



UNIVERSIDAD CENTRAL DE VENEZUELA

FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN

CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR - CENEAC

**Aplicación para el Reconocimiento de
Obras de Arte: La Ciudad Universitaria de
Caracas en tu Dispositivo Móvil**

**Trabajo Especial de Grado presentado ante la Ilustre
Universidad Central de Venezuela por el Br. Andrés Cruz Yoris (C.I. 19965431)
para optar al título de Licenciado en Computación**

Tutora: Profa. Yusneyi Y. Carballo Barrera

Ciudad Universitaria de Caracas, 17 de Octubre de 2014

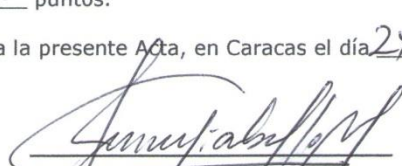
UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR - CENEAC

ACTA


Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado "**Aplicación para el Reconocimiento de Obras de Arte: La Ciudad Universitaria de Caracas en tu Dispositivo Móvil**" y presentado por el Br. **Andrés Cruz Yoris (C.I. 19965431)**, a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 24 de Octubre de 2014, a las 12:00 m horas, para que el autor lo defendiera en forma pública, lo que este hizo en PB III de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondió a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 20 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 24 de Octubre de 2014.



Prof. Yusneyi Carballo Barrera
(Tutora)



Prof. Mariantonia Palacios
(Jurado)



Prof. Esmitt Ramirez
(Jurado)

Mis más profundos agradecimientos a mi Tutora de la Escuela de Computación: Yusneyi Carballo Barrera y a la profesora de la Escuela de Arte: Mariantonia Palacios por guiarme en el desarrollo del proyecto.

A mi madre Thais Yoris por darme su arduo apoyo y esfuerzo a lo largo de la carrera.

A mi hermano Anselmo Cruz Yoris por ayudarme con el diseño de la aplicación.

A mis compañeros de clases y amigos por ayudarme a probar la aplicación.

A todo profesor y profesora de la Escuela de Computación, por otorgarme los conocimientos necesarios para realizar este trabajo y convertirme en un buen profesional.



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Enseñanza Asistida por Computador - CENEAC

Aplicación para el Reconocimiento de Obras de Arte: La Ciudad Universitaria de Caracas en tu Dispositivo Móvil

Autor: Andrés Cruz Yoris

Correo-e: andres29111990@gmail.com

Tutora: Profa. Yusneyi Carballo Barrera

Correo-e: Yusneyi.carballo@ciens.ucv.ve

Fecha: 17 de Octubre de 2014

RESUMEN

El presente Trabajo Especial de Grado se enfoca en la creación de una aplicación que permite a través de la realidad aumentada el despliegue de información sobre las obras de la colección "Síntesis de las Artes Mayores" ubicadas en la Ciudad Universitaria de Caracas (CUC) en el dispositivo móvil inteligente del usuario, empleando el reconocimiento de imágenes y el posicionamiento geográfico como tecnologías fundamentales para realizar esta labor. La aplicación utiliza la cámara del dispositivo móvil inteligente para identificar las obras de arte, también es capaz de ubicar las obras de arte por sus coordenadas geográficas con el GPS del dispositivo, por la red telefónica y mediante puntos de acceso WiFi, de esta forma se busca aumentar las posibilidades de identificación; luego de identificada la obra se procede a recuperar -desde la base de datos del portal "El Arte en la Ciudad Universitaria de Caracas"- información de la obra de arte, desplegándose finalmente una ficha informativa sobre ella, otras obras relacionadas y otras obras del autor. La aplicación permite realizar recorridos a través de mapas virtuales, ver listados de las obras de arte de la CUC y compartir información de las obras de arte a través de distintos medios, principalmente, redes sociales y correos electrónicos. También se incluye un módulo para la administración de los contenidos de la aplicación. Está dirigida a aquellos usuarios con interés en conocer las obras de arte que se encuentran en la CUC utilizando la tecnología de los dispositivos móviles. En su desarrollo se utilizó el *framework* PHP conocido como CodeIgniter, el Sistema Manejador de base de datos MySQL, la herramienta de desarrollo de software Wikitude para las funcionalidades de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada, y la API de los mapas de Google para la elaboración de recorridos.

Palabras Clave: Realidad Aumentada, Reconocimiento de Imágenes, Posicionamiento Geográfico, Puntos de Interés, Aplicaciones Móviles, Colección "Síntesis de las Artes Mayores", Arte de la Ciudad Universitaria de Caracas UCV.

Índice

INTRODUCCIÓN.....	I
1. MARCO TEÓRICO CONCEPTUAL.....	- 1 -
1.1 DEFINICIONES.....	- 1 -
1.2 COMPONENTES FUNDAMENTALES DE UN SISTEMA DE RECONOCIMIENTO DE IMÁGENES ..	- 1 -
1.3 COMPONENTES FUNDAMENTALES DE UN SISTEMA DE REALIDAD AUMENTADA.....	- 1 -
1.4 FUNCIONAMIENTO DE UN SISTEMA DE RECONOCIMIENTO DE IMÁGENES	- 2 -
1.5 FASES FUNDAMENTALES DE UN SISTEMA DE REALIDAD AUMENTADA CON RECONOCIMIENTO DE IMÁGENES.....	- 2 -
1.6 APLICACIONES PARA DISPOSITIVOS MÓVILES INTELIGENTES EMPLEADAS POR MUSEOS ...	- 3 -
1.7 EJEMPLOS DE APLICACIONES CON REALIDAD AUMENTADA, RECONOCIMIENTO DE IMÁGENES Y POSICIONAMIENTO GEOGRÁFICO SIMILARES A LA APLICACIÓN DESARROLLADA	- 5 -
1.8 FORMATO, DIMENSIONES Y CONTRASTE DE LAS IMÁGENES DE RECONOCIMIENTO	- 6 -
1.8.1 Características de las imágenes de reconocimiento.....	- 6 -
1.8.2 Herramientas para el diseño gráfico digital	- 8 -
1.9 TECNOLOGÍAS PARA EL DESARROLLO DE LA APLICACIÓN	- 9 -
2. MARCO METODOLÓGICO	- 13 -
2.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE	- 13 -
2.2 PRINCIPIOS	- 13 -
2.3 CICLO DE VIDA	- 13 -
2.3.1 Fase de Inicio.....	- 14 -
2.3.2 Fase de Elaboración.....	- 14 -
2.3.3 Fase de Construcción.....	- 15 -
2.3.4 Fase de Transición	- 15 -
2.3.5 Prácticas a tomar en la metodología.....	- 15 -
2.4 ROLES.....	- 16 -
2.5 ARTEFACTOS	- 17 -
3. DESARROLLO DE LA APLICACIÓN MACUC	- 18 -
3.1 OBJETIVO GENERAL DE LA APLICACIÓN.....	- 18 -
3.2 OBJETIVOS ESPECÍFICOS DE LA APLICACIÓN	- 18 -

3.3 ALCANCE DE LA APLICACIÓN.....	- 18 -
3.4 PLATAFORMA UTILIZADA EN EL DESARROLLO DE LA APLICACIÓN.....	- 19 -
3.5 DISPOSITIVOS UTILIZADOS PARA PROBAR EL MÓDULO DE FUNCIONALIDADES PARA DISPOSITIVOS MÓVILES	- 20 -
3.6 METODOLOGÍA DE DESARROLLO DE SOFTWARE	- 20 -
3.6.1 Fase de Inicio.....	- 20 -
3.6.2 Fase de Elaboración.....	- 30 -
3.6.3 Fase de Construcción.....	- 35 -
3.6.4 Fase de Transición.....	- 49 -
3.6.5 Principales interfaces de la aplicación.....	- 51 -
CONCLUSIONES.....	- 59 -
RECOMENDACIONES Y TRABAJOS FUTUROS	- 60 -

Índice de Figuras

Figura N° 1 Captura de pantalla de la aplicación MoMA en Android.	3 -
Figura N° 2 Captura de pantalla de la aplicación “Segundo Lienzo” en el sistema operativo móvil IOS.	4 -
Figura N° 3 Captura de pantalla de la aplicación Rijksmuseum en Android.	4 -
Figura N° 4 Obra de la exposición del “Mar de Fondo”.	5 -
Figura N° 5 Captura de pantalla de la aplicación Louvre-Lens.	6 -
Figura N° 6 Imagen con dimensiones óptimas.	7 -
Figura N° 7 Imagen muy pequeña.	7 -
Figura N° 8 Imagen con un buen nivel de contraste.	8 -
Figura N° 9 Imagen con poco contraste.	8 -
Figura N° 10 Statcounter: Estadísticas de los sistemas operativos móviles en Venezuela septiembre a octubre de 2014.	11 -
Figura N° 11 Statcounter: Estadísticas de los sistemas operativos móviles en Venezuela junio de 2013 – octubre de 2014.	12 -
Figura N° 12 Fases de la Metodología OpenUp.	14 -
Figura N° 13 Ciclo de vida de un proyecto con la Metodología OpenUP.	16 -
Figura N° 14 Diagrama de Casos de Uso, nivel 0.	26 -
Figura N° 15 Diagrama de Casos de Uso nivel 1 del módulo de funcionalidades para dispositivos móviles inteligentes.	27 -
Figura N° 16 Diagrama de Casos de Uso nivel 1 del módulo de gestión del administrador con perfil Administrador.	28 -
Figura N° 17 Diagrama de Casos de Uso nivel 1 del módulo de gestión del administrador con perfil Editor. Fuente: Creada por el Autor.	29 -
Figura N° 18 Modelo de Datos del módulo de funcionalidades para dispositivos móviles.	33 -
Figura N° 19 Modelo de Datos del módulo de funcionalidades para dispositivos móviles.	33 -
Figura N° 20 Código para generar una contraseña codificada desde el portal.	36 -
Figura N° 21 Diagrama de secuencia para cargar, crear y subir imágenes de reconocimiento, recorridos sugeridos por el Editor y paquetes de imágenes de reconocimiento.	38 -
Figura N° 22 Diagrama de secuencia para listar las imágenes de reconocimiento y los recorridos sugeridos por el Editor.	38 -
Figura N° 23 Llamada Ajax para cargar datos en las tablas administrativas.	38 -
Figura N° 24 Diagrama de secuencia para la obtención de datos de las tablas administrativas.	39 -

Figura N° 25 Diagrama de secuencia para actualizar y modificar los recorridos sugeridos por el Editor y del estilo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada.	39 -
Figura N° 26 Diagrama de Secuencia para eliminar imágenes de reconocimiento y recorridos sugeridos por el Editor.....	40 -
Figura N° 27 Código para cargar una imagen en un lienzo.....	40 -
Figura N° 28 Código para recortar imágenes.	41 -
Figura N° 29 Código para guardar una imagen dibujada en un lienzo en una imagen.	42 -
Figura N° 30 Diagrama de secuencia para la construcción de la herramienta para aumentar el contraste.....	43 -
Figura N° 31 Código para aumentar el contraste de una imagen.	43 -
Figura N° 32 Código para dibujar un mapa de Google en una vista.	44 -
Figura N° 33 Código para heredar de la API de los mapas de Google.	44 -
Figura N° 34 Código para cargas recorridos desde la base de datos del portal.....	45 -
Figura N° 35 Código para cargas recorridos desde SQLite.....	45 -
Figura N° 36 Diagrama de secuencia para conectarse a una base de datos.....	45 -
Figura N° 37 Código para agregar un marcador en un mapa de Google.....	46 -
Figura N° 38 Firma de la función para crear un listado.	47 -
Figura N° 39 Código para inicializar el Rastreador.	47 -
Figura N° 40 Código para crear una ficha informativa.	48 -
Figura N° 41 Código para cargar las obras de arte según la posición.	48 -
Figura N° 42 Código para crear un marcador.....	48 -
Figura N° 43 Diagrama de Despliegue de la aplicación.....	51 -
Figura N° 44 Interfaz de bienvenida a la aplicación para el sistema operativo móvil Android.-	51 -
-	
Figura N° 45 Interfaz principal de la aplicación para el sistema operativo móvil Android.	52 -
Figura N° 46 Interfaz para la creación de un recorrido del Visitante.....	52 -
Figura N° 47 Interfaz para el conjunto arquitectónico CUC.....	53 -
Figura N° 48 Interfaz para el listado de las Síntesis de las Artes Mayores.	53 -
Figura N° 49 Interfaz de la ficha informativa en el sistema operativo móvil Android.	54 -
Figura N° 50 Interfaz para mostrar un recorrido en el sistema operativo móvil Android.....	54 -
Figura N° 51 Interfaz del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada (panel de obras relacionadas).	55 -
Figura N° 52 Interfaz del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada (ficha informativa).....	55 -

Figura N° 53 Interfaz de autenticación del módulo de gestión del administrador. - 56 -

Figura N° 54 Interfaz del listado de los recorridos sugeridos por el Editor. - 56 -

Figura N° 55 Interfaz para hacer un recorte en una imagen. - 57 -

Figura N° 56 Interfaz para crear o actualizar un recorrido sugerido por el Editor. - 57 -

Figura N° 57 Logo de jQuery. - 66 -

Figura N° 58 Logo de jQuery UI. - 67 -

Figura N° 59 Logo de jQuery Mobile. - 68 -

Figura N° 60 Imagen representativa de los mensajes. - 69 -

Figura N° 61 Logo de Android. - 71 -

Figura N° 62 Logo de Wikitude. - 74 -

Figura N° 63 Mapas de Google. - 75 -

Figura N° 64 Logo de SQLite. - 76 -

Figura N° 65 Pendiente de la recta de una imagen. - 77 -

Figura N° 66 Recorte de una imagen con GIMP. - 78 -

Figura N° 67 Prueba de aceptación para MACUC. - 78 -

Índice de Tablas

Tabla 1 Artefactos de la metodología OpenUP.	- 17 -
Tabla 2 Participantes del proyecto.....	- 21 -
Tabla 3 Roles de los participantes del proyecto.	- 22 -
Tabla 4 Operaciones permitida en la aplicación según el tipo de usuario.....	- 24 -
Tabla 5 Lista de actividades.	- 31 -
Tabla 6 Descripción de algunas tablas de la base de datos del portal “el arte en la CUC” y algunas de sus Relaciones.....	- 32 -
Tabla 7 Descripción de las nuevas tablas de la base de datos del portal “el arte en la CUC” y sus relaciones.	- 32 -
Tabla 8 Comparación entre las versiones gratuitas y las versiones de pago de Wikitude SDK. -	73 -
Tabla 9 Lista de características de Wikitude SDK.	- 73 -
Tabla 10 Resultados de las encuestas.	- 79 -

Anexos

ANEXO A: TECNOLOGÍAS PARA EL DESARROLLO DE APLICACIONES WEB - 66 -	
ANEXO A.A: BIBLIOTECA DE DESARROLLO DE SOFTWARE JQUERY	- 66 -
ANEXO A.B: JQUERY UI.....	- 66 -
ANEXO A.C: JQUERY MOBILE.....	- 67 -
ANEXO A.D: JQUERY MULTISELECT	- 68 -
ANEXO A.E: JQUERY TOASTMESSAGE.....	- 68 -
ANEXO A.F: FANCYBOX.....	- 69 -
ANEXO A.G: STICKUP	- 69 -
ANEXO A.H: LOADING OVERLAY.....	- 69 -
ANEXO A.I: JSON.....	- 70 -
ANEXO B: TECNOLOGÍAS PARA EL DESARROLLO DE APLICACIONES MÓVILES - 70 -	
ANEXO B.A: ANDROID.....	- 70 -
ANEXO B.B: WIKITUDE.....	- 71 -
ANEXO C: TECNOLOGÍAS PARA EL DESARROLLO DE APLICACIONES WEB Y MÓVILES - 74 -	
ANEXO C.A: MAPAS DE GOOGLE.....	- 74 -
ANEXO C.B: SQLITE.....	- 75 -
ANEXO D: TÉCNICAS PARA EL PROCESAMIENTO DIGITAL DE IMÁGENES..- 76 -	
ANEXO D.A: CONTRASTE.....	- 76 -
ANEXO D.B: RECORTAR IMÁGENES	- 77 -
ANEXO E: PRUEBA DE ACEPTACIÓN PARA MACUC.....	- 78 -
ANEXO F: RESULTADOS DE LAS ENCUESTAS.....	- 79 -
ANEXO G: MANUAL DEL USUARIO ADMINISTRADOR	- 80 -

Introducción

En el año 2000, la Ciudad Universitaria de Caracas pasó a formar parte de la lista de Patrimonio Mundial de la UNESCO debido a su valor excepcional como bien cultural y por ser una obra maestra de arquitectura, urbanismo y arte moderno. Desde que Carlos Raúl Villanueva la concibiera, nuestra Ciudad Universitaria ha incentivado el cultivo de las artes. La Facultad de Humanidades y Educación ha jugado un papel importante en este sentido, acogiendo en su seno al antiguo Instituto de Arte y a la Escuela de Artes, esta última institución que se ha convertido, hoy por hoy, en uno de los centros de investigación y generación de conocimiento sobre arte más importante de Latinoamérica. Por eso, y dada su vocación y naturaleza, la catalogación y registro del patrimonio artístico de la Ciudad Universitaria de Caracas es un proyecto de una innegable pertinencia. Esta información se puso a disposición del público general de manera masiva a través de la implementación de un portal web en colaboración con la Escuela de Computación de la Facultad de Ciencias: “El Arte en la Ciudad Universitaria de Caracas”.

Este proyecto forma parte del portal web “El Arte en la Ciudad Universitaria de Caracas”, en el cual se compilan las obras de arte de la Ciudad Universitaria de Caracas, tanto las pertenecientes a la colección “Síntesis de las Artes”, como las que se encuentran en los espacios internos de las Facultades y Dependencias; el portal fue colocado al alcance del público en general en mayo de 2014 mostrando un conjunto de obras en su núcleo¹. El portal, además de servir como un sitio divulgativo de información, permite una completa gestión de las obras: agregar información básica, imágenes simples, imágenes panorámicas, autores y coordenadas geográficas de las obras de arte, incluso permite vincular contenido propio con otras páginas de Internet para enriquecer aún más el contenido propio. Resulta interesante, dado el auge de dispositivos móviles inteligentes, una aplicación que permita que el usuario combine elementos virtuales con su entorno físico para crear una realidad mixta en tiempo real, digital e interactiva a través de la cámara y sensores de los dispositivos móviles, vinculada con las obras de arte que se encuentran en la Ciudad Universitaria de Caracas. Este tipo de aplicaciones son usadas con frecuencia para proyectos educativos en museos y exhibiciones, lo cual aplica perfectamente para la Ciudad Universitaria de Caracas, puesto que, como ya sea dicho antes, esta alberga en sus espacios una importante colección de obras de arte. Este trabajo consiste en desarrollar una aplicación para dispositivos móviles inteligentes que permita el reconocimiento de obras de arte a través del reconocimiento de imágenes o posicionamiento geográfico de las mismas; permite a los usuarios con un dispositivo móvil inteligente visualizar información a través de la realidad aumentada; permite crear, ver y compartir recorridos por las obras de arte.

Así, el objetivo general de este Trabajo Especial de Grado consiste en: Desarrollar una aplicación que permita reconocer obras de arte *in situ* en la Ciudad Universitaria de Caracas (CUC).

Para desarrollar este Trabajo Especial de Grado se cumplieron los siguientes objetivos específicos:

¹ Enlace del portal “El Arte en la Ciudad Universitaria de Caracas”: http://orpheus.human.ucv.ve/paseo_virtual/

1. Realizar una lista de requerimientos y funcionalidades de la aplicación.
2. Aplicar la metodología para desarrollar la aplicación conforme a los requerimientos.
3. Realizar el diseño lógico y físico de la aplicación.
4. Utilizar patrones, estándares y lineamientos de usabilidad para la creación de aplicaciones especialmente orientadas a dispositivos móviles.
5. Desarrollar las funcionalidades de la aplicación.
6. Documentar el análisis y desarrollo de la investigación en el informe del Trabajo Especial de Grado.
7. Realizar pruebas de aceptación a través de demostraciones, entrevistas y encuestas con usuarios para obtener su opinión con aspectos relacionados con el diseño y la funcionalidad de la aplicación.
8. Incorporar las mejoras determinadas en el análisis de las demostraciones, entrevistas y las encuestas.
9. Aplicar pruebas para verificar el correcto funcionamiento de la aplicación.

La aplicación desarrollada está compuesta de un módulo de funcionalidades para dispositivos móviles que permite reconocer, obtener, visualizar, divulgar información, realizar recorridos y visualizar listados del conjunto arquitectónico de la CUC, además de un módulo de gestión para el administrador que permite gestionar: recorridos e imágenes entre otras funcionalidades; ambos módulos son explicados en detalle en el punto 3.7.2.3.

A continuación se indican los capítulos que conforman este documento y que permiten describir el desarrollo de la aplicación:

- **Capítulo I, Marco Teórico Conceptual:** En este capítulo se describen las teorías en las cuales se encuentra sustentado este Trabajo Especial de Grado y procesos relacionados con la solución a resolver.
- **Capítulo II, Marco Metodológico:** En este capítulo se describen los métodos, metodología, fases y lineamientos correspondientes al método ágil de desarrollo elegido para este trabajo de investigación el cual es conocido como OpenUp.
- **Capítulo III, Desarrollo de la Aplicación:** En este último capítulo se procede a explicar la aplicación ya desarrollada; los objetivos, alcance, tecnologías, plataformas, descripción de la aplicación y fases en el desarrollo son los principales puntos de este capítulo.

- Por último se presentan los resultados y conclusiones del trabajo realizado, recomendaciones para trabajos futuros, fuentes bibliográficas e información complementaria (Anexos).

CAPÍTULO I

1. MARCO TEÓRICO CONCEPTUAL

1.1 Definiciones

- **Reconocimiento de imágenes (RI):** Es una tecnología que permite adquirir, procesar y analizar imágenes para producir información obtenida a través de imágenes digitales. Héctor Ruiz Cabeza (2010) define el reconocimiento de imágenes como “método para extraer la información necesaria para controlar un proceso o actividad”; también define el RI como “procedimiento de adquisición de imágenes, sin contacto y mediante sistemas ópticos, donde se realiza el análisis automático de las mismas”.
- **Punto de interés (PDI, o POI por las siglas en inglés de *Point of Interest*):** Son lugares o sitios específicos que son considerados de interés para alguien; como por ejemplo una obra de arte, un sitio turístico, hospitales, etc. Para crear un PDI es preciso como mínimo la latitud y la longitud aunque también es posible utilizar la altitud para ubicar un sitio en su espacio 3D (Wikipedia, s.f.).
- **Realidad aumentada (RA):** Alejandro Sacristán (2011) director del proyecto: “La Voz de la Ciencia”, define la RA como “tecnología que permite añadir información virtual a la información física ya existente”, en otras palabras, añade una capa virtual a lo real. La realidad aumentada es la incorporación de datos e información digital en un entorno real por medio del reconocimiento de imágenes o alguna otra tecnología que se realiza mediante un software.

1.2 Componentes fundamentales de un sistema de reconocimiento de imágenes

El reconocimiento de imágenes necesita de dos componentes:

- **Hardware:** Los dispositivos que permiten el reconocimiento de imágenes contienen una cámara con capacidad de tomar fotos o hacer vídeos.
- **Software:** Se encarga de reconocer las imágenes.

1.3 Componentes fundamentales de un sistema de realidad aumentada

La realidad aumentada necesita de dos componentes:

- **Hardware:** Los dispositivos que permiten incluir capas de realidad aumentada contienen una pantalla para mostrar al usuario la información virtual que se añade a la real. El hardware debe de llevar incorporado una cámara con capacidad de tomar fotos o hacer vídeos, además de esto; los sistemas modernos de realidad aumentada pueden utilizar los

siguientes componentes: sensores ópticos, acelerómetros, GPS, giroscopios, brújulas, entre otros.

- **Software:** Superpone información digital a la realidad que el usuario ve a través de la pantalla del dispositivo (Daniel Wagner, s.f.).

1.4 Funcionamiento de un sistema de reconocimiento de imágenes

Según Peter Norvig (2011), Director de Investigación en Google, el reconocimiento de imágenes funciona mediante el aprendizaje basado en conocimiento; es decir, con información acerca de cómo está compuesta la imagen. Un sistema de reconocimiento de imágenes funciona de la siguiente manera:

1. Primero identifica puntos de interés en una imagen; líneas y patrones que contrastan o hacen sobresalir algo del fondo de la imagen.
2. Paso siguiente, identifica cómo estos puntos se relacionan entre sí (la geometría de los puntos en conjunto).
3. Por último, el sistema compara ese modelo con otros modelos dentro de la base de datos en busca de un modelo con el que corresponda. Esos otros modelos provienen de imágenes que ya fueron analizadas.

Es importante que el sistema sea flexible, para que no importe tanto si la imagen está volteada, escalada, torcida, etc. Tomando en cuenta que diferentes fotos de un mismo objeto serán distintas.

1.5 Fases fundamentales de un sistema de realidad aumentada con reconocimiento de imágenes

Según López Pomboo, H (2013), para desarrollar un sistema de realidad aumentada con reconocimiento de imágenes es necesario pasar por cuatro fases previas:

- **Captación de escena:** Consiste en captar el escenario que se va a aumentar a través de un dispositivo que posea una cámara con capacidad de tomar fotos o hacer videos, para ello debemos analizar el ambiente y procesarlo.
- **Identificar la escena:** En esta parte, básicamente lo que se hace es identificar el área física que se desea aumentar. Para ello podemos utilizar marcadores, que es cuando el sistema reconoce la escena por su geometría, color o las dos características. Este proceso escanea escenas en búsqueda de dicho marcador predeterminado, por lo tanto, previamente debemos configurar el sistema de manera que le demos un margen de error o un rango más amplio para evitar problemas con el movimiento de la imagen y nos agilice dicha identificación. Además de los marcadores, para realizar la segunda fase podemos emplear otra opción; que es mediante la posición geográfica o reconocimiento de imágenes. Este proceso, en muchos casos puede ser más complejo que utilizar marcadores y por lo tanto, necesita más cálculos para reconocer la escena.

- **Mezclar la realidad real y la realidad virtual:** Para ello ponemos la información digital que identificamos en la fase anterior, en la imagen capturada que puede ser en tiempo real (video) o en una fotografía.
- **Visualización de la escena:** que consiste en ver el espacio físico, junto a la información digital que le agregamos, para así obtener la realidad aumentada.

1.6 Aplicaciones para dispositivos móviles inteligentes empleadas por museos

Desde hace mucho los museos han utilizado la web para facilitar visitas virtuales y para ofrecer exposiciones y guías multimedia; con la popularización de los dispositivos móviles inteligentes han comenzado a surgir aplicaciones con un enfoque al de los sitios web; aprovecha al máximo la ventaja de la portabilidad de estos dispositivos y de esta forma, ganar visibilidad en sus sitios con las nuevas tecnologías.

Museo de Arte Moderno (*La Tate Modern*) de Londres dispone de varias aplicaciones para el sistema operativo móvil IOS (cerca de dieciséis aplicaciones) que permiten utilizar el teléfono para desplegar contenido de sus exposiciones (Ana Baeza, 2013).

El Museo del Arte Moderno de Nueva York (MoMA) también dispone de aplicaciones para los sistemas operativos móviles IOS y Android; dispone de guías de audio para recorrer todos los pisos; es posible utilizar la aplicación MoMA para encontrar las exhibiciones, planes de visita, buscar palabras claves en la colección, recorridos multimedia, y aprender sobre artes; requiere conexión a Internet para funcionar (Google Play, s.f.).



Figura N° 1 Captura de pantalla de la aplicación MoMA en Android.

Fuente: <http://artecontusmanitas.files.wordpress.com/2014/03/app-museo-moma.jpg>

El “**Segundo Lienzo**” (*Second Canvas*) es una aplicación creada por el Museo del Prado en Madrid y la empresa Madpixe; con esta aplicación es posible descubrir, navegar y compartir a través de las redes sociales con una tableta iPad, o un teléfono inteligente iPhone (Aplicaciones, s.f.).



Figura N° 2 Captura de pantalla de la aplicación “Segundo Lienzo” en el sistema operativo móvil IOS.

Fuente: http://www.aplicaciones.org/wp-content/uploads/2014/03/Second-Canvas-Museo-del-Prado_sc_2.jpg

El museo de Rijksmuseum ubicado en Ámsterdam tiene más de 8000 obras de arte en 80 sales de exposiciones que narran diversas historias de arte; la aplicación Rijksmuseum proporciona textos detallados e imágenes de alta resolución de su arte expuesta; la aplicación está especialmente construida para su uso en el museo en sí, no todas las funcionalidades trabajarán fuera del museo; además incluye composiciones de audio, guardado de obras de arte y descarga de imágenes (Google Play, s.f.).



Figura N° 3 Captura de pantalla de la aplicación Rijksmuseum en Android.

Fuente: <http://a4.mzstatic.com/eu/r30/Purple/v4/4b/b5/79/4bb579cb-4cc2-9e9c-e54f-0d876643c71c/screen568x568.jpeg>

1.7 Ejemplos de aplicaciones con realidad aumentada, reconocimiento de imágenes y posicionamiento geográfico similares a la aplicación desarrollada

En este punto se describen diferentes aplicaciones usadas dentro de los museos para enriquecer la experiencia del visitante a través de la realidad aumentada, guías, recorridos, etc. Similares a la solución planteada en el Trabajo de Investigación de Seminario y llevado a cabo en este Trabajo Especial de Grado; son aplicaciones que buscaba ampliar la experiencia del público con las obras de arte a través de una tableta y/o teléfono inteligente.

El Museo de Mataró ubicado en la ciudad de Madrid en España, integró un sistema de realidad aumentada con la exposición llamada “Mar de Fondo”, la cual consistía en ampliar la experiencia del público con las obras a través de una tableta o teléfono inteligente; así el visitante apunta sobre la obra para que la aplicación proporcionase más información sobre la misma: nombre, autor y una foto del artista. A partir de ahí se puede ampliar la experiencia accediendo a los contenidos creados para cada autor y en algunos casos la información temática relacionada. La figura N° 4 muestra un ejemplo de lo explicado (Musas20, 2013).



Figura N° 4 Obra de la exposición del “Mar de Fondo”.

Fuente: <http://www.digitalavmagazine.com/wp-content/uploads/2012/04/Museo-Mataro.jpg>

El Museo del Louvre-Lens ubicado en la ciudad de Lens en Francia, en colaboración con la telefónica Orange, lanza la aplicación llamada "Louvre-Lens". Esta aplicación permite navegar por la historia universal del arte a través de una selección de obras de la Galería del Tiempo, descubrir museo del parque y su pasado minero con la realidad aumentada y localizar cualquier punto con un plano del museo (Google Play, s.f.).

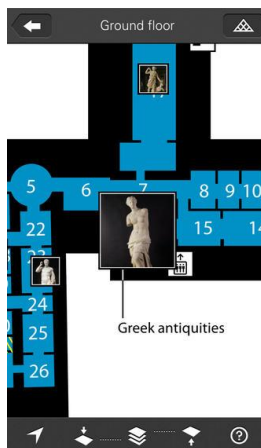


Figura N° 5 Captura de pantalla de la aplicación Louvre-Lens.

Fuente: <http://a5.mzstatic.com/us/r30/Purple/v4/f1/de/e4/f1dee496-be89-2215-080c-4cb237664dd6/screen568x568.jpeg>

1.8 Formato, dimensiones y contraste de las imágenes de reconocimiento

Los formatos permitidos para las imágenes a almacenar en la aplicación como imágenes de reconocimiento (ver punto 3.7.1.6) son los formatos JPG y PNG sin transparencia; Wikitude (que es la herramienta de desarrollo de software (SDK) seleccionada para la elaboración del módulo de funcionalidades para dispositivos móviles; más información en el Capítulo III) no puede emplear imágenes con fondo transparente para el proceso de reconocimiento; son elegidos estos formatos debido a que son formatos de compresiones comunes. En el caso del formato de compresión PNG permite almacenar imágenes con una profundidad de contraste mayor que otros formatos; en cuanto al formato de compresión JPG, es un formato común empleado por las cámaras fotográficas.

1.8.1 Características de las imágenes de reconocimiento

En esta sección veremos algunas características que deben cumplir las imágenes de reconocimiento en la aplicación:

1. Las imágenes almacenadas en la aplicación deben de estar entre los 500 a 1000 píxeles de ancho y largo:



Figura N° 6 Imagen con dimensiones óptimas.

Fuente:http://www.wiktitude.com/external/doc/documentation/latest/android/images/guide_dimension_good.png



Figura N° 7 Imagen muy pequeña.

Fuente:http://www.wiktitude.com/external/doc/documentation/latest/android/images/guide_dimension_wrong.png

2. Deben de tener un alto contraste para un mejor enfoque y claridad de la imagen:



Figura N° 8 Imagen con un buen nivel de contraste.

Fuente: http://www.wiktitude.com/external/doc/documentation/latest/android/images/low_contrast_good.png



Figura N° 9 Imagen con poco contraste.

Fuente: http://www.wiktitude.com/external/doc/documentation/latest/android/images/low_contrast_wrong.png

Todas estas limitantes se deben a que Wikitude recomienda imágenes que cumplan las características anteriores para que el reconocimiento de imágenes sea efectivo (Wikitude, s.f.).

1.8.2 Herramientas para el diseño gráfico digital

Estas herramientas tienen una importancia significativa para la aplicación que se lleva a cabo en este Trabajo Especial de Grado; aunque no se tenga previsto que el usuario emplee estas herramientas para editar las imágenes de reconocimiento de la aplicación (la aplicación tiene lo necesario para realizar las operaciones de recortes -para capturar el objeto de interés ver Anexo D parte -B- y aumento de contraste -para un mejor enfoque y claridad de la imagen ver Anexo D parte -A-), son necesarias para entender cómo funcionan estas herramientas diseñadas en la aplicación; es decir, es importante mencionarlas debido a la gran similitud que existe en el uso de

estas herramientas provistas en la aplicación desarrollada y en el uso de estas herramientas en la mayoría de los software de edición de imágenes como los que veremos a continuación:

- **Programa de Manipulación de Imágenes (GIMP, GNU *Image Manipulation Program*):** Es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito. Está englobado en el proyecto GNU y disponible bajo la licencia pública general de GNU; fue desarrollado como una herramienta libre para trabajar con imágenes y se ha convertido en una alternativa libre y eficaz a Photoshop; además de poseer su propio formato de almacenamiento de archivos, XCF (Gimp.es, s.f.).
- **Phostoshop:** es un programa de edición de imágenes digitales. Es un programa comercial desarrollado por Adobe Systems; los formatos propios de Photoshop son PSD y PDD.

Ambos software de edición de imágenes poseen las siguientes características:

- Leen y escriben en la mayoría de los formatos de archivos gráficos, entre ellos JPG, GIF, PNG, PCX, TIFF.
- Poseen varias herramientas de selección (rectangular, esférica, lazo manual, varita mágica, por color), tijeras inteligentes, herramientas de pintado como pincel, brocha, aerógrafo, relleno, texturas, etc.
- Poseen herramientas de modificación de escala, de inclinación, de deformación, clonado en perspectiva o brocha de curado.
- Posee herramientas de manipulación de texto.
- Dispone también de muchas herramientas o filtros para la manipulación de los colores y el aspecto de las imágenes, como enfoque y desenfoco, eliminación o adición de manchas, sombras, mapeado de colores, etc (Gimp.es, s.f.).

1.9 Tecnologías para el desarrollo de la aplicación

Ya definidos los objetivos y evaluadas el conjunto de funcionalidades en el Trabajo de Investigación Seminario, se han seleccionado el siguiente conjunto de herramientas y tecnologías para llevar a cabo la aplicación:

- **Lenguaje de Marcado y Lenguaje de Estilo:** HTML5 y CSS3 respectivamente; ambos son usados en conjunto para el desarrollo de páginas web; de tal manera se logra separar el contenido de la presentación.
- **Lenguaje de Programación del lado cliente:** JavaScript permite agregar en las páginas web validaciones del lado del cliente; se usara específicamente la biblioteca de desarrollo de software jQuery (ver Anexo A parte -A-); la misma permite simplificar la manera de interactuar con los elementos que conforman el DOM, mejora la compatibilidad entre los navegadores web, mejora sustancial en el uso del DOM, incorpora efectos visuales, mejora en el manejo de los eventos y cuenta con una enorme cantidad de *plugin*, para agregar

nuevas funcionalidades DataTable (ver Anexo A parte -J- del Trabajo de Investigación de Seminario) como una completa tabla administrativa y dinámica altamente personalizable, Fancybox (ver Anexo A parte -F-) para la creación de galerías, además de que permite mantener la compatibilidad entre distintos navegadores web, StickUp (ver Anexo A parte -G-) para tener elementos HTML siempre visible y Loading Overlay (ver Anexo A parte -H-) para la superposición de capas que son ideales para las vistas de carga de elementos; además de jQuery UI (ver Anexo A parte -B-) para las interfaces de usuario.

- **Lenguaje de Programación del lado servidor:** PHP (ver Anexo A parte -C- del Trabajo de Investigación de Seminario) es ampliamente utilizado para el desarrollo de páginas web dinámicas, posee integración con la mayoría de los sistemas administradores de base de datos del mercado; específicamente se usará el *framework* CodeIgniter (ver Anexo A parte -D- del Trabajo de Investigación de Seminario); permite desarrollar proyectos de pequeña envergadura de forma rápida, organizada y flexible; su curva de aprendizaje es muy corta.
- **Sistema Administrador de Base de Datos:** MySQL (ver Anexo A parte -A- del Trabajo de Investigación de Seminario) es un completo Sistema Administrador de Base de Datos flexible, rápido y ampliamente utilizado en conjunto con PHP.
- **Tecnologías para el desarrollo en aplicaciones móviles:**
 - Wikitude (ver Anexo B parte -B-) como herramienta de desarrollo de software para el reconocimiento de imágenes, posicionamiento geográfico y realidad aumentada; la herramienta de desarrollo de software Wikitude está construida en gran medida con tecnologías Web (HTML, JavaScript, CSS, etc.) lo que permite crear una plataforma cruzada. Se ha seleccionado la herramientas de desarrollo de software Wikitude para la elaboración del módulo de funcionalidades para dispositivos móviles en vez de la herramientas de desarrollo de software Vuforia -como estaba especificado en el Trabajo de Investigación de Seminario- debido al hecho de que para utilizar el reconocimiento de imágenes en una aplicación realizada en Vuforia SDK es necesario alojar el paquete de imágenes de reconocimiento en sus servidores con un mínimo de 99\$ al mes; por otro lado Wikitude SDK permite ubicar el paquete de imágenes de reconocimiento en un Servidor Web o local a la aplicación; en ambos casos; puede ser accedido mediante una URL.
 - jQuery Mobile (ver Anexo A parte -C-) como *framework* para la creación de interfaces, las mismas serán empleadas para mostrar las capas de información a través de la realidad aumentada vinculadas al sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada.
- **Otras tecnologías para el desarrollo de la aplicación:**
 - Los mapas de Google (ver Anexo C parte -A-) están disponibles tanto para web como para el desarrollo de aplicaciones en dispositivos móviles; ofrecen una completa plataforma para interactuar con los mapas agregando marcadores en cualquier ubicación del mapa y marcar rutas entre ellos.

Gran parte de estas tecnologías son utilizadas por el portal “El Arte en la Ciudad Universitaria de Caracas”; se busca lograr la mayor compatibilidad entre ambos proyectos; debido a que la aplicación a desarrollar forma parte de dicho portal.

Se propone realizar desarrollos para el sistema operativo móvil Android (ver Anexo B parte - A-); Android es el sistema operativo móvil más utilizado en nuestro país, según Statcounter² en Venezuela lidera Android con un 65,65% del mercado para los meses septiembre y octubre de 2014:

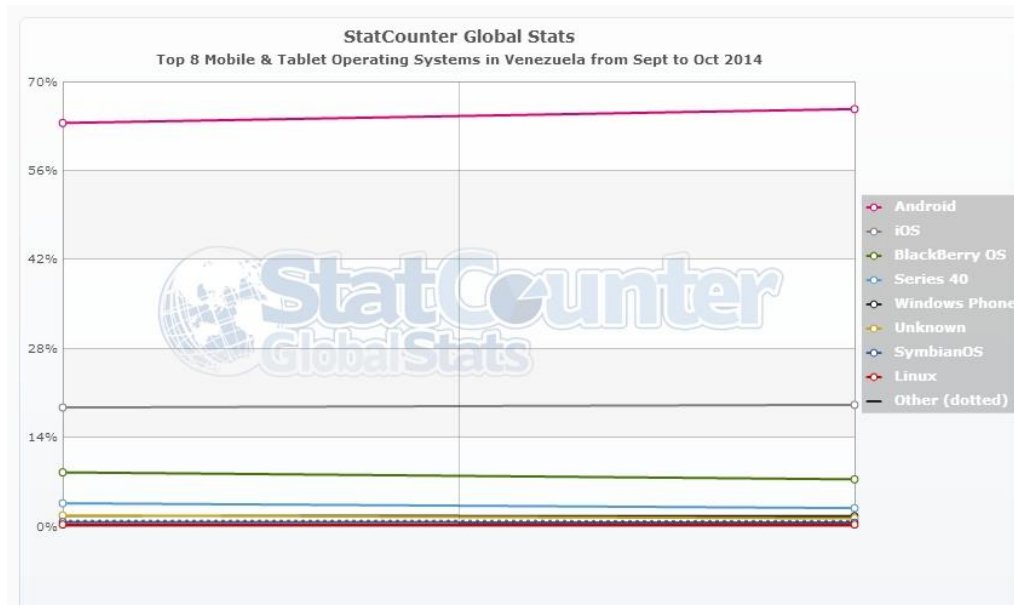


Figura N° 10 Statcounter: Estadísticas de los sistemas operativos móviles en Venezuela septiembre a octubre de 2014.

Fuente: <http://gs.statcounter.com/#mobile+tablet-os-VE-monthly-201409-201410>

Para el desarrollo del módulo de funcionalidades para dispositivos móviles, el cual consiste en desarrollar una aplicación nativa para el sistema operativo móvil más utilizados en Venezuela; según el Trabajo de Investigación de Seminario se propuso llevar a cabo un desarrollo para los sistemas operativos móviles Android e IOS; sin embargo; debido a las siguientes razones expuestas:

- El porcentaje de usuarios en Venezuela que utiliza el sistema operativo móvil IOS como sistema operativo móvil han venido reduciendo constantemente desde la elaboración del Trabajo de Investigación de Seminario (ver figura N° 11).
- Una aplicación para el sistema operativo móvil IOS es más costosa de mantener en el tiempo que una aplicación para el sistema operativo móvil Android; es necesario una cuota anual de al menos 99\$ para mantener activa una aplicación desarrollada para el sistema operativo móvil IOS en la Tienda de Aplicaciones de Apple (*App Store*); a su vez,

² Statcounter es un servicio estadístico para navegadores, sistemas operativos, resoluciones de pantalla, medios sociales, etc.

es imprescindible que la aplicación este alojada en dicha tienda para poder ser instalada legalmente en un dispositivo con el sistema operativo móvil IOS; en caso de que se desee instalar un aplicación que no se encuentre en la tienda en dispositivo con sistema operativo móvil IOS se debe de utilizar otros métodos para instalarla en el dispositivo como el *jailbreak*³;

Se ha decidido solo desarrollar para el sistema operativo móvil Android; para poder cargar aplicaciones en la Tienda de Aplicaciones de Google (*Google Play*) consisten en comprar una licencia de 25\$ sin fecha de vencimiento; además, si no se desea comprar la licencia todavía existe la posibilidad de instalar aplicaciones que no se encuentren dentro de la tienda con solo copiar el instalador (apk) de la aplicación dentro del dispositivo y ejecutarlo.

En el siguiente diagrama podemos ver la cuota del mercado de los sistemas operativos móviles en Venezuela desde junio de 2013 hasta septiembre de 2014; el sistema operativo móvil Android ha tenido un crecimiento constante, mientras que el sistema operativo móvil IOS ha venido reduciendo su cuota:

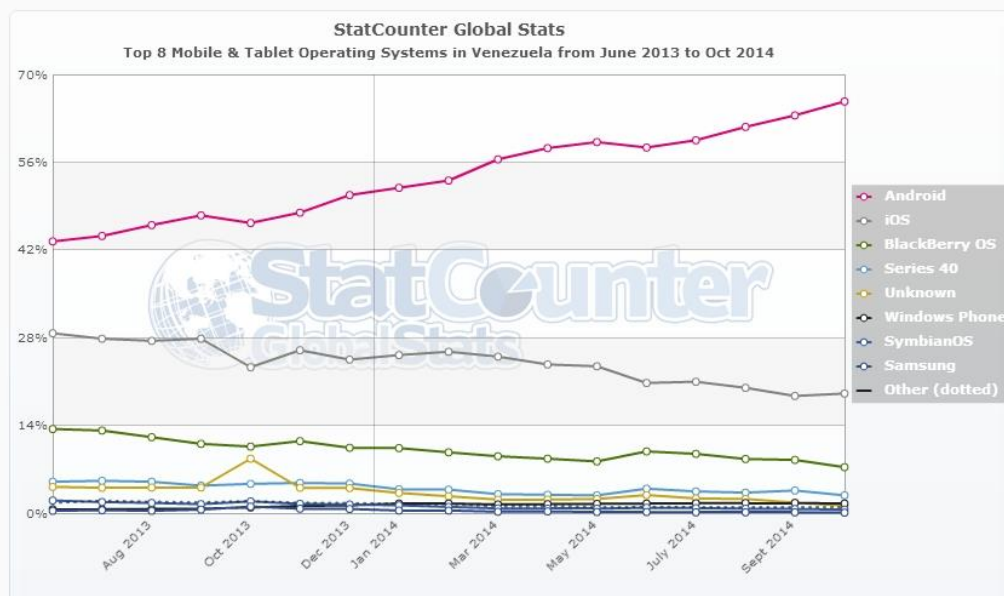


Figura N° 11 Statcounter: Estadísticas de los sistemas operativos móviles en Venezuela junio de 2013 – octubre de 2014.

Fuente: <http://gs.statcounter.com/#mobile+tablet-os-VE-monthly-201306-201410>

³ Jailbreak: Proceso de suprimir algunas de las limitaciones impuestas por Apple en dispositivos que utilicen el sistema operativo móvil IOS; por ejemplo es posible descargar aplicaciones, extensiones y temas que no estén disponibles en la Tienda de Aplicaciones de Apple.

CAPÍTULO II.

2. MARCO METODOLÓGICO

Según Arias F. (1999, p.19) la metodología del proyecto debe de incluir las técnicas y los procedimientos que será utilizados para llevar la investigación. Es el “cómo” se realizará el estudio para responder al problema planteado.

2.1 Metodología de Desarrollo de Software

Dentro de los objetivos principales de este trabajo está el de cumplir las tareas planteadas de la mejor manera y con la mejor calidad; por lo tanto es recomendable el uso de una metodología que permita desarrollar de forma eficiente las funcionalidades del mismo.

El uso de esta metodología permite controlar el desarrollo en cuatro fases bien definidas, lo que conlleva a minimizar los errores y optimizar el trabajo en cada fase (Tuza, 4 de mayo de 2012); a continuación se describe la metodología utilizada para la implementación de la aplicación.

El método OpenUP es un proceso mínimo y suficiente, lo que significa que sólo el contenido fundamental y necesario es incluido. Por lo tanto no provee lineamientos para todos los elementos que se manejan en un proyecto pero tiene los componentes básicos que pueden servir de base a procesos específicos (Tuza, 4 de mayo de 2012).

2.2 Principios

- Colaborar para sincronizar intereses y compartir conocimiento.
- Equilibrar las prioridades para maximizar el beneficio obtenido por los interesados en el proyecto.
- Usar escenarios y casos de uso para capturar requerimientos
- Centrarse en la arquitectura de forma temprana para minimizar el riesgo y organizar el desarrollo.
- Desarrollo evolutivo para obtener retroalimentación y mejoramiento continuo, permitiendo demostrarles incrementos progresivos en la funcionalidad (Universidad Distrital Francisco José de Caldas, 16 de octubre de 2012).

2.3 Ciclo de vida

Todo proyecto en OpenUP consta de cuatro fases: inicio, elaboración, construcción y transición. A su vez estas fases se dividen en iteraciones.

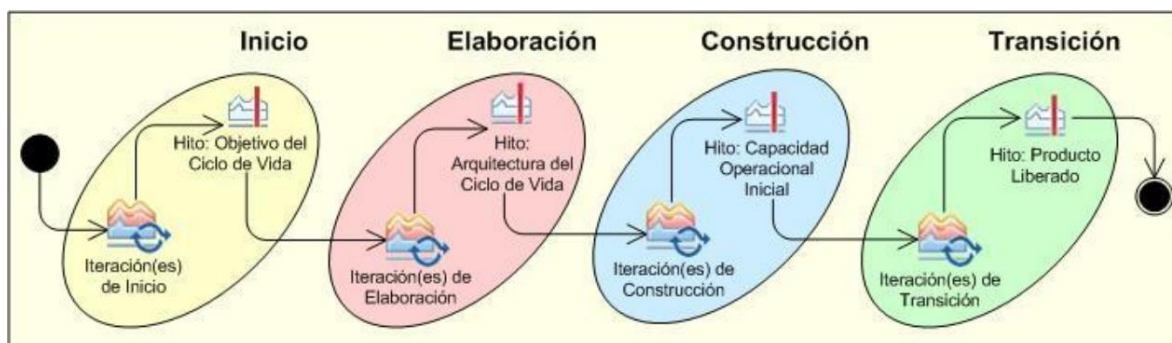


Figura N° 12 Fases de la Metodología OpenUp.

Fuente: http://upload.wikimedia.org/wikipedia/commons/2/2c/Openup-basic_lifecycle.jpg

2.3.1 Fase de Inicio

- En esta fase se logra comprender el alcance del proyecto, se definen para el proyecto:
 - Entender que es lo que se desea construir, el alcance, ámbito y el límite del proyecto.
 - Identificar los requerimientos y funcionalidades del sistema.
 - Encontrar al menos una arquitectura candidata y su viabilidad como solución.
 - Finalmente calcular el coste y los riesgos asociados al sistema (Universidad Distrital Francisco José de Caldas, 16 de octubre de 2012).

2.3.2 Fase de Elaboración

- En esta fase se realizan tareas de análisis del dominio y definición de la arquitectura del sistema:
 - Se elabora un plan de proyecto, estableciendo unos requisitos y una arquitectura estable.
 - Por otro lado, el proceso de desarrollo, las herramientas, infraestructura y el entorno de desarrollo.
- Se produce un cronograma detallado.
- Diseñar, implementar y probar un prototipo. Aunque la funcionalidad no sea completa aún, muchas de las interfaces entre los bloques de construcción se implementan y prueban (Universidad Distrital Francisco José de Caldas, 16 de octubre de 2012).

Al final de la fase se debe tener una definición clara y precisa de los casos de uso, los actores, la arquitectura del sistema y un prototipo ejecutable de la misma.

2.3.3 Fase de Construcción

- En esta fase se desarrolla el sistema basado en la arquitectura anterior:
 - Todos los componentes y funcionalidades del sistema que falten por implementar son realizados, probados e integrados en esta fase.
 - Liberar la primera versión operativa del software (beta) del sistema.
 - Los resultados obtenidos en forma de incrementos ejecutables deben ser desarrollados de la forma más rápida posible sin dejar de lado la calidad de lo desarrollado (Salgado y otros, s.f.).

2.3.4 Fase de Transición

- Esta fase corresponde a la introducción del producto en la comunidad de usuarios, cuando el producto está lo suficientemente maduro:
 - La fase de la Transición consta de las subfases de pruebas de versiones beta, pilotaje y capacitación de los usuarios finales y de los encargados del mantenimiento del sistema.
 - En función de la respuesta obtenida por los usuarios puede ser necesario realizar cambios en las entregas finales o implementar alguna funcionalidad más.
 - El desarrollo se considera completo luego de realizar varias pruebas con los usuarios, para poder colocar el producto en producción.

La prueba beta valida que se satisfagan las expectativas del usuario. Generalmente requieren de algunas actividades de afinamiento, como depuración de errores y mejora del desempeño y la usabilidad (Universidad Distrital Francisco José de Caldas, 16 de septiembre de 2011).

2.3.5 Prácticas a tomar en la metodología

- **Reuniones:** OpenUp está orientado a realizar reuniones de forma regular; de esta forma es posible captar o prevenir posibles riesgos sobre el proyecto y poder plantear estrategias para su mitigación.
- **Conocer a los Interesados:** No se puede generar un sistema que sea útil si no se conoce las necesidades y el problema a resolver; se debe de identificar, conocer y trabajar de cerca con los grupos de interés para asegurarse que sus necesidades son claramente definidas e incrementalmente satisfechas.
- **Conocer el problema:** Se debe estar seguro que se conoce el problema antes de definir una solución; de esta forma es sencillo mantener un enfoque y encontrar vías alternativas para solucionar el problema.
- **Mantener prioridades:** Se deben priorizar los requisitos al seleccionar qué es lo que se debe hacer en el próximo micro-incremento o iteración.

- **Contribuir con micro-incrementos:** A nivel personal, los miembros de equipo en un proyecto contribuyen con pequeñas entregas llamadas micro-incrementos que vienen siendo una iteración; los mismos representan el resultado de unas pocas horas o pocos días de trabajo. La aplicación evoluciona con cada micro-entrega en el tiempo y el progreso es visto efectivamente cada día.
- **Conocer cuándo se debe parar:** Recargar de características un sistema no sólo es una pérdida de tiempo y recursos sino que conduce a sistemas innecesariamente complejos (Universidad Distrital Francisco José de Caldas, 16 de octubre de 2012).

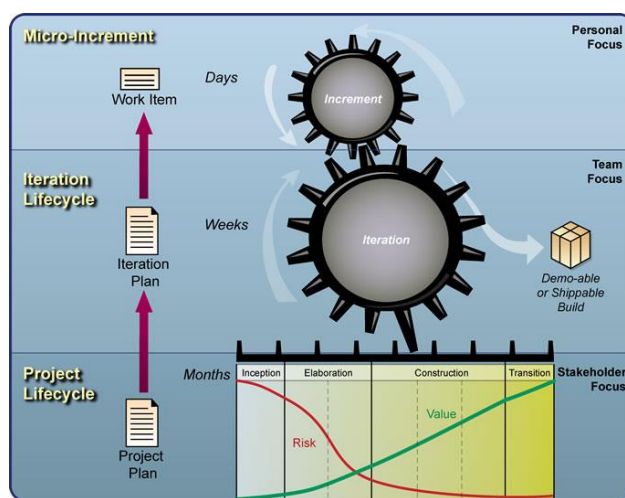


Figura N° 13 Ciclo de vida de un proyecto con la Metodología OpenUP.

Fuente: http://process.osellus.com/Process%20Documents/OpenUP/OpenUP_Images/three_layers.jpg

2.4 Roles

- **Interesados:** Representa el grupo interesado cuyas necesidades deben ser satisfechas por el proyecto; es un rol que puede ser tomado por cualquiera que vaya a ser afectado por el resultado del proyecto.
- **Analista:** Representa el cliente y las preocupaciones del usuario final; permite comprender el problema a resolver y la captura, prioridades y el establecimiento de los requisitos.
- **Arquitecto:** Es responsable por el diseño la arquitectura del software, que incluye la toma de decisiones claves técnica que limitan el diseño en general y ejecución del proyecto.
- **Desarrollador:** Es el responsable por el desarrollo de una parte del sistema incluyendo el diseño. También se encarga de la implementación, las pruebas unitarias y la integración de componentes que forman parte de la solución.
- **Probador:** Es el responsable de las actividades básicas de la prueba de esfuerzo, como la identificación, definir, implementar y llevar a cabo las pruebas necesarias, así como el registro de la resultados de las pruebas y análisis de los resultados.

- **Líder del proyecto:** Lleva la planificación del proyecto con colaboración con la parte interesada; mantiene al equipo de desarrollo enfocado en alcanzar los objetivos del proyecto (Eclipse, 2007).

2.5 Artefactos

Nombre	Descripción
Plan General del Proyecto	Este artefacto define los parámetros para realizar el direccionamiento y seguimiento al proyecto. Especifica los objetivos de alto nivel de las iteraciones y sus correspondientes hitos.
Casos de Uso	Captura la secuencia de acciones que un sistema realiza y que genera un resultado observable que es de valor para aquellos que interactúan con el sistema.
Listado de requerimientos y requisitos	En este documento se registran los requerimientos y requisitos que surjan a lo largo del proyecto, y sirve para priorizar y organizar las tareas, objetivos y metas del mismo.

Tabla 1 Artefactos de la metodología OpenUP.

Fuente: <http://www.udistrital.edu.co/files/dependencias/oas/GuiaRapidaOpenUPOAS.pdf>

En el próximo capítulo se indica con mayor detalle como la Metodología OpenUP fue utilizada en el desarrollo de la aplicación.

CAPÍTULO III

3. DESARROLLO DE LA APLICACIÓN MACUC

3.1 Objetivo general de la aplicación

Difundir y profundizar en el conocimiento de las obras de arte de la Ciudad Universitaria de Caracas bajo una experiencia digital didáctica *in situ* que permita enriquecer la visión del visitante a través del reconocimiento de las obras de arte de la Ciudad Universitaria de Caracas con un dispositivo móvil.

3.2 Objetivos específicos de la aplicación

1. Ofrecer herramientas para mejorar la concientización de conservación y comprensión sobre las obras de arte presentes en la Ciudad Universitaria de Caracas.
2. Exponer las riquezas en cuanto a las obras de arte presentes en la Ciudad Universitaria de Caracas.
3. Reconocer obras de arte a partir del reconocimiento de imágenes o posicionamiento geográfico.
4. Mostrar capas de información a través de la realidad aumentada con información proveniente del portal “El Arte en la Ciudad Universitaria de Caracas”.
5. Crear recorridos virtuales a través de las obras de arte.
6. Visualizar recorridos a través de las obras de arte.
7. Mostrar listados de obras pertenecientes al conjunto arquitectónico de la CUC.
8. Compartir información a través de distintos medios.

3.3 Alcance de la aplicación

La aplicación desarrollada consta de dos módulos explicados en el punto 3.7.2.3. Está orientada a la obtención y manejo de información sobre las obras expuestas a través de la realidad aumentada; específicamente obras de la colección "Síntesis de las Artes Mayores".

El alcance establecido para la aplicación es:

1. Elaboración de un conjunto de imágenes de reconocimiento.
2. Mostrar listados y mapas virtuales con obras de arte en donde destaquen las pinturas, mosaicos, esculturas, murales y edificaciones ubicados en la Ciudad Universitaria de Caracas.
3. Mostrar fichas informativas con respecto a las obras de arte y compartirlas.
4. Visualizar recorridos por las obras de arte mediante mapas virtuales.
5. Permitir la creación, visualización y compartición de recorridos virtuales.

6. Reconocer obras de arte a través del reconocimiento de imágenes y el posicionamiento geográfico.
7. Mostrar capas de información y funcionalidades mediante la realidad aumentada.
8. Permitir la gestión de la aplicación mediante un módulo administrativo.
9. Ofrecer la posibilidad de modificar el estilo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada, así como actualizar el paquete de imágenes de reconocimiento y realizar copias de seguridad.
10. Realización de pruebas de funcionalidad, navegabilidad y visualización del módulo de gestión del administrador en varios navegadores web.
11. Realización de pruebas de funcionalidad, navegabilidad y visualización del módulo de funcionalidades para dispositivos móviles en varias resoluciones de pantallas con distintas densidades de píxeles.

3.4 Plataforma utilizada en el desarrollo de la aplicación

- **Computador:**
 - Hardware:
 - Procesador: Intel Core i5-2430.
 - Memoria RAM: 4 Gb.
 - Software:
 - Sistema operativo: Linux Fedora 18.
 - Netbeans.
 - Eclipse con el *plugin* ADT.
- **Tableta:** Asus Transform TF300.
 - Hardware:
 - Procesador: NVIDIA Tegra 3.
 - Memoria RAM: 1 Gb.
 - Software:
 - Sistema operativo móvil: Android en su versión 4.2.1.
- **Teléfono:** Asus Zenfone 5.
 - Hardware:
 - Procesador: Intel Atom Z2560.
 - Memoria RAM: 2 Gb.
 - Software:
 - Sistema operativo móvil: Android en su versión 4.3.
- Dispositivo empleado para probar las funcionalidades del módulo de funcionalidades para dispositivos móviles:
 - **Tableta.**
 - **Teléfono.**

- Navegadores web utilizados para probar las funcionalidades del módulo de gestión del administrador:
 - **Computador:**
 - Google Chrome, Firefox, Opera e Internet Explorer 9 o superior.
 - **Tableta:**
 - Google Chrome, Firefox y Opera Mini.

3.5 Dispositivos utilizados para probar el módulo de funcionalidades para dispositivos móviles

- Asus Transformer TF300
- Asus Zenfone 5
- HTC One X
- HTC One S
- Samsung Galaxy Note 3
- Sony Xperia Z1

3.6 Metodología de Desarrollo de Software

Dentro de los objetivos principales de este trabajo está el tratar de cumplir las tareas planteadas de la mejor manera y con la mejor calidad; por lo tanto es recomendable el uso de una metodología que permita desarrollar de forma eficiente las funcionalidades del mismo. En la revisión de metodologías de desarrollo de software realizada en el trabajo de Seminario se seleccionó para tal fin OpenUp. Esta metodología indica las actividades a realizar para coordinar pequeños proyectos y gestionar distintos requerimientos asociados a la creación de software y su ciclo de vida. A continuación se explican los pasos que se realizaron en cada fase de la metodología OpenUp:

3.6.1 Fase de Inicio

En esta fase se comprendió el alcance del proyecto: lista de requerimientos, funcionalidades, arquitectura y módulos a construir.

Además se evaluaron las posibles restricciones para el desarrollo del proyecto lo que llevo un cambio en la elección inicial de dos tecnologías (ver punto 1.9) propuestas en el Trabajo de Investigación de Seminario.

Se asociaron los roles de la metodología OpenUp Analista, Probador, Desarrollador, Arquitecto y parte de las responsabilidades del Líder de Proyecto fueron asumidos por el Estudiante, la Tutora ocupó las responsabilidades restantes del rol de Líder de Proyecto y la Escuela de Arte de la Universidad Central de Venezuela tomó el rol de Grupo Interesado y Analista; en la tabla 2 se hace la descripción y especificación de las responsabilidades dentro del sistema de cada uno de los participantes del proyecto:

Nombre	Descripción	Roles
Estudiante	Es la persona encargada de analizar y desarrollar los distintos módulos, además de la elección de las tecnologías a emplear en la aplicación y desarrollo de funcionalidades.	Líder del Proyecto, Analista, Probador, Desarrollador y Arquitecto.
Tutora	Es la persona encargada de regir los avances del proyecto.	Líder de proyecto.
Escuela de Arte UCV	Grupos de interés cuyas necesidades deben ser satisfechas por el proyecto.	Grupo Interesado y Analista.

Tabla 2 Participantes del proyecto.

Fuente: Creada por el Autor.

Siguiendo como modelo las actividades presentadas en el documento de la Universidad Distrital Francisco José de Caldas (s.f.), en la siguiente tabla se realiza una adaptación para especificar las actividades que cumplen cada uno de los participantes del equipo de este proyecto:

Nombre	Descripción	Roles
Tutora	<ul style="list-style-type: none"> Estudia y aprueba las propuestas que se definan para el método y proceso de desarrollo de software. Direccionamiento y aprobación general de las actividades definidas en el proyecto. Seguimiento a objetivos, metas y resultados arrojados en el proyecto. Proponer y participar en el diseño de las soluciones asociadas a los requerimientos. Supervisa la calidad técnica y tecnológica del desarrollo de software. Es responsable del resultado del proyecto y la aceptación del producto por parte del interesado. 	Líder del Proyecto
Estudiante	<ul style="list-style-type: none"> Se encarga de establecer las condiciones de trabajo. Seguimiento a objetivos, metas y resultados arrojados en el proyecto. Aplica conocimientos de gestión, competencias, herramientas y técnicas a un amplio rango de tareas con el propósito de alcanzar los resultados deseados para un proyecto en particular en tiempo oportuno. Proponer y participar en el diseño de las soluciones asociadas a los requerimientos. Es responsable del resultado del proyecto y la aceptación del producto por parte del interesado. 	Líder del Proyecto
Estudiante	<ul style="list-style-type: none"> Este rol es el responsable de diseñar la arquitectura del software, la cual incluye tomar las principales decisiones técnicas que condicionan globalmente el diseño y la implementación del proyecto. Este rol lidera o coordina el diseño técnico del sistema y tiene la responsabilidad general de manejar las principales decisiones técnicas, expresadas en la arquitectura del software. Esto típicamente incluye identificar y documentar los aspectos arquitecturalmente significativos del sistema, incluyendo vistas de requerimientos, diseño, implementación y despliegue. El rol de arquitecto es también responsable de razonar sobre estas decisiones, equilibrando las preocupaciones de los interesados, reduciendo los riesgos técnicos y asegurando que las decisiones se comunican eficazmente. 	Arquitecto

Nombre	Descripción	Roles
Estudiante	<ul style="list-style-type: none"> La persona en este rol es responsable por desarrollar una parte del sistema, incluyendo diseñar esta para que se ajuste a la arquitectura, crear un prototipo de la interfaz de usuario y entonces implementar, hacer pruebas unitarias e integrar los componentes que son parte de la solución. 	Desarrollador
Estudiante	<ul style="list-style-type: none"> Este rol es responsable de las actividades principales del esfuerzo de las pruebas. Estas actividades incluyen identificar, definir, implementar y dirigir las pruebas necesarias, así como verificar y analizar sus resultados. 	Probador
Escuela de Artes UCV	<ul style="list-style-type: none"> Las personas en este rol representan al cliente y los usuarios finales involucrados, obteniendo información desde el grupo interesado interesados para entender el problema a ser resuelto y capturar y ajustar las prioridades para los requerimientos. Realizar tareas de relevamiento, análisis y diseño de los requerimientos y requisitos en el proyecto. Supervisar las pruebas de la aplicación. 	Analista
Escuela de Artes UCV	<ul style="list-style-type: none"> Este rol representa los grupos de interés cuyas necesidades deben ser satisfechas por el proyecto. Esto es un rol que podría ser desempeñado por cualquiera que esté o potencialmente estará materialmente afectado por el resultado del proyecto. Definición de requerimientos. 	Grupo Interesado

Tabla 3 Roles de los participantes del proyecto.

Fuente: Creada por el Autor.

3.6.1.1 Usuarios destino de la aplicación

Público en general que visita la Ciudad Universitaria de Caracas con intereses culturales; así como estudiantes, arquitectos, artistas, docentes e investigadores. Especialmente aquellos que hacen uso de dispositivos móviles inteligentes, que tengan interés en el patrimonio artístico y en las obras de la Ciudad Universitaria de Caracas.

3.6.1.2 Requerimientos funcionales

1. Reconocer obras de arte de la Ciudad Universitaria de Caracas a través del reconocimiento de imágenes o el posicionamiento geográfico utilizando un dispositivo móvil y mostrar la ficha informativa de la obra reconocida, las obras relacionadas y otras obras del autor.
2. Crear, editar, cargar, ver y compartir recorridos del Visitante.
3. Visualizar los recorridos sugeridos por el Editor en los mapas de Google.
4. Compartir la ficha informativa de las obras de arte a través de distintos medios, principalmente redes sociales y correos electrónicos.
5. Visualizar las obras del conjunto arquitectónico de la CUC a través de consultas al repositorio de datos del portal “El Arte en la Ciudad Universitaria de Caracas”.
6. Gestionar los datos de la aplicación mediante un módulo administrativo.

3.6.1.3 Requerimientos no funcionales

1. **Fiabilidad:** La aplicación deberá ser fiable, ya que la misma fue desarrollada con tecnologías probadas, estándares y bajo una metodología de software evolutiva.
2. **Disponibilidad:** La aplicación será accesible continuamente siempre que se disponga de una conexión a Internet.
3. **Seguridad:** La aplicación deberá ser segura, ya que para acceder al módulo de gestión del administrador es necesario utilizar una contraseña cifrada; además la aplicación en toda su envergadura utiliza *framework*, librerías de desarrollo de software y APIs; códigos que han sido sometidos a procesos de prueba y actualización respecto a lineamientos de seguridad.
4. **Escalabilidad:** La aplicación deberá ser escalable, estará diseñada bajo un esquema modular lo que permite integrar nuevos módulos a los ya desarrollados e inclusive utilizar servicios ya desarrollados.

3.6.1.4 Perfiles creados en la aplicación:

- **Visitante:** Usuarios que posean un dispositivo móvil inteligente con el sistema operativo móvil Android que deseen conocer las obras pertenecientes a la colección “Síntesis de las Artes Mayores” y/o compartir la información obtenida.
- **Administrador:** Usuarios con los permisos de acceso necesarios para ingresar al módulo de gestión del administrador para listar, crear, modificar y eliminar los recorridos e imágenes con las cuales la aplicación móvil identificará cada una de las obras; además de la modificación del estilo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada, actualizar el paquete de reconocimiento de imágenes y realizar respaldos de los datos de la aplicación.
- **Editor:** Son usuarios con acceso a todas las funcionalidades del módulo de gestión del administrador, con la excepción de que no pueden eliminar o modificar ningún tipo de información y actualizar el paquete de imágenes de reconocimiento.

A continuación se muestra en la siguiente tabla las principales operaciones que pueden realizar cada tipo de usuario en la aplicación:

Sub-módulo	Funcionalidad	Visitante	Administrador	Editor
Gestión de imágenes de reconocimiento	Registrar imágenes de reconocimiento.		X	X
	Ver imágenes de reconocimiento.		X	X
	Eliminar imagen de reconocimiento.		X	
	Aumentar contraste a las imágenes de reconocimiento.		X	
	Aplicar la operación de recordé a las		X	

Sub-módulo	Funcionalidad	Visitante	Administrador	Editor
	imágenes de reconocimiento.			
	Descargar imágenes de reconocimiento.		X	X
Gestión de recorridos del Editor	Registrar recorridos.		X	X
	Ver recorridos.	X	X	X
	Eliminar recorridos.		X	
	Modificar recorridos.		X	
Estilo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada	Modificar el estilo.		X	
Actualizar el paquete de imágenes de reconocimiento	Subir paquete.		X	
Respaldo	Respaldo de la base de datos y de las imágenes de reconocimiento de la aplicación.		X	X
Recorridos del Visitante	Crear recorridos.	X		
	Eliminar recorridos.	X		
	Editar recorridos.	X		
	Compartir recorridos.	X		
	Ver recorridos.	X		
	Cargar recorridos.	X		
RI y posicionamiento geográfico con RA	Despliegue de información a través de la realidad aumentada.	X		
Compartir	Fichas informativas de las obras de arte.	X		
Listado del conjunto arquitectónico	Obtener listado.	X		

Tabla 4 Operaciones permitida en la aplicación según el tipo de usuario.

Fuente: Creada por el Autor.

3.6.1.5 Lista de funcionalidades

Basado los objetivos y requerimientos que debe de cumplir la aplicación, se elabora la siguiente lista de funcionalidades; la cual se encuentra agrupada en dos conjuntos principales:

Funcionalidades a nivel de usuario con perfil Administrador y Editor:

1. Gestionar las imágenes de reconocimiento:

- 1.1. **Listar imágenes:** Despliega la información en una tabla. Para cada registro es mostrado el nombre del autor, nombre de la obra de arte, tipo de la obra de arte y las acciones relacionadas (ver, eliminar, agregar contraste, operación de recorte y descargar); además es posible filtrar los datos por cada campo del registro (exceptuando el menú de acciones), búsqueda general por texto, mostrar un número determinado de registros, paginación y ordenación por campos (exceptuando el menú de acciones).
- 1.2. **Visualizar:** Permite mostrar imágenes representativas de los objetos a reconocer; estas imágenes son modelos o metadata para el reconocimiento de las obras reales.
- 1.3. **Eliminar imágenes:** Permite eliminar imágenes de reconocimiento de la aplicación y el registro de la base de datos.
- 1.4. **Agregar contraste:** Permite aumentar el contraste de las imágenes de reconocimiento.
- 1.5. **Recortar:** Permite aplicar la operación de recorte sobre las imágenes de reconocimiento.
- 1.6. **Descargar:** Permite descargar imágenes de reconocimiento en la computadora o dispositivo móvil.
- 1.7. **Registrar imágenes de reconocimiento:** Permite subir imágenes a la aplicación.

2. Gestionar los recorridos creados por el Editor:

- 2.1. **Listar recorridos:** Despliega la información en una tabla, para cada registro es mostrado el nombre del recorrido y las acciones relacionadas (vista previa, eliminar y modificar); además permite búsqueda general por texto, mostrar un número determinado de registros, paginación y ordenación por campos (exceptuando el menú de acciones).
- 2.2. **Visualizar:** Permite mostrar recorridos en los mapas de Google.
- 2.3. **Eliminar recorrido:** Permite eliminar recorridos de la base de datos.
- 2.4. **Modificar recorrido:** Permite modificar el nombre, descripción y los objetos que conforman el recorrido.

3. Modificar el estilo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada: Permite modificar el color de fondo, color del texto, tipografía, tamaño de la letra y estilo del borde.

4. Actualizar el paquete de imágenes de reconocimiento: Permite actualizar el paquete de imágenes de reconocimiento generado en el portal de Wikitude.

Funcionalidades a nivel de usuario con perfil Visitante:

1. **Reconocimiento de objetos:** Permite identificar obras de arte a través del reconocimiento de los mismos o por su posición geográfica y mostrar fichas informativas, obras relacionadas y otras obras del autor.
2. **Recorridos sugeridos por el Editor y recorridos del Visitante:** Permite visualizar recorridos en un mapa de Google mediante marcadores y obtener más información sobre los objetos expuestos a través de una ficha informativa de la obra de arte expuesta.
3. **Gestionar los recorridos del Visitante:** Este sub-módulo está compuesto por siete funcionalidades las cuales permiten listar, crear, agregar, eliminar, compartir, cargar y ver los recorridos que hayan sido creados por los usuarios; dentro del módulo de funcionalidades para dispositivos móviles este sub-módulo es llamado con el nombre de “Mis Recorridos”; esto se debe a que el usuario es capaz de crear y modificar recorridos como los desee utilizando las siguientes funcionalidades:
 - 3.1. **Crear:** Esta funcionalidad permite crear recorridos del Visitante; estos recorridos se consideran como “vacíos” al no contener ninguna obra de arte dentro de los mismos.
 - 3.2. **Agregar:** Para agregar obras de arte a recorridos del Visitante es solo posible desde la ficha informativa de alguna obra de arte.
 - 3.3. **Listar:** Permite visualizar los recorridos creados por el Visitante.
 - 3.4. **Eliminar:** Permite eliminar los recorridos u obras de arte del recorrido del Visitante.
 - 3.5. **Compartir:** Permite compartir recorridos del Visitante por distintos medios.
 - 3.6. **Cargar:** Permite cargar recorridos del Visitante compartidos por otros usuarios de la aplicación.
4. **Listados del conjunto arquitectónico de la CUC:** Este sub-módulo permite mostrar listados clasificados por autores, tipos de obras de arte, edificaciones, zonas del conjunto arquitectónico de la CUC y fichas informativas, con datos registrados en el portal “El Arte en la Ciudad Universitaria de Caracas”.

3.6.1.6 Diagramas de Casos de Uso

A continuación se muestra el nivel 0 de los módulos que conforman a la aplicación:

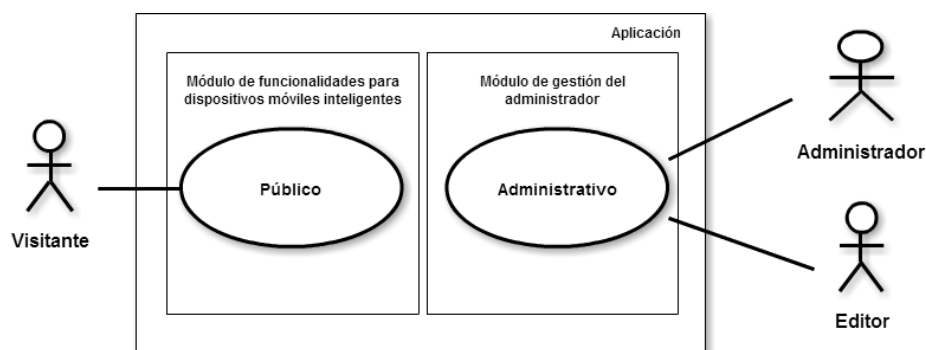


Figura N° 14 Diagrama de Casos de Uso, nivel 0.

Fuente: Creada por el Autor.

A continuación se muestra el nivel 1 del módulo de funcionalidades para dispositivos móviles inteligentes:

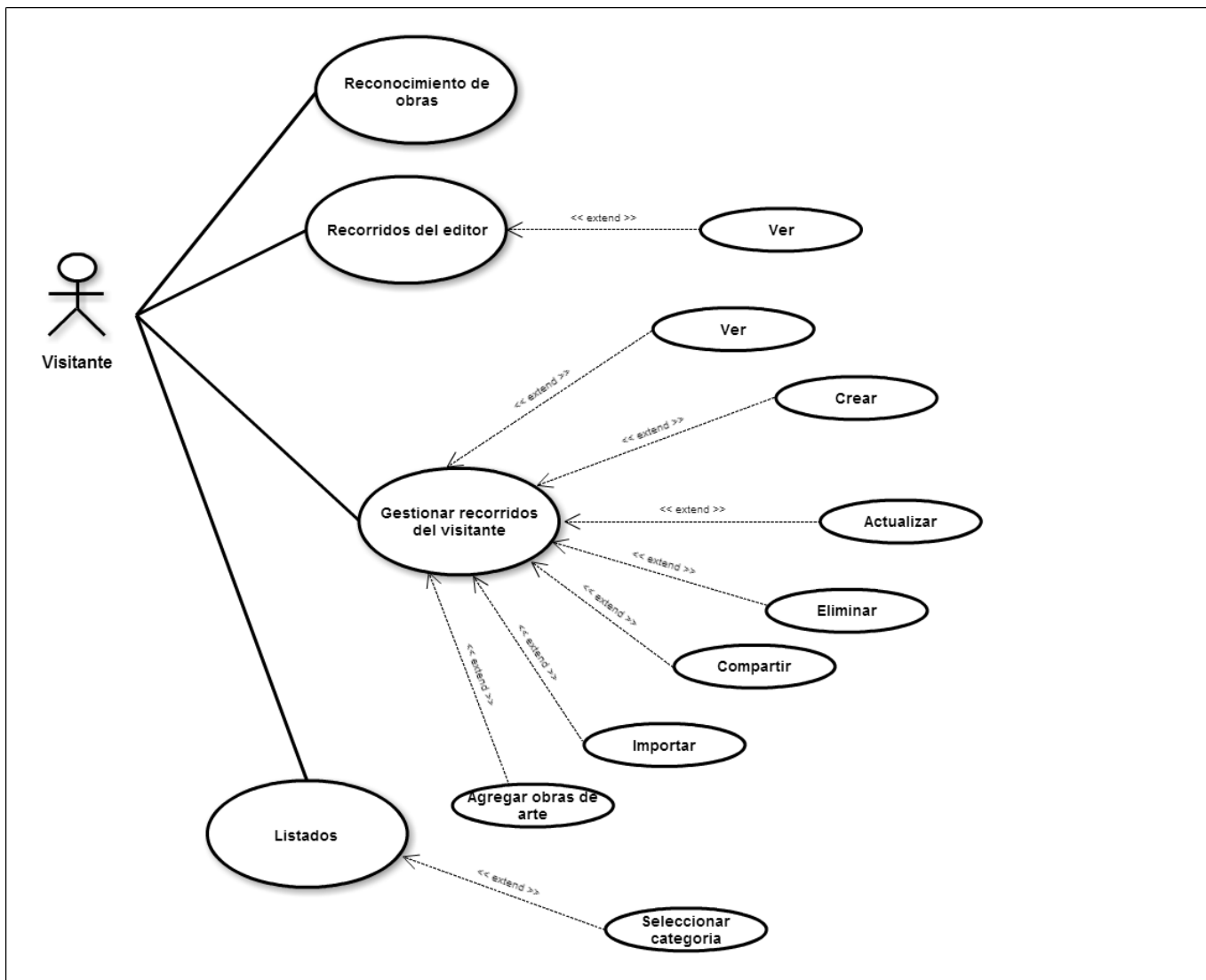


Figura N° 15 Diagrama de Casos de Uso nivel 1 del módulo de funcionalidades para dispositivos móviles inteligentes.

Fuente: Creada por el Autor.

A continuación se muestra el nivel 1 del módulo de gestión del administrador con perfil Administrador:

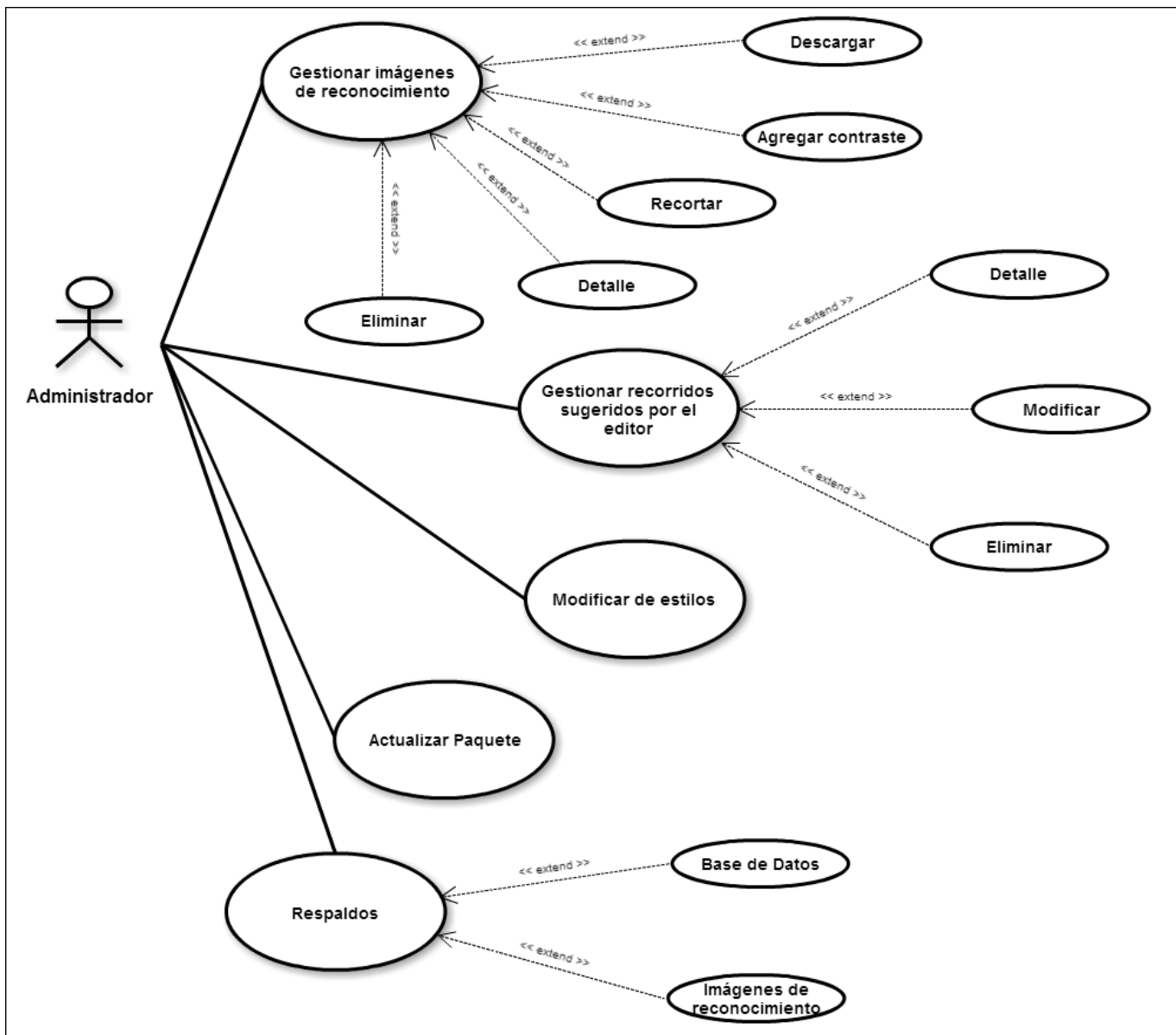


Figura N° 16 Diagrama de Casos de Uso nivel 1 del módulo de gestión del administrador con perfil Administrador.

Fuente: Creada por el Autor.

A continuación se muestra el nivel 1 del módulo de gestión del administrador con perfil Editor:

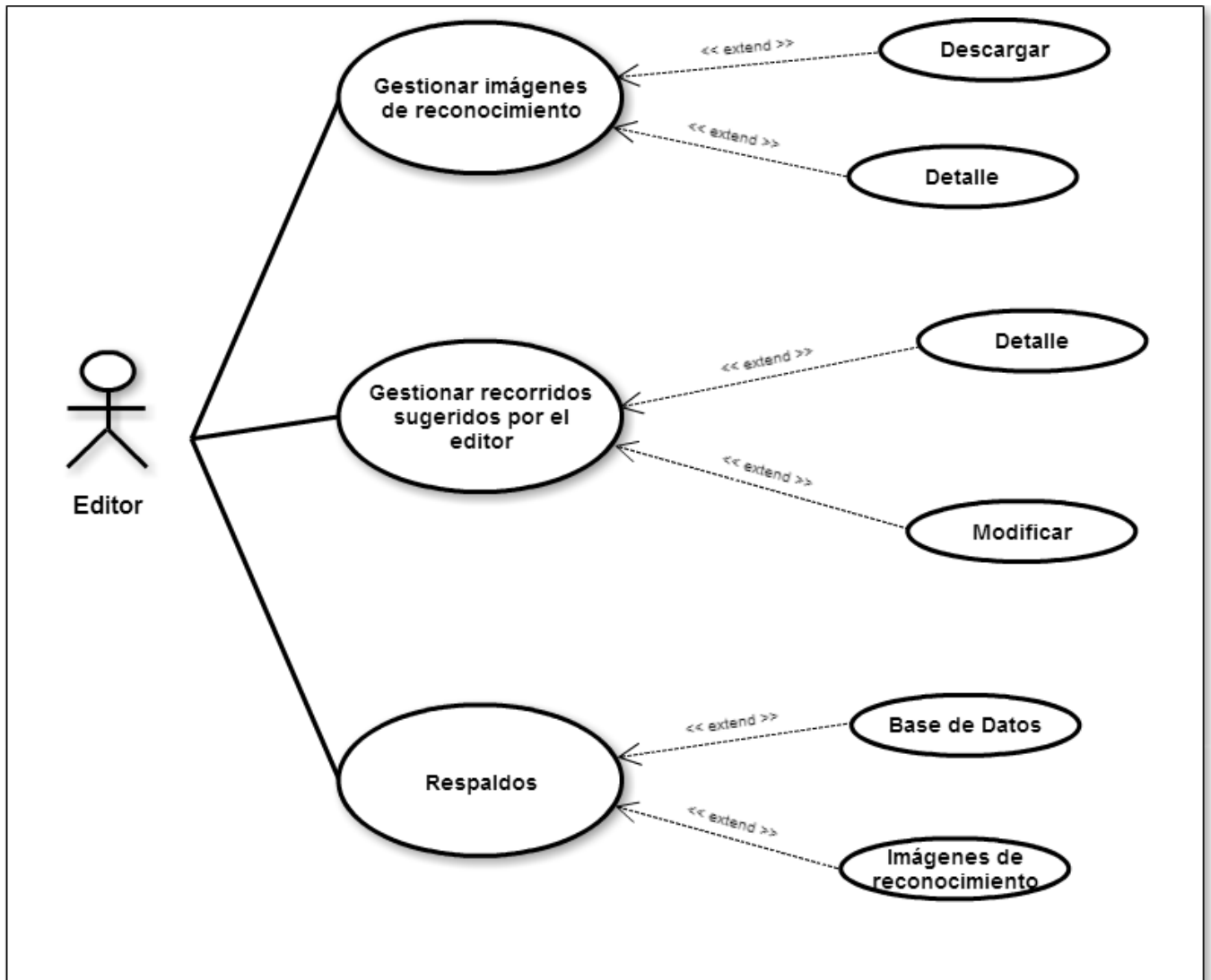


Figura N° 17 Diagrama de Casos de Uso nivel 1 del módulo de gestión del administrador con perfil Editor.
Fuente: Creada por el Autor.

3.6.1.7 Terminología empleada para el nombramiento de los elementos de la aplicación desarrollada

- **Recorridos del Visitante:** Son recorridos creados por un visitante a través del módulo de funcionalidades para dispositivos móviles inteligentes con el objetivo de administrar recorridos propios y compartirlos a otros usuarios del módulo de funcionalidades para dispositivos móviles.
- **Recorridos sugeridos por el Editor:** Son recorridos creados por el editor a través del módulo de gestión del administrador con el objetivo de ser accedidos por cualquier visitante que utilice el módulo de funcionalidades para dispositivos móviles inteligentes a través de los mapas de Google.

- **Imágenes de reconocimiento:** Imágenes con el formato, definición, tamaño y contraste adecuados para ser utilizadas como metadata para generar el paquete de imágenes de reconocimiento y de esta forma reconocer obras de arte a través del reconocimiento de imágenes de Wikitude.
- **Paquete de imágenes de reconocimiento:** Es un archivo compuesto de información sobre imágenes que utiliza la herramienta de desarrollo de software (SDK) Wikitude con el objetivo de reconocer objetos del mundo real a través de información de imágenes.
- **Ficha informativa:** Son bloque de texto que permite registrar información de una manera organizada sobre un tema determinado; en otras palabras, contienen información organizada e importante sobre un tema; para este trabajo las fichas informativas manejan información sobre obras de arte.

3.6.1.8 Costos y Riesgos

La aplicación a desarrollar tiene como limitante la necesidad de estar constantemente conectada a Internet para extraer de información provenientes de la base de datos del portal “El Arte en la Ciudad Universitaria de Caracas” y así poder utilizar completamente las funcionalidades del sistema. En cuanto al módulo de funcionalidades para dispositivos móviles es necesario que el dispositivo cumpla con los requerimientos mínimos provistos en la documentación oficial de *Wikitude Developer*:

- Sistema operativo móvil Android en su versión 4 o superior.
- Alta densidad por píxeles (*high density per pixel*).
- El dispositivo debe tener una cámara trasera.

A diferencia del módulo de gestión del administrador que se encuentra alojado en un Servidor Web y se accede mediante un navegador web; el módulo de funcionalidades para dispositivos móviles requiere ser descargado e instalado en un dispositivo.

El módulo de gestión del administrador al ser una aplicación Web, permite ser accedido y utilizado en múltiples plataformas (sistemas operativos) con distintos navegadores web como lo son: Opera, Google Chrome, Internet Explorer (versión 9 en adelante) y Firefox; es necesario un navegador web con soporte a HTML5 y CSS3; muchas de las funcionalidades desarrolladas en el módulo emplean estas tecnologías como el caso del lienzo (*Canvas*) para el desarrollo de las funcionalidades de recorte y del aumento del contraste de las imágenes de reconocimiento.

3.6.2 Fase de Elaboración

En esta fase se definió la arquitectura del sistema. Se obtuvo un mejor entendimiento de los requerimientos y funcionalidades de la aplicación y de esta forma sobre lo que se desea construir; se logró crear un plan de proyecto para lograr una organización de trabajo más preciso y de esta forma un ajuste de los tiempos de desarrollo del sistema propuesto.

En esta fase se define como es la arquitectura en base de los requerimientos y funcionalidades ya identificadas y definidas en la fase anterior; fue posible identificar y definir la planeación basada en micro-incrementos.

3.6.2.1 Planeación basada en micro-incrementos (Plan del Proyecto)

La metodología OpenUp permite un desarrollo ágil cuando el tiempo de desarrollo no es muy largo; consiste en hacer pequeñas entregas llamadas micro-incrementos en cortos periodos de tiempo (horas, días o unas pocas semanas); lo que permite desarrollar un gran número de funcionalidades en un corto período de tiempo; por lo tanto, la planificación de proyecto fue creada bajo este esquema de pequeñas entregas:

Nro.	Descripción	Duración
1	Diseño lógico y físico de la base de datos del módulo de gestión del administrador.	2 días
2	Diseño de la hoja de estilo del módulo de gestión del administrador.	2 días
3	Diseño lógico del módulo de gestión del administrador.	4 días
4	Desarrollo de la interfaz del módulo de gestión del administrador.	3 día
5	Desarrollo de las funcionalidades de administración de las imágenes de reconocimiento.	3 semana
6	Desarrollo de las funcionalidades de administración de los recorridos sugeridos.	1 semana
7	Desarrollo de las funcionalidades de administración del estilo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada.	2 día
8	Desarrollo de la funcionalidad para la carga del paquete de imágenes de reconocimiento.	1 día
9	Desarrollo de las funcionalidades para la autenticación.	1 día
10	Diseño lógico del módulo de funcionalidades para dispositivos móviles inteligentes.	3 días
11	Diseño lógico y físico de la base de datos del módulo de funcionalidades para dispositivos móviles.	1 día
12	Diseño del estilo del módulo de funcionalidades para dispositivos móviles inteligentes.	5 día
13	Desarrollo de las funcionalidades para la realidad aumentada, reconocimiento de imágenes y posicionamiento geográfico de las obras de arte.	4 semanas
14	Desarrollo de las funcionalidades para la visualización de los recorridos en móviles.	3 días
15	Desarrollo de las funcionalidades para crear, listar, eliminar, modificar, ver, cargar y compartir los recorridos del Visitante.	2 semanas
16	Desarrollo de las funcionalidades para compartir contenido.	1 día
17	Desarrollo del menú de navegación para el sistema operativo móvil Android.	3 días
18	Desarrollo de los listados del conjunto arquitectónico de la CUC y las fichas informativas.	5 días
19	Secciones de ayuda.	2 días
20	Pruebas de aceptación con usuarios.	1 mes
21	Incorporación de mejoras según el análisis de las encuestas.	1 semana
22	Pruebas para verificar el correcto funcionamiento de la aplicación.	1 semana

Tabla 5 Lista de actividades.

Fuente: Creada por el Autor.

3.6.2.2 Diseño de la Base de Datos

El siguiente paso luego de estudiar los requerimientos y funcionalidades a desarrollar en los distintos módulos que conforman a la aplicación, consiste en crear un diseño lógico de la base

de datos; este trabajo consiste en una extensión del portal Informativo y Divulgativo del Patrimonio Artístico de la Ciudad Universitaria de Caracas llamado: “El Arte en la Ciudad Universitaria de Caracas”; lo que permite reutilizar una gran parte de las tablas provistas en dicho portal:

Tabla	Descripción	Relación
objetos	Incluye cualquier tipo de objeto del portal; cada uno de los elementos del conjunto, es decir, las obras de arte, paisajismos, entre otros. Y las fotos panorámicas.	
metaelementos	Incluye información específica para los objetos que son elementos del conjunto, es decir, las obras de arte, paisajismos, entre otros.	objetos.
autores	Incluye la lista de autores.	
autores_objetos	Incluye las relaciones entre un autor y un objeto.	objetos, autores.
imagenes	Incluye la información de las imágenes para cada objeto, panorámicas y las múltiples imágenes que puede tener un elemento del conjunto.	objetos.
users	Incluye usuarios del sistema en representación de las credenciales de acceso.	

Tabla 6 Descripción de algunas tablas de la base de datos del portal “el arte en la CUC” y algunas de sus Relaciones.

Fuente: Creada por el Autor.

Sin embargo fue necesaria la creación de un conjunto de tablas para agregar las funcionalidades descritas en la primera fase del desarrollo:

Tabla	Descripción	Relación
meta_imagenes	Incluye datos específicos a las imágenes de reconocimiento.	objetos.
recorridos	Incluye datos específicos sobre los recorridos creados por el Editor.	objetos.
hoja_estilo	Incluye datos específicos sobre el estilo (color de fondo, color de borde, tamaño de la letra y tipografía).	
variables_personalizadas	Incluye datos específicos sobre la aplicación que pueden cambiar en el tiempo; por ejemplo la ubicación de las imágenes de las obras de arte o la carpeta para subir imágenes a la aplicación.	

Tabla 7 Descripción de las nuevas tablas de la base de datos del portal “el arte en la CUC” y sus relaciones.

Fuente: Creada por el Autor.

En la figura N° 18, se define la estructura de la base de datos que se utilizó en el desarrollo de la aplicación:

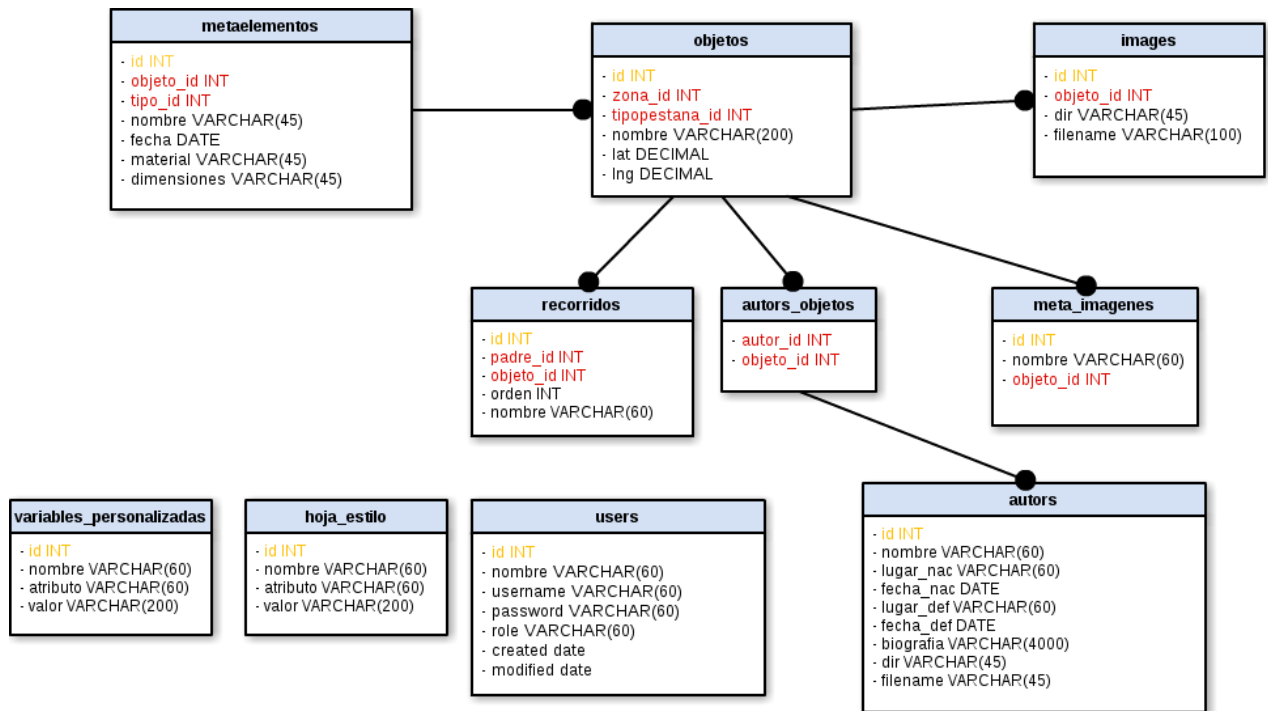


Figura N° 18 Modelo de Datos del módulo de funcionalidades para dispositivos móviles.

Fuente: Creado por el autor

En la figura N° 19 se define la estructura de la base de datos que se utilizó en el desarrollo de la aplicación para el sistema operativo móvil Android:

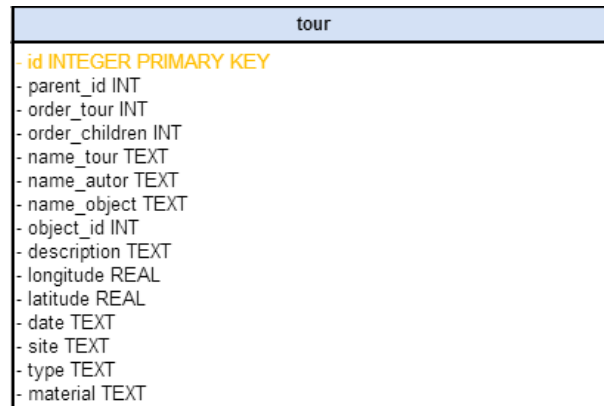


Figura N° 19 Modelo de Datos del módulo de funcionalidades para dispositivos móviles.

Fuente: Creada por el Autor.

3.6.2.3 Arquitectura de la aplicación

La aplicación desarrollada consta de dos módulos:

1. Módulo de funcionalidades para dispositivos móviles, el cual constará con cinco sub-módulos para:

- 1.1. **Reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada:** Permite reconocer obras de arte de la CUC que el usuario está visualizando a través de la cámara de su dispositivo móvil inteligente y mostrar la ficha informativa y obras relacionadas a través de la realidad aumentada.
 - 1.2. **Recorridos sugeridos por el Editor:** Permite mostrar al visitante los recorridos creado por el Editor por las obras de arte de la CUC y mostrarlos a través de un mapa de Google.
 - 1.3. **Recorridos del Visitante:** Permite al visitante crear y editar sus propios recorridos por las obras de arte de la CUC y verlos en un mapa de Google; además de poder compartirlos.
 - 1.4. **Listados del conjunto arquitectónico de la CUC:** Permite visualizar las obras de arte de la CUC en un listado filtrado por autores, tipos de obra de arte, zonas y fechas.
 - 1.5. **Compartir la ficha informativa:** Permite compartir la ficha informativa de las obras de arte de la CUC a través de las redes sociales, correos electrónicos, entre otros.
2. Módulo de gestión del administrador, aplicación Web constará de cinco sub-módulos para:
 - 2.1. Gestionar (listar, crear, modificar y eliminar) las imágenes con las cuales el módulo de funcionalidades para dispositivos móviles identificará cada una de las obras de la CUC (imágenes de reconocimiento).
 - 2.2. Gestionar (listar, crear, modificar y eliminar) los recorridos del Editor.
 - 2.3. Modificar el estilo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada
 - 2.4. Actualizar el paquete de imágenes de reconocimiento.
 - 2.5. Respalidar los datos de la aplicación.

La arquitectura para cada módulo varía y es una variante del patrón Modelo–Vista–Controlador (MVC):

Arquitectura para el módulo de funcionalidades para dispositivos móviles: MVC en el sistema operativo móvil Android:

Al desarrollar aplicaciones para el sistema operativo móvil Android se utiliza el patrón MVC para separar los datos de una aplicación, la interfaz de usuario y la lógica de negocios en tres componentes distintos que se relacionarán para al final tener como resultado una aplicación para el sistema operativo móvil Android:

- **Modelo:** Son las entidades o cualquier otro objeto de negocio que describen los componentes de las aplicaciones.
- **Vista:** Son los XML de diseño (*layouts*) necesarios para definir la estructura visual de la interfaz de usuario; además permite al usuario manipular la data de los Modelos.
- **Controlador:** Son las actividades (*activity*); las actividades son definidas para indicar las acciones del usuario; además son las intermediarias entre los Modelos y las Vistas.

Arquitectura para el módulo de gestión del administrador: MVC en CodeIgniter:

El *framework* CodeIgniter está basado en el patrón de desarrollo Modelo-Vista-Controlador (MVC); permite separar la lógica de la aplicación de la presentación; en la práctica, permite que las páginas de presentación contengan una cantidad mínima de código PHP; permitiendo la modularización de las aplicaciones y la reutilización de componentes:

- **Modelo:** Representa la data estructurada; generalmente contiene funciones de inserción, actualización y recuperación de datos de la base de datos.
- **Vista:** Es la información que es presentada al usuario final; normalmente es una página web con HTML-CSS-JavaScript; pero en CodeIgniter, puede ser la cabecera, la sección de contenido, el pie de página, menús o cualquier otro elemento.
- **Controlador:** Es el intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición de HTTP.

3.6.3 Fase de Construcción

Construcción basada en micro-incrementos: Los micro-incrementos son pequeñas secciones de código con una funcionalidad específica que permite una evolución constante de la aplicación a desarrollar; los siguientes puntos a desarrollar, consisten en explicar las principales funcionalidades a nivel de código.

En los siguientes puntos se explica el desarrollo de las funcionalidades de la aplicación en un detalle más técnico; se ha dividido la sección en dos puntos, uno por cada módulo que conforman a la aplicación; cada módulo cuenta con sub-módulos que a su vez se fragmentan en funcionalidades como veremos a continuación:

3.6.3.1 Módulo de gestión del administrador

En esta sección explicaremos las principales funcionalidades desarrolladas en el módulo de gestión del administrador:

1. Construcción de la herramienta para la autenticación:

Se desea poder utilizar los usuarios que se encuentran actualmente registrados en la base de datos del portal cuyo módulo administrativo maneja distintos algoritmos de codificación, entre ellos el SHA1 para generar las claves codificadas a partir de una clave registrada en un archivo de configuración de CakePHP, al ser un algoritmo hash se debe codificar la clave

introducida por el usuario cada vez que se desee verificar sus credenciales al momento de la autenticación; se desarrolló un controlador en el módulo de administración del portal que dada una cadena de texto (la clave del usuario) en texto plano, la codifique y retorne; de esta forma es posible consumir el controlador como si fuese un servicio y comparar la contraseña codificada devuelta con la contraseña registrada en la base de datos del portal desde cualquier lenguaje de programación web; el controlador desarrollado es el siguiente:

```
function get-hash-password(){
    //desabilito el layout
    $this->layout = null;
    // si es post
    if ($this->request->is('post'))
        echo json_encode(array('password' => Security::hash($_POST['password'], NULL, true)));
}
```

Figura N° 20 Código para generar una contraseña codificada desde el portal.

Fuente: Creada por el Autor.

2. Construcción de la herramienta para la gestión de las imágenes de reconocimiento (operaciones de listar, registrar, descargar, recortar, aumentar contraste, eliminar y visualizar):

- 2.1. **Listar:** Vista (*index*) – controlador (*image*) – modelo (*meta_imagenes*): La tabla de la vista de la función *index()* del controlador *image* se conecta mediante tecnología Ajax con la vista *json()* para obtener un listado de todos los autores, sus objetos y sus imágenes de reconocimiento mediante la función *getAllMetaImageAutorAndObject()* y de esta forma construir una tabla con dichos datos; esta vista presenta los enlaces que permiten el acceso al resto de las funcionalidades de este sub-módulo.
- 2.2. **Registrar:** Vista (*create*) – controlador (*image*) – modelo (*meta_imagenes*): Esta vista muestra un formulario que permite registrar una imagen de reconocimiento por vez dado su autor y el objeto representado.
- 2.3. **Recortar imágenes:** Vista (*cut_image*) - controlador (*image*) – modelo (*meta_imagenes*): Esta vista está compuesta por una imagen de reconocimiento dibujada en un lienzo; de tal forma que el usuario seleccione la zona de interés y obtenga una imagen a partir de la selección.
- 2.4. **Descargar:** Vista (*download*) - controlador (*image*) - modelo (no tiene): Esta operación permite descargar imágenes de reconocimiento redimensionadas a no más de 1000 píxeles de alto y/o ancho.
- 2.5. **Aumentar contraste:** Vista (*add_contrast*) - controlador (*image*) - modelo (*meta_imagenes*): Esta vista está compuesta por una imagen de reconocimiento dibujada en un lienzo; permite aumentar el contraste de una imagen de reconocimiento a través de un control de tipo rango (*range*).

- 2.6. **Eliminar:** Vista (no tiene) - controlador (*image*) - modelo (*meta_imagenes*): El controlador utiliza la función *delete()* para eliminar el registro de la base de datos; además elimina físicamente la imagen de reconocimiento de la aplicación.
 - 2.7. **Visualizar:** Vista (no tiene) - controlador (no tiene) - modelo (no tiene): Esta operación permite mostrar imágenes de reconocimiento registradas en la aplicación.
- 3. Construcción de la herramienta para la gestión de recorridos sugeridos por el Editor (operaciones de listar, registrar, eliminar y visualizar):**
- 3.1. **Listar:** Vista (*index*) – controlador (*tour*) - modelo (*tour*): La tabla de la vista de la función *index()* del controlador *tour* se conecta mediante Ajax con la vista *json()* para obtener un listado de todos los recorridos, sus objetos y sus imágenes mediante la función *getAllParent()* del modelo *tour* y de esta forma construir una tabla con dichos datos; esta vista presenta los enlaces que permiten el acceso al resto de las funcionalidades de este sub-módulo.
 - 3.2. **Registrar y actualizar:** Vista (*create_update*) – controlador (*tour*) – modelo (*tour*): Esta vista muestra un formulario que permite registrar y/o actualizar un recorrido; el recorrido está compuesto por objetos que son filtrados por autores o por tipos; los recorridos son desplegados en un mapa de Google para su selección y visualización y así formar un recorrido.
 - 3.3. **Eliminar:** Vista (no tiene) - controlador (*tour*) - modelo (*tour*): El controlador utiliza la función *delete()* para eliminar los registros que conforman al recorrido de la base de datos.
 - 3.4. **Visualizar:** Vista (no tiene) - controlador (*tour*) - modelo (*tour*): Esta operación permite mostrar los recorridos creados.
- 4. Construcción del proceso para cargar imágenes de reconocimiento, crear recorridos sugeridos por el Editor y subir paquetes de imágenes de reconocimiento:**

Esta funcionalidad puede ser realizada por un usuario con perfil Administrador y Editor; consisten en enviar datos ingresados por el usuario al controlador a través de un formulario; podemos resumir su funcionamiento en el siguiente diagrama de secuencia:

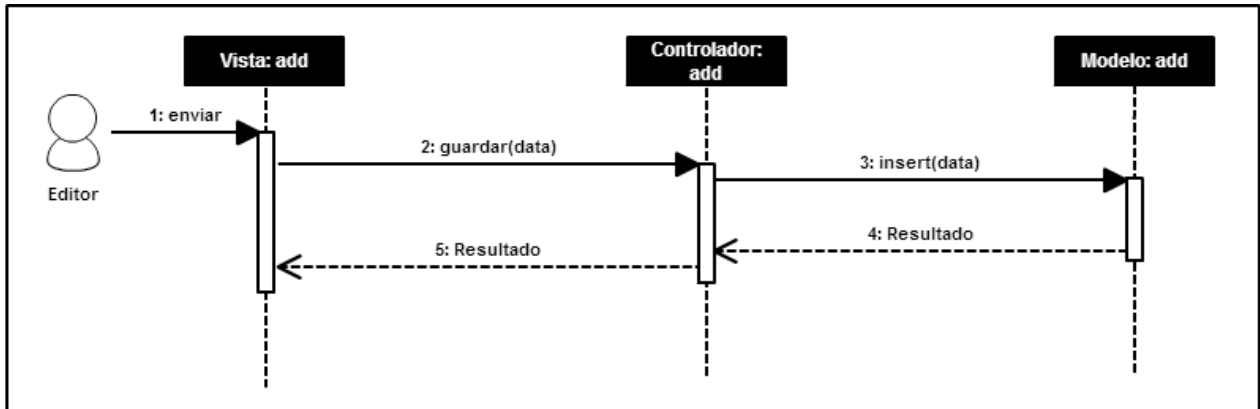


Figura N° 21 Diagrama de secuencia para cargar, crear y subir imágenes de reconocimiento, recorridos sugeridos por el Editor y paquetes de imágenes de reconocimiento.

Fuente: Creada por el Autor.

5. Funcionalidad para listar las imágenes de reconocimiento y los recorridos sugeridos por el Editor:

Esta funcionalidad puede ser realizada por un perfil Administrador y Editor; consisten en mostrar un listado en una tabla con diversos datos; podemos resumir su funcionamiento en el siguiente diagrama de secuencia:

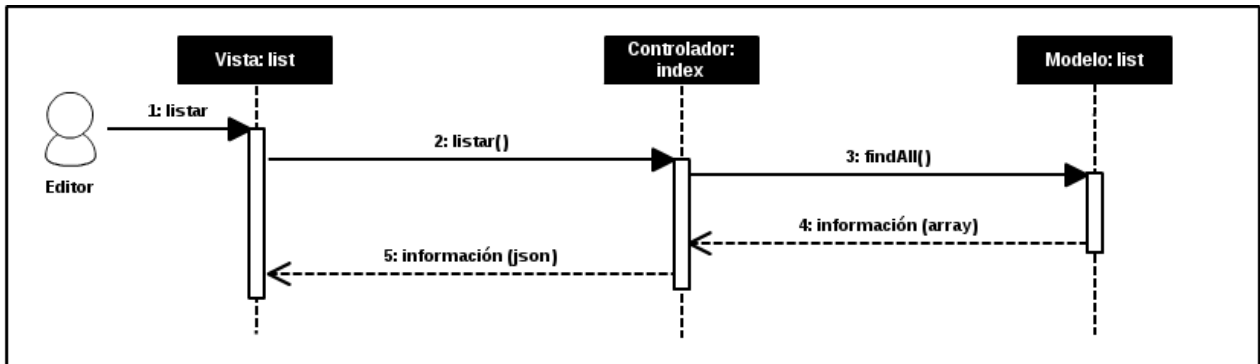


Figura N° 22 Diagrama de secuencia para listar las imágenes de reconocimiento y los recorridos sugeridos por el Editor.

Fuente: Creada por el Autor.

6. Funcionalidad para obtener los datos de las tablas administrativas:

Todas las tablas administrativas que se encuentren en la aplicación son alimentadas mediante un JSON (ver Anexo A parte -I-); el cual está compuesto por datos a mostrar en las tablas administrativas; el JSON permite ser accedido mediante el siguiente código asíncrono JavaScript:

```
"sAjaxSource": 'http://tesis.com/image/json',
```

Figura N° 23 Llamada Ajax para cargar datos en las tablas administrativas.

Fuente: Creada por el Autor.

El procesamiento que permite crear las columnas y filas a partir del JSON se realiza de manera automática desde el núcleo del *plugin DataTable*.

Podemos ver como interaccionan los distintos componentes en el siguiente diagrama de secuencia:

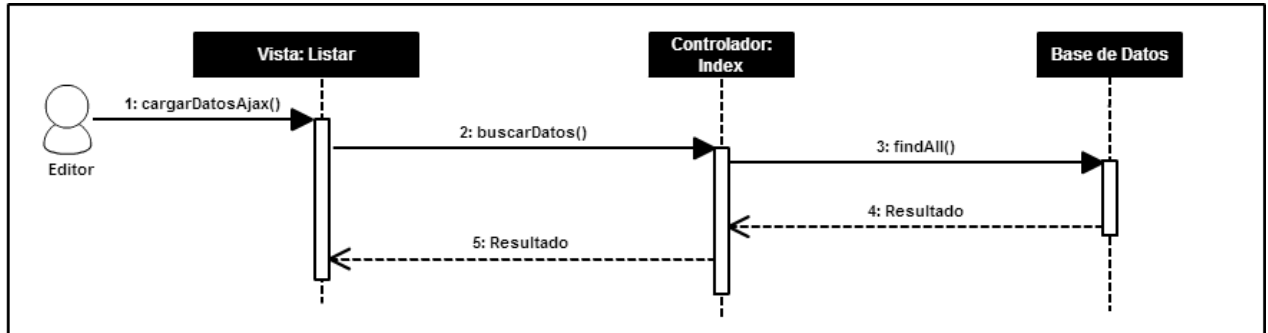


Figura N° 24 Diagrama de secuencia para la obtención de datos de las tablas administrativas.

Fuente: Creada por el Autor.

7. Construcción de la funcionalidad para actualizar los recorridos sugeridos por el Editor y modificar el estilo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada:

Esta funcionalidad puede ser realizada por un perfil Administrador; consisten en actualizar datos ya existentes en la aplicación por el usuario a través de un formulario; podemos resumir su funcionamiento en el siguiente diagrama de secuencia:

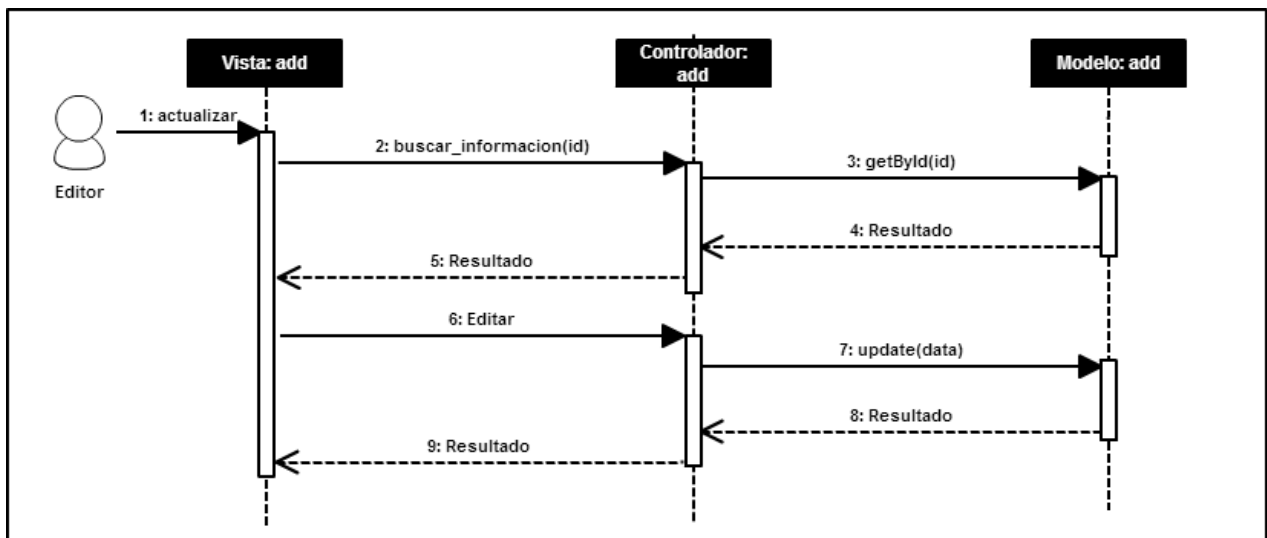


Figura N° 25 Diagrama de secuencia para actualizar y modificar los recorridos sugeridos por el Editor y del estilo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada.

Fuente: Creada por el Autor.

8. Funcionalidad para eliminar imágenes de reconocimiento y recorridos sugeridos por el Editor:

Esta funcionalidad puede ser realizada por un perfil Administrador; consisten básicamente en eliminar imágenes de reconocimiento y recorridos a través de un enlace; podemos resumir su funcionamiento en el siguiente diagrama de secuencia:

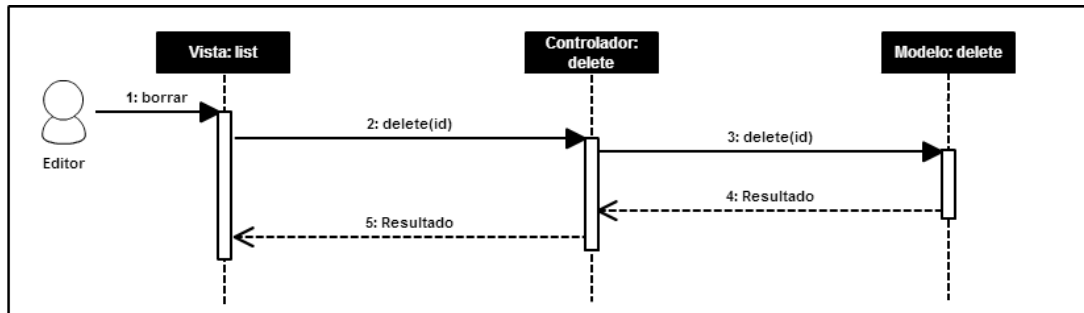


Figura N° 26 Diagrama de Secuencia para eliminar imágenes de reconocimiento y recorridos sugeridos por el Editor.

Fuente: Creada por el Autor.

9. Construcción de la funcionalidad para el recorte de imágenes:

Esta funcionalidad permite emplear la herramienta de recorte provista por otros programas de edición de imágenes como GIMP o Photoshop; consiste en seleccionar un área de interés de forma rectangular y generar una imagen a partir de la selección; es necesario definir una función que cargue el contenido de una imagen en un lienzo (*Canvas*; elemento usado para dibujar gráficos) para poder inicializar el mismo:

```

// carga la imagen al canvas
function loadImage() {
    // declaro la imagen y la almaceno en una variable
    newImg = new Image();
    newImg.src = imgSource;
    // muestro la imagen sólo si se ha terminado de cargar
    newImg.onload = function() {

        canvas.width = newImg.width;
        canvas.height = newImg.height;
        ctx.drawImage(this, 0, 0);
    };
}
    
```

Figura N° 27 Código para cargar una imagen en un lienzo.

Fuente: Creada por el Autor.

La “imagen” que edita el usuario, en realidad es un lienzo (*Canvas*) que permite interactuar con el usuario; luego de cargar la imagen en el lienzo, el siguiente paso consiste en definir una función que capture dos clics del usuario sobre el lienzo; los clics permiten determinar la imagen a generar; en la siguiente imagen podemos ver una sección del código de esta función:

```

// click sobre el canvas
$(canvas).click(function(e) {
    var offset = $(this).offset();

    auxX = e.clientX - offset.left + $("body").scrollLeft();
    auxY = e.clientY - offset.top + $("body").scrollTop();

    // primer click
    if (iniX < 0) {
        // primer punto
        iniX = auxX;
        iniY = auxY;
        // segundo click
    } else {
        // segundo punto
        endX = auxX;
        endY = auxY;

        // dimensiones de la nueva imagen
        imgW = endX - iniX;
        imgH = endY - iniY;

        // guardo el corte en una imagen
        saveCanvasToImage();
        // cargo de nuevo la imagen en el canvas
        loadImageToCanvas();
    }
});

```

Figura N° 28 Código para recortar imágenes.

Fuente: Creada por el Autor.

El cuerpo de la función manejada por el evento clic sobre el lienzo está dividida en dos bloques; la variable **iniX** es empleada como indicador para saber si se trata del primer clic del usuario sobre el lienzo (primer bloque encerrado por el condicional) o el segundo (segundo bloque encerrado por el condicional). Si el valor de la variable es menor a cero significa que es el primer clic del usuario sobre el lienzo; caso contrario, el clic corresponde al segundo clic del usuario sobre el lienzo.

El objetivo principal de la función es capturar la posición de los clics del usuario:

1. Son capturadas las coordenadas de los clics del usuario sobre la ventana del navegador con los eventos **e.clientX** y **e.clientY**.
2. El método **offset()** permite obtener las coordenadas **X** y **Y** relativas a un elemento.
3. El método **scrollTop()** y **scrollLeft()** permite obtener la posición de la barra de desplazamiento de un elemento; en este caso del cuerpo de la página web.
4. También es necesaria la definición de una función que permita cargar el contenido del lienzo a una imagen para poder guardar la imagen en la aplicación:

```
// guarda la imagen en el servidor
function saveImage() {

    var dataURL = canvasImg.toDataURL();

    $.ajax({
        type: 'POST',
        enctype: 'multipart/form-data',
        url: "<?php echo base_url() ?>image/upload_image_modify",
        data: {
            'image': dataURL,
            'id': "<?php echo $id ?>",
            success: function(response) {
                $('#target').loadingOverlay("remove");
                showNoticeToast(response);
            }
        }
    });
}
```

Figura N° 29 Código para guardar una imagen dibujada en un lienzo en una imagen.

Fuente: Creada por el Autor.

10. Construcción de la funcionalidad para el guardado, nombramiento, descarga y escalado de las imágenes de reconocimiento almacenadas en la aplicación:

Para registrar una imagen de reconocimiento en la aplicación:

1. Tiene que tener la extensión PNG o JPG y sus dimensiones no pueden exceder de 2000 píxeles de alto y/o largo.
2. La imagen es renombrada siguiente la siguiente convención: prefijo “final_image_” seguido del identificador numérico del objeto (objeto_id); debido a que puede haber múltiples imagen que representen un objeto, es necesario colocar luego del identificador el carácter “_” seguido de una numeración ascendente comenzando desde uno; por último es colocada la extensión de la imagen (PNG o JPG)⁴.
3. Al momento de descargar una imagen de la aplicación:
 - 3.1. La imagen es escalada a no más de 1000 píxeles por ancho o alto siguiendo las recomendaciones de la documentación oficial de Wikitude.
 - 3.2. La imagen es copiada y renombrada con el prefijo “image_” quitando el prefijo utilizado al momento de registrar la imagen anterior “final_image_”; el resto del nombre se conserva.

11. Construcción de la herramienta para aumentar el contraste:

⁴ Ejemplo: Identificador del objeto = 50

El nombre de la imagen es final_image_50_1.png

Esta funcionalidad permite aumentar el contraste de las imágenes de reconocimiento que se encuentren registradas en la aplicación; podemos ver como interaccionan los distintos componentes de la aplicación en el siguiente diagrama de secuencia:

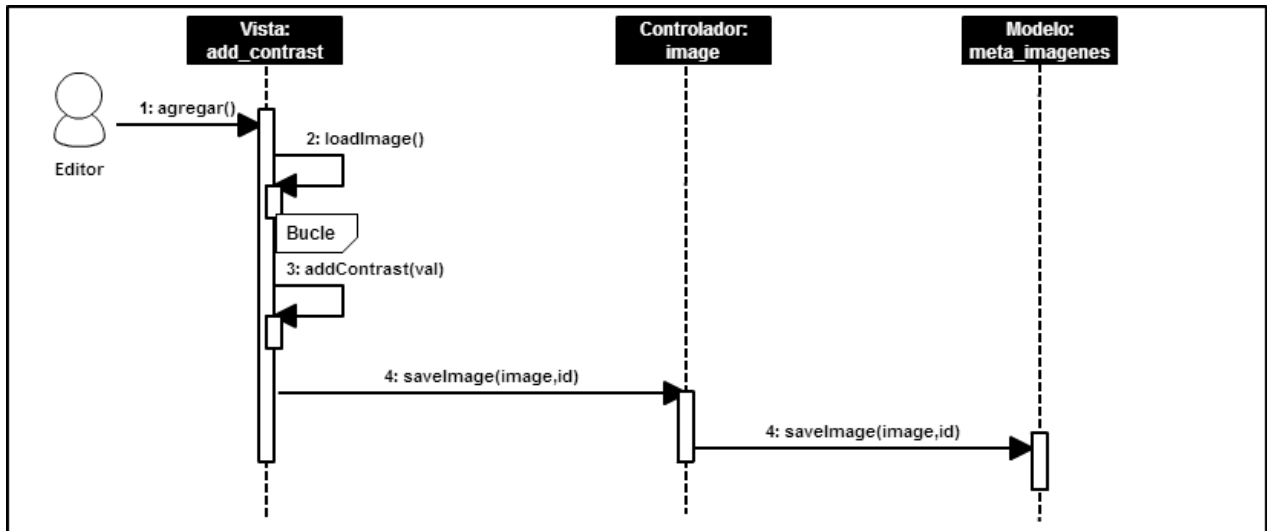


Figura N° 30 Diagrama de secuencia para la construcción de la herramienta para aumentar el contraste.

Fuente: Creada por el Autor.

En la figura N° 31 podemos ver la función que permite aumentar el contraste de las imágenes de reconocimiento; la función consiste en aplicar la fórmula del contraste (ver Anexo D parte -A-) a una imagen:

```

function Addcontrast(idCanvas, srcImg, val) {
    img = new Image();
    img.src = srcImg;
    canvas = document.getElementById(idCanvas);
    ctx = canvas.getContext('2d');
    //combinó la fórmula para obtener el contraste con el valor obtenido del elemento ranges
    var contrast = Math.tan(val * Math.PI / 180.0);

    img.onload = function() {
        ctx.drawImage(this, 0, 0);
        var imgd = ctx.getImageData(0, 0, canvas.width, canvas.height);
        var pix = imgd.data;
        for (var i = 0, n = pix.length; i < n; i += 4) {
            //incremento los valores rojo, verde y azul de cada pixel
            pix[i] = rangeColor(128 + (pix[i] - 128) * contrast);
            pix[i + 1] = rangeColor(128 + (pix[i + 1] - 128) * contrast);
            pix[i + 2] = rangeColor(128 + (pix[i + 2] - 128) * contrast);
        }
        ctx.putImageData(imgd, 0, 0);
    };
}
    
```

Figura N° 31 Código para aumentar el contraste de una imagen.

Fuente: Creada por el Autor.

3.6.3.2 Módulo de funcionalidades para dispositivo móviles inteligentes

A diferencia de la arquitectura del módulo de gestión del administrador en donde cada funcionalidad estaba acompañado de una vista y está a un controlador y este a un modelo, esto difiere de cómo es programado en el sistema operativo móvil Android, aunque también cumple con un patrón MVC el desarrollo para el sistema operativo móvil Android está basado en eventos;

así que cada acción que aplique el usuario sobre menús, PDIs, iconos y botones estos desencadenan en un evento que están contenidos dentro de un Controlador (actividad –*activity*-) o un Adaptator (*Adapter*).

1. Construcción de la herramienta para mostrar los PDIs a través de los mapas de Google

Esta herramienta permite visualizar en un mapa de Google los recorridos sugeridos por el Editor y los recorrido del Visitante; para poder utilizar la API de los mapas de Google en una aplicación para el sistema operativo móvil Android es necesario obtener una clave (*API Key*) accesible desde la página de Desarrolladores de Google, registrar el código de certificado de la aplicación configurable desde eclipse con el *plugin* ADT y registrar la clave dentro de la aplicación con el sistema operativo móvil Android; luego es posible incluir el código en la vista (*layout*) donde se mostrará el mapa de Google:

```
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android" >
```

Figura N° 32 Código para dibujar un mapa de Google en una vista.

Fuente: Creada por el Autor.

Ahora debemos hacer compatible el controlador con la API de los mapas de Google indicando al controlador que herede de:

```
public class ShowInMap extends android.support.v4.app.FragmentActivity
    implements AdapterView.OnItemClickListener {
```

Figura N° 33 Código para heredar de la API de los mapas de Google.

Fuente: Creada por el Autor.

El siguiente paso consiste en obtener los PDIs de las obras de artes; hay dos escenarios posibles:

1. Si el visitante está accediendo a los recorridos sugeridos por el Editor, los recorridos serán cargados desde Internet; específicamente desde la base de datos del portal “El Arte en la Ciudad Universitaria de Caracas”, utilizando el siguiente código asíncrono⁵:

⁵ Es necesario emplear un hilo (*thread*) u algún proceso asíncrono para no bloquear el proceso principal y así evitar interrumpir el ciclo de vida de una aplicación para Android y ocasionar una excepción en la misma.

```

// traigo de Internet los recorridos
Ion.with(this).load(URLEData.URL_GET_ALL TOURS.toString())
    .setJsonObjectBody(new JsonObject()).asJsonObject()
    .setCallback(new FutureCallback<JsonObject>() {
        @Override
        public void onCompleted(Exception e, JsonObject result) {}

        // creo un List para que sea mas manejable el JSON
        List<ReconidoSugerido> reconidoSugeridos = CastJsonToList
            .castJSONArrayToReconidoSugeridos(result.get(
                "aaData").getAsJSONArray());
    });
    });

```

Figura N° 34 Código para cargas recorridos desde la base de datos del portal.

Fuente: Creada por el Autor.

- Si el visitante está accediendo a los recorrido del visitante o recorridos creados por los visitantes; los recorridos se encuentran almacenados dentro de la propia aplicación; específicamente en la base de datos local SQLite (ver Anexo C parte -B-); por lo tanto no se requiere una conexión a Internet:

```
myTours = DbDao.getByParentId(idMyTour, ShowInMap.this);
```

Figura N° 35 Código para cargas recorridos desde SQLite.

Fuente: Creada por el Autor.

En ambos casos, la aplicación se conecta a una base de datos para obtener los recorridos; podemos ver como interaccionan los distintos componentes en el siguiente diagrama de secuencia:

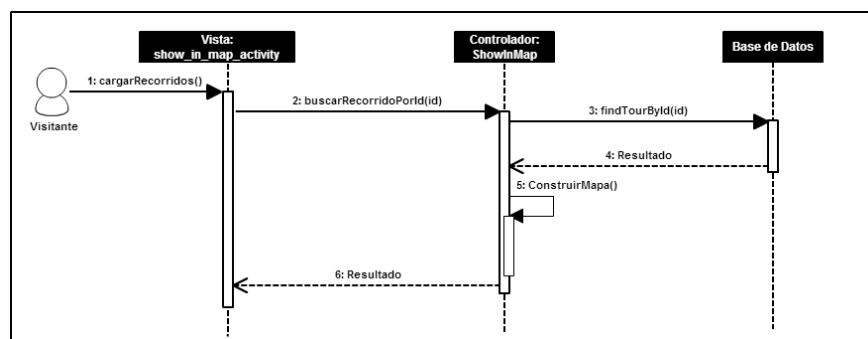


Figura N° 36 Diagrama de secuencia para conectarse a una base de datos.

Fuente: Creada por el Autor.

Para agregar en el mapa de Google los PDIs de las obras de arte obtenidas anteriormente, se emplea el siguiente código:

```

/*
 * Agrego un POI al mapa
 */
private void initPOI(double latitude, double longitude, String title,
String snippet, int id, int i) {

    // create marker
    Marker marker = googleMap.addMarker(new MarkerOptions()
        .position(new LatLng(latitude, longitude))
        .title(title)
        .icon(BitmapDescriptorFactory.fromResource(Information
            .getResourcePOIByNumber(i))));

    // agrego el marcador
    markerList.add(marker);
    markerId.add(id);
}

```

Figura N° 37 Código para agregar un marcador en un mapa de Google.

Fuente: Creada por el Autor.

Básicamente es creado un marcador (*marker*) con información de la obra de arte y es agregado a una lista para el control interno de los marcadores.

2. Construcción del sub-módulo de recorridos del Visitante:

Este sub-módulo está compuesto por siete funcionalidades las cuales permiten listar, crear, agregar, eliminar, compartir, cargar y ver los recorridos que hayan sido creados por el visitante dentro del módulo de funcionalidades para dispositivos móviles este sub-módulo es llamado con el nombre de “Mis recorridos”, debido a que el usuario puede crea sus propios recorridos agregándolos desde las fichas informativas de las obras de arte; fue necesario almacenar información clave sobre la obra de arte como su nombre, nombre del autor, año de creación, dimensiones, tipo, etc. Debido a que para visualizar los recorridos del Visitante en un mapa de Google y parte de la ficha informativa de las obras de arte (excluyendo la imagen) no requieren de una conexión a Internet; las siguientes operaciones están provistas en este sub-módulo:

- 2.1. **Crear recorrido:** (activity_create_my_tour) - controlador (CreateMyTourActivity) - modelo (*tour*): Esta funcionalidad permite crear un recorrido del Visitante y guardarla en una base de datos local de la aplicación; son solicitados un nombre y descripción para crear el recorrido.
- 2.2. **Agregar recorrido:** (tour_activity - ra_activity - show_in_map_activity) - controlador (TourActivity - RAActivity - ShowInMap) - modelo (*tour*): Para agregar una obra a un recorrido del Visitante es solo posible desde la ficha informativa de las obras de arte.
- 2.3. **Listar recorridos:** (my_tour_activity) - controlador (MyTourActivity) - modelo (*tour*): Permite visualizar todos los recorridos creados por el usuario con una breve descripción de los mismos.
- 2.4. **Eliminar vista:** (my_tours_activity - my_tour_activity) - controlador (MyToursActivity - MyTourActivity) - modelo (*tour*): Permite eliminar recorridos u obras de arte de recorridos.

- 2.5. **Compartir:** (my_tour_activity) - controlador (MyTourActivity) - modelo (*tour*): Al visualizar la ficha informativa de la obra es posible compartir recorridos del Visitante por distintos medios.
- 2.6. **Cargar recorrido:** (import_my_tour_activity) - controlador (ImportMyTourActivity) - modelo (*tour*): Permite cargar recorridos del Visitante en módulo de funcionalidades para dispositivos móviles.
- 2.7. **Ver recorrido:** (show_in_map_activity) - controlador (ShowInMap) - modelo (recorridos): Permite visualizar recorridos del Visitante en un mapa de Google.

3. Creación del listado para los recorrido del Visitante y las obras de arte de la Ciudad Universitaria de Caracas:

Para la construcción de los listados, se emplearon un tipo de interfaz llamado Adaptadores, los cuales son provistos por la API de Android y permiten mostrar datos a través de listas; el método constructor de la clase recibe la instancia de la clase y una colección de *ítems* para crear los listados y sus eventos asociados:

```
// Ahora rellenamos el ListView
ListAdapterListTours = new ListAdapterListTours(this, informationLists);
list.setAdapter(listAdapterListTours);
```

Figura N° 38 Firma de la función para crear un listado.

Fuente: Creada por el Autor.

4. Construcción del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada:

1. El primer paso a realizar es el de acceder al paquete de imágenes de reconocimiento para inicializar el Rastreador (*Tracker*) de Wikitude SDK el cual permitirá reconocer las obras de arte de la CUC:

```
// inicializamos el Tracker con el Target Collection
this.tracker = new AR.Tracker("http://desarrollolibre.net/tesis/target_collection/targetcollection.wtc", {onLoaded: this.worldLoaded});
```

Figura N° 39 Código para inicializar el Rastreador.

Fuente: Creada por el Autor.

2. Paso siguiente, consiste en emplear un evento llamado “*onEnterFieldOfVision*” que permite ejecutar una sección de código cuando ocurre el reconocimiento de imágenes; en este caso, invocar una función para generar la ficha informativa de la obra de arte:

```
World.updateStatusMessage('<p class="id">Obra: El Atleta (1951) - Francisco Narváez</p>'+
'<p class="id">Autor: Francisco Narváez</p>'+
'<p class="id">Año de creación: 1951</p>'+
'<p class="id">Ubicación: Edif. del Estadio Olímpico</p>'+
'<p class="id">Tipo: Escultura</p>'+
'<p class="id">Material: Piedra de Cumarebo</p>'+
'<p class="id">Dimensiones: 500 x 250 cm. Base 200 x 200 cm</p>');
```

Figura N° 40 Código para crear una ficha informativa.

Fuente: Creada por el Autor.

- Para poder utilizar la característica que permite ubicar un sitio de interés por su posición geográfica, es necesario tener las coordenadas geográficas del mismo; en este caso, de las obras de arte, esta información ya se encuentra disponible en la base de datos del portal y es accesible mediante el siguiente código asíncrono:

```
// cargo los pois segun la posicion
$.getJSON(serverUrl, function(data) {
    World.loadPoisFromJsonData(data['aaData']);
})
.error(function(err) {
    World.updateStatusMessage("Ruta invalida para obtener los objetos", true);
    World.isRequestingData = false;
})
.complete(function() {
    World.isRequestingData = false;
});|
```

Figura N° 41 Código para cargar las obras de arte según la posición.

Fuente: Creada por el Autor.

- Luego se crean tantos objetos *Marker* como sitios se deseen ubicar:

```
World.markerList.push(new Marker(singlePoi));
```

Figura N° 42 Código para crear un marcador.

Fuente: Creada por el Autor.

- Al momento de reconocer una obra de arte o presionar un marcador, se activa un evento que permite mostrar la ficha informativa, obras del mismo tipo y otras obras del autor, estas dos últimas, son obtenidas mediante un código asíncrono.

Este proceso es autogenerado desde el módulo de gestión del administrador según las imágenes de reconocimiento y obras de arte que se encuentren almacenadas en la aplicación; al encontrarse ubicado en un Servidor Web permite ser actualizado constantemente cada vez que se modifiquen estos datos y ser consumido mediante un enlace por un dispositivo con el sistema operativo móvil Android; de esta forma se logra reflejar los cambios en tiempo real sin la necesidad de volver a generar la **apk** (instalador de una aplicación para el sistema operativo móvil Android).

3.6.4 Fase de Transición

En esta fase se colocó una versión beta de la aplicación en un servidor privado y temporal que cumplía con las características de un Servidor Web con el objetivo de introducir el producto a la comunidad de usuario y facilitar el ciclo de pruebas.

3.6.4.1 Demostraciones, entrevistas y encuesta de aceptación a los usuarios finales

Se empleó un servicio que permite la creación y distribución mediante la Web de formularios para la creación de encuestas en el ciclo de pruebas con los usuarios, el servicio es conocido como Google Formulario o *Google Form* por sus siglas en inglés; se hicieron Pruebas de Usabilidad con el propósito de conocer diversas opiniones en cuanto a la usabilidad, navegabilidad, utilidad y satisfacción de la aplicación que se encuentra compuesta de dos módulos explicados en el punto 3.7.2.3:

1. Para el módulo de funcionalidades para dispositivos móviles los tipos de pruebas realizadas fueron:
 - 1.1. **Demostraciones al usuario:** Permite realizar pruebas “en vivo” con usuarios potenciales con la idea de mostrar el funcionamiento, pantallas y funcionalidades. Se realizaron demostraciones a la Profa. Mariantonia Palacios y la Profa. Yusneyi Carballo Barrera.
 - 1.2. **Entrevistas al usuario:** Según Álvarez Juan. (2003, p.109) “una entrevista es una conversación que tiene una estructura y un propósito”, además dice que “la entrevista busca entender el mundo desde la perspectiva del entrevistado, y desmenuzar los significados de sus experiencias”. Las entrevistas a los usuarios estuvieron compuesta por un grupo de ocho usuarios tanto expertos como inexpertos en el área de computación pero todos usuarios del sistema operativo Android o IOS se les pidió que instalaran la aplicación en su dispositivo móvil inteligente con el sistema operativo móvil Android (se les facilitó un dispositivo con el sistema operativo móvil Android en caso de que no tuviesen uno) y que dieran sus impresiones sobre la aplicación a medida que utilizaban la misma, además se les pidió que contestaran parte de las preguntas de la encuesta de manera oral y de esta forma poder afinar los resultados obtenidos.
 - 1.3. **En base a encuestas (ver Anexo E):** Permiten recopilar datos mediante cuestionarios o formularios previamente diseñados. Se realizaron pruebas en base a encuestas a un total de seis usuarios pertenecientes al área de computación (ver Anexo F), se les envió por Internet el enlace de la encuesta y el enlace para poder instalar la aplicación para el sistema operativo móvil Android alojada en la Tienda de Aplicaciones de Google (Google Play).

Analizadas las consideraciones, comentarios y respuestas, es posible indicar las modificaciones más importantes que se aplicaron en el módulo de funcionalidades para dispositivos móviles:

1. Distribución a los laterales los botones para el despliegue de los paneles de otras obra del autor y otras obras relacionadas en el sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada.
 2. Incorporación de un botón para ocultar los paneles de otras obra del autor y otras obras del mismo tipo en el sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada.
 3. Eliminación del botón “Deshabilitar imágenes” del menú principal para ciertas pantallas.
 4. Mejoras en la redacción y la inclusión de títulos descriptivos en algunas pantallas.
 5. Posibilidad de reorganizar los recorridos del Visitante entre otras características para mejorar la usabilidad de la aplicación.
 6. Agrupación de las preguntas en la sección de ayuda.
 7. Incorporación de una sección llamada “Acerca de” que muestra datos sobre la aplicación, los distintos grupos involucrados, desarrollador y derechos de autor.
2. Para el módulo de gestión del administrador, se realizaron demostraciones a la Profa. Mariantonia Palacios y la Profa. Yusneyi Carballo Barrera que son consideradas usuarias de tipo Administrador en este trabajo.

El módulo de funcionalidades para dispositivos móviles tuvo buena aceptación en los usuarios finales (ver Anexo F), cumple con las características deseadas en una aplicación móvil: sencillez y efectividad, obtuvo buenas ponderaciones y críticas en las respuestas a las preguntas formuladas en las encuestas y entrevistas en los distintos campos como los colores seleccionados para la interfaz de la aplicación, navegabilidad y utilidad de ciertos sub-módulos como visualizar recorridos en el mapa de Google, la sección de ayuda, entre otros.

3.6.4.2 Dispositivos utilizados por los usuarios en las entrevistas y encuestas para probar el módulo de funcionalidades para dispositivos móviles

A continuación se muestra una recopilación de los distintos dispositivos empleados por los usuarios para instalar el módulo de funcionalidades para dispositivos móviles según datos obtenidos en la encuesta y en la entrevista con los usuarios, podemos nombrar los siguientes dispositivos:

- Asus Transformer TF300
- Asus ZenFone 5
- HTC One S
- HTC One X
- Motorola Moto G
- Samsung Galaxy Note 3
- Samsung Galaxy S3 mini
- Samsung Galaxy Tab 2 7”
- Sony Ericsson Xperia Arc
- Sony Ericsson Xperia Z1

3.6.4.3 Diagrama de Despliegue de la aplicación

A continuación se observa el diagrama que muestra el despliegue de la aplicación:

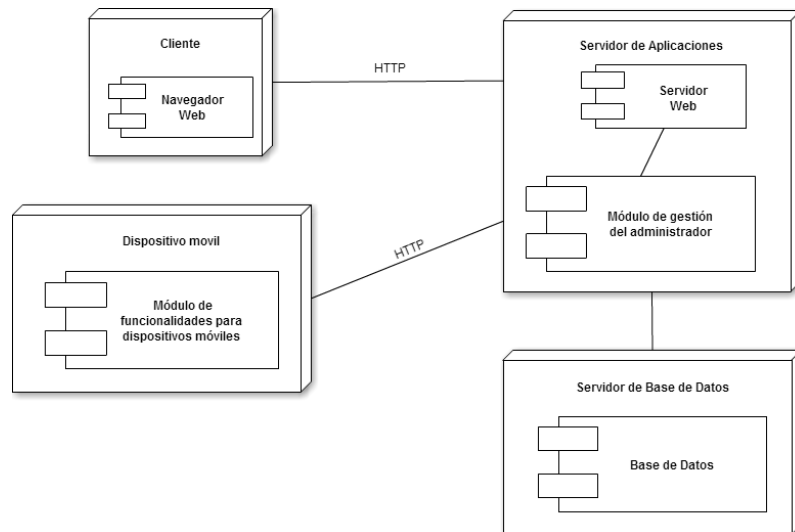


Figura N° 43 Diagrama de Despliegue de la aplicación.

Fuente: Creada por el Autor.

3.6.5 Principales interfaces de la aplicación

En esta sección se mostrará dos tipos de interfaces que varían según al módulo que correspondan; el primer tipo de interfaces corresponden al módulo de funcionalidades para dispositivos móviles; el segundo tipo pertenecen al módulo de gestión del administrador.

3.6.5.1 Principales interfaces del módulo de funcionalidades para dispositivos móviles

A continuación se muestra la interfaz principal del módulo de funcionalidades para dispositivos móviles ejecutándose en un dispositivo con el sistema operativo móvil Android:

Esta es la pantalla inicial del módulo, configurada para que se oculte pasado los cinco segundos:

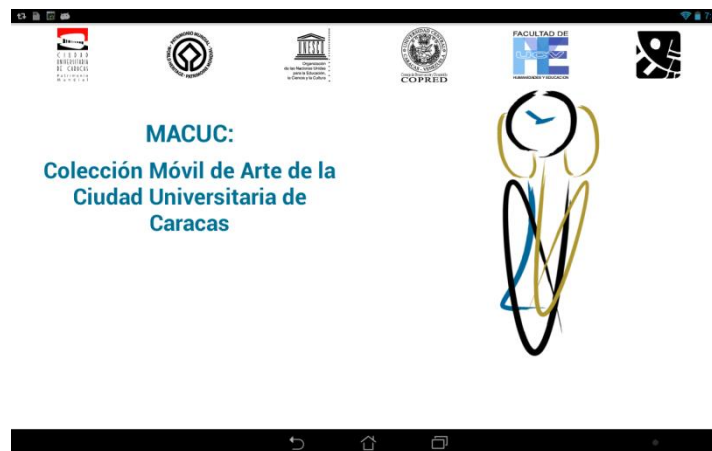


Figura N° 44 Interfaz de bienvenida a la aplicación para el sistema operativo móvil Android.

Fuente: Creada por el Autor.

Esta es la pantalla principal del módulo:

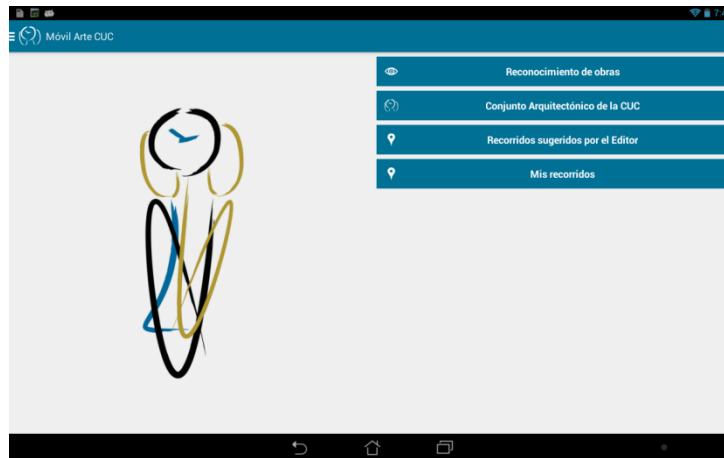


Figura N° 45 Interfaz principal de la aplicación para el sistema operativo móvil Android.

Fuente: Creada por el Autor.

En la figura N° 46 se muestra la interfaz para crear o actualizar los recorridos del Visitante; los cuales consisten en un nombre y una descripción:

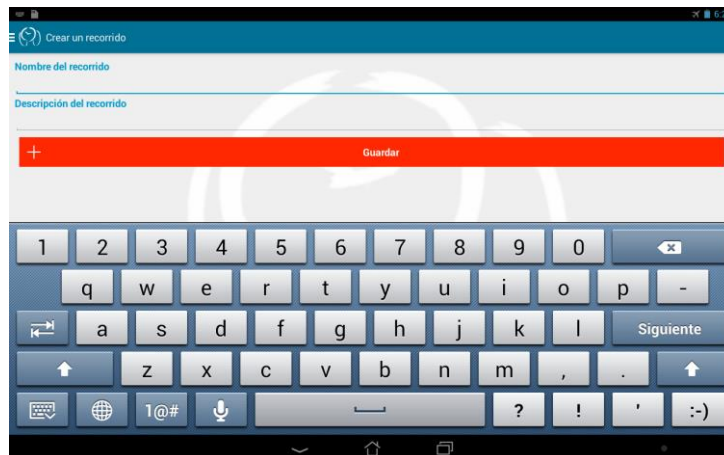


Figura N° 46 Interfaz para la creación de un recorrido del Visitante.

Fuente: Creada por el Autor.

En la figura N° 47 se muestra la clasificación del conjunto arquitectónico de la CUC:



Figura N° 47 Interfaz para el conjunto arquitectónico CUC.

Fuente: Creada por el Autor.

En la figura N° 48 se muestra la interfaz del listado de las Síntesis de las Artes Mayores pertenecientes al conjunto arquitectónico de la CUC; al seleccionar uno de los elementos del listado, se desplegará la ficha informativa de la obra:

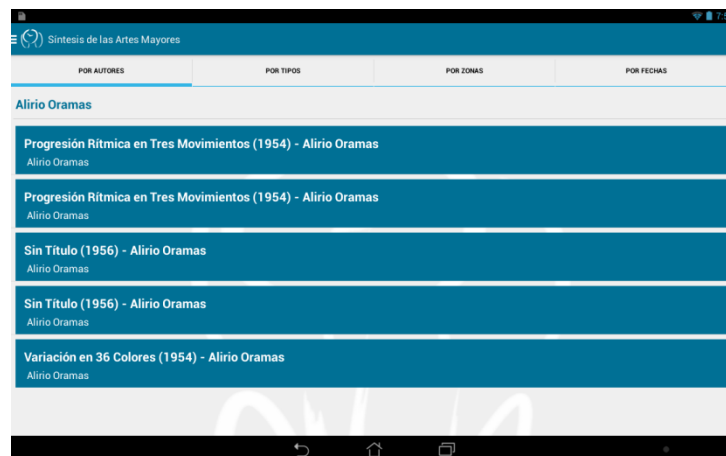


Figura N° 48 Interfaz para el listado de las Síntesis de las Artes Mayores.

Fuente: Creada por el Autor.

En la figura N° 49 se muestra la ficha informativa de una obra de arte:

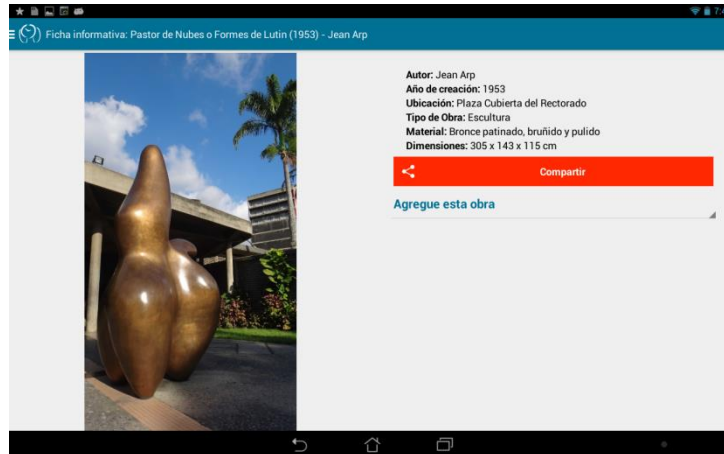


Figura N° 49 Interfaz de la ficha informativa en el sistema operativo móvil Android.

Fuente: Creada por el Autor.

En la figura N° 50 se muestra un ejemplo de un recorrido en un mapa de Google:

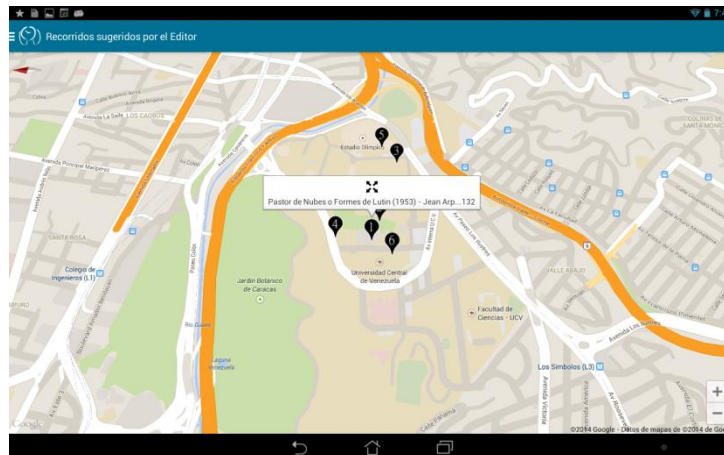


Figura N° 50 Interfaz para mostrar un recorrido en el sistema operativo móvil Android.

Fuente: Creada por el Autor.

En la figura N° 51 y la figura 52 se muestra un ejemplo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada:

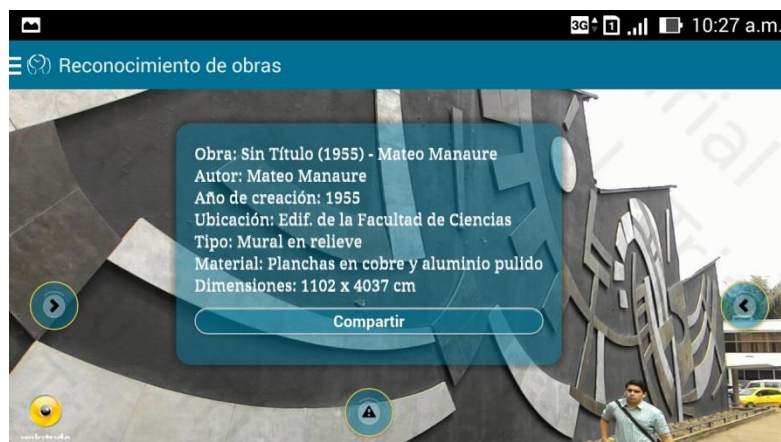


Figura N° 51 Interfaz del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada (panel de obras relacionadas).

Fuente: Creada por el Autor.

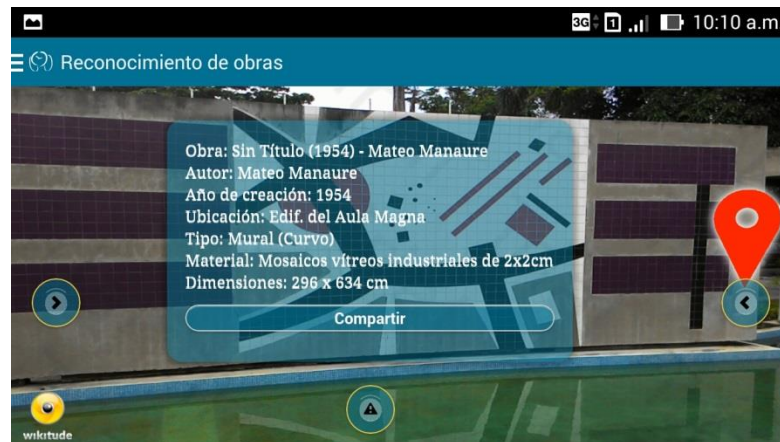


Figura N° 52 Interfaz del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada (ficha informativa).

Fuente: Creada por el Autor.

3.6.5.2 Principales interfaces del módulo de gestión del administrador

En todas las interfaces mostradas (salvo la interfaz de autenticación) se encuentra un menú lateral en el cual puede acceder a todas las funcionalidades del módulo de gestión del administrador:

- Inicio.
- Gestión de imágenes de reconocimiento.
- Gestión de los recorridos sugeridos por el Editor.
- Modificación del estilo del sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada.
- Actualización del paquete de imágenes de reconocimiento.
- Copias de seguridad.
- Ayuda.
- Cerrar sección.

En la figura N° 53 se muestra la interfaz para la autenticación del usuario:

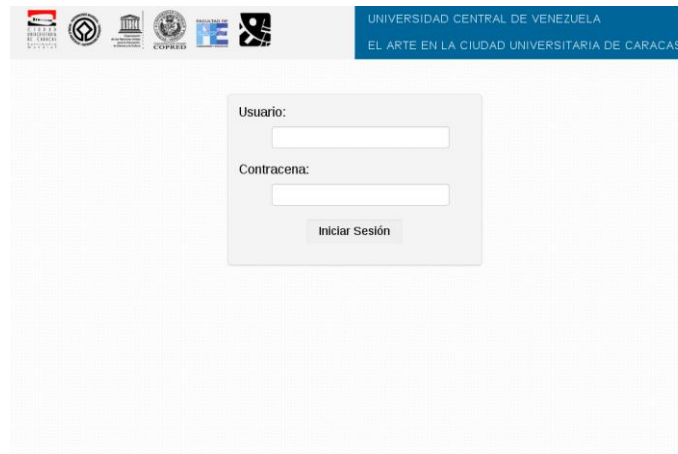


Figura N° 53 Interfaz de autenticación del módulo de gestión del administrador.

Fuente: Creada por el Autor.

En la figura N° 54 se muestra la interfaz para listar los recorridos sugeridos por el Editor que se han creado en la aplicación; en esta sección es posible acceder a las herramientas que permiten gestionar los recorridos sugeridos por el Editor luego de que son creados:



Figura N° 54 Interfaz del listado de los recorridos sugeridos por el Editor.

Fuente: Creada por el Autor.

En la figura N° 55 se muestra la interfaz para aplicar la operación de recorte a una imagen almacenada en la aplicación:



Figura N° 55 Interfaz para hacer un recorte en una imagen.

Fuente: Creada por el Autor.

En la figura N° 56 se muestra la interfaz para crear y/o actualizar los recorridos sugeridos por el Editor:

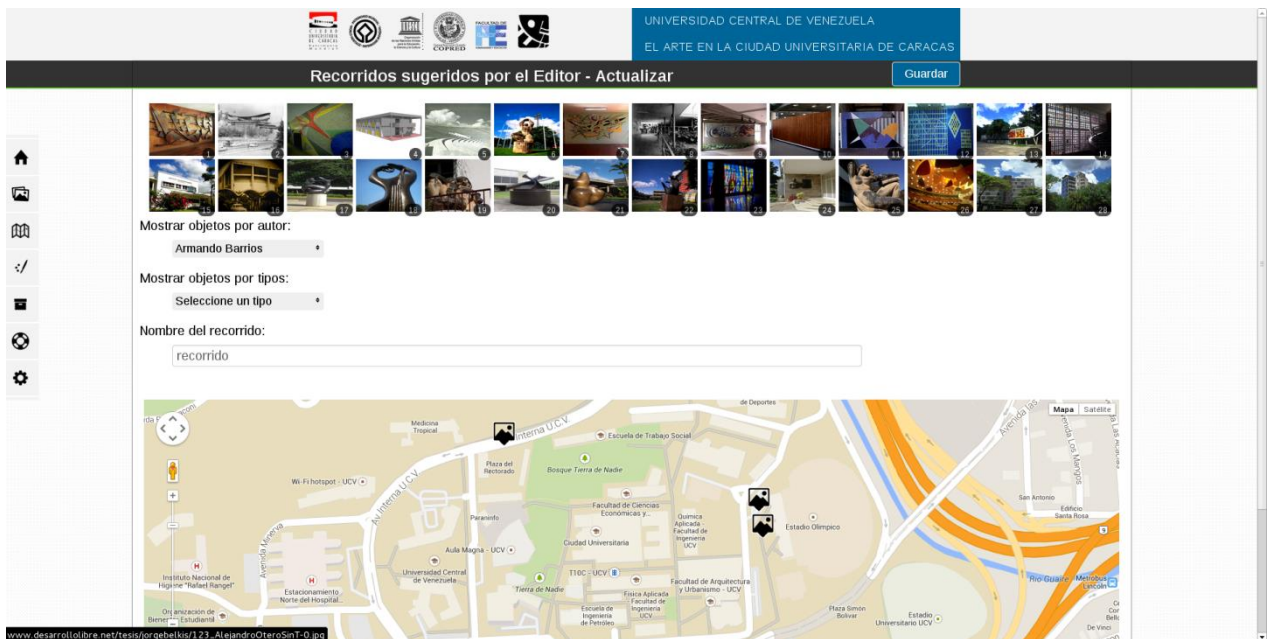


Figura N° 56 Interfaz para crear o actualizar un recorrido sugerido por el Editor.

Fuente: Creada por el Autor.

3.6.5.3 Manual del usuario Administrador

El manual de usuario del Administrador contiene el detalle del módulo de gestión del administrador así como la explicación en detalle de cómo utilizar este módulo; ver Anexo G.

3.6.5.4 Resultados

Se logró cumplir con el objetivo general de este trabajo que consistía en construir una aplicación que permita visualizar, divulgar información, además de realizar recorridos a través de mapas virtuales por las obras de arte, ver listados del conjunto arquitectónico de la CUC y ofrecer una completa administración de la aplicación.

De esta forma se logró cumplir con todos los requerimientos expuestos en este trabajo; de tal forma podemos decir que este trabajo permite:

1. Una administración completa de los recorridos creados por los Editores y de las imágenes de reconocimiento.
2. Visualizar, crear, compartir, cargar y realizar recorridos en el módulo de funcionalidades para dispositivos móviles a través de un mapa de Google.
3. Obtener las fichas informativas de las obras de arte a través de listados.
4. Visualizar imágenes de las obras de arte.
5. Visualizar las fichas informativas, obras relacionadas y otras obras del autor a través del proceso de reconocimiento de imágenes, posicionamiento geográfico y realidad aumentada.
6. Ofrecer un mapa del sitio.

La aplicación cumple con los siguientes requisitos no funcionales:

1. **Fiabilidad:** Fue desarrollada bajo tecnologías probadas y en constante actualización en el tiempo.
2. **Disponibilidad:** Se encuentra disponible en un Servidor Web accesible desde cualquier dispositivo con acceso a Internet las 24 horas del día los 365 días del año.
3. **Seguridad:** Para acceder al módulo de gestión del administrador es necesario utilizar una contraseña que se encuentra cifrada; además la aplicación en toda su envergadura utiliza *framework*, librerías de desarrollo de software y APIs.
4. **Escalabilidad:** Se desarrolló bajo un esquema modular que permite acoplar nuevos módulos sin que esto afecte al resto de los ya desarrollados.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Al concluir la realización de este Trabajo Especial de Grado, se presenta la construcción de un módulo para dispositivos móviles denominado “MACUC” por sus siglas de Colección Móvil de Arte de la Ciudad Universitaria de Caracas y un módulo para la administración denominado “Módulo del administrador MACUC” que en conjunto cumplen con los requerimientos planteados al principio de este trabajo que permitieron llevar una tecnología como la realidad aumentada a los teléfonos y tabletas; lo cual permite a los usuarios obtener información sobre las obras expuestas en la Ciudad Universitaria de Caracas, específicamente, la colección "Síntesis de las Artes Mayores". Al colocar al alcance del público en general una aplicación de estas características se logrará un cambio en la interacción obra-usuario. En la mayoría de los museos o colecciones de arte solo se encuentran las esculturas u obras con una breve descripción de cada uno de estos elementos y ningún otro recurso que ayude a mejorar su comprensión. La idea de este trabajo es lograr que los usuarios interactúen o visualicen las obras de arte desde una perspectiva distinta brindada a través de dispositivos muy utilizados en la actualidad como lo son los teléfonos inteligentes y las tabletas. Con solo enfocar la cámara sobre las obras de arte se despliega al instante un conjunto de información; además el módulo permite ver, crear y compartir recorridos a través de mapas virtuales.

Las tecnologías, técnicas y patrones con las que se llevó a cabo el desarrollo del trabajo son de gran importancia; para que la experiencia final del usuario sea satisfactoria al momento de interactuar con la aplicación y con las obras de arte es necesario que sea intuitiva; con que el usuario posicione la cámara del dispositivo sobre las obras de arte hay un despliegue de información y funcionalidades asociadas; es de vital importancia de que sea correctamente gestionada por el administrador; si las tecnologías a escoger y la lógica empleada, dan como resultado que su administración sea un proceso complejo, la aplicación no se actualizará, y esto, para una aplicación de este tipo, es lo mismo que desaparecer en un corto periodo de tiempo. Es por ello la elección de bibliotecas de desarrollo de software y herramientas de desarrollo de software para el desarrollo de la aplicación, debe permitir evitar a los programadores detalles de bajo nivel, para que se centren principalmente en identificar y desarrollar los requerimientos de software.

Después de haber desarrollado el Trabajo Especial de Grado con la metodología OpenUp, se pudo constatar que facilitó el proceso en la toma de requerimientos, definición de la aplicación en su todas sus fases, organización, diseño y desarrollo; todo esto trajo como resultado una buena planificación que se ve reflejado en los cortos tiempos de desarrollo los cuales se pueden medir en su mayoría en pocos días de desarrollo; los que superan la semana de duración, se deben a que son funcionalidades que requieren un alto grado de investigación y pruebas.

Los objetivos de este Trabajo Especial de Grado se ha cumplido satisfactoriamente, el cual consiste en el desarrollo de una aplicación que permita visualizar, divulgar información, además de realizar recorridos a través de mapas virtuales por las obras de arte, ver listados del conjunto

arquitectónico de la CUC y ofrecer una completa administración de la aplicación; se cumplieron con todos los requerimientos planteados en este trabajo y la creación de una aplicación para el conocimiento.

Para la elaboración de la aplicación se pasaron por distintas fases definidas y desarrolladas en el capítulo dos y tres respectivamente; la fase de Construcción se caracteriza por ser la que mayor tiempo consumido en lo que a su elaboración se refiere, una de las iteraciones o micro-incrementos que consiste en desarrollar las funcionalidades para la realidad aumentada, reconocimiento de imágenes y posicionamiento geográfico de las obras de arte se convirtió en un verdadero reto debido a lo poco usual en su lógica y la complejidad en probar su funcionamiento, aun así se logró culminarlo en un tiempo acorde a su nivel de dificultad y con todas las funcionalidades especificadas en este trabajo incluidas; culminada la fase de Construcción, la siguiente y última de las fases llamada Transición, consistió en introducir la aplicación en la comunidad de usuarios y realizar pruebas con los mismos a través de distintos recursos como entrevistas a los usuarios y en base a encuestas vía web.

Dicho todo esto y en el ámbito personal considero la utilización de este Trabajo Especial de Grado como modelo para aplicarlo en diferentes contextos académicos debido a la potencialidad, utilidad e innovación del mismo.

Según lo afirmado en los párrafos anteriores y en base a los buenos resultados obtenidos en las encuestas, podemos decir que se cumplieron los objetivos planteados al comienzo de este documento, y atribuimos en gran parte el logro de estos a la planificación y organización de requerimientos y actividades necesarias para la creación de la aplicación.

Recomendaciones y Trabajos Futuros

Como recomendaciones para trabajos futuros, podemos mencionar el concepto de una opción híbrida para el sub-módulo de funcionalidades para dispositivos móviles; en otras palabras poder alojar una versión del paquete de imágenes de reconocimiento en el dispositivo junto con la información de las obras de arte; de esta manera poder reconocer obras de artes sin la necesidad de estar conectados a Internet; el paquete de imágenes de reconocimiento podría ser actualizado cada vez que el dispositivo móvil inteligente sea conectada a Internet con previa aprobación del usuario.

También se podría considerar la creación de las aplicaciones para los sistemas operativos móviles IOS y Blackberry 10, Wikitude SDK permite desarrollar para estas plataformas, dando de esta manera la posibilidad de llegar a una mayor cantidad de usuarios; resulta sencillo consumir la capa de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada (esta capa es llamada en la Documentación Oficial de Wikitude como *Architect World*) por otros sistemas operativos móviles; al tratarse de tecnologías no nativas (tecnologías Web) o que no forman parte de la API de desarrollo de cada sistema operativo móvil se puede abstraer de esta capa y solo desarrollar la capa de compatibilidad que representa el código nativo de la aplicación.

Se podrían agregar otras características al módulo de funcionalidades para dispositivos móviles como:

- Acercamientos y alejamientos en las imágenes.
- Reordenamiento de los recorridos del Visitante mediante la técnica de Arrastrar y Soltar.

- Pase entre las ficha informativa de las obras de arte con un solo toque en la pantalla.
- Filtros y rangos de distancia para la carga de los PDIs para el sub-módulo de reconocimiento de imágenes y posicionamiento geográfico con realidad aumentada.
- Agregar la ubicación de la posición del usuario Visitante en los mapas.

Referencias bibliográficas.

Arias Fidiás. (1999). Proyecto de Investigación. *Introducción a la Metodología Científica*. Caracas, Venezuela, Episteme.

Alvarez, Juan L; Jurgenson, Gayou. (2003) *Cómo Hacer Investigación Cualitativa*. Fundamentos y Metodología. Editorial Paidós Educador. México- D. F

Gómez, Marcelo M (2006). *Introducción a la Metodología de la Investigación*. Primera Edición. Córdoba.

Hernández Sampieri, Roberto. (1991). *Metodología de la Investigación*. Universidad de Montreal, Canadá McGraw-Hill.

Otros documentos digitales

Báez, Manuel. *Introducción a Android*. Universidad Complutense de Madrid.

Barrios Dueñas, Joel. (2013). *Configuración De Servidores Con GNU/Linux*. México D.F.

Colléll Puig, Jordi. *CSS3 y JavaScript Avanzado*. España Universidad Abierta de Cataluña.

Mateu, Carles. (2004). *Desarrollo de Aplicaciones Web*. España Universidad Abierta de Cataluña.

Murphey, Rebecca. (2010). *jQuery Fundamentals*. Estados Unidos de América.

Oliver, Salvador Gómez. (2011). *Manual-Programacion-Android-v2*. España.

Pérez, Javier Eguíluz. (2009). *Introducción a JavaScript*. España.

Referencias de Internet

Akquinet (s.f.). *jquery-toastmessage-plugin*. Recuperado el 12 de mayo de 2014, de <http://akquinet.github.io/jquery-toastmessage-plugin/widget>

Alejandro Sacristán (27 de septiembre de 2011). *Realidad Aumentada: una convergencia de lo real y lo virtual para un mundo mejor*. Recuperado el 08 de enero de 2014, de <http://www.lavozdelaciencia.com/general/realidad-aumentada-una-convergencia-de-lo-real-y-lo-virtual-para-un-mundo-mejor.html>

Ana Baeza (21 de noviembre de 2013). *Apps sobre museos*. Recuperado el 1 de junio de 2014, de <http://www.biblogtecarios.es/anabaeza/apps-sobre-museos>

Aplicaciones (s.f.). *Second Canvas Museo del Prado*. Recuperado el 2 de junio de 2014, de <http://www.aplicaciones.org/app/second-canvas-museo-del-prado>.

Aurelio Ceballos (18 de marzo de 2013). *Fases para el desarrollo de sistemas de realidad aumentada*. Recuperado el 9 de agosto de 2013, de <http://informatic2you.wordpress.com/tag/aumentada>

Daniel Wagner (s.f.). *First Steps Towards Handheld Augmented Reality*. Recuperado el 11 de septiembre de 2014, de <https://www.ims.tuwien.ac.at/publications/tuw-138074.pdf>

DesarrolloWeb (06 de diciembre de 2011). *Introducción a jQuery Mobile*. Recuperado el 24 de agosto de 2014, de <http://www.desarrolloweb.com/articulos/introduccion-jquery-mobile.html>

Desarrollador Wikitude (s.f.). *Documentación Oficial Wikitude SDK*. Recuperado el 30 de agosto de 2013, de <http://developer.wikitude.com>

Eclipse (Agosto, 2007). *Información sobre la Metodología de Software Openup*. Recuperado el 24 de agosto de 2013, de <http://www.eclipse.org/epf/general/OpenUP.pdf>

Eric Hynds (s.f.). *jQuery UI MultiSelect Widget*. Recuperado el 25 de abril de 2014, de <http://www.erichynds.com/blog/jquery-ui-multiselect-widget>

Héctor Ruiz Cabeza (25 de noviembre de 2010) *Visión artificial aplicada a la industria*. Recuperado el 8 de enero de 2014, de http://www.jcee.upc.edu/JCEE2010/pdf_ponencias/PDFs/25_11_10/INFAIMON-Vision%20artificial.pdf.

FancyBox (s.f.). *Documentación*. Recuperado el 25 de abril de 2014, de <http://fancyapps.com/fancybox/>

Gimp (s.f.). *Gimp*. Recuperado el 12 de mayo de 2014, de <http://gimp.es>

Gimp (s.f.). *Gimp*. Recuperado el 24 de agosto de 2014, de <http://docs.gimp.org/es/gimp-tool-crop.html>

Google Play (s.f.). *MoMA*. Recuperado el 2 de junio de 2014, de <https://play.google.com/store/apps/details?id=org.moma.android>

Google Play (s.f.). *Rijksmuseum*. Recuperado el 4 de junio de 2014, de <https://play.google.com/store/apps/details?id=nl.rijksmuseum.mmt>

Google Play (s.f.). *Musée du Louvre-Lens*. Recuperado el 4 de junio de 2014, de <https://play.google.com/store/apps/details?id=com.orange.itlabs.louvrelens>.

Hassan Montero, Yusef Martín Fernández, Francisco J (14 de Julio de 2003). *Qué es la Accesibilidad Web*. Recuperado el 19 de agosto de 2014, de <http://www.nosolousabilidad.com/articulos/accesibilidad.htm>

jQuery Mobile (s.f.). *About*. Recuperado el 24 de agosto de 2014, de <http://jquerymobile.com/about/>

JSON (s.f.). *Introducing JSON*. Recuperado el 28 de junio de 2014, de <http://json.org/>

Loading Overlay (s.f.). *Documentación*. Recuperado el 28 de junio de 2014, de <http://jgerigmeyer.github.io/jquery-loading-overlay/>

Santiago Ríos Salgado, Cecilia Hinojosa Raza & Ramiro Delgado Rodríguez (s.f.). *Metodología de Software Openup*. Recuperado el 09 de agosto de 2013, de <http://repositorio.espe.edu.ec/bitstream/21000/6316/1/AC-SISTEMAS-ESPE-047042.pdf>

Ricardo Arce (21 de noviembre de 2009). *Evaluación de la Usabilidad de un sitio web*. Recuperado el 18 de agosto de 2014, de <http://es.slideshare.net/ricardoarce/evaluacin-de-la-usabilidad-de-un-sitio-web>

Rosson & Carroll (2002). *Evaluación de la usabilidad*. Recuperado el 18 de agosto de 2014, de <http://evaluausabilidad.sprinterweb.net/>

SQLite (s.f.). Latino America. *SQLite: Rápido, ágil, liviano y robusto*. Recuperado el 15 de mayo de 2014, de <http://sqlite-latino.blogspot.com/>

Stalin Tuza (4 de mayo de 2012). *Metodología de Software Openup*. Recuperado el 24 de agosto de 2013, de <http://www.slideshare.net/StalinTuza/ingenieriasoftwareopenup>

Santa Marta (15 de marzo de 2010). *Metodología de Software Openup*. Recuperado el 24 de agosto de 2013, de <http://www.slideshare.net/samith/metodologia-upen-up-3439131>

Statcounter (s.f.). *Estadísticas Globales*. Recuperado el 14 de septiembre de 2014, de <http://gs.statcounter.com>

Peter Norvig (30 de noviembre de 2011). *La tecnología de reconocimiento de imágenes de Google explicada*. Recuperado el 9 de agosto de 2013, de http://tecnologiayproductosgoogle.blogspot.com/2011/11/la-tecnologia-de-reconocimiento-de.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+LatinAmericaProductBlog+%28Latin+Americba+Product+Blog%29

Jenaro C. Paz (s.f.). *Procesamiento digital de imágenes*. Recuperado el 24 de mayo de 2014, de <http://www2.uacj.mx/Publicaciones/GeneracionImagenes/imagenesCap8.pdf>

Musas20 (28 de noviembre de 2013). *Realidad aumentada para el patrimonio, museos... y para el mercado del arte*. Recuperado el 24 de mayo de 2014, de <http://www2.uacj.mx/Publicaciones/GeneracionImagenes/imagenesCap8.pdf>

Universidad Distrital Francisco José de Caldas (16 de septiembre de 2011). *Guía Rápida Proceso de Desarrollo OPENUP/OAS*. Recuperado el 09 de mayo de 2014, de <http://www.udistrital.edu.co/files/dependencias/oas/GuiaRapidaOpenUPOAS.pdf>

Universidad Distrital Francisco José de Caldas (16 de octubre de 2012). *Capítulo 1. Generalidades del Proceso OPENUD*. Recuperado el 11 de septiembre de 2014, de <http://www.udistrital.edu.co:8080/documents/276352/356568/Cap1GeneralidadesProcesoOpenup.pdf>

Universidad Distrital Francisco José de Caldas (16 de octubre de 2012). *Capítulo 2. Ciclo de Vida del Proceso de Desarrollo*. Recuperado el 11 de septiembre de 2014, de <http://www.udistrital.edu.co:8080/documents/276352/356568/Cap2CicloVidaProcesoDesarrollo>

Perurena Cancio (2013). *Usabilidad de los sitios Web, los métodos y las técnicas para la evaluación*. Recuperado el 18 de agosto de 2014, de <http://www.acimed.sld.cu/index.php/acimed/article/view/405/306>

W3c (s.f.). *Guía Breve de Accesibilidad Web*. Recuperado el 19 de agosto de 2014, de <http://w3c.es/Divulgacion/GuiasBreves/Accesibilidad>

Wikipedia (s.f.). *Google Maps*. Recuperado el 15 de mayo de 2014, de http://en.wikipedia.org/wiki/Google_Maps

Wikipedia (s.f.). *SQLite*. Recuperado el 15 de mayo de 2014, de <http://en.wikipedia.org/wiki/SQLite>

Wikipedia (s.f.). *Point of interés*. Recuperado el 12 de mayo de 2014, de http://en.wikipedia.org/wiki/Point_of_interest

Wikitude (s.f.). *Best practice for target images*. Recuperado el 24 de agosto de 2014, de <http://www.wikitude.com/developer/documentation/android>

ANEXOS

Anexo A: Tecnologías para el desarrollo de aplicaciones Web

Anexo A.A: Biblioteca de desarrollo de software jQuery

jQuery es una biblioteca de desarrollo de software de JavaScript de código abierto que posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos; jQuery consiste en un único archivo JavaScript que contiene funcionalidades comunes; entre ellas:

Permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM. De hecho, la función `$()` nos permite recuperar referencias a elementos del DOM.

Manejar eventos, desarrollar animaciones, agregar interacción con la técnica AJAX.

Agregar nuevas funcionalidades mediante *plugin*; además de ofrecer ya muchas funcionalidades basadas en JavaScript que de otra manera requeriría de mucho más código, es decir, con las funciones propias de esta biblioteca de desarrollo de software se logran grandes resultados en menos tiempo y espacio (Murphey, Rebecca, 2010).



Figura N° 57 Logo de jQuery.

Fuente: http://activ.com.mx/wp-content/uploads/2013/05/jquery_logo.png

Anexo A.B: jQuery UI

Es un conjunto de interfaces de interacción del usuario, efectos, *widgets*, y temas creados en la biblioteca de desarrollo de software jQuery, recomendando para aplicaciones altamente interactivas y para la creación de Aplicaciones Web dinámicas.

Con jQuery UI podemos Añadir comportamientos complejos a los elementos:

- **Draggable:** Hace al elemento arrastrable.
- **Droppable:** Permite que el elemento responda a elementos arrastrables.
- **Resizable:** Permite redimensionar el elemento.
- **Selectable:** Permite seleccionar entre una lista de elementos.
- **Sortable:** Ordena una lista de elementos.

También ofrece un conjunto completo de controles para la Interfaz de Usuario, los *Widgets* o pequeñas aplicaciones que tienen un conjunto de opciones configurable:

- Acordeón.
- Autocompletado.
- Botones.
- Ventana de Diálogos.
- *Slider* (Elemento para elegir en un rango de valores).
- *Tabs* o pestañas.
- *Datepicker* (Calendario gráfico).
- Barra de progreso.
- *Tooltip* o descripción emergente.
- Menús.



Figura N° 58 Logo de jQuery UI.

Fuente: http://upload.wikimedia.org/wikipedia/en/thumb/7/72/JQuery_UI_Logo.svg/559px-JQuery_UI_Logo.svg.png

Anexo A.C: jQuery Mobile

jQuery Mobile es un *framework* de interfaces basada en HTML5 utilizado para escribir aplicaciones para los sistemas operativos móviles como Android, IOS, Window Phone, BlackBerry, etc. Algunas características claves que podemos encontrar:

- Permite realizar sitios que se adapten a distintos tipos de resoluciones de pantallas y dispositivos.
- Está construido sobre jQuery y con la arquitectura de jQuery UI.
- Está desarrollado para trabajar en conjunto con HTML5.
- Repleto de automatismos; es decir con tan solo agregar unos atributos claves sobre los elementos, el *framework* se encarga de realizar las adaptaciones sobre ese elemento (transformar enlaces en botones, transformar contenedores en paneles, etc.).
- Personalizable con temas o desde el CSS (jQuery Mobile, s.f.).



Figura N° 59 Logo de jQuery Mobile.

Fuente: <http://academiaos.com/imagen/logo/jqueryM.png>

Anexo A.D: jQuery multiselect

Permite reemplazar el comportamiento y apariencia de las listas de opciones definida por defecto en por los navegadores web en una lista de opciones más personalizable y estética utilizando el estilo definido en jQuery UI; entre las principales características de jQuery multiselect se tiene:

- Personalizar la cabecera.
- Seleccionar Todos (para lista de opciones múltiples).
- Deseleccionar Todos (para lista de opciones múltiples).
- Selección simple.
- Selección múltiple.
- Deshabilitar y habilitar la lista de opciones.
- Filtrar la lista opciones.
- Animaciones en la apertura y cierre de la lista de opciones (Hynds, s.f.).

Anexo A.E: JQuery toastmessage

Permite introducir información oportuna y puntual al usuario de una manera natural, no intrusiva y sin interrumpir al usuario sobre las tareas que esté realizando; trae cuatro diferentes formas predeterminadas de mostrar errores: noticia, éxito, error, advertencia; ofrece las siguientes características:

- Personalización con CSS.
- Facilidad de instalación.
- Facilidad de uso (Akquinet, s.f.).



Figura N° 60 Imagen representativa de los mensajes.

Fuente: <http://akquinet.github.io/jquery-toastmessage-plugin/img/Toastmessages.png>

Anexo A.F: Fancybox

Es una herramienta utilizada para mostrar imágenes, contenido HTML y multimedia en páginas web de una forma atractiva; está construido en jQuery lo que garantiza compatibilidad entre los navegadores estas son algunas de sus características:

- Mostrar elementos HTML incluyendo imágenes.
- Personalización con CSS.
- Soporte de transiciones.
- Extensible utilizando Helpers.
- Diseño Web Adaptativo.
- Utiliza CSS3.
- Facilidad de instalación y uso (FancyBox, s.f.).

Anexo A.G: StickUp

Permite ubicar elementos HTML en el tope de la ventana del navegador aun cuando se utilice la barra de desplazamiento; de esta forma es posible ubicar elementos que son importantes para el usuario y así mejorar su experiencia en el sitio.

Anexo A.H: Loading overlay

Permite utilizar JavaScript para agregar una superposición de carga de un elemento; se puede crear la capa de superposición en cualquier elemento que deba recibir la superposición de carga; además es personalizable a través de estilos CSS (Loading Overlay, s.f.).

Anexo A.I: JSON

JSON de las siglas de Notación de Objetos JavaScript por sus siglas en ingles de *JavaScript Object Notation*; logrado generalizar su uso como formato ligero para el intercambio de datos; al contrario de otras tecnología como XML en donde es necesario construir un analizador para poder escribir o leer de él, JSON puede ser analizado y validado fácilmente desde JavaScript con la función **eval()**; es utilizado en conjunto con otras tecnologías como AJAX.

JSON puede ser construido en dos estructuras:

- Una colección de pares nombre/valor: Esto se puede interpretar desde un Lenguaje de Programación como JavaScript como un objeto, registro, estructura, diccionario, tabla hash, lista con clave, o una matriz asociativa.
- Una lista ordenada de valores: Esto se puede interpretar desde un Lenguaje de Programación como una matriz, vector, lista o secuencia (JSON, s.f.).

Anexo B: Tecnologías para el desarrollo de aplicaciones móviles

Anexo B.A: Android

Android es un sistema operativo para móviles basado en Linux inteligentes: teléfonos y tabletas principalmente; Android permite programar bajo el Lenguaje de Programación Java corriendo las aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución).

Estructura de un proyecto Android

Todos los proyectos Android eventualmente se construyen dentro de un .apk que es el archivo que se instala en el dispositivo; contiene cosas como el código fuente y recursos; algunos se crean de forma predeterminada y otros se agregan según lo amerite. Los siguientes archivos y carpetas componen un proyecto Android:

- **Carpeta src:** Contiene todo el código fuente (Java) de la aplicación, entre ellas nuestra actividad principal; que se almacenan en: src/nombre-paquete/Actividad.java.
- **Carpeta bin:** Directorio de salida; Aquí se encuentra el .apk y otros recursos procedentes de la compilación el proyecto.
- **Carpeta res:** Contiene los recursos (imágenes, vídeos, audio, etc.) necesarios para generar una aplicación Android; los diferentes tipos de recursos de deberán distribuir entre las siguientes carpetas:
 - **res/drawable/:** Guarda las imágenes y se divide en: drawable-ldpi, drawable-mdpi y drawable-hdpi, que dependerá de la resolución del dispositivo.
 - **res/raw/:** Contiene archivos de propósito general, la única diferencia con el directorio assets es que pueden ser referenciados usando el archivo R.
 - **res/layout/:** Incluye los archivos que definen el diseño de la interfaz gráfica, siempre en XML.

- **res/values/:** Guarda los datos y tipos que utiliza la aplicación, tales como colores, cadenas de texto, estilos, dimensiones, etc.
- **Carpeta gen:** Ésta carpeta guarda un conjunto de archivos (de código Java) creados automáticamente cuando se compila el proyecto, para poder dirigir los recursos de la aplicación. El archivo R ajusta automáticamente todas las referencias a archivos y valores de la aplicación (guardados en la carpeta res).
- **Carpeta assets:** Guarda el resto de archivos necesarios para el correcto funcionamiento de la aplicación, como los archivos de datos o de configuración. La principal diferencia entre los recursos que almacena ésta carpeta y los que guarda la carpeta res, es que los recursos de ésta última genera un identificador por recurso, que se encargará de gestionar el archivo R y sólo se podrá acceder a ellos a través de determinados métodos de acceso, mientras que los recursos almacenados en la carpeta assets no generan identificador alguno y se accede a ellos a través de su ruta, como se hace con cualquier otro archivo.
- **Archivo AndroidManifest.xml:** Este archivo es uno de los más importantes de cualquier aplicación Android. Se genera automáticamente al crear el proyecto, y en él se encuentra definida la configuración del proyecto en XML (Oliver Salvador, 2011).



Figura N° 61 Logo de Android.

Fuente: <http://fin6.com/wp-content/uploads/2013/08/07641ade31d49d2589152ba9940dd319.png>

Anexo B.B: Wikitude

Wikitude es un conjunto de herramientas de desarrollo de software (SDK) gratuito para proyecto no comerciales y paga en caso contrario; que permite la creación de aplicaciones de realidad aumentada para los sistemas operativos móviles Android, IOS y Blackberry 10. Soporta geolocalización y reconocimiento de imágenes; además permite utilizar tecnologías Web (HTML, JavaScript, CSS) para desarrollar aplicaciones de realidad aumentada; esta forma cruzada (utilizando el lenguaje nativo y tecnologías web para desarrollar una aplicación para alguna de las plataformas móviles anteriores) de construir la realidad aumentada se conoce con el nombre de *ARchitect Worlds*.

Para integrar el conjunto de herramientas de desarrollo de software (SDK) en una aplicación nativa solo hay que añadir a la aplicación nativa un componente de software que hará de puente entre la aplicación nativa y el *ARchitect Worlds* llamado *ArchitectView* en el proyecto; y este último, notificará todos los eventos que puedan ocurrir.

Creación de una colección de objetivos:

Para poder generar la **colección de objetivos** es necesario tener una cuenta en el portal de Wikitude, puede crearse una cuenta de forma gratuita en el siguiente enlace: <http://developer.wikitude.com/login>.

Creada la cuenta es posible crear la **colección de objetivos** la cual se puede generar de forma gratuita en el siguiente enlace: <http://developer.wikitude.com/tools/target-manager> y seguimos los siguientes pasos:

- Crear un proyecto: clic en el botón “*Add Project*”.
- Asignar un nombre y presionamos en el botón “*Create*”.
- Agregar imágenes en la opción “*Add images*”.
- Click en el botón “*Generate target collection*”.
- Características deseables en una imagen para su detección
- Entre 500 a 1000 píxeles en cada dimensión.
- Un alto contraste.
- Muchas esquinas como estructuras.
- Texturas distribuidas de forma uniforme.

Licenciamiento:

Wikitude SDK está disponible de forma gratuita cuando se utiliza para proyectos no comerciales; para proyectos comerciales, Wikitude ofrece varias opciones:

Paquete	Distribución	Características (Ver Tabla N° 8)	Soporte Técnico	Observaciones
SDK FLEX	200 usuarios.	Básico.	Comunidad.	<ul style="list-style-type: none"> • Versión de pago. • Puedes actualizar a cualquiera de los demás paquetes de pago.
SDK LITE	Ilimitada.	Básico.	Comunidad.	<ul style="list-style-type: none"> • Versión de pago.
SDK PRO	Ilimitada.	Profesional.	Profesionales en el área.	<ul style="list-style-type: none"> • Versión de pago.
SDK EDU	200 usuarios.	Profesional.	Comunidad.	<ul style="list-style-type: none"> • Versión gratuita. • Inicia la aplicación con una animación del logo de Wikitude. • El logo de Wikitude está presente en la pantalla (vista de la cámara).

Paquete	Distribución	Características (Ver Tabla N° 9)	Soporte Técnico	Observaciones
SDK TRIAL	Ilimitada.	Profesional.	Comunidad.	<ul style="list-style-type: none"> • Versión gratuita. • Inicia la aplicación con una animación del logo de Wikitude. • El logo de Wikitude está presente en la pantalla (vista de la cámara). • Una marca de agua está presente en la pantalla (vista de la cámara).

Tabla 8 Comparación entre las versiones gratuitas y las versiones de pago de Wikitude SDK.

Fuente: http://www.wikitude.com/products/wikitude-sdk/pricing/#feature_set

		Básico	Profesional
Dispositivos	Teléfonos Inteligentes.	Sí.	Sí
	Tabletas.	Sí.	Sí.
Características principales	Plataforma Cruzada (JavaScript API).	Sí.	Sí.
	Comunicación con el código nativo de la aplicación.	Sí.	Sí.
	Reconocimiento de imagen.	No.	Sí.
	Documentación y ejemplos.	Sí.	Sí.
	Geolocalización basada en realidad aumentada.	Sí.	Sí.
	Número de paquetes.	N/A	Ilimitado.
	Número de objetivos con imágenes en un paquete dentro de un dispositivo.	N/A	1000

Tabla 9 Lista de características de Wikitude SDK.

Fuente: http://www.wikitude.com/products/wikitude-sdk/pricing/#feature_set

(Desarrollador Wikitude, s.f.).



Figura N° 62 Logo de Wikitude.

Fuente: <http://realidadaumentadabarcelona.com/wp-content/uploads/2013/11/wikitude-logo.png>

Anexo C: Tecnologías para el desarrollo de aplicaciones Web y móviles

Anexo C.A: Mapas de Google

Los mapas de Google consisten en un servidor de aplicaciones de mapas que pueden ser incrustadas en una página web mediante JavaScript o en una aplicación para Android. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes

Los mapas de Google permiten hacer clic en cualquier parte para que se muestre más información según la ubicación en la que nos encontremos, en general los siguientes elementos están presentes en ellos:

- **Cuadro de búsqueda:** Busca lugares, negocios, intersecciones, direcciones etc.
- **Historial:** Ver un historial reciente de búsquedas y páginas visitadas.
- **Resultados de búsqueda:** Mediante marcadores para obtener información.
- **Compartir enlaces:** Puedes compartir la URL de tu mapa en cualquier momento.
- **Regla para medir distancias:** Te ayudará a entender la escala del mapa y las distancias entre ubicaciones en distintas magnitudes.
- **Recorridos virtuales:** Permite realizar recorridos virtuales por las calles, ciudades, plazas, etc.
- **Contenido multimedia:** Permite incluir elementos multimedia (enlaces a páginas web, imágenes, videos, etc.) dentro de las explicaciones emergentes de marcas de posiciones y zonas.
- **Acercamientos y alejamientos:** Para mostrar el mapa.
- **Crear mapas personalizados.**

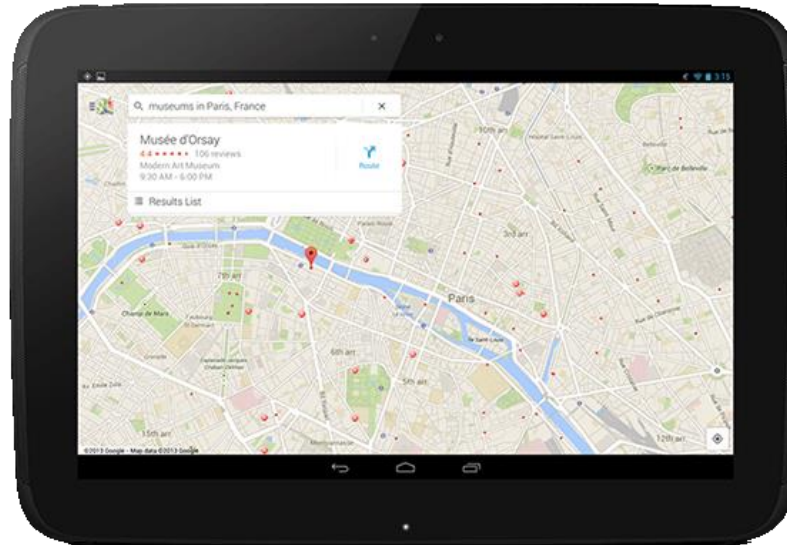


Figura N° 63 Mapas de Google.

Fuente: <http://www.google.com/mobile/maps/images/c2.png>

Anexo C.B: SQLite

SQLite es un sistema gestor de bases de datos relacional, contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos.

El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo archivo estándar en la máquina host.

SQLite usa un sistema de tipos inusual. En lugar de asignar un tipo a una columna como en la mayor parte de los sistemas de bases de datos SQL, los tipos se asignan a los valores individuales. Por ejemplo, se puede insertar una cadena de texto en una columna de tipo entero (a pesar de que SQLite tratará en primera instancia de convertir la cadena en un entero).

Razones para elegir SQLite:

- **Tamaño:** SQLite tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- **Rendimiento de base de datos:** SQLite realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- **Portabilidad:** SQLite se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.

- **Estabilidad:** SQLite es compatible con ACID, reunión de los cuatro criterios de Atomicidad, Consistencia, Aislamiento y Durabilidad.
- **SQL:** SQLite implementa un gran sub-conjunto de la ANSI - 92 SQL estándar, incluyendo sub-consultas, generación de usuarios, vistas y triggers.
- **Interfaces:** SQLite cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, Perl, Python, etc.
- **Costo:** SQLite es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.

Debido a su pequeño tamaño, SQLite es muy adecuado para los sistemas integrados, y también está incluido en:

- Android.
- BlackBerry.
- Windows Phone 8.
- iOS.
- Google Chrome.
- Mozilla Firefox (SQLite, s.f.).



Figura N° 64 Logo de SQLite.

Fuente: http://www.sqlite.org/images/sqlite370_banner.gif

Anexo D: Técnicas para el procesamiento digital de imágenes

Anexo D.A: Contraste

El contraste es una variación de intensidades que permite incrementar la luminosidad entre las zonas oscuras y claras de una imagen permitiendo un mejor enfoque y claridad de la imagen.

Nótese que esta fórmula representa una familia de rectas que pasan por el punto (128,128) con diferentes pendientes.

Si observamos la siguiente figura, aumentar o disminuir el contraste en una imagen consiste en aumentar o disminuir la pendiente de la línea recta con pendiente a 45 grados que representa los grises, cuidando siempre de nunca rebasar los límites 0 y 255:

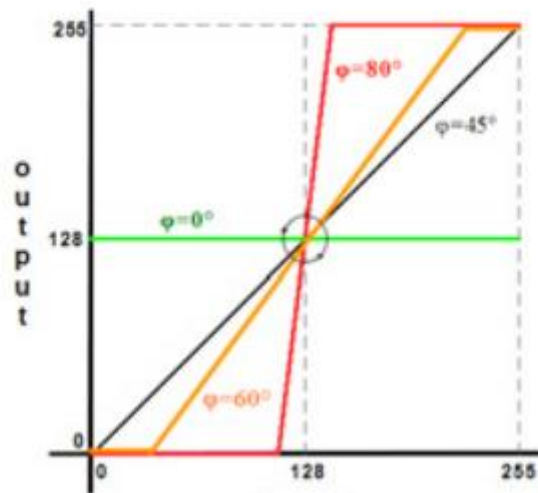


Figura N° 65 Pendiente de la recta de una imagen.

Fuente: <http://www2.uacj.mx/Publicaciones/GeneracionImagenes/imagenesCap8.pdf>

Así, la fórmula que tenemos que aplicar en este tipo de transformación tiene la forma siguiente:

$$\text{pixelConContraste} = (\text{valorOriginal} - 128) * \tan(\text{ángulo}) + 128;$$

\tan = Tangente.

$\text{PI} \approx 3.141$.

valorOriginal = Valor original del píxel de una imagen.

En donde el ángulo es calculado como: $\text{ángulo} = \tan(\text{val} * \text{PI} / 180.0)$ y donde “val” es un valor negativo o positivo que aumenta o reduce el contraste (Jenaro C. Paz ,s.f.).

Anexo D.B: Recortar imágenes

Hay muchas razones para recortar una imagen; por ejemplo, ajustar una imagen para rellenar un marco, quitar una parte del fondo para resaltar el sujeto, quitar bordes, o para eliminar áreas no deseadas para proporcionar un área de trabajo más concentrada, etc. También, es útil si necesita un tamaño específico de imagen distinto de las dimensiones originales de su imagen (Gimp, s.f.).

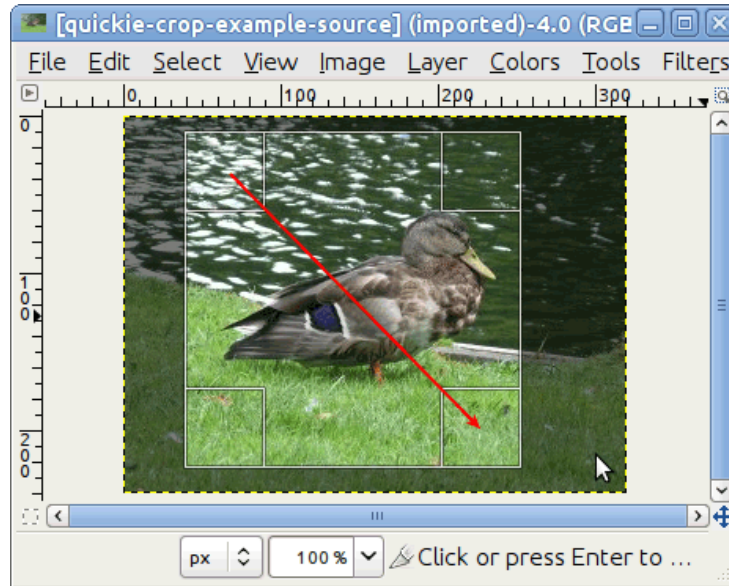


Figura N° 66 Recorte de una imagen con GIMP.

Fuente: <http://docs.gimp.org/es/images/tutorials/quickie-crop-step1.png>

Anexo E: Prueba de aceptación para MACUC

Test de aceptación para MACUC

*Obligatorio

Tras una primera mirada, ¿Le queda claro cuál es el objetivo del sitio y el tipo de contenido que ofrece? *

Sí
 No

¿Encuentra una interfaz intuitiva? *

Sí
 No

Tras una primera mirada, ¿Es posible saber a qué institución corresponde la aplicación? *

Sí
 No

De una escala del 1-10, ¿qué le parece la navegabilidad de la aplicación? *

En donde 1 significa que la navegabilidad es mala y 10 que es excelente

1 2 3 4 5 6 7 8 9 10

De una escala del 1-10, ¿qué le parece los colores de la aplicación? *

En donde 1 significa que no le gustan los colores y 10 que es si le gustan los colores

1 2 3 4 5 6 7 8 9 10

De una escala del 1-10, ¿considera que la interfaz es simple o recargada? *

En donde 1 significa que la interfaz es simple y 10 que es recargada

1 2 3 4 5 6 7 8 9 10

De una escala del 1-10, ¿Cómo considera los tiempos para la carga de información y datos en los distintos módulos? *

En donde 1 significa que son malos y 10 que son excelentes

1 2 3 4 5 6 7 8 9 10

¿Alguna vez se ha perdido dentro de la aplicación o no supo cómo realizar alguna actividad? *

¿Cómo considera el proceso para crear un recorrido y agregar Puntos de Interés en el mismo? *

¿Cómo considera el proceso para exportar un recorrido e importarlo en la aplicación? *

¿Le es útil visualizar los recorridos en un mapa digital? *

¿La sección de ayuda le fue útil en algún momento? *

¿Qué le parece la estructura de la sección de ayuda? *

¿Fondos con o sin marca de agua? *

¿En que dispositivo(s) ha probado la aplicación? *

En caso de ser más de uno, sepárelos por guión

¿Qué nombre prefiere o piensa que se identifica mejor con la aplicación? *

¿Alguna consideración u observación que desee añadir? *

Figura N° 67 Prueba de aceptación para MACUC.

Fuente: Creada por el Autor.

Anexo F: Resultados de las encuestas

¿Queda claro cuál es el objetivo de la aplicación?	Navegabilidad de la aplicación	Interfaz de la aplicación	Tiempos de carga de información	Proceso para exportar un recorrido	¿Le es útil visualizar los recorridos en un mapa virtual?
Sí.	8	10	10	Excelente.	Sí.
No.	5	9	7	Depende de la aplicación que se use para importar el recorrido.	Sí.
Sí.	9	10	10	Sencillo.	Sí, excelente funcionalidad.
Sí.	7	10	10	Fácil y eficiente.	Sí, eso hace más intuitivo al usuario ubicarse.
Sí.	9	10	7	Rápido y sencillo.	Sí, es muy útil porque permite que las personas puedan ubicar fácilmente las obras que forman parte de la Ciudad Universitaria de Caracas.
Sí.	10	10	9	Sencillo.	Sí

Tabla 10 Resultados de las encuestas.

Fuente: Creada por el Autor.

Anexo G: Manual del usuario Administrador



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR - CENEAC

Colección Móvil de Arte de la Ciudad Universitaria de Caracas Manual del Administrador de MACUC

*Desarrollado por Andrés Cruz Yoris
Correo-e: andres29111990@gmail.com*