen una vez que se cargó la imagen a la aplicación se ve de la siguiente manera.



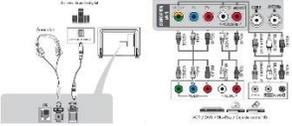
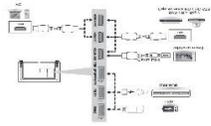
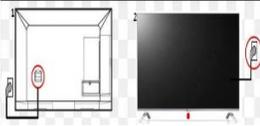
11. Cargar imágenes que van a proporcionar información contextual.

Se eligieron las siguientes tres imágenes de información contextual, una imagen por elemento detectado, antes de cargar estas imágenes de información contextual, para definir el nombre de cada imagen se utilizó la siguiente nomenclatura:

Deberá iniciar con las letras “obj”, seguido de un número consecutivo 01,02,.. (haciendo referencia a la base de datos que ya se cargó con anterioridad, para este caso sería obj07) y el símbolo especial “_”, continua con la palabra “elemento” seguida del número consecutivo 01,02, ..., n dependiendo del número de elementos que se hayan identificado. Después deberá llevar el símbolo “_” y por último el nombre de la base de datos de Vuforia.

En la siguiente tabla se puede observar las imágenes correspondientes a cada elemento, así como el nombre de cada imagen de información contextual.

Imágenes de información contextual

Elemento 1	Elemento 2	Elemento 3
 obj07_elemento01_obj07_SmartTVLG.jpg	 obj07_elemento02_obj07_SmartTVLG.jpg	 obj07_elemento03_obj07_SmartTVLG.jpg

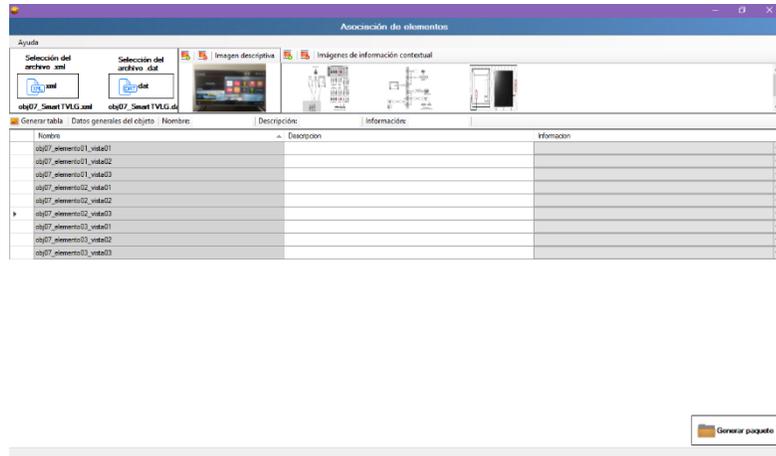
Para cargar las imágenes se da clic en el botón “Agregar imagen” el cual abre una ventana para la búsqueda de la imagen. Una vez que se cargaron las imágenes a la aplicación se puede ver una vista previa de las imágenes seleccionadas como se muestra a continuación.



12. Generar tabla para la asociación de información contextual.

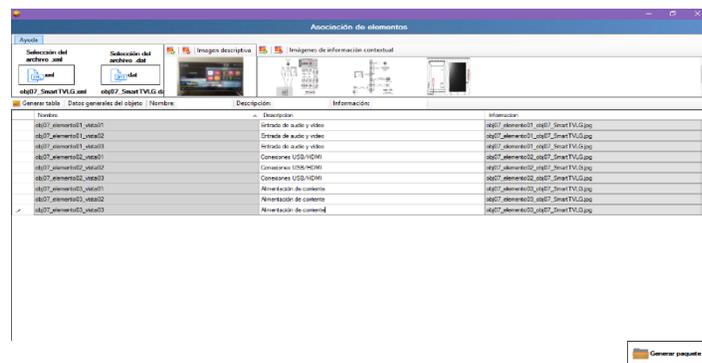
Para la generación de la tabla se da clic al botón “Generar tabla” se genera una tabla de forma automática que contiene tres columnas que son:

- Nombre: Son los nombres de las imágenes que se subieron al portal web de Vuforia.
- Descripción: En esta columna se escribe una breve descripción del elemento.
- Información: En esta columna se muestra una lista con los nombres de las imágenes de información contextual que se cargaron en el punto anterior.



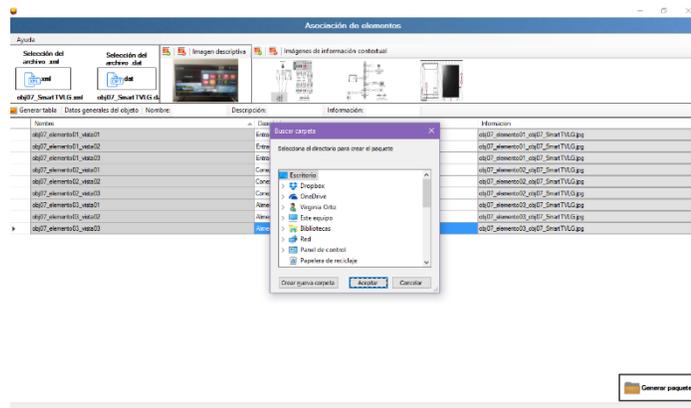
De igual manera se llenan los siguientes campos de texto que hacen referencia al objeto de interés, los cuales se describen a continuación.

- Nombre: En esta caja se escribió el nombre del objeto analizado “SmartTVLG”.
- Descripción: En esta caja se proporcionó una descripción breve sobre el objeto analizado, en este caso el objeto “SmartTVLG”.
- Información: En esta caja se proporciona información detallada sobre el objeto que se analizó, en este caso el objeto “SmartTVLG”.



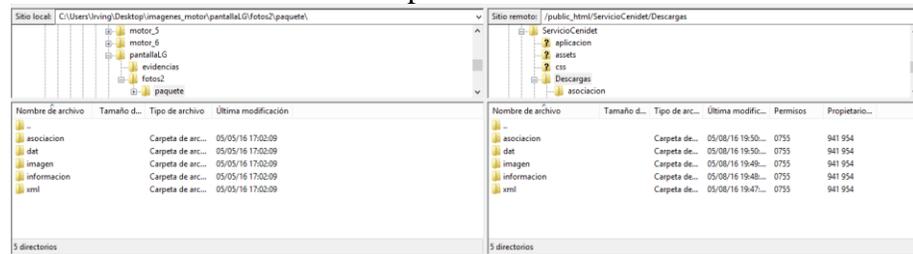
13. Generación del directorio que contiene los archivos de asociación.

Para generar el directorio del paquete se da clic en el botón “Generar paquete”, se abre una ventana para seleccionar la ruta donde se guardará el directorio de asociación



14. Subir paquete generado al servidor

Una vez que se realizó la asociación con la aplicación de escritorio y se obtuvo el directorio con sus respectivos archivos se subirán al servidor FTP:



Observación

Siguiendo los pasos de la secuencia de este caso de prueba se pudo implementar la fase 1 de la metodología propuesta de manera exitosa. De igual forma se puede observar que la aplicación de escritorio para realizar la asociación de la información contextual con los elementos de las vistas funciona correctamente.

Responsable de la prueba

Autor de tesis

Tabla 6.4 Caso de prueba multifuncional HP.

Caso de prueba: 2	
Descripción	Comprobar que la fase 1 de la metodología propuesta puede ser implementada.
Secuencia	<ol style="list-style-type: none"> 1. Selección de objeto de interés 2. Identificar vistas 3. Obtener Imágenes de las vistas 4. Identificar elementos dentro de las vistas 5. Generación de patrones de las vistas 6. Asociación de la información contextual de los elementos
Resultado	<ol style="list-style-type: none"> 1. Objeto seleccionado: multifuncional HP 2. Se identificaron 3 vistas: Superior, frontal y lateral derecha 3. A continuación, se muestran las imágenes de las vistas identificadas. <div style="display: flex; justify-content: space-around; align-items: center;">    </div> <ol style="list-style-type: none"> 4. Se identificaron los siguientes elementos dentro de las vistas: <p>Se identificaron cuatro elementos, por cada elemento se obtuvieron tres vistas, para definir el nombre de las vistas de cada elemento se utilizó la siguiente nomenclatura:</p> <p>Deberá iniciar con las letras “<i>obj</i>”, seguido de un número consecutivo 01,02,.. (haciendo referencia a la base de datos que ya se cargó con anterioridad, en este caso sería obj14) y el símbolo especial “_”, continua con la palabra “<i>elemento</i>” seguida del número consecutivo 01,02, ..., n dependiendo del número de elementos que identificaron. Después deberá llevar el símbolo “_” y por último la palabra “<i>vista</i>” seguido del número de vista a la que se haga referencia.</p> <p>En las siguientes tablas se pueden ver los elementos identificados, así como las vistas de cada elemento y el nombre de cada vista.</p>

Elemento 1 con tres vistas		
Vista 1	Vista 2	Vista 3
		
obj14_elemento01_vista01	obj14_elemento01_vista02	obj14_elemento01_vista03
Elemento 2 con tres vistas		
Vista 1	Vista 2	Vista 3
		
obj14_elemento02_vista01	obj14_elemento02_vista02	obj14_elemento02_vista03
Elemento 3 con tres vistas		
Vista 1	Vista 2	Vista 3
		
obj14_elemento03_vista01	obj14_elemento03_vista02	obj14_elemento03_vista03
Elemento 4 con tres vistas		
Vista 1	Vista 2	Vista 3
		
obj14_elemento04_vista01	obj14_elemento04_vista02	obj14_elemento04_vista03

5. En este paso se creó la base de datos y se subieron las vistas de cada elemento detectado al portal web de Vuforia.

Para definir el nombre de la base de datos se utilizó la siguiente nomenclatura:

Deberá iniciar con las letras **“obj”**, seguido de un número consecutivo 01,02,.. (Dependiendo del número de bases de datos que existan), después deberá llevar el símbolo **“_”** junto con el nombre que el usuario quiera dar a la base de datos, es recomendable que el nombre dado a la base de datos tenga relación con el objeto analizado.

Siguiendo la nomenclatura el nombre de la base de datos quedó de la siguiente manera.

Obj14_multifuncionalHP

En la siguiente imagen se puede observar las vistas cargadas en el portal web Vuforia.

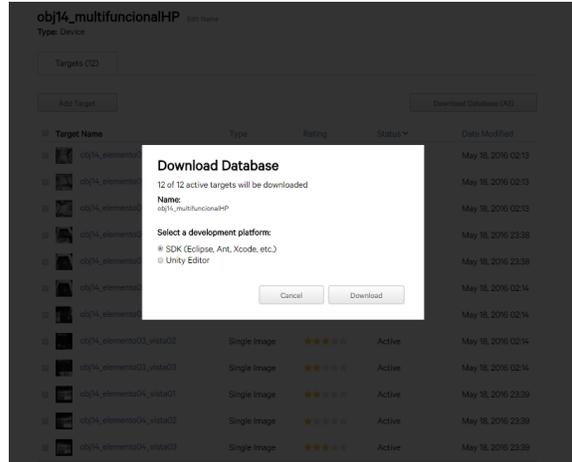
obj14_multifuncionalHP [Edit Name](#)
 Type: Device

Targets (12)

Add Target Download Database (All)

Target Name	Type	Rating	Status	Date Modified
 obj14_elemento01_vista01	Single Image	★★★★★	Active	May 18, 2016 02:13
 obj14_elemento01_vista02	Single Image	★★★★★	Active	May 18, 2016 02:13
 obj14_elemento01_vista03	Single Image	★★★★★	Active	May 18, 2016 02:13
 obj14_elemento02_vista01	Single Image	★★★★★	Active	May 18, 2016 23:38
 obj14_elemento02_vista02	Single Image	★★★★★	Active	May 18, 2016 23:38
 obj14_elemento02_vista03	Single Image	★★★★★	Active	May 18, 2016 02:14
 obj14_elemento03_vista01	Single Image	★★★★★	Active	May 18, 2016 02:14
 obj14_elemento03_vista02	Single Image	★★★★★	Active	May 18, 2016 02:14
 obj14_elemento03_vista03	Single Image	★★★★★	Active	May 18, 2016 02:14
 obj14_elemento04_vista01	Single Image	★★★★★	Active	May 18, 2016 23:39
 obj14_elemento04_vista02	Single Image	★★★★★	Active	May 18, 2016 23:39
 obj14_elemento04_vista03	Single Image	★★★★★	Active	May 18, 2016 23:39

Una vez que se cargaron todas las vistas de los elementos en el portal web de Vuforia se descarga la base de datos. Se obtiene un archivo .rar que al descomprimirlo contiene los archivos .xml y .dat que se utilizarán en el siguiente paso.

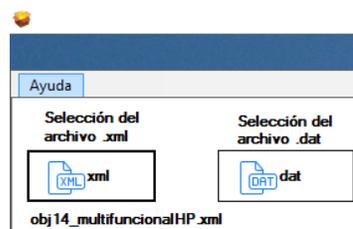


6. Se muestra la pantalla de la aplicación de escritorio en la cual se llevó a cabo la asociación de las imágenes.

Para ejecutar esta actividad se debe realizar los siguientes pasos

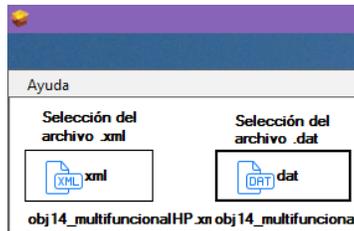
15. Cargar el archivo .xml que se descargó del portal de Vuforia.

Para cargar el archivo .xml se da clic al botón “xml” el cual abre una ventana en donde se debe buscar y seleccionar el archivo. Una vez que se ha cargado el archivo aparecerá el nombre debajo del botón “xml” como se observa en la siguiente imagen.



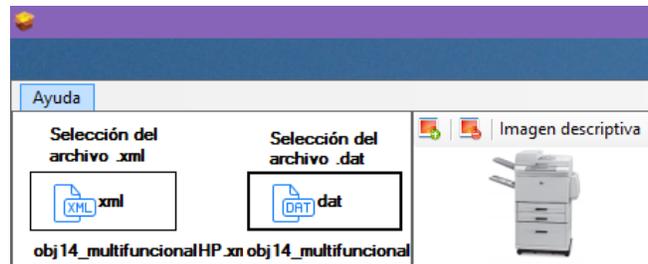
16. Cargar el archivo .dat que se descargó del portal de Vuforia.

Para cargar el archivo .dat se da clic al botón “*dat*” el cual abre una ventana en donde se debe buscar y seleccionar el archivo. Una vez que se ha cargado el archivo aparecerá el nombre debajo del botón “*dat*” como se observa en la siguiente imagen.



17. Imagen representativa del objeto analizado, en este caso el objeto “multifuncionalHP”.

Para cargar la imagen se da clic en el botón “Agregar imagen” el cual abre una ventana para la búsqueda de la imagen una vez que se cargó la imagen a la aplicación se ve de la siguiente manera.

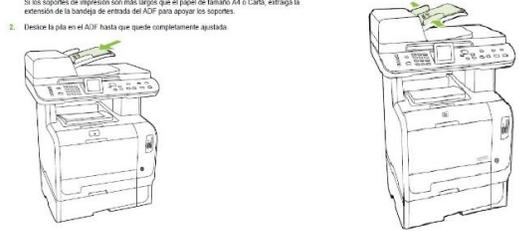
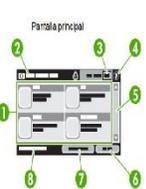
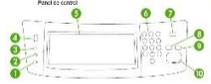


18. Cargar imágenes que van a proporcionar información contextual.

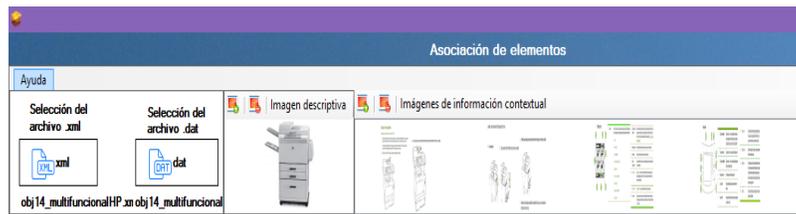
Se eligieron las siguientes cuatro imágenes de información contextual, una imagen por elemento detectado, antes de cargar estas imágenes de información contextual, para definir el nombre de cada imagen se utilizó la siguiente nomenclatura:

Deberá iniciar con las letras “**obj**”, seguido de un número consecutivo 01,02,.. (haciendo referencia a la base de datos que ya se cargó con anterioridad, en nuestro caso sería obj14) y el símbolo especial “_”, continua con la palabra “**elemento**” seguida del número consecutivo 01,02, ..., n dependiendo del número de elementos que se hayan identificado. Después deberá llevar el símbolo “_” y por último el nombre de la base de datos de Vuforia.

En la siguiente tabla se puede observar las imágenes correspondientes a cada elemento, así como el nombre de cada imagen de información contextual

<p>Elemento 1</p>	<p>obj14_elemento01_obj14_multifuncionalHP.jpg</p> <p>Carga de originales</p> <p>Alimentador automático de documentos (ADF)</p> <ol style="list-style-type: none"> 1. Inserte el borde superior de la pila de originales en la bandeja de entrada del ADF, con la pila de soporte de impresión boca arriba y la primera página que se va a copiar encima de la pila. Si los soportes de impresión son más largos que el papel de tamaño A4 o Carta, estraga la extensión de la bandeja de entrada del ADF para apoyar los soportes. 2. Deslice la pila en el ADF hasta que quede completamente ajustada. 3. Ajuste las guías de los soportes de impresión hasta que queden totalmente cerradas a los soportes. 																																				
<p>Elemento 2</p>	<p>obj14_elemento02_obj14_multifuncionalHP.jpg</p> <p>La bandeja puede contener hasta 100 hojas de papel o 10 sobres.</p> <ol style="list-style-type: none"> 1. Abra la bandeja 1. 2. Cargue el papel en función del tamaño y de las opciones de acabado. 3. Ajuste las guías de papel para que toquen ligeramente la pila de papel, pero sin llegar a doblarlo. 4. Asegúrese de que el papel carga debajo de las lengüetas de las guías y que no sobrepase los indicadores de nivel de carga. 																																				
<p>Elemento 3</p>	<p>obj14_elemento03_obj14_multifuncionalHP.jpg</p> <p>Panel principal</p>  <ol style="list-style-type: none"> 1. Inicio 2. Copiar 3. Escanear 4. Enviar 5. Configuración 6. Estado de papel 7. Estado de tinta 8. Estado de mantenimiento <table border="1"> <tr> <td>1 Inicio</td> <td>En función de la pantalla en que el administrador del sistema haya configurado el dispositivo, las funciones que aparecen en esta página pueden variar en los dispositivos conectados.</td> <td>2 Estado de estado del dispositivo</td> <td>El estado de funcionamiento de copias, tareas y número de copias que se han realizado.</td> </tr> <tr> <td>• Copiar</td> <td></td> <td>3 Reciente de estado</td> <td>El estado de ejecución de copias, tareas y número de copias que se han realizado.</td> </tr> <tr> <td>• Para copiar</td> <td></td> <td>4 Bandeja Ajuste</td> <td>Trague el botón Ajuste para abrir el estante de ajuste incorporado.</td> </tr> <tr> <td>• Copiar electrónico</td> <td></td> <td>5 Bandeja de alimentación</td> <td>Trague las flechas arriba y abajo de la bandeja de alimentación para ver la lista completa de funciones de alimentación.</td> </tr> <tr> <td>• Copiar electrónico secundario</td> <td></td> <td>6 Control de estado</td> <td>Trague Control para ver la lista de dispositivos de la lista para acceder a funciones de estado. Después de cerrar la pantalla, el dispositivo recordará todas las opciones preseleccionadas.</td> </tr> <tr> <td>• Copiar de red</td> <td></td> <td>7 Estado de red</td> <td>Trague Estado de red para buscar información acerca de la conexión de red.</td> </tr> <tr> <td>• Administración de dispositivos</td> <td></td> <td>8 Fecha y hora</td> <td>Ajuste la hora y la fecha. El administrador del sistema puede seleccionar el formato que desea el dispositivo para mostrar la hora y la fecha. Por ejemplo, el formato de "Día de 24 horas".</td> </tr> <tr> <td>• Estado de los consumibles</td> <td></td> <td></td> <td></td> </tr> <tr> <td>• Administración</td> <td></td> <td></td> <td></td> </tr> </table>	1 Inicio	En función de la pantalla en que el administrador del sistema haya configurado el dispositivo, las funciones que aparecen en esta página pueden variar en los dispositivos conectados.	2 Estado de estado del dispositivo	El estado de funcionamiento de copias, tareas y número de copias que se han realizado.	• Copiar		3 Reciente de estado	El estado de ejecución de copias, tareas y número de copias que se han realizado.	• Para copiar		4 Bandeja Ajuste	Trague el botón Ajuste para abrir el estante de ajuste incorporado.	• Copiar electrónico		5 Bandeja de alimentación	Trague las flechas arriba y abajo de la bandeja de alimentación para ver la lista completa de funciones de alimentación.	• Copiar electrónico secundario		6 Control de estado	Trague Control para ver la lista de dispositivos de la lista para acceder a funciones de estado. Después de cerrar la pantalla, el dispositivo recordará todas las opciones preseleccionadas.	• Copiar de red		7 Estado de red	Trague Estado de red para buscar información acerca de la conexión de red.	• Administración de dispositivos		8 Fecha y hora	Ajuste la hora y la fecha. El administrador del sistema puede seleccionar el formato que desea el dispositivo para mostrar la hora y la fecha. Por ejemplo, el formato de "Día de 24 horas".	• Estado de los consumibles				• Administración			
1 Inicio	En función de la pantalla en que el administrador del sistema haya configurado el dispositivo, las funciones que aparecen en esta página pueden variar en los dispositivos conectados.	2 Estado de estado del dispositivo	El estado de funcionamiento de copias, tareas y número de copias que se han realizado.																																		
• Copiar		3 Reciente de estado	El estado de ejecución de copias, tareas y número de copias que se han realizado.																																		
• Para copiar		4 Bandeja Ajuste	Trague el botón Ajuste para abrir el estante de ajuste incorporado.																																		
• Copiar electrónico		5 Bandeja de alimentación	Trague las flechas arriba y abajo de la bandeja de alimentación para ver la lista completa de funciones de alimentación.																																		
• Copiar electrónico secundario		6 Control de estado	Trague Control para ver la lista de dispositivos de la lista para acceder a funciones de estado. Después de cerrar la pantalla, el dispositivo recordará todas las opciones preseleccionadas.																																		
• Copiar de red		7 Estado de red	Trague Estado de red para buscar información acerca de la conexión de red.																																		
• Administración de dispositivos		8 Fecha y hora	Ajuste la hora y la fecha. El administrador del sistema puede seleccionar el formato que desea el dispositivo para mostrar la hora y la fecha. Por ejemplo, el formato de "Día de 24 horas".																																		
• Estado de los consumibles																																					
• Administración																																					
<p>Elemento 4</p>	<p>obj14_elemento04_obj14_multifuncionalHP.jpg</p> <p>Panel de control</p>  <ol style="list-style-type: none"> 1. Copiar 2. Escanear 3. Enviar 4. Configuración 5. Estado de papel 6. Estado de tinta 7. Estado de mantenimiento 8. Fecha y hora 9. Estado de red 10. Inicio <table border="1"> <tr> <td>1 Copiar</td> <td>En función de la pantalla en que el administrador del sistema haya configurado el dispositivo, las funciones que aparecen en esta página pueden variar en los dispositivos conectados.</td> <td>7 Estado de estado del dispositivo</td> <td>El estado de funcionamiento de copias, tareas y número de copias que se han realizado.</td> </tr> <tr> <td>2 Copiar</td> <td></td> <td>8 Fecha y hora</td> <td>Ajuste la hora y la fecha. El administrador del sistema puede seleccionar el formato que desea el dispositivo para mostrar la hora y la fecha. Por ejemplo, el formato de "Día de 24 horas".</td> </tr> <tr> <td>3 Copiar</td> <td></td> <td>9 Estado de red</td> <td>Trague Estado de red para buscar información acerca de la conexión de red.</td> </tr> <tr> <td>4 Copiar</td> <td></td> <td>10 Inicio</td> <td>Trague Inicio para volver al menú principal.</td> </tr> <tr> <td>5 Copiar</td> <td></td> <td></td> <td></td> </tr> <tr> <td>6 Copiar</td> <td></td> <td></td> <td></td> </tr> </table>	1 Copiar	En función de la pantalla en que el administrador del sistema haya configurado el dispositivo, las funciones que aparecen en esta página pueden variar en los dispositivos conectados.	7 Estado de estado del dispositivo	El estado de funcionamiento de copias, tareas y número de copias que se han realizado.	2 Copiar		8 Fecha y hora	Ajuste la hora y la fecha. El administrador del sistema puede seleccionar el formato que desea el dispositivo para mostrar la hora y la fecha. Por ejemplo, el formato de "Día de 24 horas".	3 Copiar		9 Estado de red	Trague Estado de red para buscar información acerca de la conexión de red.	4 Copiar		10 Inicio	Trague Inicio para volver al menú principal.	5 Copiar				6 Copiar															
1 Copiar	En función de la pantalla en que el administrador del sistema haya configurado el dispositivo, las funciones que aparecen en esta página pueden variar en los dispositivos conectados.	7 Estado de estado del dispositivo	El estado de funcionamiento de copias, tareas y número de copias que se han realizado.																																		
2 Copiar		8 Fecha y hora	Ajuste la hora y la fecha. El administrador del sistema puede seleccionar el formato que desea el dispositivo para mostrar la hora y la fecha. Por ejemplo, el formato de "Día de 24 horas".																																		
3 Copiar		9 Estado de red	Trague Estado de red para buscar información acerca de la conexión de red.																																		
4 Copiar		10 Inicio	Trague Inicio para volver al menú principal.																																		
5 Copiar																																					
6 Copiar																																					

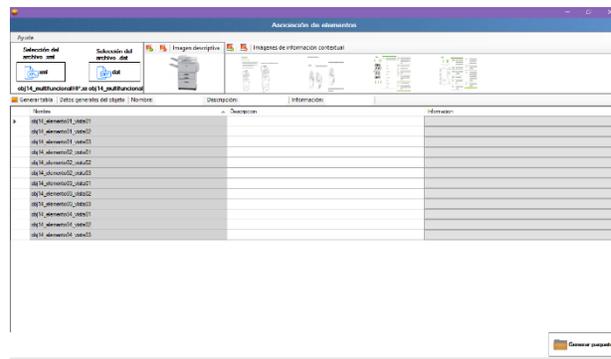
Para cargar las imágenes se da clic en el botón “Agregar imagen” el cual abre una ventana para la búsqueda de la imagen. Una vez que se cargaron las imágenes a la aplicación se puede ver una vista previa de las imágenes seleccionadas como se muestra a continuación.



19. Generar tabla para la asociación de información contextual.

Para la generación de la tabla se da clic al botón “*Generar tabla*” se genera una tabla de forma automática que contiene tres columnas que son:

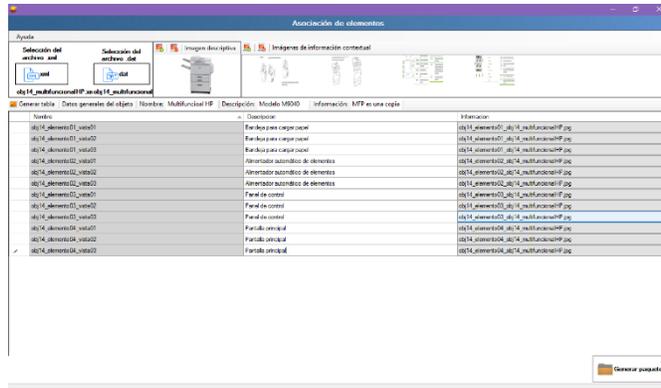
- Nombre: Son los nombres de las imágenes que se subieron al portal web de Vuforia.
- Descripción: En esta columna se escribe una breve descripción del elemento.
- Información: En esta columna se muestra una lista con los nombres de las imágenes de información contextual que se cargaron en el punto anterior.



De igual manera se llenan los siguientes campos de texto que hacen referencia al objeto de interés, los cuales se describen a continuación.

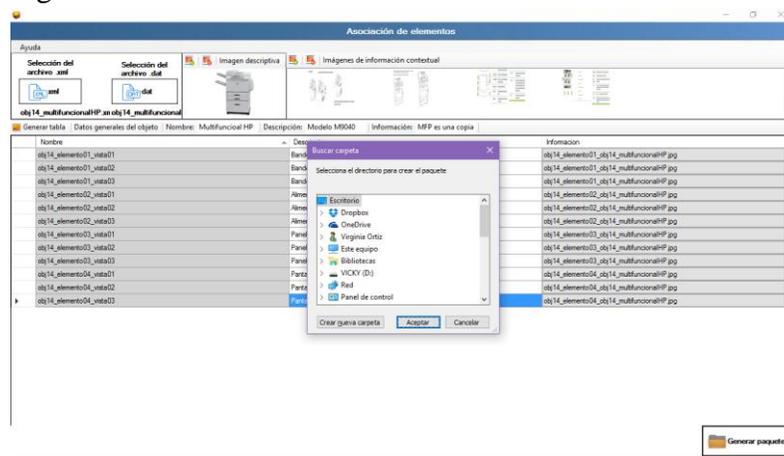
- Nombre: En esta caja se escribió el nombre del objeto analizado “*multifuncionalHP*”.
- Descripción: En esta caja se proporcionó una descripción breve sobre el objeto analizado, en este caso el objeto “*multifuncionalHP*”.

- Información: En esta caja se proporciona información detallada sobre el objeto que se analizó, en este caso el objeto “*multifuncionalHP*”.



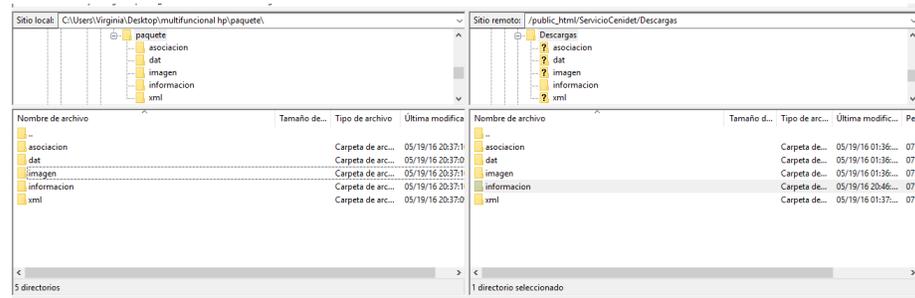
20. Generación del directorio que contiene los archivos de asociación.

Para generar el directorio del paquete se da clic en el botón “Generar paquete”, se abre una ventana para seleccionar la ruta donde se guardará el directorio de asociación



21. Subir paquete generado al servidor

Una vez que se realizó la asociación con la aplicación de escritorio y se obtuvo el directorio con sus respectivos archivos se subirán al servidor FTP:



Observación

Siguiendo los pasos de la secuencia de este caso de prueba se pudo implementar la fase 1 de la metodología propuesta de manera exitosa. De igual manera se puede observar que la aplicación de escritorio para realizar la asociación de la información contextual con los elementos de las vistas funciona correctamente.

Responsable de la prueba

Autor de tesis

Tabla 6.5 Caso de prueba bocina Sony.

Caso de prueba: 3	
Descripción	Comprobar que la fase 1 de la metodología propuesta puede ser implementada.
Secuencia	<ol style="list-style-type: none"> 1. Selección de objeto de interés 2. Identificar vistas 3. Obtener Imágenes de las vistas 4. Identificar elementos dentro de las vistas 5. Generación de patrones de las vistas 6. Asociación de la información contextual de los elementos
Resultado	<p>Objeto seleccionado: Bocina Sony Se identificaron 3 vistas: Superior, frontal y trasera A continuación, se muestran las imágenes de las vistas identificadas.</p>  <p>Se identificaron los siguientes elementos dentro de las vistas:</p> <p>Se identificaron seis elementos, por cada elemento se obtuvieron tres vistas, para definir el nombre de las vistas de cada elemento se utilizó la siguiente nomenclatura:</p> <p>Deberá iniciar con las letras “obj”, seguido de un número consecutivo 01,02,.. (haciendo referencia a la base de datos que ya se cargó con anterioridad, para este caso sería obj15) y el símbolo especial “_”, continua con la palabra “elemento” seguida del número consecutivo 01,02, ..., n dependiendo del número de elementos que identificaron. Después deberá llevar el símbolo “_” y por último la palabra “vista” seguido del número de vista a la que se haga referencia.</p>

En las siguientes tablas se pueden ver los elementos identificados, así como las vistas de cada elemento y el nombre de cada vista.

Elemento 1 con tres vistas

Vista 1	Vista 2	Vista 3
		
obj15_elemento01_vista01	obj15_elemento01_vista02	obj15_elemento01_vista03

Elemento 2 con tres vistas

Vista 1	Vista 2	Vista 3
		
obj15_elemento02_vista01	obj15_elemento02_vista02	obj15_elemento02_vista03

Elemento 3 con tres vistas

Vista 1	Vista 2	Vista 3
		
obj15_elemento03_vista01	obj15_elemento03_vista02	obj15_elemento03_vista03

Elemento 4 con tres vistas

Vista 1	Vista 2	Vista 3
		
obj15_elemento04_vista01	obj15_elemento04_vista02	obj15_elemento04_vista03

Elemento 5 con tres vistas

Vista 1	Vista 2	Vista 3
		
obj15_elemento00_vista01	obj15_elemento00_vista02	obj15_elemento05_vista03

Elemento 6 con tres vistas

Vista 1	Vista 2	Vista 3
		
obj15_elemento06_vista01	obj15_elemento06_vista02	obj15_elemento06_vista03

En este paso se creó la base de datos y se subieron las vistas de cada elemento detectado al portal web de Vuforia.

Para definir el nombre de la base de datos se utilizó la siguiente nomenclatura:

Deberá iniciar con las letras “**obj**”, seguido de un número consecutivo 01,02,.. (Dependiendo del número de bases de datos que existan), después deberá llevar el símbolo “_” junto con el nombre que el usuario quiera dar a la base de datos, es recomendable que el nombre dado a la base de datos tenga relación con el objeto analizado.

Siguiendo la nomenclatura el nombre de la base de datos quedó de la siguiente manera.

obj15_Bocina

En la siguiente imagen se puede observar las vistas cargadas en el portal web Vuforia.

obj15_Bocina Edit Name

Type: Device

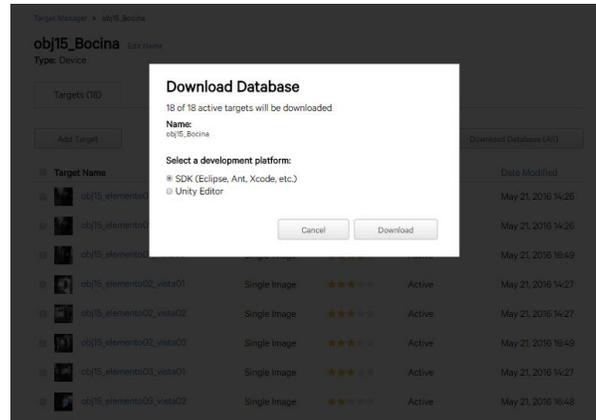
Targets (18)

Add Target

Download Database (All)

<input type="checkbox"/> Target Name	Type	Rating	Status	Date Modified
<input type="checkbox"/>  obj15_elemento01_vista01	Single Image	★★★★★	Active	May 21, 2016 14:26
<input type="checkbox"/>  obj15_elemento01_vista02	Single Image	★★★★★	Active	May 21, 2016 14:26
<input type="checkbox"/>  obj15_elemento01_vista03	Single Image	★★★★★	Active	May 21, 2016 16:49
<input type="checkbox"/>  obj15_elemento02_vista01	Single Image	★★★★★	Active	May 21, 2016 14:27
<input type="checkbox"/>  obj15_elemento02_vista02	Single Image	★★★★★	Active	May 21, 2016 14:27
<input type="checkbox"/>  obj15_elemento02_vista03	Single Image	★★★★★	Active	May 21, 2016 16:49
<input type="checkbox"/>  obj15_elemento03_vista01	Single Image	★★★★★	Active	May 21, 2016 14:27
<input type="checkbox"/>  obj15_elemento03_vista02	Single Image	★★★★★	Active	May 21, 2016 16:48
<input type="checkbox"/>  obj15_elemento03_vista03	Single Image	★★★★★	Active	May 21, 2016 14:28
<input type="checkbox"/>  obj15_elemento04_vista01	Single Image	★★★★★	Active	May 21, 2016 14:28
<input type="checkbox"/>  obj15_elemento04_vista02	Single Image	★★★★★	Active	May 21, 2016 16:48
<input type="checkbox"/>  obj15_elemento04_vista03	Single Image	★★★★★	Active	May 21, 2016 14:30
<input type="checkbox"/>  obj15_elemento05_vista01	Single Image	★★★★★	Active	May 21, 2016 14:30
<input type="checkbox"/>  obj15_elemento05_vista02	Single Image	★★★★★	Active	May 21, 2016 16:41
<input type="checkbox"/>  obj15_elemento05_vista03	Single Image	★★★★★	Active	May 21, 2016 14:30
<input type="checkbox"/>  obj15_elemento06_vista01	Single Image	★★★★★	Active	May 21, 2016 14:30
<input type="checkbox"/>  obj15_elemento06_vista02	Single Image	★★★★★	Active	May 21, 2016 14:31
<input type="checkbox"/>  obj15_elemento06_vista03	Single Image	★★★★★	Active	May 21, 2016 14:31

Una vez que se cargaron todas las vistas de los elementos en el portal web de Vuforia se descarga la base de datos. Se obtiene un archivo .rar que al descomprimirlo contiene los archivos .xml y .dat que se utilizarán en el siguiente paso.

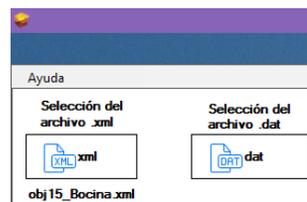


Se muestra la pantalla de la aplicación de escritorio en la cual se llevó a cabo la asociación de las imágenes.

Para ejecutar esta actividad se debe realizar los siguientes pasos

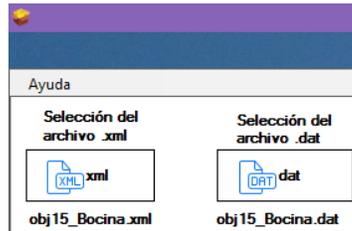
22. Cargar el archivo .xml que se descargó del portal de Vuforia.

Para cargar el archivo .xml se da clic al botón “xml” el cual abre una ventana en donde se debe buscar y seleccionar el archivo. Una vez que se ha cargado el archivo aparecerá el nombre debajo del botón “xml” como se observa en la siguiente imagen.



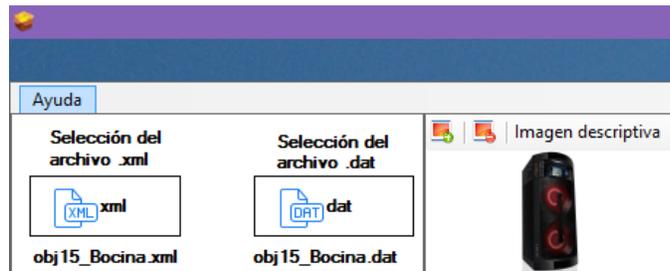
23. Cargar el archivo .dat que se descargó del portal de Vuforia.

Para cargar el archivo .dat se da clic al botón “*dat*” el cual abre una ventana en donde se debe buscar y seleccionar el archivo. Una vez que se ha cargado el archivo aparecerá el nombre debajo del botón “*dat*” como se observa en la siguiente imagen.



24. Imagen representativa del objeto analizado, en este caso el objeto “multifuncionalHP”.

Para cargar la imagen se da clic en el botón “Agregar imagen” el cual abre una ventana para la búsqueda de la imagen una vez que se cargó la imagen a la aplicación se ve de la siguiente manera.

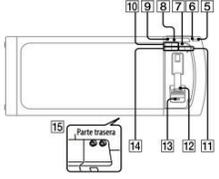
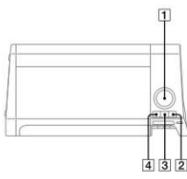
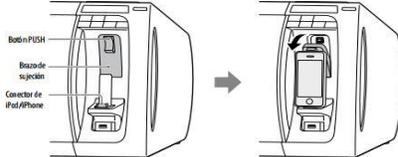
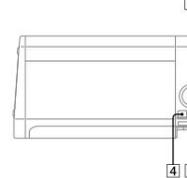
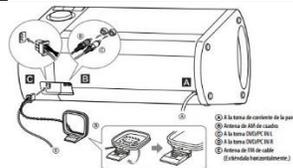
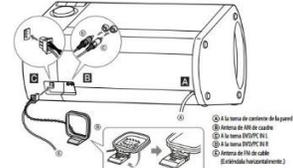


25. Cargar imágenes que van a proporcionar información contextual.

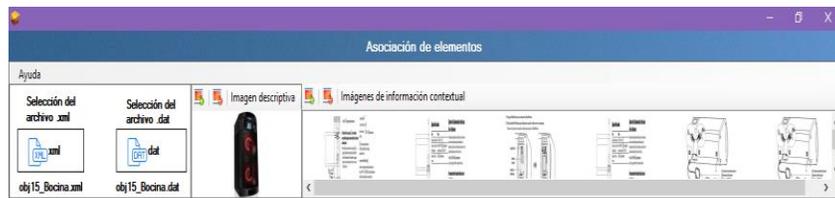
Se eligieron las siguientes seis imágenes de información contextual, una imagen por elemento detectado, antes de cargar estas imágenes de información contextual, para definir el nombre de cada imagen se utilizó la siguiente nomenclatura:

Deberá iniciar con las letras “**obj**”, seguido de un número consecutivo 01,02,.. (haciendo referencia a la base de datos que ya se cargó con anterioridad, en nuestro caso sería obj15) y el símbolo especial “_”, continua con la palabra “**elemento**” seguida del número consecutivo 01,02, ..., n dependiendo del número de elementos que se hayan identificado. Después deberá llevar el símbolo “_” y por último el nombre de la base de datos de Vuforia.

En la siguiente tabla se puede observar las imágenes correspondientes a cada elemento, así como el nombre de cada imagen de información contextual.

<p>Elemento 1</p>	<p>obj15_elemento01_obj15_Bocina.jpg</p>  <p>Pulse I/⏻ [5] para encender el sistema.</p> <p>El indicador de en espera [8] se mantiene encendido después de desconectar el cable de alimentación.</p> <p>• Es posible que el indicador de en espera [8] no se apague inmediatamente después de desconectar el cable de alimentación. El indicador se apagará después de unos 40 segundos. Esto no es un mal funcionamiento.</p> <p>Pulse FUNCTION +/- [7] repetidamente para seleccionar la función USB.</p> <p>visualizador [6] sensor del sistema [9] Seleccionar una carpeta +/- [10] repetidamente. Para avanzar rápidamente o rebobinar rápidamente, mantenga pulsado el botón. conector de iPod/iPhone [11]. Conecte un componente de audio adicional a las tonas DVD/PC IN L/R [12] de la unidad utilizando un cable de audio analógico (no suministrado). Conecte el dispositivo USB al puerto ↔ (USB) [13].</p>
<p>Elemento 2</p>	<p>obj15_elemento02_obj15_Bocina.jpg</p>  <p>Ajuste del sonido</p> <p>Para Pulse</p> <p>Ajustar el volumen VOLUME +/- [1] Generar un sonido más dinámico BASS BOOST [2] repetidamente para seleccionar "BASS ON". Ajustar el efecto sonoro EQ [3] repetidamente.</p> <p>Ajuste de la iluminación de las luces de los altavoces</p> <p>Puede ajustar la iluminación de las luces de los altavoces entre una variedad de patrones de iluminación. La iluminación de las luces de los altavoces creará efectos de iluminación basados en la fuente de música. Pulse LED SPEAKER [4] repetidamente. Cada vez que pulse el botón, el patrón cambiará.</p> <p>Para desactivar la función de las luces de los altavoces Pulse LED SPEAKER [4] repetidamente hasta que aparezca "LED OFF".</p>
<p>Elemento 3</p>	<p>obj15_elemento03_obj15_Bocina.jpg</p> <p>1 Ponga el iPod/iPhone en el conector de iPod/iPhone. 2 Pulse el botón PUSH hasta que el brazo de sujeción se libere con un chasquido. El brazo de sujeción se abate hacia afuera para sujetar el iPod/iPhone.</p>  <p>Botón PUSH Brazo de sujeción Conector de iPod/iPhone</p>
<p>Elemento 4</p>	<p>obj15_elemento04_obj15_Bocina.jpg</p>  <p>Ajuste del sonido</p> <p>Para Pulse</p> <p>Ajustar el volumen VOLUME +/- [1] Generar un sonido más dinámico BASS BOOST [2] repetidamente para seleccionar "BASS ON". Ajustar el efecto sonoro EQ [3] repetidamente.</p> <p>Ajuste de la iluminación de las luces de los altavoces</p> <p>Puede ajustar la iluminación de las luces de los altavoces entre una variedad de patrones de iluminación. La iluminación de las luces de los altavoces creará efectos de iluminación basados en la fuente de música. Pulse LED SPEAKER [4] repetidamente. Cada vez que pulse el botón, el patrón cambiará.</p> <p>Para desactivar la función de las luces de los altavoces Pulse LED SPEAKER [4] repetidamente hasta que aparezca "LED OFF".</p>
<p>Elemento 5</p>	 <p>① A la toma de corriente de la parrilla ② Antena de AM de cassette ③ A la toma DVD/PC IN L/R ④ A la toma DVD/PC IN R ⑤ Antena de FM de cable (Ejemplo de configuración)</p>
<p>Elemento 6</p>	 <p>① A la toma de corriente de la parrilla ② Antena de AM de cassette ③ A la toma DVD/PC IN L/R ④ A la toma DVD/PC IN R ⑤ Antena de FM de cable (Ejemplo de configuración)</p>

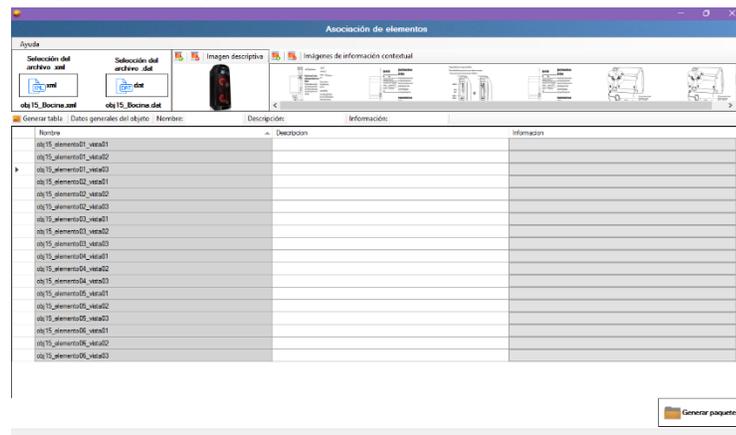
Para cargar las imágenes se da clic en el botón “Agregar imagen” el cual abre una ventana para la búsqueda de la imagen. Una vez que se cargaron las imágenes a la aplicación se puede ver una vista previa de las imágenes seleccionadas como se muestra a continuación.



26. Generar tabla para la asociación de información contextual.

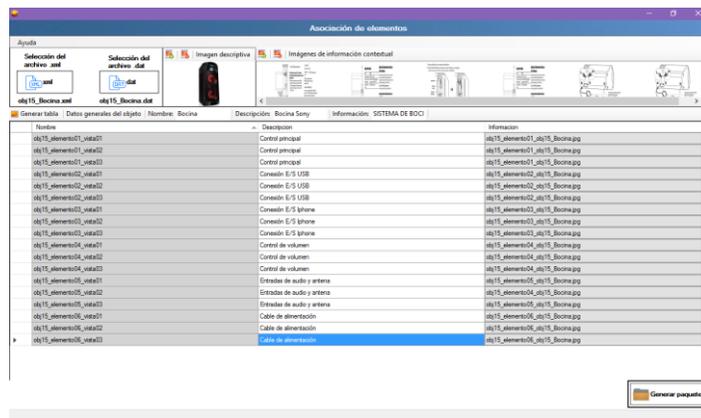
Para la generación de la tabla se da clic al botón “Generar tabla” se genera una tabla de forma automática que contiene tres columnas que son:

- Nombre: Son los nombres de las imágenes que se subieron al portal web de Vuforia.
- Descripción: En esta columna se escribe una breve descripción del elemento.
- Información: En esta columna se muestra una lista con los nombres de las imágenes de información contextual que se cargaron en el punto anterior.



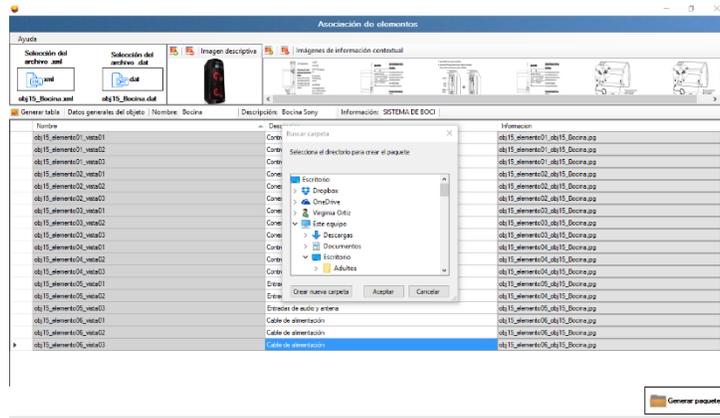
De igual manera se llenan los siguientes campos de texto que hacen referencia al objeto de interés, los cuales se describen a continuación.

- Nombre: En esta caja se escribió el nombre del objeto analizado “Bocina”.
- Descripción: En esta caja se proporcionó una descripción breve sobre el objeto analizado, en este caso el objeto “Bocina”.
- Información: En esta caja se proporciona información detallada sobre el objeto que se analizó, en este caso el objeto “Bocina”.



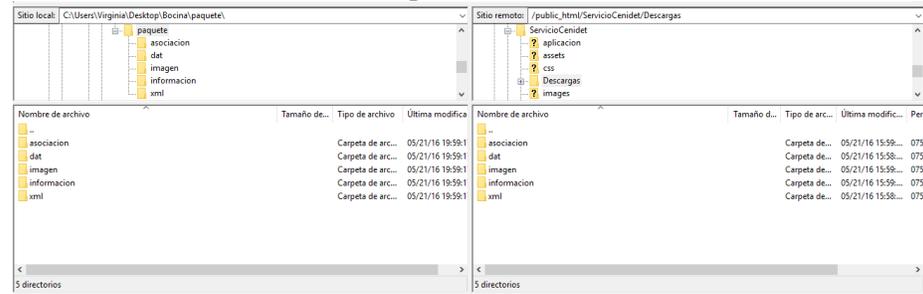
27. Generación del directorio que contiene los archivos de asociación.

Para generar el directorio del paquete se da clic en el botón “Generar paquete”, se abre una ventana para seleccionar la ruta donde se guardará el directorio de asociación



28. Subir paquete generado al servidor

Una vez que se realizó la asociación con la aplicación de escritorio y se obtuvo el directorio con sus respectivos archivos se subirán al servidor FTP:



Observación

Siguiendo los pasos de la secuencia de este caso de prueba se pudo implementar la fase 1 de la metodología propuesta de manera exitosa. De igual manera se puede observar que la aplicación de escritorio para realizar la asociación de la información contextual con los elementos de las vistas funciona correctamente.

Responsable de la prueba

Autor de tesis

6.2. Plan de pruebas de la aplicación Android

A continuación, se muestra el plan de pruebas establecido para llevar a cabo la evaluación de la aplicación móvil mediante pruebas funcionales de software.

6.2.1. Plan de pruebas

La documentación del plan de pruebas se guiará por el estándar *IEEE 829-1998*, el cual está diseñado para documentar las pruebas que se elaboran en un sistema de software comprobando la funcionalidad del mismo.

Para las pruebas del sistema se ejecutaron las funcionalidades desarrolladas, comprobando su comportamiento en el entorno de implementación y así poder corregir los errores que puedan surgir.

6.2.2. Enfoque

En estas pruebas se verificó que las funcionalidades implementadas en el prototipo Android cumplen con la funcionalidad capaz de proporcionar información contextual al usuario utilizando la tecnología de realidad aumentada.

6.2.3. Características a probar

Las características a probar en la aplicación móvil permitieron cumplir con los requerimientos funcionales del proyecto, las cuales se definen a continuación:

- **Descarga de paquetes disponibles:** la aplicación debe ser capaz de realizar la descarga de los paquetes que se encuentran en el servidor FTP.
- **Identificación de los elementos de un objeto:** la aplicación debe ser capaz de identificar elementos de los paquetes descargados y desplegar dos elementos virtuales al momento que identifica dichos elementos.
- **Visualización de elementos virtuales de información contextual:** la aplicación debe ser capaz de mostrar elementos con información contextual al momento de tocar alguno de los elementos virtuales que se despliegan cuando se realiza el reconocimiento de los elementos de los paquetes descargados.

6.2.4. Características excluidas

Las características que no serán consideradas al evaluar el sistema son las siguientes:

- **Precisión de la aplicación al identificar elementos.** Las pruebas que se realizaron en la aplicación no evaluaron la precisión con la que se reconocen los elementos, así como la visualización de los elementos virtuales eso debido a factores naturales como la luz, el uso de la cámara, la distancia con la que se enfoca el objeto, etc.

6.2.5. Elementos de prueba

- Deben existir paquetes disponibles para realizar la descarga.

6.2.6. Requerimientos para realizar las pruebas

- Smartphone con sistema operativo *Android*. Se requiere una versión mínima 4.4.
- Conexión a internet 3G/4G o mediante Wifi

6.2.7. Criterio éxito/fracaso de los casos de prueba

Se considera que el sistema funciona de manera correcta si sus funcionalidades cumplen con lo siguiente:

1. La aplicación móvil es capaz de conectarse al servidor FTP para tener acceso a la información proporcionada.
2. La aplicación móvil es capaz de mostrar las opciones de “*Configuración*” cuando el usuario presiona la pantalla dos veces seguidas.
3. La aplicación móvil es capaz de activar y desactivar la opción “*Flash*”.
4. La aplicación móvil es capaz de intercambiar el uso de la cámara trasera y delantera.
5. La aplicación móvil es capaz de mostrar los paquetes disponibles para descarga una vez que el usuario presiona la opción “*Agregar elemento*”
6. La aplicación móvil es capaz de descargar los paquetes disponibles.
7. La aplicación móvil muestra la pantalla “*Configuración*” cuando el usuario presiona la opción “*Regresar*” una vez que ha finalizado la descarga de los paquetes.
8. El sistema es capaz de “*Activar/Eliminar*” alguno de los paquetes descargados en la aplicación.
9. Una vez que se ha elegido uno de los paquetes descargados, el usuario pulsa la opción “*Regresar*” la aplicación móvil es capaz de regresar a la pantalla principal donde se encuentra activa la cámara trasera para empezar a realizar el reconocimiento de elementos.
10. La aplicación móvil es capaz de identificar los elementos del paquete que el usuario descargó.
11. La aplicación móvil es capaz de proporcionar información contextual cuando el usuario toca alguno de los dos botones virtuales.

Si alguno de los aspectos anteriores no se cumple se marcará que una de las funcionalidades ha fracasado y se revisarán los aspectos técnicos de está para corregir los posibles errores.

6.2.8. Responsabilidades

El tesista Irving Ulises Herrera Molina será el responsable de elaborar cada una de las pruebas diseñadas para esta aplicación móvil, así como las correcciones al sistema a partir de los fracasos que puedan surgir en las pruebas.

6.2.9. Reporte de pruebas

En esta sección se presentan algunas pruebas realizadas a la aplicación, mostrando los resultados obtenidos y las observaciones que se hicieron.

6.3. Casos de prueba SmartTV

A continuación, se muestran los casos de prueba implementados para el objeto SMARTVLG.

Tabla 6.6 Caso de prueba descarga del paquete SamrtTV.

Caso de prueba: 1	
Descripción: Descarga de paquetes disponibles	
Resultado:	
 <p>The figure shows three screenshots of the application. The top-left screenshot shows the 'Descargar paquetes' screen with 'camara' and 'SmartTVLG' listed, each with a download icon and a 'Regresar' button at the bottom. The top-right screenshot shows the same screen during a download, with a progress indicator and 'Descargando...' text. The bottom screenshot shows the 'Configuracion' screen with 'Paquetes' section, where 'SmartTVLG' is listed with 'Activar' and 'Eliminar' options, and a 'Regresar' button.</p>	
Observaciones: En esta prueba se realizó de manera satisfactoria la descarga de un paquete disponible de la pantalla “ <i>Descargar paquete</i> ”	
Responsable de la prueba: Irving Ulises Herrera Molina	Cargo: Autor de la prueba

Tabla 6.7 Caso de prueba identificación de elementos SmartTVLG.

Caso de prueba: 2

Descripción: Identificación de los elementos del paquete SmartTVLG

Resultado:



Observaciones: Se puede observar que se identificaron de manera correcta los elementos del paquete SmartTVLG, también se puede ver como se visualizan dos botones virtuales que proporcionan información contextual.

Responsable de la prueba: Irving Ulises Herrera Molina

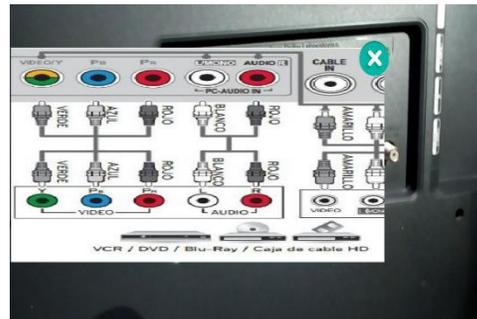
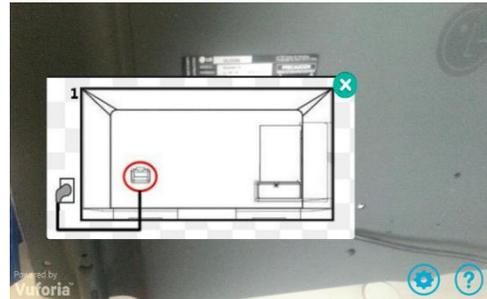
Cargo: Autor de la prueba

Tabla 6.8 Caso de uso información aumentada SmartTV.

Caso de prueba: 3

Descripción: Verificar que los botones virtuales proporcionan información contextual de los elementos del paquete SmartTVLG

Resultado:



Observaciones: En esta prueba se verificó que tocando los botones virtuales que se visualizan al identificar los elementos del paquete SmartTVLG proporcionan información contextual

Responsable de la prueba: Irving Ulises Herrera Molina

Cargo: Autor de la prueba

6.4. Casos de prueba multifuncional HP

Tabla 6.9 Caso de prueba descarga de paquete multifuncional HP.

Caso de prueba: 1	
Descripción: Descarga de paquetes disponibles	
Resultado:	
Observaciones: En esta prueba se realizó de manera satisfactoria la descarga de un paquete disponible de la pantalla “ <i>Descargar paquete</i> ”	
Responsable de la prueba: Irving Ulises Herrera Molina	Cargo: Autor de la prueba

Tabla 6.10 Caso de uso identificación de elementos multifuncional HP.

Caso de prueba: 2	
Descripción: Identificación de los elementos del paquete multifuncional HP	
Resultado:	
	
Observaciones: Se puede observar que se identificaron de manera correcta los elementos del paquete multifuncional HP, también se puede ver como se visualizan dos botones virtuales que proporcionan información contextual.	
Responsable de la prueba: Irving Ulises Herrera Molina	Cargo: Autor de la prueba

Tabla 6.11 Caso de usos elementos virtuales multifuncional HP.

Caso de prueba: 3

Descripción: Verificar que los botones virtuales proporcionan información contextual de los elementos del paquete multifuncional HP

Resultado:



Observaciones: En esta prueba se verificó que tocando los botones virtuales que se visualizan al identificar los elementos del paquete multifuncional HP proporcionan información contextual

Responsable de la prueba: Irving Ulises Herrera Molina

Cargo: Autor de la prueba

6.5. Casos de prueba Bocina

Tabla 6.12 Caso de prueba descargar paquete bocina.

Caso de prueba: 1	
Descripción: Descarga de paquetes disponibles	
Resultado:	
Observaciones: En esta prueba se realizó de manera satisfactoria la descarga de un paquete disponible de la pantalla “ <i>Descargar paquete</i> ”	
Responsable de la prueba: Irving Ulises Herrera Molina	Cargo: Autor de la prueba

Tabla 6.13 Caso de prueba identificación de elementos de la bocina

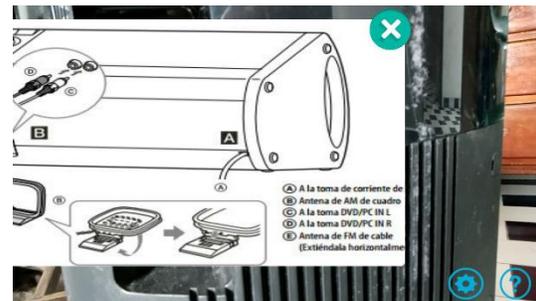
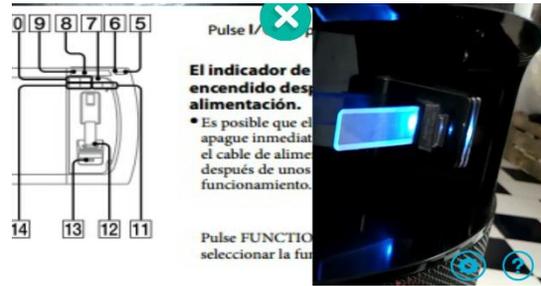
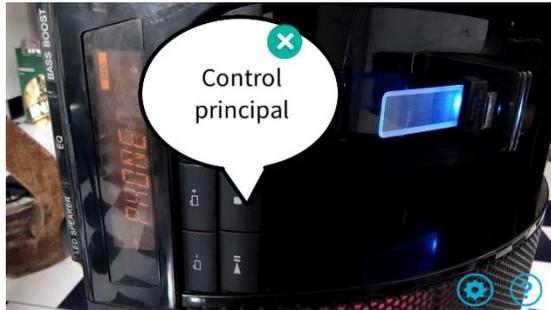
Caso de prueba: 2	
Descripción: Identificación de los elementos del paquete Bocina	
Resultado:	
	
<p>Observaciones: Se puede observar que se identificaron de manera correcta los elementos del paquete Bocina, también se puede ver como se visualizan dos botones virtuales que proporcionan información contextual.</p>	
<p>Responsable de la prueba: Irving Ulises Herrera Molina</p>	<p>Cargo: Autor de la prueba</p>

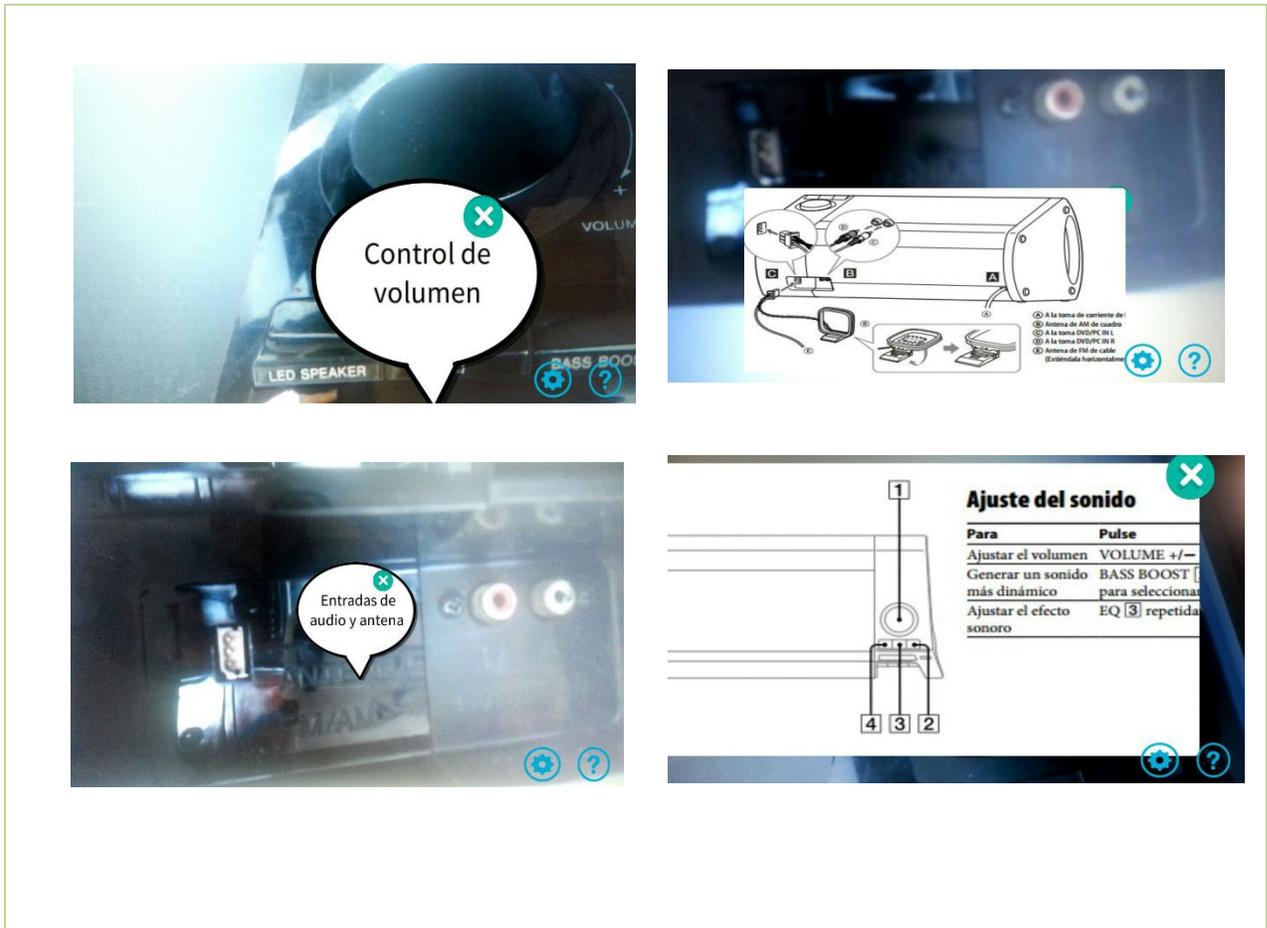
Tabla 6.14 Caso de uso elementos virtuales de la bocina.

Caso de prueba: 3

Descripción: Verificar que los botones virtuales proporcionan información contextual de los elementos del paquete Bocina

Resultado:





Observaciones: En esta prueba se verificó que tocando los botones virtuales que se visualizan al identificar los elementos del paquete Bocina proporcionan información contextual

Responsable de la prueba: Irving Ulises Herrera Molina

Cargo: Autor de la prueba

Capítulo 7

Conclusiones y Trabajos futuros

En esta sección se presentan las conclusiones generadas a partir de este proyecto de investigación. Así mismo, se describen los trabajos futuros que se pueden derivar a partir de esta tesis.

7.1 Conclusiones

Con este trabajo de investigación se desarrolló una metodología con la cual se pueden caracterizar objetos multivista, por cada vista del objeto analizado se puede realizar una desintegración de los elementos que componen la vista para posteriormente ser detectados de manera individual.

Se utilizó la tecnología de realidad aumentada que, en los últimos años, se está posicionando como un área de desarrollo tecnológico de interés para diversas áreas: capacitación, operación de maquinaria, juegos interactivos (PokemonGo), ambientes con contexto enriquecido (Museos), entornos virtuales de aprendizaje, por mencionar algunas. Esta tecnología se utilizó para poder representar información mediante elementos virtuales al momento de reconocer los elementos de un objeto.

La metodología desarrollada consta de dos fases, la fase 1 es la encargada de realizar el proceso de análisis del objeto multivista y la desintegración de los componentes de cada una de las vistas del objeto, también se lleva a cabo la asociación de los elementos de información para cada elemento que compone cada vista, en la fase 2 se desarrolló un prototipo Android que hace uso de la información generada en la fase 1 que mediante técnicas de realidad aumentada muestra la información asociada a cada elemento del objeto caracterizado.

El prototipo Android fue desarrollado con la finalidad de probar y validar la metodología desarrollada en esta tesis, la cual se puede utilizar en una aplicación de información aumentada, en donde se presentan elementos virtuales con los cuales puede interactuar el usuario y a la vez puede proporcionar información sobre los elementos detectados.

Además, esta aplicación se considera una aplicación dinámica debido a que no es necesario rehacer la aplicación si se requiere detectar nuevos objetos ya que la información generada en la metodología está disponible en la nube y puede ser descargada en la aplicación dándole así nuevas funcionalidades para detectar una variedad de objetos.

Para poder tener una metodología depurada se realizaron pruebas heurísticas de usabilidad mediante el estándar y la metodología de (Llanes, 2014), estas pruebas permitieron la evaluación de la funcionalidad y usabilidad ante usuarios reales. Las pruebas se realizaron en dos iteraciones, en la primera de ellas los usuarios detectaron las heurísticas más relevantes,

posteriormente en la segunda iteración, las fallas de diseño fueron corregidas, lo cual conllevó a tener como resultado una metodología estable.

Finalmente en este trabajo de investigación se obtuvo como resultado una metodología para la caracterización de objetos multivista así como un prototipo Android que implementa la información generada por la metodología desarrollada en esta tesis, la metodología desarrollada tiene una variedad de campos de aplicación como son: la caracterización de objetos para simuladores, manuales, guías interactivas para museos, exposiciones, para el uso de anuncios interactivos, entre otros, de igual manera puede ser implementada en diferentes dispositivos como los Smartphone, Tables, SmartGlass, etc.

7.2 Trabajos futuros

A continuación, se presentan algunos de los posibles trabajos futuros que se pueden desarrollar como continuación o mejoras para esta investigación:

- Mejoras en la aplicación de escritorio para la realización de la asociación de información a los elementos del objeto analizado, para automatizar lo más posible los pasos de la metodología propuesta.
- Implementación de la aplicación de información aumentada a nuevos dispositivos como son los smart glass (lentes inteligentes) y recientemente los lentes de contacto inteligentes.
- Desarrollo de librerías de visión artificial para realizar la caracterización de los objetos para no depender de librerías de terceros.
- Desarrollar librerías de visión artificial robustas para poder caracterizar la mayor cantidad de objetos posible.

Referencias

- Abbar, S., Bouzeghoub, M., & Lopez, S. (2009). Context-Aware Recommender Systems: A service-Oriented Approach - Artículo ACM. VLDB '09, August 24-28, 2009, Lyon, France.
- Acuna, E., Mena, P., & Torres, C. C. (2013). ForestTime: una aplicación móvil para el estudio de tiempos de trabajo de maquinaria forestal a través de teléfonos inteligentes. *SCIELO*, 359-366.
- antiago, A. &. (2014). *Detección de marcas para realidad aumentada sobre dispositivos móviles* Directoro. Madrid: Universidad Politécnica de Madrid.
- Aranda, P. J. (2015). *Aplicación Android de realidad aumentada para mostrar imágenes históricas de lugares turísticos de interés*. Valencia: Universitat Politècnica de València.
- Azuma, R. T. (1997). A Survey of Augmented Reality. *Mossachusetts Institute of Technology*, 355-385.
- Bartolome. (06 de 2015). *Dibujo Tecnico*. Obtenido de Dibujo Tecnico: <http://www.dibujotecnico.com/obtencion-de-las-vistas-de-un-objeto/>
- Carrasco Ochoa Jesús Ariel, M. T. (2011). Reconocimiento de patrones. *Komputer Sapiens*, 5-8.
- Cobo Ceballos, E. (2013). *Diseño e integración en Android de un sistema de realidad aumentada y reconocimiento de imágenes para un sistema de domótica asistencial*. Obtenido de Diseño e integración en Android de un sistema de realidad aumentada y reconocimiento de imágenes para un sistema de domótica asistencial.: <http://e-archivo.uc3m.es/handle/10016/17659>
- DAQRI Smart Helmet. (2014). Obtenido de DAQRI Smart Helmet: <http://hardware.daqri.com/smarthelmet/>
- Dawei, L., & Mooi Choo, C. (2013). EMOVIS - An Efficient Mobile Visual Search System for Landmark Recognition. *IEEE*, 53-60.
- Encalada, J. C. (2015). *MAESTRÍA EN CONTROL Y AL RECONOCIMIENTO DE ROSTROS EN VÍDEO VIGILANCIA DEL SIS-ECU-911*. Salesiana: Universidad Politécnica Salesiana.
- EPSON. (2015). *EPSON*. Obtenido de EPSON: <http://www.epson.com/cgi-bin/Store/jsp/Landing/moverio-bt-200-smart-glasses.do?UseCookie=yes>
- Framework, Y. (2015). *Yii Framework*. Obtenido de Yii Framework: <http://www.yiiframework.com/about/>
- Girod, B., & Chen, D. (2014). MemoryEfficient Image Databases for Mobile Visual Search. *IEEE*, 13-23.
- Gonzalez Rafael C., E. W. (2001). *Digital Image*. United States of America: Prentice-Hall.

- Herrero, M. I. (2013). *Realidad Aumentada: ARToolkit para animación de personajes*. . Valencia: Universidad politecnica de valencia.
- HoloLens, M. (2015). *Microsoft HoloLens*. Obtenido de Microsoft HoloLens: <https://www.microsoft.com/microsoft-hololens/en-us>
- Houqiang , L., Yang , W., Tao , M., Jingdong , W., & Shipeng, L. (2013). Interactive Multimodal Visual Search. *IEEE*, 594-607.
- Ilbay, M. Ñ. (2013). *ANÁLISIS DE EFICIENCIA EN ALGORITMOS DE RECONOCIMIENTO DE IMÁGENES DIGITALES APLICABLES A DISPOSITIVOS MÓVILES BAJO LA PLATAFORMA ANDROID*. Obtenido de ANÁLISIS DE EFICIENCIA EN ALGORITMOS DE RECONOCIMIENTO DE IMÁGENES DIGITALES APLICABLES A DISPOSITIVOS MÓVILES BAJO LA PLATAFORMA ANDROID: <http://repositorio.espe.edu.ec/handle/21000/7770>
- Jakob Nielsen, R. M. (1990). Heuristic Evaluation of User Interfaces. *ACM*, 249-256.
- Javier Fombona Cadavieco, M. Á. (2012). *REALIDAD AUMENTADA, UNA EVOLUCIÓN DE LAS APLICACIONES DE LOS DISPOSITIVOS MÓVILES*.
- Javier Fombona Cadavieco, M. Á. (2012). Realidad aumentada, una evolución de las aplicaciones de los dispositivos móviles - Artículo ISSN: 1133-8482 Píxel-Bit. *Revista de Medios y Educación*.
- Ketan Parmar (KP Bird)*. (2015). Obtenido de Ketan Parmar (KP Bird): <http://www.kpbird.com/2013/09/android-augmented-reality.html>
- Kioskea. (Junio de 2014). *Kioskea.net*. Obtenido de Kioskea.net: <http://es.kioskea.net/contents/642-sistema-de-informacion>
- labs, A. (2015). *Atheer labs*. Obtenido de Atheer labs: <https://www.atheerlabs.com/>
- Leal, R. A. (2012). *Servicios de Recomendación Contextual para Instituciones de Educación Superior Mediante Realidad Aumentada Utilizando Smartphones y Ontologías Organizacionales - Tesis de maestría*. CENIDET Cuernavaca, Morelos: CENIDET.
- Llanes, J. Y. (2014). *Metodología para evaluación de SRSC centrada en el usuario, basada en características de efectividad, confianza y satisfacción mediante interfaces multimodales sobre dispositivos móviles multisensoriales*. Cuernavaca, Morelos: CENIDET.
- Manta, W. A. (2014). *DESARROLLO DE UN APLICATIVO PARA LA GENERACION DE MOSAICOS 2D ENFOCADO AL SEGUIMIENTO DE INFRAESTRUCTURAS LINEALES DE TRANSPORTE DE ENERGIA*. Granada: Granada.
- Nielsen, J. (1995). 10 Heuristics for User Interface Design: Article by Jakob Nielsen. *ACM*.

- ODG. (2015). *ODG GROUP*. Obtenido de ODG GROUP: <http://www.osterhoutgroup.com/system>
- Parmar, K. (2015). *Ketan Parmar (KP Bird)*. Obtenido de Ketan Parmar (KP Bird): <http://www.kpbird.com/2013/09/android-augmented-reality.html>
- Ramos, B. R. (2014). *SISTEMA DE INFORMACIÓN AUMENTADA PARA NAVEGACIÓN INDOOR-OUTDOOR BASADO EN MODELOS SEMÁNTICOS ORGANIZACIONALES*. Cuernavaca.
- Ruiz-Shulcloper J., G. A.-T. (1999). *Enfoque Logico Combinatorio al Reconocimiento*. Mexico: IPN.
- Serrano, A. (2012). *Herramientas de desarrollo libres para aplicaciones de Realidad Aumentada con Android*. Valencia: Valencia.
- Shiliang Zhang, Q. T. (2013). Edge-SIFT: Discriminative Binary Descriptor for Scalable Partial-Duplicate Mobile Search. *IEEE*, 2889-2902.
- Slick, J. (17 de 07 de 2016). *About Tech*. Obtenido de About Tech: <http://3d.about.com/od/3d-101-The-Basics/a/Rendering-Finalizing-The-3d-Image.htm>
- SQLite. (2015). *SQLite*. Obtenido de SQLite: <https://www.sqlite.org/about.html>
- Tanacs, A., & Varjas, V. (2013). Car Recognition from Frontal Images in Mobile Environment. *ISPA*, 819-823.
- Torre, M. B. (2015). *Framework de realidad aumentada para la extraccion y clasificacion de informacion visual de contexto - Tesis de maestria*. Cuernavaca, Mexico: CENIDET.
- Unity. (2016). *Unity*. Obtenido de Unity: <http://unity3d.com/es/public-relations>
- Vuforia. (2016). *Vuforia*. Obtenido de Vuforia: <https://developer.vuforia.com/support>
- Wikitude. (2015). *Wikitude*. Obtenido de Wikitude: <http://www.wikitude.com/>
- X. Basogain, M. O. (2007). *Realidad Aumentada en la Educación: una tecnología emergente*.

Anexos

Anexo A. Análisis de los principales SDK's de Realidad Aumentada y detección de objetos (Amin & Govilkar, 2008)

Introducción

La expresión “Realidad Aumentada”, usualmente abreviada como AR, hace referencia a la tecnología emergente que permite la mezcla en tiempo real de la información digital procesada por una computadora con la información procedente del mundo real por medio de interfaces.

La realidad aumentada es una tecnología de información integral que combina el procesamiento digital de imágenes, gráficos de computadora, inteligencia artificial, la tecnología multimedia y otras áreas.

La realidad aumentada utiliza gráficos asistidos por computadora para agregar una capa adicional de información que ayuda a la comprensión y/o interacción con el mundo físico a su alrededor. La definición comúnmente aceptada para la realidad aumentada es: Generación de una imagen virtual en la parte superior de una imagen real, permitiendo la interacción en tiempo real, sin problemas mezclando objetos virtuales 3D (o 2D) con objetos reales.

La Realidad Aumentada hace explícito lo implícito; esto significa que la información que está asociada implícitamente en algún contexto se hace utilizable y directamente accesible por medio de la interfaz de AR. A continuación, se muestra un resumen de las principales características de los SDK's de realidad aumentada mayormente utilizados.

SDK's de Realidad Aumentada y detección de objetos

Metaio

Metaio SDK es un framework modular constituido por diferentes componentes como renderizado, captura, tracking e interfaz de sensor que junto con la interfaz Metaio SDK interactúan entre la aplicación y los demás componentes modulares. Los detalles de las implementaciones se encapsulan y las diferentes funcionalidades se realizan a través de APIs SDK la cual conecta con las otras partes del SDK, por tanto, se proporciona una fácil implementación para aplicaciones de RA.

El SDK Metaio es compatible con las principales plataformas de desarrollo de software: Android, iOS, Unity3D y Windows. Las interfaces específicas de la plataforma del SDK Metaio apoyan la fácil interacción con cualquier entorno de desarrollo. Su conjunto de características

contiene marcadores 2D y 3D de tracking, tracking POI's, soporte para el código QR y lectura de códigos de barras, construcción en renderizado 3D, optimizaciones para chips móviles, tracking Marker LLA, etc.

El metaio Mobile SDK está implementado de forma modular, al igual que los anteriores, de forma que divide la actividad de RA en tres componentes ver figura A.1: Tracking, Captura y Renderizado:

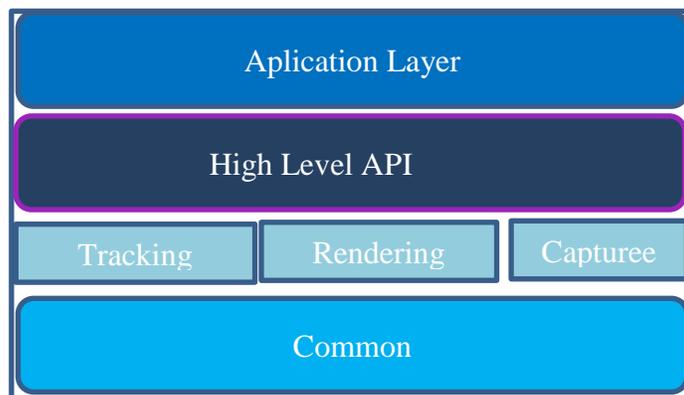


Figura A.1 Estructura del Metaio Mobile SDK.

En la versión gratuita la etapa de tracking está oculta bajo una aplicación estándar configurable mediante un archivo de configuración .XML. (Cobo Ceballos, 2013).

Algunas características:

- Proporciona un alto nivel de abstracción el cual permite al desarrollador asumir un bajo nivel de implementación en el diseño y el desarrollo de apps con mayor facilidad, eficiencia además de mejorar la productividad.
- Proporciona potente motor de renderizado 3D y tracking altamente anticipado que guía el desarrollo de app de artefactos visuales y se puede ejecutar en un plazo razonable de tiempo en dispositivos de bajos costos.
- Metaio soporta formatos OBJ, FBX y modelo md2 para objetos 3D que proporcionan una mayor flexibilidad en el desarrollo de aplicaciones.
- La carga directa de los modelos 3D es compatible con la abstracción de OpenGL que permite un uso más rápido y más eficiente de los recursos del Smartphone como la memoria del dispositivo y también minimiza los tiempos de E / S de carga tal que no se requiere ninguna conversión para formato 3D.
- Incluye AREL, un lenguaje de scripting de gran alcance que proporciona experiencias de Realidad Aumentada altamente interactivas basadas en tecnologías web comunes, como XML, HTML 5 y JavaScript.

Vuforia

Vuforia es un SDK desarrollado por Qualcomm, una empresa productora de chipsets para tecnología móvil. En 2010 la empresa lanzó algunas aplicaciones propias que hacían uso de tecnologías de RA, y finalmente ese mismo año anunció que ponía a disposición de los desarrolladores sus frameworks de desarrollo al que denominaron Vuforia. Está disponible para Android e iOS y se basa en el reconocimiento de imágenes basado en características especiales, por lo que también soporta marcadores naturales (targets) o RA sin marcadores. Además dispone de un plugin para interactuar con Unity3D y ofrece la posibilidad de crear botones virtuales para ampliar las vías de interacción con el usuario (Serrano, 2012).

La plataforma Vuforia utiliza una visión por computadora basado en el reconocimiento de imagen superior, estable y técnicamente eficiente y ofrece el más completo conjunto de características y capacidades, dando a los desarrolladores la libertad de extender sus puntos de vista sin limitaciones técnicas.

La plataforma Vuforia consta de diferentes componentes como Target Management System (Sistema de gestión de objetivos) disponible en el portal para desarrolladores, Cloud Target Database (Base de datos de objetivos en la nube) Device Target Database (Base de Datos de dispositivos objetivos) y el motor de Vuforia, en la figura A.2 se puede apreciar los componentes que integran el SDK de Vuforia.

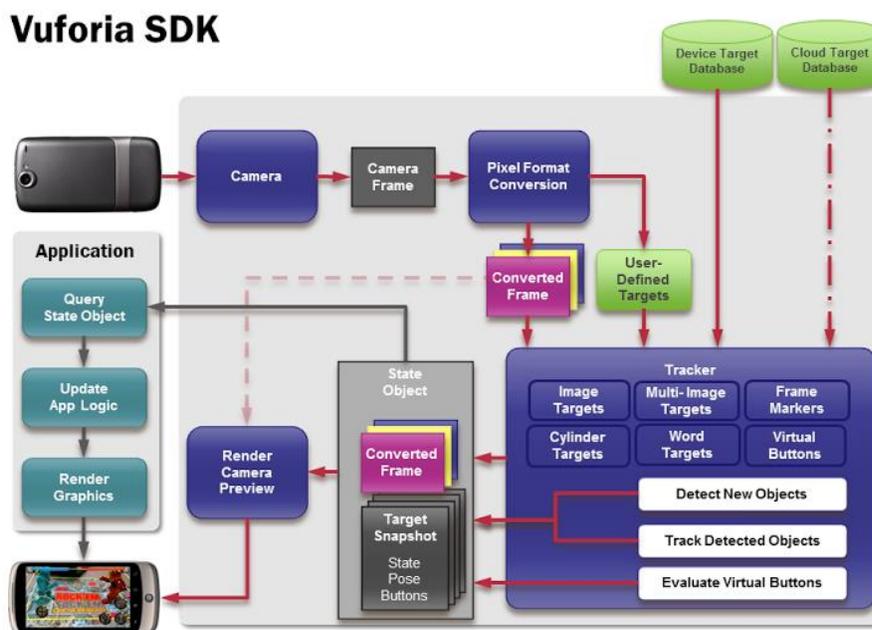


Figura A.2 Arquitectura del SDK Vuforia

Una aplicación basada en el SDK Vuforia se compone de los siguientes elementos (IñarraSagüés, 2012):

- **Cámara**

La cámara se asegura que cada frame es capturado y pasado eficientemente al Tracker. El desarrollador sólo ha de decir cuándo ha de comenzar y terminar de capturar. La imagen es entregada automáticamente en un formato y tamaño dependiendo del dispositivo.

- **ImageConverter**

Se encarga de convertir la imagen capturada por la cámara en un formato aceptable para que OpenGL ES la renderice y trate. Esta conversión también incluye una reducción del muestreo para tener la imagen de la cámara en diferentes resoluciones para diferentes procesos.

- **Tracker**

Es la parte encargada de detectar y rasterizar la imagen captada mediante algoritmos computacionales de visionado. Partiendo de la imagen captada por la cámara, se ocupa de detectar nuevos marcadores, objetos y de evaluar posibles botones virtuales. El resultado es almacenado en un objeto usado por el renderizador de video y que puede también ser accesible desde el código del desarrollador. Puede cargar distintos conjuntos de datos, patrones de reconocimiento, pero sólo uno puede estar activo al mismo tiempo.

- **Renderizador de Video de Fondo**

El renderizador de video de fondo realiza la tarea de analizar la imagen almacenada por la cámara en el objeto de estado. El rendimiento de la representación de vídeo de fondo está optimizado para cada dispositivo específico.

- **Código de Aplicación**

El desarrollador debe inicializar todos los componentes y seguir 3 pasos claves en el código de la aplicación. Para cada frame procesado, el objeto de estado es actualizado y el método de renderizado es llamado. El código debe de hacer:

- Preguntar al objeto de estado si ha detectado nuevos objetos, marcadores o actualizar el estado de estos elementos.
- Actualizar la lógica de la aplicación con los nuevos datos entrantes.
- Renderizar los gráficos correspondientes por encima de la imagen de la cámara.

- **Objeto a reconocer**

Los objetos a reconocer son creados mediante la página web con el “Target Management System”. El conjunto de datos descargados tras el proceso de creación contiene un XML de configuración que permite al desarrollador configurar algunas características del objeto y un fichero binario que contiene el patrón que permite identificar al mismo. Estos ficheros son compilados junto a la aplicación dentro del instalador y son usados en tiempo de ejecución por el SDK Vuforia.

Vuforia SDK es compatible con el desarrollo nativo para iOS y Android, al tiempo que permite el desarrollo de aplicaciones de RA en Unity que son fáciles de transportar a ambas plataformas.

El SDK Vuforia soporta diferentes tipos de objetivos, tanto en 2D y 3D, incluyendo configuraciones multi-objetivo, objetivos cilíndricos para rastrear imágenes en una superficie cilíndrica y objetivos de reconocimiento de la nube para hacer un seguimiento de 1 millón de blancos simultáneamente.

El SDK proporciona características como detecciones de oclusión localizadas utilizando los botones virtuales, selección de imágenes objetivo en tiempo real y tiene la capacidad de volver a configurar y crear conjuntos de objetivos en función del escenario. También soporta objetivo de texto para reconocer y realizar un seguimiento de palabras.

Algunas características:

- Proporciona detección local más rápida de objetivos con la capacidad de rastrear 5 objetivos simultáneamente.
- Seguimiento eficiente en condiciones de poca luz y aunque el objetivo este parcialmente cubierto.
- Capacidades de seguimiento extendido, que permiten habilitar la aplicación para mantener los objetivos de seguimiento y ayuda a mantener una referencia constante para los aumentos de objetos, incluso cuando los objetivos ya no son visibles en la vista de la cámara en tiempo real.

Wikitude

Wikitude SDK es un framework “todo en uno” que proporciona servicios de realidad aumentada. Entre estos servicios podemos encontrar (Aranda, 2015):

- Reconocimiento de imágenes
- Renderizado de modelos 3D

- Superposición de video e imágenes.
- Localización basada en realidad aumentada.

El SDK Wikitude combina capacidades basadas en geo y reconocimiento de imágenes para proporcionar tracking híbrido y que se construye en gran medida de tecnologías web (HTML, JavaScript y CSS), que permitirá escribir experiencias de realidad multiplataforma aumentada acuñados como mundos *ARquitectonicos* y son básicamente páginas HTML que pueden utilizar la API de arquitecto para crear objetos en realidad aumentada. SDK Wikitude puede ser integrado a las aplicaciones añadiendo el componente de vista específico de la plataforma llamada ARchitectView a la interfaz de la aplicación de usuario.

Wikitude SDK es una solución comercial, pero también está disponible como una versión de prueba con algunas limitaciones como el logo Wikitude en la vista de cámara etc. Wikitude SDK está disponible para plataformas Android y iOS.

El navegador Wikitude AR fue lanzado por primera vez en la plataforma Android y luego extendido a Blackberry, Bada, Windows Phone y iOS.

Wikitude también proporciona studio Wikitude que facilita el procedimiento de desarrollo, donde no se requieren conocimientos de programación y las aplicaciones se pueden crear con sólo arrastrar objetos en la pantalla de studio.

En la Figura A.3 se muestran los diferentes componentes del SDK Wikitude y los posibles enfoques para crear aplicaciones de Realidad Aumentada (<http://www.wikitude.com/>, 2015):

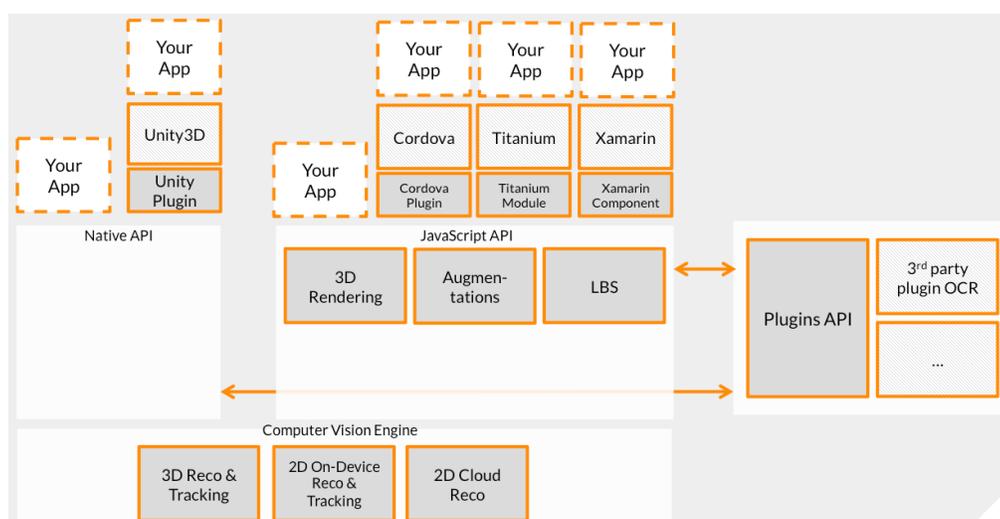


Figura A.3 Arquitectura del SDK Wikitude.

D'Fusion

La suite D'Fusión establecida por Total Inmersión es una plataforma multifacética que apoya el diseño de proyectos de Realidad Aumentada para aplicaciones basadas en web y móviles. Es una herramienta multiplataforma que soporta diferentes plataformas como Web, iPhone y Android.

La tecnología D'Fusión integra contenidos gráficos 3D interactivos en tiempo real en una secuencia de vídeo en vivo. Usando D'Fusión Exporter 3D se pueden crear y exportar objetos. Estudio de D'fusión se utiliza para diseñar y exportar los proyectos AR. D'Fusión SDK está disponible en diferentes plataformas como escritorio, móviles y también está disponible un plug-in de Flash.

El renderizado se basa en D'FusionCV; puede reconocer código de barras de 1D y 2D. La detección de movimiento característica disponible con D'Fusión SDK sustituye a la utilización de un ratón, un teclado o una pantalla táctil mediante movimientos corporales identificados. También reconoce múltiples tipos de marcadores. Se ofrecen marcador sin tracking y soporta cualquier tipo de objetos 2D y 3D. Soporta seguimiento facial que permite un reconocimiento robusto de caras, centrándose en la detección de los ojos y la boca. Proporciona gran conjunto de 500 imágenes en la plataforma móvil de datos. Los marcadores sin seguimiento se apoyan en la plataforma móvil utilizando diferentes sensores y se basa en el lenguaje LUA, la arquitectura de D'Fusión se puede observar en la figura A.4. Cualquier dispositivo que puede ser conectado al escritorio se puede utilizar como cámara para crear una escena AR.

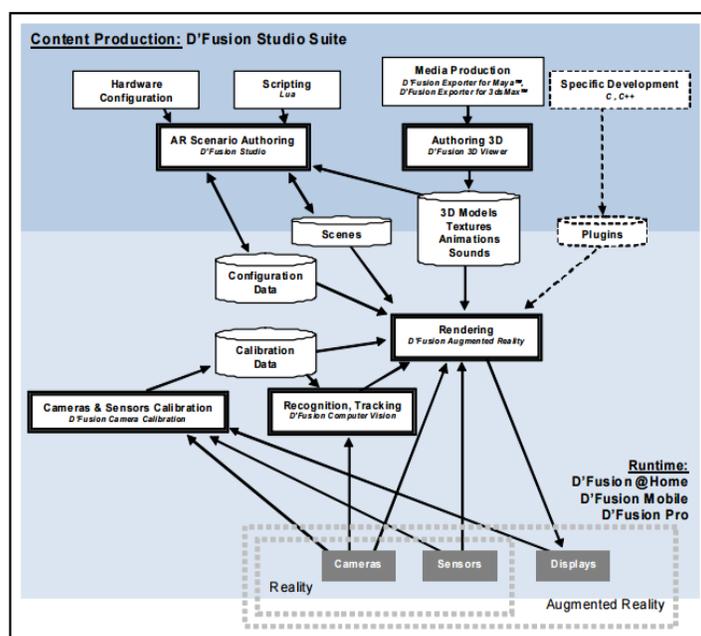


Figura A.4 Arquitectura del SDK D'Fusion

Algunas características:

- Interfaz de usuario basada (D'Fusion Studio & D Fusión CV) permite construir todo el escenario a través de la interfaz gráfica de usuario (GUI).
- Un escenario paquete funcionará tanto en Android e iPhone.
- Soporta multi-tag y Face-tracking.

ARToolkit

ARToolkit es un conjunto de librerías para C/C++ que sirven para la creación de aplicaciones de realidad aumentada. Para ello proporciona una serie de funciones para la captura de vídeo y para la búsqueda de ciertos patrones, en las imágenes capturadas, mediante técnicas de visión por computadora. También proporciona una serie de ejemplos y utilidades de gran ayuda al programador que quiera realizar este tipo de aplicaciones(Herrero, s. f.).

ARToolkit soporta múltiples plataformas y para el renderizado utiliza OpenGL; GLUT se utiliza para las ventanas/evento y una biblioteca de vídeo depende del hardware y se proporciona una API estándar para cada plataforma.

La biblioteca ARToolkit consta del módulo AR que contiene rutinas de seguimiento de marcador, la calibración y la recopilación de parámetros, frames de vídeo, sub módulo que es una colección de rutinas gráficas basadas en las librerías OpenGL y GLUT, el módulo sub_Lite que tiene una colección de rutinas gráficas, que contiene rutinas de vídeo para capturar la entrada del mismo, como se puede observar en la figura A.5.

Fue diseñado para computadoras personales y no para dispositivos embebidos. De ahí portarlo directamente a las plataformas móviles era difícil y poco práctico, ya que utiliza una gran cantidad de cálculos FPU. Pero hasta el momento se han desarrollado muchas de las extensiones de ARToolkit con soporte para plataformas móviles.

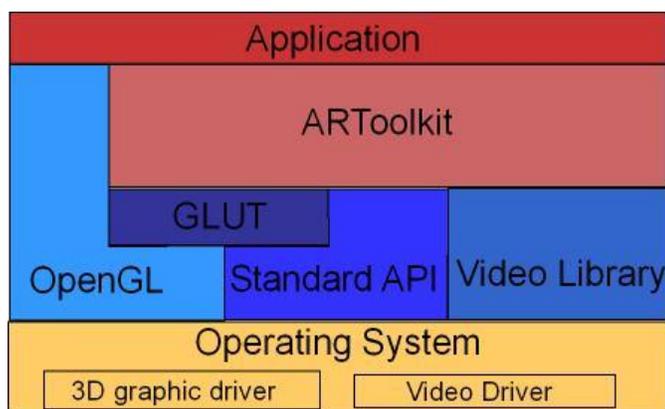


Figura A.5 Arquitectura ARToolkit SDK.

Algunas características:

- Tiene la capacidad de utilizar cualquier patrón de marcadores cuadrados también pueden ser rastreados cuadrados negros simples.
- Fácil código de calibración de la cámara.
- El seguimiento es lo suficientemente rápido para aplicaciones de RA en tiempo real.
- Proporciona videotecas multiplataforma con múltiples fuentes de entrada (USB), soporta el formato múltiple (RGB, YUV), el seguimiento de múltiples cámaras.

ARmedia

El SDK ARmedia 3D se basa en un enfoque de seguimiento de modelo en 3D que reconoce imágenes planares, así como objetos complejos en 3D con independencia de su tamaño y la geometría. La arquitectura SDK consiste en renderizadores para renderizar el modelo 3D, rastreador para rastrear el destino, la captura para la captura de fotogramas de la cámara del dispositivo y la interfaz de Android y iOS nativa. ARmedia es multiplataforma e implementado en C / C ++.

La aplicación AR se puede crear con este SDK tomando primero fotos del objeto con el cual desea interactuar el usuario final, también debe ser creado el modelo 3D que luego se superpone sobre el rastreador que se crea en el servidor SDK en línea.

El Framework incluye diferentes librerías de seguimiento como OpenCV, rastreadores de características naturales, OpenNI que proporciona el reconocimiento de gestos con las cámaras de profundidad y un potente rastreador 3D: InGlobe Tracker, ver figura A.6. La plataforma soporta varias extensiones de OpenGL para renderizado en tiempo real.

El SDK ARmedia permite la creación de aplicaciones que redefinen los límites entre la realidad aumentada interior y exterior en distintos ámbitos como el mantenimiento industrial, juegos interactivos, etc.

ARmedia SDK 3D está disponible para móviles y gafas inteligentes compatibles y pronto estará disponible también para PCs con Windows y Mac OS.

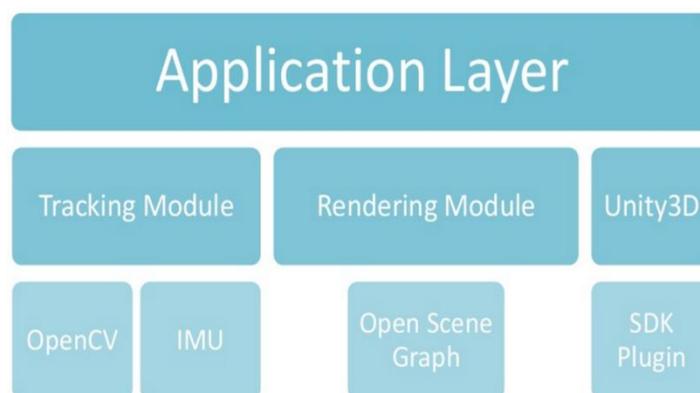


Figura A.6 Arquitectura de ARMedia SDK.

Algunas características:

- Proporciona seguimiento 3D de los objetos del mundo real en tiempo real con condiciones de luz cambiantes.
- Proporciona la modularidad para poder integrar los diferentes algoritmos de seguimiento, diferentes motores 3D y la interfaz.
- Pueden ser fijados diferentes parámetros de seguimiento para afinar los resultados.
- Puede ser integrado fácilmente con cualquier otra plataforma AR simplemente al convertir el marco a objeto OpenCV Mat y pasarlo al tracker.
- Los servicios de creación y gestión de destino en 3D están disponibles en la nube.

Comparación de los sdk's

Basada en los tipos de licencia:

SDK's	Vuforia	Metaio	Wikitude	ARToolKit	D'Fusion	ARmedia
Tipo de licencia						
Open Source	x	x	x	✓	x	x
Free	✓	✓	✓	✓	✓	✓
Comercial	✓	✓	✓	✓	✓	✓

Basada en la plataforma soportada:

SDK's	Vuforia	Metaio	Wikitude	ARToolKit	D'Fusion	ARmedia
Tipo de licencia						
iOS	x	✓	x	✓	✓	✓
Android	✓	✓	✓	✓	✓	✓
Windows	✓	✓	✓	✓	✓	✓

Basado en generación de marcadores:

SDK's	Vuforia	Metaio	Wikitude	ARToolKit	D'Fusion	ARmedia
Generación de marcadores	Administrador de objetos Online. Apoyo a la generación de frames, marcadores de imagen.	No posee herramienta Online. Proporciona 512 marcadores de identificación diferentes	Herramienta administradora Online de objetivos. Proporciona creación de colección de destino de múltiples objetivos.	Herramienta Online para crear marcadores. Proporcionar un conjunto de marcadores cuadrados predefinidos en un archivo PDF.	D'Fusion studio. Proporciona conjuntos de datos de 500 imágenes para utilizar como marcador.	Proporcionar plug-in para Google SketchUp. Proporciona un conjunto de marcadores predefinidos.

Basado en tracking

SDK's	Vuforia	Metaio	Wikitude	ARToolKit	D'Fusion	ARmedia
Tipo						
Marcadores	Marcadores Frame, objetivos de imagen, texto objetivo	ID, marcadores de imagen y LLA, QR y códigos de barra	Imágenes y seguimiento de códigos de barra.	Marcadores cuadrados, seguimiento múltiple de marcadores	Seguimiento múltiple de marcadores, seguimiento de código de barras	Seguimiento de marcador de referencia.
GPS	x	✓	✓	x	✓	✓
IMU	x	✓	✓	x	✓	✓
Face	x	✓	x	x	✓	x
N	✓	✓	✓	✓	✓	✓
Objetos 3D	✓	✓	x	x	✓	✓
Otros	Seguimiento extendido, Detección de oclusión localizada	Mapas 3D instantáneos de seguimiento, SLAM 3D, seguimiento	Seguimiento híbrido, seguimiento extendido	Marcadores de seguimiento 6D (detección plana en tiempo real)	Reconocimiento de zona interactiva a través de seguimiento señalado.	Proporciona calibración de la cámara y la profundidad IR de la cámara.

Comparación basada en la capacidad

SDK's		Vuforia	Metaio	Wikitude	ARToolKit	D'Fusion	ARmedia
Capacidad de superposición	Contenido 2D	✓	✓	✓	✓	✓	✓
	Contenido 3D	✓	✓	✓	✓	✓	✓
	Otros	Animación en 3D puede ser superpuesta en la pantalla	La cartelera puede ser superpuesta	Animaciones Sprite, transformaciones en 3D y contenido HTML	Soporte para contenido de gráficos de alto nivel u animaciones.	Animación en 3D puede ser superpuesta en la pantalla	Animación interactiva en 3D a través de SketchUp

Beneficios de los diferentes SDK's

Vuforia	Es capaz de mantener el seguimiento, incluso cuando el objetivo está fuera de la vista y verlo desde mayor distancia. La base de datos en la nube permite almacenar miles de objetivos de imagen.
Metaio	Potente motor de renderizado 3D con capacidad de cargar modelos 3D en formato .obj. No hay límite en el número de objetos rastreables depende de la memoria del dispositivo.
Wikitude	El contenido de AR se puede programar usando HTML5, JavaScript y CSS. Fácil portabilidad de aplicaciones de RA de una plataforma a otra.
ARToolKit	Múltiples plataformas AR posibles para el desarrollo de aplicaciones. Está disponible como open source, a través del cual muchos nuevos frameworks AR se desarrollan.
D'Fusion	Contenido 3D de alta calidad puede ser aumentada con soporte para múltiples formatos de objetos 3D. Proporciona encriptación para prevenir riesgos de privacidad.
ARmedia	Calibración de la cámara de profundidad prevista que crea la experiencia más envolvente.

Limitaciones de los diferentes SDK's

Vuforia	Vuforia SDK para Android no expone ninguna función de utilidad para cargar fácilmente un modelo 3D a partir de cualquier formato estándar. La base de datos del dispositivo sólo puede admitir 100 objetivos de imagen.
Metaio	Dificultad para renderizar objetos complejos en 3D también la limitación está asociada con el tamaño del modelo.
Wikitude	No rastrea modelos 3D lo cual ha utilizar sólo el seguimiento 2D. El seguimiento de las imágenes objetivo tiene que ser de colores sólidos para ser reconocido
ARToolKit	Menos precisión en marcadores de seguimiento, incluso cuando la cámara y el marcador se incluyen. No soporta localización basada en realidad aumentada.
D'Fusion	Soporta archivos de video, pero el audio asociado con el vídeo no se puede reproducir.
ARmedia	No es compatible con todo tipo de texturas de objetos 3D

OpenCV

OpenCV (*Open Source Computer Vision*) es una librería que contiene funciones de Visión por Computador en tiempo real. La primera versión alfa fue originalmente desarrollada por Intel en enero de 1999 (Cobo Ceballos, 2013).

OpenCV es multiplataforma, es decir, las interfaces C, C++, Python y Java que contiene, funcionan a la perfección en diversos Sistemas Operativos: Windows, GNU/Linux, Android y Mac OS X.

Estas librerías contienen más de 500 funciones optimizadas, que abarcan distintas áreas en el proceso de Visión por Computador, como el reconocimiento de objetos, calibración de cámaras, visión estéreo y visión robótica avanzada. Gracias a esta diversidad, y a que OpenCV fue publicada bajo la licencia BSD, que permite su uso libre tanto para propósitos comerciales como de investigación, se ha utilizado en infinidad de aplicaciones, desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere el reconocimiento de objetos.

Debido a su optimización, que hace de OpenCV una herramienta muy eficiente, y a su licencia BSD que permite, que sea usada libremente tanto para propósitos comerciales como de investigación, OpenCV goza de un buen reconocimiento y es usado en numerosos proyectos como el sistema de visión del vehículo no tripulado Stanley de la Universidad de Stanford (Santiago & García, 2015).

Las librerías OpenCV se dividen en cinco grandes grupos:

- **CXCORE**: Estructuras básicas, operadores aritméticos/lógicos (copia, Transformación), estructuras dinámicas (asignación, grafos, árboles), funciones de dibujo, errores de enlace y funciones de sistema.
- **CV**: En este archivo están definidas las funciones de procesamiento general de imágenes, filtros, conversión de colores, análisis de movimiento, operadores morfológicos, análisis, reconocimiento de patrones (detección de objetos), calibración de cámaras y reconstrucción 3D.
- **HighGUI**: en él se encuentra todo lo relacionado con la interfaz gráfica de OpenCV, brinda herramientas para trabajar con imágenes y vídeos guardados en archivos u obtenidos directamente desde cámaras
- **ML**: aquí se encuentran algoritmos de aprendizaje y clasificadores.
- **CvAux**: Correspondencia estéreo, descriptores de textura, procesamiento 2D- 3D, segmentación de fondo, morfología además de que contiene algoritmos experimentales.

Algoritmos de Reconocimiento de patrones

Actualmente existen varios algoritmos y todos se basan en detectar puntos de interés (keypoint detection”) en las imágenes que sean invariantes al escalamiento, la rotación y los colores. Entre los más reconocidos se pueden nombrar: SURF, SIFT y ORB.

A continuación, se hace una breve descripción de estos algoritmos:

Algoritmo SURF (Speeded Up Robust Features)

Es un detector robusto de características locales que fue estudiado y presentado por Herbert Bay en 2006, el cual introduce el concepto de detección de puntos de interés locales e invariantes. SURF por tanto es invariante a las transformaciones comunes de las imágenes como lo es la rotación, cambio en las escalas, cambios de iluminación y pequeños cambios en la perspectiva de visión.

Dentro de esta técnica también es utilizado el concepto de imagen integral (suma de áreas), las cuales son una representación intermedia de la imagen y contienen la suma de los valores de los píxeles de una imagen en una escala de grises, de esta manera se reduce también el tiempo de cómputo.

Este algoritmo tiene como ventaja, que el tiempo de procesamiento es menor, respecto a otros algoritmos de descripción de características como SIFT. El desarrollo del algoritmo SURF está dividido en tres etapas (Fernandez, 2014):

- Generación de punto de interés
- Obtención de los descriptores
- Comparación de los descriptores

El descriptor usa ondas de Wavelets de Haar con integrales de imágenes para decodificar la intensidad del pixel en el vecindario el primer paso de la computación del descriptor es asignarle una orientación para hacer el descriptor invariante a las rotaciones, se ubica una ventana cuadrada con centro en el punto hallado y con la orientación antes realizada, esta ventana se divide en 4x4 subregiones y a su vez cada subregión en 5x5. Cada subregión contribuye al descriptor con cuatro componentes y la suma de estas a las dos direcciones principales (Manta, 2014).

Algoritmo SIFT (Scale Invariant Feature Transform)

Es un descriptor de imágenes basado en el desarrollo realizado por David Lowe entre 1999 y 2004. Este procedimiento tiene la ventaja de ser invariante a las posibles distorsiones debidas a traslación, rotación y escala de la imagen a ser estudiada y posee cierto grado de robustez ante transformaciones debidas a las variaciones de iluminación o del ángulo de perspectiva con el cual se capte al objeto de estudio.

En base lo que este método plantea es la detección de puntos de interés dentro de una matriz de representación de un objeto en el cual, y a través de la sumatoria de las intensidades de los distintos gradientes, se obtenga al aplicar el proceso de diferencias Gaussianas a una pirámide también conocida como pirámide Gaussiana, proceso que fue propuesto por Burt y Andelson en 1983.

La pirámide Gaussiana es construida a partir de una imagen de muestra de entrada y mediante muestras suavizadas constantes de esta imagen se van realizando submuestreos de las mismas, mientras que las diferencias Gaussianas corresponden a las diferencias entre los niveles adyacentes de la pirámide. De esta manera los puntos de interés se obtienen a partir de los puntos en los que los valores de la diferencia Gaussiana asumen extremos con respecto tanto a las coordenadas espaciales en el dominio de la imagen como al nivel de la escala en la pirámide (Encalada & Jiménez, 2015).

El algoritmo descrito por Lowe consta de 4 etapas (Ilbay, Córdova, Raura, & Gualotuña, 2013):

- Detección de extremos en la escala: La primera etapa del algoritmo realiza una búsqueda sobre las diferentes escalas y dimensiones de la imagen identificando posibles puntos de interés, invariantes a los cambios de orientación y escalado. Esto se lleva a cabo mediante la función DoG (Difference-of-Gaussian).
- Localización de keypoint: los puntos claves se eligen en base a medidas de estabilidad (se eliminan los puntos clave con bajo contraste o si se encuentran localizados en los bordes).
- Asignación de la orientación: La invarianza respecto de la rotación se consigue mediante la asignación a cada uno de los puntos una orientación basada en las propiedades locales de la imagen y representando el descriptor respecto de esta orientación.
- Descriptor de keypoints: Los gradientes locales de la imagen se miden en la región que rodea al punto clave. Éstos son transformados mediante una representación que permite medir niveles de distorsión y cambios en la iluminación de forma local.

Algoritmo ORB

El descriptor local ORB fue publicado en el ICCV (International Conference on Computer Vision) del 2011 como una alternativa a los descriptores SIFT y SURF, siendo una de sus ventajas sobre estos la velocidad de cómputo.

ORB utiliza el algoritmo Features from Accelerated Segment Test (FAST) para la detección de puntos claves y BinaryRobust Independent Elementary Features (BRIEF) para la extracción de los descriptores. FAST y BRIEF son algoritmos que mejoran notablemente la velocidad de cómputo, sin embargo, carecen de un operador de orientación e invarianza a la rotación respetivamente. Para solventar estos dos problemas ORB propone:

- oFAST (FAST KeypointOrientation): Al algoritmo FAST se le agrega un componente de orientación para los keypoints detectados.
- rBRIEF (Rotation-AwareBrief): Al algoritmo BRIEF se le agrega invarianza a la rotación y mejora la velocidad de cálculo (Ilbay et al., 2013)