

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UNA APLICACIÓN MÓVIL PROTOTIPO UTILIZANDO REALIDAD AUMENTADA Y EL SISTEMA OPERATIVO ANDROID PARA BRINDAR INFORMACIÓN SOBRE PINTURAS Y ESTATUAS DE SANTUARIOS RELIGIOSOS

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN INGENIERÍA ELECTRÓNICA Y REDES DE INFORMACIÓN**

YESSICA MARÍA CRUZ VILLEGAS

DIRECTOR: GABRIEL ROBERTO LÓPEZ FONSECA

CODIRECTOR: JORGE EDUARDO CARVAJAL RODRIGUEZ

Quito, octubre 2020

AVAL

Certificamos que el presente trabajo fue desarrollado por Yessica María Cruz Villegas, bajo nuestra supervisión.

GABRIEL ROBERTO LÓPEZ FONSECA
DIRECTOR DEL TRABAJO DE TITULACIÓN

JORGE EDUARDO CARVAJAL RODRIGUEZ
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Yessica María Cruz Villegas, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Yessica María Cruz Villegas

DEDICATORIA

Dedico el presente proyecto de titulación a mi mami, Magdalena Villegas, quien estuvo a mi lado en cada paso apoyándome cuando más lo necesitaba. Este logro es obra tuya, gracias a tu paciencia, sacrificio y amor incondicional.

Te amo.

AGRADECIMIENTO

Agradezco:

A Dios por brindarme la fortaleza y sabiduría necesaria a lo largo de mi vida universitaria.

A mi mami por todas las noches de desvelo y cuidado que me brindó durante toda mi vida estudiantil.

A mi hermano por el apoyo que me brindó durante mi carrera y aún más en el desarrollo del presente trabajo de titulación.

A mi padre por enseñarme a ser una persona perseverante y a nunca rendirme.

A mis tíos (Gladys y Luis) por brindarme su cariño, apoyo y consejos durante todo este tiempo

Al Ingeniero Freddy Malatay por ser parte fundamental en el desarrollo de este proyecto porque gracias a su experiencia, apoyo y consejos lograron consolidar la idea de SantuRA.

Al Padre Carlos Solarte, Párroco del Santuario del Señor de la Salud por brindarme las facilidades para desarrollar el presente trabajo de titulación.

A los docentes de la Escuela Politécnica Nacional quienes, con paciencia, responsabilidad compromiso me otorgaron los conocimientos necesarios para mi vida profesional.

Al Magíster Gabriel López y al Magíster Jorge Carvajal, director y codirector respectivamente del presente proyecto de titulación, por estar pendientes en el desarrollo de mi tesis y darme las directrices y conocimientos necesarios para solventar el problema existente en los santuarios a nivel nacional.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
ÍNDICE DE FIGURAS.....	IX
ÍNDICE DE TABLAS.....	XII
ÍNDICE DE CÓDIGO	XIII
RESUMEN.....	XIV
ABSTRACT	XV
1. INTRODUCCIÓN.....	1
1.1. OBJETIVOS	2
1.2. ALCANCE	3
1.3. MARCO TEÓRICO	6
1.3.1. METODOLOGÍA DE DESARROLLO RÁPIDO DE APLICACIONES	6
1.3.1.1. Definición.....	6
1.3.1.2. Ventajas	6
1.3.1.3. Desventajas.....	6
1.3.1.4. Fases de la Metodología	7
1.3.2. REALIDAD AUMENTADA	8
1.3.2.1. Definición.....	8
1.3.2.2. Funcionamiento.....	8
1.3.2.3. Elementos de un SRA	9
1.3.2.4. Realidad Aumentada y el turismo	10
1.3.2.5. Ventajas de la Realidad Aumentada en el turismo.....	11
1.3.3. HERRAMIENTAS PARA EL DESARROLLO DE LA APLICACIÓN.....	11
1.3.3.1. Vuforia.....	11
1.3.3.1.1. Arquitectura de Vuforia	12
1.3.3.2. Unity	14
1.3.3.3. Vegas Pro.....	15
1.3.4. PORTAL CAUTIVO	16
1.3.4.1. Definición.....	16

1.3.4.2.	Tipos de Portales Cautivos	16
1.3.4.3.	Ventajas	17
2.	METODOLOGÍA.....	18
2.1.	SITUACIÓN ACTUAL	18
2.2.	FASE DE DISEÑO.....	18
2.2.1.	FASE DE PLANEACIÓN DE REQUERIMIENTOS	18
2.2.1.1.	Requerimientos Funcionales	19
2.2.1.2.	Requerimientos No Funcionales.....	19
2.2.2.	HISTORIAS DE USUARIO	20
2.2.2.1.	Formato.....	20
2.2.2.2.	Detalle de Historias de Usuario	21
2.2.3.	CASOS DE USO	22
2.2.3.1.	Descripción de los Casos de Uso	22
2.2.4.	DIAGRAMAS DE SECUENCIA.....	29
2.2.5.	SKETCHES DE INTERFACES DE USUARIO	32
2.2.5.1.	Escena “Menu”	32
2.2.5.2.	Escena “EscenaRA”	33
2.2.5.3.	Escena “MenuRA”	34
2.2.5.4.	Escena “Información”.....	36
2.2.5.5.	Escena “Ayuda”	37
2.2.5.6.	Escena “EscenaDeCarga”	38
2.2.6.	DIRECCIONAMIENTO IP	39
2.2.7.	ELECCIÓN DEL ACCESS POINT	40
2.2.8.	PARÁMETROS DE CONFIGURACIÓN DEL ACCESS POINT.....	41
2.2.8.1.	Configuración Básica.....	41
2.2.8.2.	Configuración IP	42
2.2.8.3.	Ajustes de Radio	43
2.2.8.4.	Balance de carga.....	45
2.2.9.	PARÁMETROS DE CONFIGURACIÓN DEL PORTAL CAUTIVO	46
2.2.9.1.	Configuración Básica del Portal Cautivo	46
2.2.9.2.	Configuración de la Página de Login	48
2.2.9.3.	Configuración de anuncios	49
2.2.10.	PARÁMETROS DE CONFIGURACIÓN DEL SERVIDOR DE DESCARGA.....	50
2.2.10.1.	Servicio DNS	50
2.2.10.2.	Servicio DHCP.....	50
2.2.10.3.	Servicio Web	50
2.3.	FASE DE IMPLEMENTACIÓN	51

2.3.1.	ENTREVISTAS.....	51
2.3.2.	GRABACIÓN Y EDICIÓN DE AUDIO.....	51
2.3.3.	BASE DE DATOS EN VUFORIA.....	54
2.3.4.	CONFIGURACIÓN DE LA PLATAFORMA DE DESARROLLO.....	60
2.3.5.	CONFIGURACIÓN PARA LA CONSTRUCCIÓN DE LA APLICACIÓN.....	61
2.3.5.1.	Build Settings.....	61
2.3.5.2.	Player Settings.....	62
2.3.6.	DESARROLLO DE ESCENAS.....	64
2.3.1.1	Escena Información.....	67
2.3.1.2	Escena EscenaRA.....	69
2.3.7.	CODIFICACIÓN.....	76
2.3.7.1.	Script Menu Controller.....	76
2.3.7.2.	Script LoadScene.....	78
2.3.7.3.	Script Scene Controller.....	79
2.3.7.4.	Script DefaultTrackableEventHandler.....	79
2.3.7.5.	Script GuardarEstado.....	81
2.3.7.6.	Script MenuRAcontroller.....	82
2.3.7.7.	Script InfoController.....	83
2.3.7.8.	Script HelpController.....	87
2.3.7.9.	Script VideoController.....	88
2.3.8.	CONFIGURACIÓN MÓDULO DE DESCARGA.....	89
2.3.8.1.	Configuración del servidor.....	89
2.3.8.1.1.	Servicio DNS.....	89
2.3.8.1.2.	Servicio DHCP.....	91
2.3.8.1.3.	Servicio WEB.....	93
2.3.8.2.	Configuración del Access Point.....	96
2.3.8.3.	Configuración del Portal Cautivo.....	99
3.	RESULTADOS Y DISCUSIÓN.....	100
3.1.	EVALUACIÓN DE LOS PROTOTIPOS.....	100
3.1.1.	PROTOTIPO NÚMERO UNO.....	101
3.1.2.	PROTOTIPO NÚMERO DOS.....	105
3.1.3.	PROTOTIPO NÚMERO TRES.....	107
3.1.4.	PROTOTIPO NÚMERO CUATRO.....	109
3.2.	PRUEBAS DE LA APLICACIÓN.....	111
3.2.1.	PRUEBAS DE LA ESCENA MENU.....	113
3.2.2.	PRUEBAS DE LA ESCENA AYUDA.....	113
3.2.3.	PRUEBAS DE LA ESCENA DE CARGA.....	114

3.2.4.	PRUEBAS DE LA EscenaRA	114
3.2.5.	PRUEBAS DE LA ESCENA MenuRA.....	115
3.2.6.	PRUEBAS DE LA ESCENA INFORMACIÓN.....	116
3.3.	PRUEBAS DEL MÓDULO DE DESCARGA	116
3.4.	CORRECCIÓN DE ERRORES.....	119
4.	CONCLUSIONES Y RECOMENDACIONES	123
4.1.	CONCLUSIONES.....	123
4.2.	RECOMENDACIONES.....	124
5.	REFERENCIAS BIBLIOGRÁFICAS.....	125
	ANEXOS.....	¡Error! Marcador no definido.

ÍNDICE DE FIGURAS

Figura 1.1	Funcionamiento de la arquitectura de la aplicación	4
Figura 1.2	Topología para descargar el APK de la aplicación	5
Figura 1.3	Etapas de la metodología RAD	5
Figura 1.4	Fases de la Metodología RAD.....	7
Figura 1.5	Funcionamiento del SRA	8
Figura 1.6	Elementos de un SRA.....	9
Figura 1.7	Ventana principal de Unity	14
Figura 1.8	Tipos de Portales Cautivos	16
Figura 2.1	Diagrama de casos de uso.....	23
Figura 2.2	Diagrama de Secuencia para Escena de Menú y Ayuda.....	30
Figura 2.3	Diagrama de Secuencia para Escena de Menú, EscenaRA, MenuRA e Información.....	31
Figura 2.4	Diagrama de Secuencia para Escena de Menú, MenuRA e Información	30
Figura 2.5	Escena Menú	33
Figura 2.6	Escena EscenaRA	34
Figura 2.7	Escena MenuRA	35
Figura 2.8	Escena Información	37
Figura 2.9	Escena Ayuda.....	38
Figura 2.10	Escena EscenaDeCarga.....	39
Figura 2.11	Arquitectura para la descarga de la aplicación	40
Figura 2.12	Configuración del nombre del Access Point	41
Figura 2.13	Configuración IP dinámica del Access Point	42
Figura 2.14	Configuración IP estática del Access Point	42
Figura 2.15	Ajustes de Radio del Access Point.....	43
Figura 2.16	Configuración del Modo del Access Point	44
Figura 2.17	Configuración del ancho del canal del Access Point	44
Figura 2.18	Configuración del canal del Access Point.....	45
Figura 2.19	Configuración de la potencia de transmisión	45
Figura 2.20	Configuración del balance de carga del Access Point.....	46
Figura 2.21	Configuración básica del portal cautivo	47
Figura 2.22	Tipo de autenticación del portal cautivo	48
Figura 2.23	Configuración de la página de login	49
Figura 2.24	Configuración de anuncios publicitarios	49
Figura 2.25	Insertar pista de audio.....	52
Figura 2.26	Grabar el audio	52
Figura 2.27	Ventana para renderizar la pista	53
Figura 2.28	Pista después de cortar los espacios sin audio.	53
Figura 2.29	Cortar el audio.....	53
Figura 2.30	Portal de desarrolladores de Vuforia	56
Figura 2.31	Ventana Target Manager	56
Figura 2.32	Ventana Crear base de datos.....	57
Figura 2.33	Ventana Target Manager con la base de datos creada	57
Figura 2.34	Ventana de la base de datos.....	58
Figura 2.35	Ventana Add Target	58
Figura 2.36	Ventana de la base de datos con el marcador creado.....	59
Figura 2.37	Ventana con la información del marcador	59
Figura 2.38	Ventana de descarga de la base de datos	60
Figura 2.39	Ventana Preferences de Unity.....	60

Figura 2.40	Ventana Build Settings	61
Figura 2.41	Ventana Player Settings.....	62
Figura 2.42	Configuración Splash Screen	63
Figura 2.43	Configuración Other Settings	63
Figura 2.44	Configuración XR Settings	64
Figura 2.45	Ventana Game	65
Figura 2.46	Configuración del Canvas	66
Figura 2.47	Ventana Inspector del Panel	66
Figura 2.48	Nivel de jerarquía de los elementos UI del Canvas	66
Figura 2.49	Ventana Game de la escena Información.....	67
Figura 2.50	Ventana Jerarquía de la escena Información	67
Figura 2.51	Ventana Inspector del elemento Text	68
Figura 2.52	Configuración del panel para el elemento txtInformación	68
Figura 2.53	Elementos del Slider	69
Figura 2.54	Elementos del Slider (Ventana de Jerarquía)	69
Figura 2.55	Incluir la AR Camera para la escena EscenaRA	70
Figura 2.56	Ventana Inspector de AR Camera.....	70
Figura 2.57	Ventana Inspector de la configuración Vuforia	71
Figura 2.58	Ventana License Manager	71
Figura 2.59	Ventana License Manager con la licencia creada.....	72
Figura 2.60	Ventana para crear la licencia	72
Figura 2.61	Ventana License Key	73
Figura 2.62	Insertar un Image Target en Unity	73
Figura 2.63	Ventana Inspector del Image.....	74
Figura 2.64	Image Targets de la escena EscenaRA	74
Figura 2.65	Ventana Jerarquía de la escena EscenaRA	75
Figura 2.66	Ventana Game de la escena EscenaRA	75
Figura 2.67	Ventana Build Settings con las escenas de la aplicación	76
Figura 2.68	Añadir la función cargarEscenaRA() al botón Iniciar	77
Figura 2.69	Archivo named.conf.local	90
Figura 2.70	Archivo db.santura	90
Figura 2.71	Resultado del comando named-checkzone para el archivo db.santura	90
Figura 2.72	Archivo db.192	91
Figura 2.73	Resultado del comando named-checkzone para el archivo db.192.....	91
Figura 2.74	Archivo dhcpd.conf (PARTE I).....	92
Figura 2.75	Archivo dhcpd.conf (PARTE II).....	92
Figura 2.76	Archivo apache2.conf.....	93
Figura 2.77	Archivo dir.conf	93
Figura 2.78	Directorios creados en /var/www/html	94
Figura 2.79	Regla CSS	94
Figura 2.80	AP activo en Omada Controller	97
Figura 2.81	Configuración básica AP	97
Figura 2.82	Configuración IP del AP	97
Figura 2.83	Configuración de Radio del AP	98
Figura 2.84	Configuración de balance de carga	98
Figura 2.85	Configuración SSID.....	99
Figura 2.86	Configuración portal cautivo	99
Figura 3.1	Escena Menu (Prototipo 1).....	102
Figura 3.2	Escena Ayuda (Prototipo 1)	102
Figura 3.3	EscenaRA (Prototipo 1).....	103

Figura 3.4 MenuRA de una pintura (Prototipo 1)	103
Figura 3.5 MenuRA de una escultura (Prototipo 1)	104
Figura 3.6 Escena Información con la oración de una escultura (Prototipo 1)	104
Figura 3.7 Escena Información con los milagros de una escultura (Prototipo 1).....	104
Figura 3.8 Escena Información con la historia de una escultura (Prototipo 1)	104
Figura 3.9 Escena Información (Prototipo 2)	106
Figura 3.10 Escena EscenaRA sin detectar el marcador	108
Figura 3.11 Escena EscenaRA con el marcador identificado	108
Figura 3.12 Escena Ayuda - Canvas Video (Prototipo 4)	110
Figura 3.13 Escena Ayuda - Canvas Principal (Prototipo 4)	110
Figura 3.14 Visualización de la red Santuario en el dispositivo móvil	118
Figura 3.15 IP que fue asignado al dispositivo móvil	117
Figura 3.16 Página del portal cautivo	118
Figura 3.17 Página en dónde se podrá descargar el APK de la aplicación.....	119
Figura 3.18 Cambio del valor de Orientation en Player Settings	120
Figura 3.19 Posición de los anclajes de los elementos del canvas.....	121
Figura 3.20 Configuración del Canvas	121
Figura 3.21 Ventana Inspector del audio.....	122
Figura C.1 Banco de preguntas - Requerimientos.....	¡Error! Marcador no definido.
Figura D.1 Banco de Preguntas - Párrocos del Santuario ..	¡Error! Marcador no definido.
Figura E.1 Banco de preguntas - Pintor Ivo Mora.....	¡Error! Marcador no definido.
Figura F.1 Ventana de Jerarquía de la escena Menu.....	¡Error! Marcador no definido.
Figura F.2 Ventana Game de la escena Menu	¡Error! Marcador no definido.
Figura F.4 Ventana Game de la escena MenuRA.....	¡Error! Marcador no definido.
Figura F.3 Ventana Jerarquía de la escena MenuRA.....	¡Error! Marcador no definido.
Figura F.5 Ventana de Jerarquía de la escena EscenaDeCarga.....	¡Error! Marcador no definido.
Figura F.6 Ventana Game de la escena EscenaDeCarga..	¡Error! Marcador no definido.
Figura F.7 Ventana de Jerarquía de la escena Ayuda.....	¡Error! Marcador no definido.
Figura F.8 Ventana Game del canvas principal de la escena Ayuda .	¡Error! Marcador no definido.
Figura F.9 Ventana Game del canvas video de la escena Ayuda.....	¡Error! Marcador no definido.

ÍNDICE DE TABLAS

Tabla 1.1	Targets de reconocimeinto de Vuforia	13
Tabla 2.1	Formato de las Historias de Usuario	20
Tabla 2.2	Historia de Usuario: Reconocimiento de imágenes.....	21
Tabla 2.3	Historia de Usuario: Información Adicional.....	21
Tabla 2.4	Historia de Usuario: Audio	21
Tabla 2.5	Historia de Usuario: Visualización en tiempo real	22
Tabla 2.6	Historia de Usuario: Instrucciones de la aplicación	22
Tabla 2.7	Formato de la Tabla de descripción de casos de uso	24
Tabla 2.8	Caso de Uso: Activar cámara	25
Tabla 2.9	Caso de Uso: Enfocar imagen	25
Tabla 2.10	Caso de Uso: Reconocer imagen	26
Tabla 2.11	Caso de Uso: Ver Información de imágenes.....	26
Tabla 2.12	Caso de Uso: Ver Historia	26
Tabla 2.13	Caso de Uso: Ver Milagros	27
Tabla 2.14	Caso de Uso: Ver Oración	27
Tabla 2.15	Caso de Uso: Reproducir audio	27
Tabla 2.16	Caso de Uso: Ver información adicional de imágenes	28
Tabla 2.17	Caso de Uso: Biografía Pintor Ivo Mora.....	28
Tabla 2.18	Caso de Uso: Historia del santuario.....	28
Tabla 2.19	Caso de Uso: Ver Ayuda	29
Tabla 2.20	Direccionamiento IP para la red de descarga	39
Tabla 2.21	Tabla comparativo de Access Points	41
Tabla 2.22	Lista de imágenes y pinturas	55
Tabla 3.1	Formato para la evaluación de los prototipos.....	101
Tabla 3.2	Evaluación del prototipo número uno.....	105
Tabla 3.3	Evaluación del prototipo número dos	107
Tabla 3.4	Evaluación del prototipo número tres.....	109
Tabla 3.5	Evaluación del prototipo número cuatro.....	111
Tabla 3.6	Características del celular Samsung J5 Prime.....	112
Tabla 3.7	Características del celular Xiaomi Redmi Note 8	112
Tabla 3.8	Formato para las pruebas de funcionamiento	112
Tabla 3.9	Formato para las pruebas de reconocimiento de las imagenes	113
Tabla 3.10	Resultados de las pruebas de funcionamiento de la escena Menu.....	113
Tabla 3.11	Resultados de las pruebas de funcionamiento de la escena Ayuda.....	114
Tabla 3.12	Resultados de las pruebas de funcionamiento de la escena de carga	114
Tabla 3.13	Resultados de las pruebas de funcionamiento de la EscenaRA	114
Tabla 3.14	Resultados de las pruebas de reconocimiento de las imágenes	115
Tabla 3.15	Resultados de las pruebas de funcionamiento de la escena MenuRA.....	115
Tabla 3.16	Resultados de las pruebas de funcionamiento de la escena Información	116
Tabla A.1	Estatuas del Santuario del Señor de la Salud	¡Error! Marcador no definido.
Tabla B.1	Pinturas del Santuario del Señor de la Salud	¡Error! Marcador no definido.

ÍNDICE DE CÓDIGO

Código 2.1 Función Start() del script Menu Controller.....	77
Código 2.2 Funciones para cambiar de escena del script MenuController	77
Código 2.3 Función Start() del script LoadScene	78
Código 2.4 Función Update() del script LoadScene	78
Código 2.5 Función Start() y Update() del script SceneController	79
Código 2.6 Funciones de cambio de escena del script SceneController	79
Código 2.7 Declaración de variables y definición de métodos del script DefaultTrackableEventHandler	80
Código 2.8 Método OnTrackingFound()	80
Código 2.9 Método OnTrackingLost()	81
Código 2.10 Función Awake() del script GuardarEstado.....	81
Código 2.11 Función Update() del script GuardarEstado	82
Código 2.12 Función Start() del script MenuRAcontroller.....	82
Código 2.13 Función cambiarEscenaH() del script MenuRAcontroller	83
Código 2.14 Función Regresar() del script MenuRAcontroller.....	83
Código 2.15 Función Start() del script InfoController.....	84
Código 2.16 Función Update y ActualizarTimer del script InfoController	84
Código 2.17 Función ChangeSlider del script InfoController	85
Código 2.18 Función Play() del script InfoController	85
Código 2.19 Funciones Stop, Pause, movePosition y Regresar del script InfoController	86
Código 2.20 Función Lectura del script InfoController.....	86
Código 2.21 Función Buscar del script Info Controller.....	87
Código 2.22 Función Start() del script HelpController	88
Código 2.23 Función Regresar y HabilitarCanvas del script HelpController	88
Código 2.24 Función cargarEscena() del script VideoController	89
Código 2.25 Función Update() del script VideoController.....	89
Código 2.26 Archivo estilos.css	95
Código 2.27 Archivo index.html	96
Código 3.1 Modificación del código del Script LoadScene	122

RESUMEN

El presente proyecto de titulación tiene como objetivo crear un prototipo de una aplicación en Android, para brindar información de estatuas y pinturas en Santuarios Religiosos, mediante el uso de realidad aumentada. Este prototipo utilizará como base la información del Santuario del Señor de la Salud en la provincia Bolívar.

El prototipo, fue desarrollado en Unity y se utilizó Vuforia para la implementación de realidad aumentada. Consta de un menú principal en el cual se podrá acceder al módulo de realidad aumentada, instrucciones e información adicional. El prototipo presenta una arquitectura autónoma y el usuario podrá acceder al APK mediante un servidor de descarga ubicado en el Santuario.

El primer capítulo describe los objetivos, alcance y marco teórico del proyecto. La sección teórica detalla las herramientas, así como la metodología utilizada para el desarrollo prototipo. En el segundo capítulo muestra la metodología con sus fases de diseño y desarrollo. El tercer capítulo, corresponde a pruebas y resultados, y por último el cuarto capítulo detalla las conclusiones y recomendaciones obtenidas al finalizar el desarrollo del presente proyecto de titulación.

PALABRAS CLAVE: Unity, Vuforia, Realidad Aumentada, Android

ABSTRACT

The objective of this degree project is to create a prototype of an application on Android, to provide information on statues and paintings in Religious Shrines, through the use of augmented reality. This prototype will use the information from the Sanctuary of the Lord of Health in Bolívar province as a basis.

The prototype was developed in Unity and Vuforia was used for the implementation of augmented reality. It consists of a main menu in which you can access the augmented reality module, instructions and additional information. The prototype has an autonomous architecture and the user will be able to access the APK through a download server located in the Sanctuary.

The first chapter describes the objectives, scope and theoretical framework of the project. The theoretical section details the tools, as well as the methodology used for the prototype development. The second chapter shows the methodology with its design and development phases. The third chapter corresponds to tests and results, and finally the fourth chapter details the conclusions and recommendations obtained at the end of the development of this degree project.

KEYWORDS: Unity, Vuforia, Augmented Reality, Android

1. INTRODUCCIÓN

Los santuarios religiosos se encuentran distribuidos en diferentes provincias de nuestro país, estos santuarios atraen a propios y extranjeros fortaleciendo la actividad turística. El Ministerio de Turismo impulsa el turismo religioso, mediante la fusión de turismo de fe y el turismo cultural [1]. En la actualidad los smartphones se han convertido en parte de nuestro diario vivir, es por ello que el turismo y el avance de la tecnología permite que se fusionen para brindar una mejor experiencia al turista, una de estas tecnologías es la realidad aumentada.

Por otro lado, los santuarios religiosos no proporcionan información sobre estatuas y pinturas que reposan en ellas, perjudicando la experiencia del visitante y por consecuencia afectan al turismo de la parroquia o la ciudad donde se encuentre la misma. El proyecto propuesto brinda una solución tecnológica e innovadora para mejorar la entrega de información al turista del santuario, mostrando la historia de cada imagen que reposa en el templo, en especial del Santo al que representa. Este prototipo será accesible para todo visitante que tenga un Smartphone con el sistema operativo Android, donde podrá obtener información relevante de los objetos de interés del templo.

El desarrollo de este prototipo utilizará como base la información del Santuario del Señor de la Salud ubicado en la parroquia Santiago de la provincia Bolívar.

1.1. OBJETIVOS

El objetivo general de este proyecto de titulación es desarrollar una aplicación móvil prototipo utilizando realidad aumentada y el sistema operativo Android para brindar información sobre pinturas y estatuas de santuarios religiosos.

Los objetivos específicos de este proyecto de titulación son:

- Analizar la metodología y las plataformas de desarrollo de aplicaciones multiplataforma de Realidad Aumentada.
- Diseñar los módulos que conforman el prototipo.
- Implementar los módulos diseñados para el prototipo.
- Analizar los resultados de las pruebas de funcionamiento.

1.2. ALCANCE

El presente trabajo de titulación utilizará como base la información del Santuario del Señor de la Salud ubicado en Santiago de Bolívar, parroquia perteneciente al Cantón San Miguel (Provincia Bolívar). El principal atractivo turístico de la parroquia es la iglesia, denominada "Santuario del Señor de la Salud". Santiago, se diferencia con el resto de las parroquias por alojarse en ella la imagen del Señor de la Salud, nombre propicio para atraer a feligreses en busca de sanación física y espiritual; siendo la celebración de su llegada el primer y segundo domingo de julio, donde acuden miles de personas para agradecer los milagros recibidos y a implorar sanación [2]. La parroquia en la actualidad no cuenta con un guía turístico para dar a conocer datos importantes sobre el santuario, ni con leyendas para identificar estatuas o pinturas. Los inconvenientes mencionados de parte de la parroquia provocan que los visitantes de la tierra del Señor de la Salud no cuenten con información relevante de las obras, ni de la historia de la misma.

La información que se desplegará en la aplicación será la siguiente:

Estatuas: Nombre, historia, milagros y la oración perteneciente a cada estatua si la tuviera. Para la imagen del Señor de la Salud, el cual no tiene un libro del cual basarse para su historia, se realizarán entrevistas al menos a tres párrocos que administraron los sacramentos en el Santuario del Señor de la Salud.

Pinturas: Autor e historia.

Iglesia: Historia.

Biografía del Pintor Ivo Mora, para lo cual se realizará una entrevista con el pintor.

La aplicación presentará una arquitectura autónoma (Figura 1.1), es decir la totalidad de las operaciones y tareas del sistema se llevarán a cabo en el terminal del usuario. El reconocimiento utilizando Realidad Aumentada se basa en detección visual mediante Vuforia.

La arquitectura interna estará compuesta por, al menos, los siguientes módulos:

- **Módulo de Realidad Aumentada:** En este módulo se realizarán las configuraciones y el diseño de las escenas para incluir los marcadores de Vuforia e implementarlos en Unity.
- **Módulo de Interacciones con el Usuario:** Este módulo está constituido de varios scripts que permitirán desarrollar el comportamiento de cada elemento que está

conformando la escena, y de las transiciones entre las mismas, además de la programación de eventos de los elementos visuales de la aplicación.

- Módulo de Audio: Este módulo permitirá incluir el audio en la aplicación, ya que la información visualizada por el usuario podrá ser reproducida a través del dispositivo móvil.
- Módulo de Ayuda: Permitirá describir el funcionamiento de la aplicación, y una guía de ayuda al usuario.
- Módulo de Descarga de Aplicación: En este módulo se realizará la configuración del servidor y el portal cautivo necesarios para que el usuario pueda descargar la aplicación.

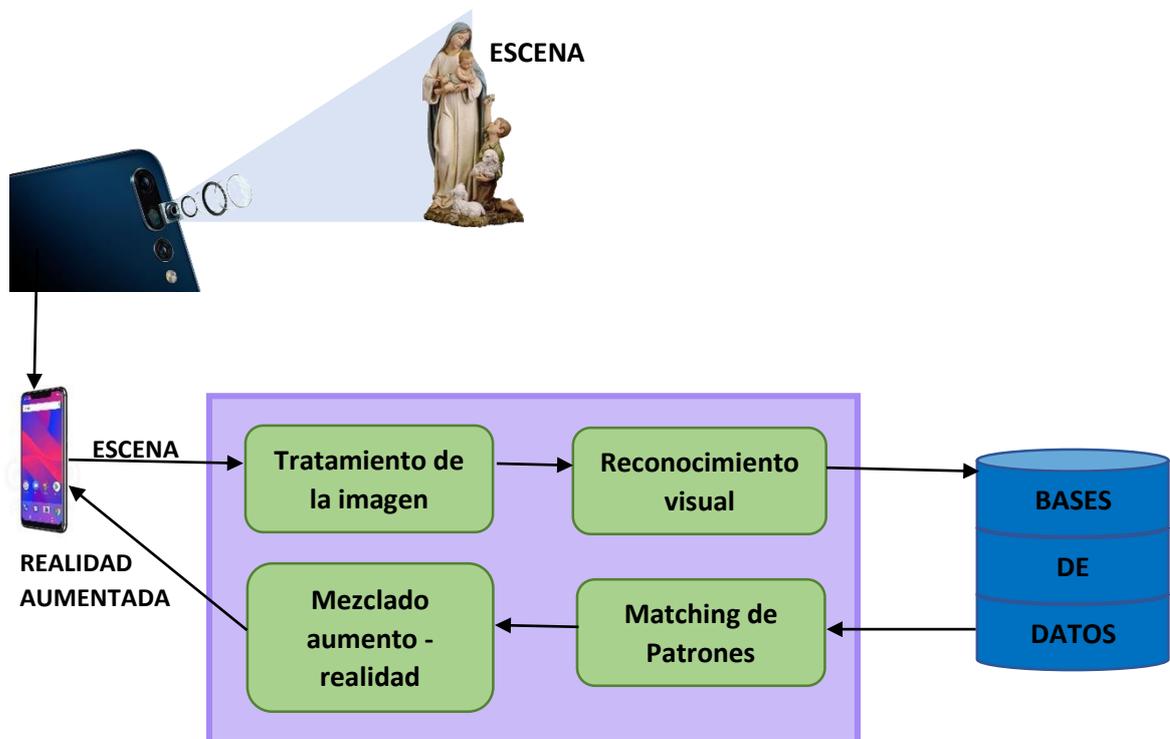


Figura 1.1 Funcionamiento de la arquitectura de la aplicación [3]

La arquitectura autónoma fue seleccionada debido a que ésta no requiere de agentes externos, lo que permite garantizar el servicio ante problemas, como falta de cobertura de Internet. El problema de cobertura está presente en la Parroquia, ya que no hay buena señal de telefonía celular. Esta es la razón por la cual no se encontrará la aplicación en Play Store, sino que mediante un portal cautivo se direccionarán a todos los usuarios a un servidor de descarga donde podrán acceder al APK de la aplicación, tal como se muestra en la **Figura 1.2**.

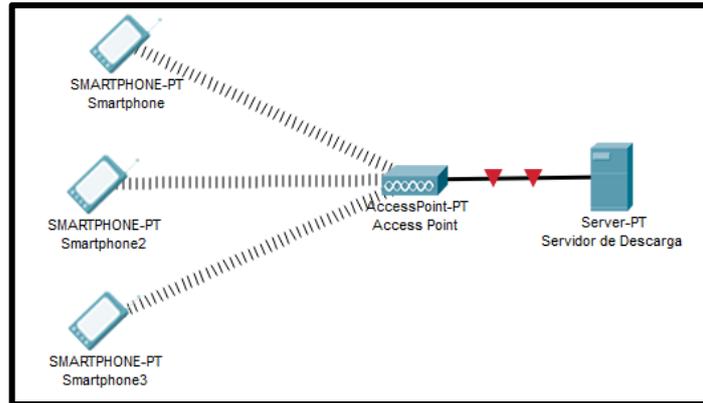


Figura 1.2 Topología para descargar el APK de la aplicación

La aplicación será desarrollada en Android y permitirá reconocer al menos diez imágenes entre estatuas y pinturas de la iglesia, posterior al reconocimiento se abrirá una nueva escena con la descripción de la imagen con la opción de poder escuchar dicha información. Además, el prototipo será desarrollado utilizando una metodología ágil de diseño de software denominada RAD (Rapid Application Development). Esta metodología presenta cuatro etapas tal como se muestra en la Figura 1.3.

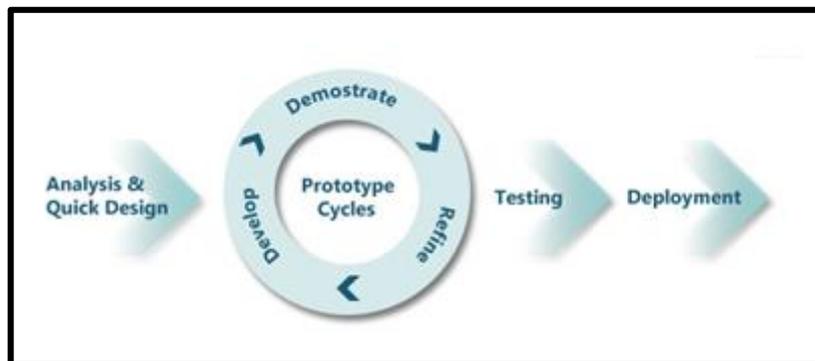


Figura 1.3 Etapas de la metodología RAD [4]

1.3. MARCO TEÓRICO

1.3.1. METODOLOGÍA DE DESARROLLO RÁPIDO DE APLICACIONES

En 1970 fue desarrollada esta metodología por James Martin, como la evolución de las metodologías en cascada. Esta evolución surgió debido a que, en el enfoque tradicional del modelo en cascada los usuarios intervienen únicamente en la fase inicial, luego se ejecutan las etapas siguientes y el usuario únicamente verá la implementación final. Esto provoca un alto costo al efectuar cambios, cuando el usuario cambia los requerimientos [5].

1.3.1.1. Definición

La metodología de desarrollo rápido de aplicaciones o RAD (Rapid Application Development) por sus siglas en inglés, se desarrolló para reducir el tiempo de entrega de sistemas de información. Esta metodología se basa en la creación de prototipos y se utiliza para sistemas de información pequeños o medianos. RAD requiere el uso de prototipos para interactuar con el usuario, debido a que no suelen expresar de forma clara cuáles son sus necesidades y de esta manera poder definir los requisitos del usuario para diseñar el sistema final [6].

1.3.1.2. Ventajas

Las ventajas de la metodología RAD son las siguientes [6] [7] :

- La principal ventaja de RAD es obtener un sistema final que cumpla con los requerimientos del usuario, debido a que el usuario podrá ver lo que está pidiendo paulatinamente hasta que el sistema final esté terminado.
- Reducción de costos de desarrollo.
- Aumenta la calidad del sistema, debido al involucramiento del usuario en la etapa de análisis y de diseño.
- Mayor flexibilidad en el momento de hacer cambios en comparación a otras metodologías.
- Mejor comunicación con el usuario.
- Los usuarios tendrán expectativas más cercanas a la realidad, debido a que el usuario es parte del desarrollo del prototipo y tendrá la oportunidad de establecer dichas expectativas de forma adecuada.

1.3.1.3. Desventajas

Las desventajas de la metodología RAD son las siguientes [6] [7]:

- El usuario al ver los cambios que se pueden hacer en el prototipo, puede generar más de los necesarios en busca del sistema “perfecto”.
- El usuario percibe que el sistema final está terminado dado el último prototipo, pero aún falta otros aspectos, como es el control de errores, entre otros.

1.3.1.4. Fases de la Metodología

La metodología RAD se divide en tres fases tal como se muestra en la Figura 1.4.

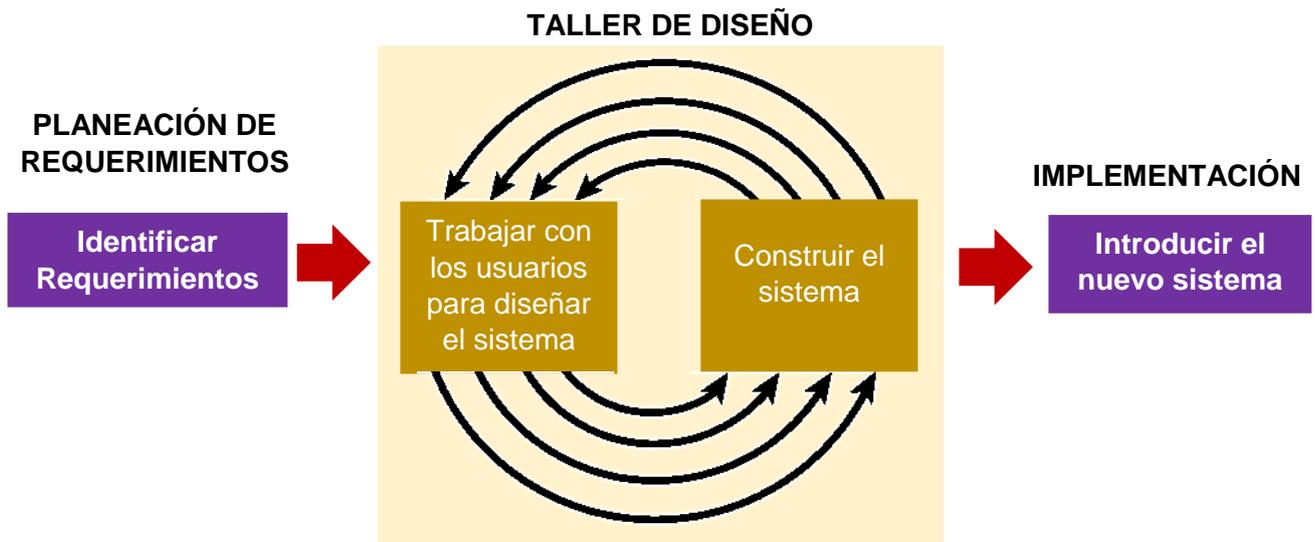


Figura 1.4 Fases de la Metodología RAD [7]

- **Planeación de Requerimientos**

El descubrimiento de los requerimientos, es una de las tareas más importantes y difíciles, esto se debe a que las necesidades del cliente cambian a medida que el proyecto avanza. En esta etapa se requiere la participación del cliente y los analistas [7].

- **Taller de Diseño**

Esta etapa está constituida por un ciclo entre la fase de trabajar con los usuarios y construir el sistema. Los usuarios evalúan los prototipos funcionales, realizan comentarios y los analistas efectúan el cambio, esta fase se repetirá tantas veces como cambios solicite el usuario [7].

- **Implementación**

En esta etapa se ha llegado a un acuerdo de la funcionalidad del sistema, se procede a crear y refinar el sistema para luego realizar pruebas, y finalmente se introduce a la organización [7].

1.3.2. REALIDAD AUMENTADA

Desde hace varios años las tecnologías y las comunicaciones han ido evolucionando, como consecuencia están cambiando la forma de nuestro diario vivir, e incluso la percepción de la realidad. Hace aproximadamente 28 años [8] era imposible obtener información de forma digital de un objeto de la vida real, pero hoy en día se lo puede realizar gracias a la realidad aumentada.

1.3.2.1. Definición

La Realidad Aumentada es una tecnología que permite añadir sobre un escenario real cualquier tipo de contenido digital, puede ser visual, auditivo etc. Esto permite generar nuevas experiencias que aportan conocimiento relevante de nuestro entorno.

1.3.2.2. Funcionamiento

El funcionamiento de la realidad aumentada está constituido básicamente por cuatro tareas tal como se visualiza en la Figura 1.5.



Figura 1.5 Funcionamiento del SRA¹ [3]

- Captura de escenario

La captura de escenario es una de las etapas más importantes del SRA, ya que será el lugar donde se realizará la fusión entre lo virtual y la realidad.

- Identificación de la escena

Es el proceso por el cual se identifica qué escenario del mundo real es el que el usuario desea conocer datos importantes mediante información digital. Este proceso se lo puede realizar básicamente mediante dos formas:

¹ Sistema de Realidad Aumentada

- ❖ Reconocimiento con marcadores².
- ❖ Reconocimiento sin marcadores: El reconocimiento sin marcadores, se refiere a realizarlo a través de imágenes, estimación de posición o una combinación de las dos que sería la forma híbrida.

El presente proyecto utilizará reconocimiento sin marcadores y particularmente el reconocimiento a través de imágenes.

- Realidad + Aumento

Proceso por el cual se agrega la información digital sobre la escena real ya identificada en el paso anterior. Esta información digital puede ser auditivo o visual.

- Visualización

Esta etapa permite consolidar lo que es realidad aumentada, ya que permitirá visualizar la fusión entre lo real y lo virtual.

1.3.2.3. Elementos de un SRA

Los elementos que permiten obtener una experiencia en realidad aumentada son cuatro y están relacionados con las etapas de funcionamiento del SRA, tal como se puede visualizar en la Figura 1.6 [9].

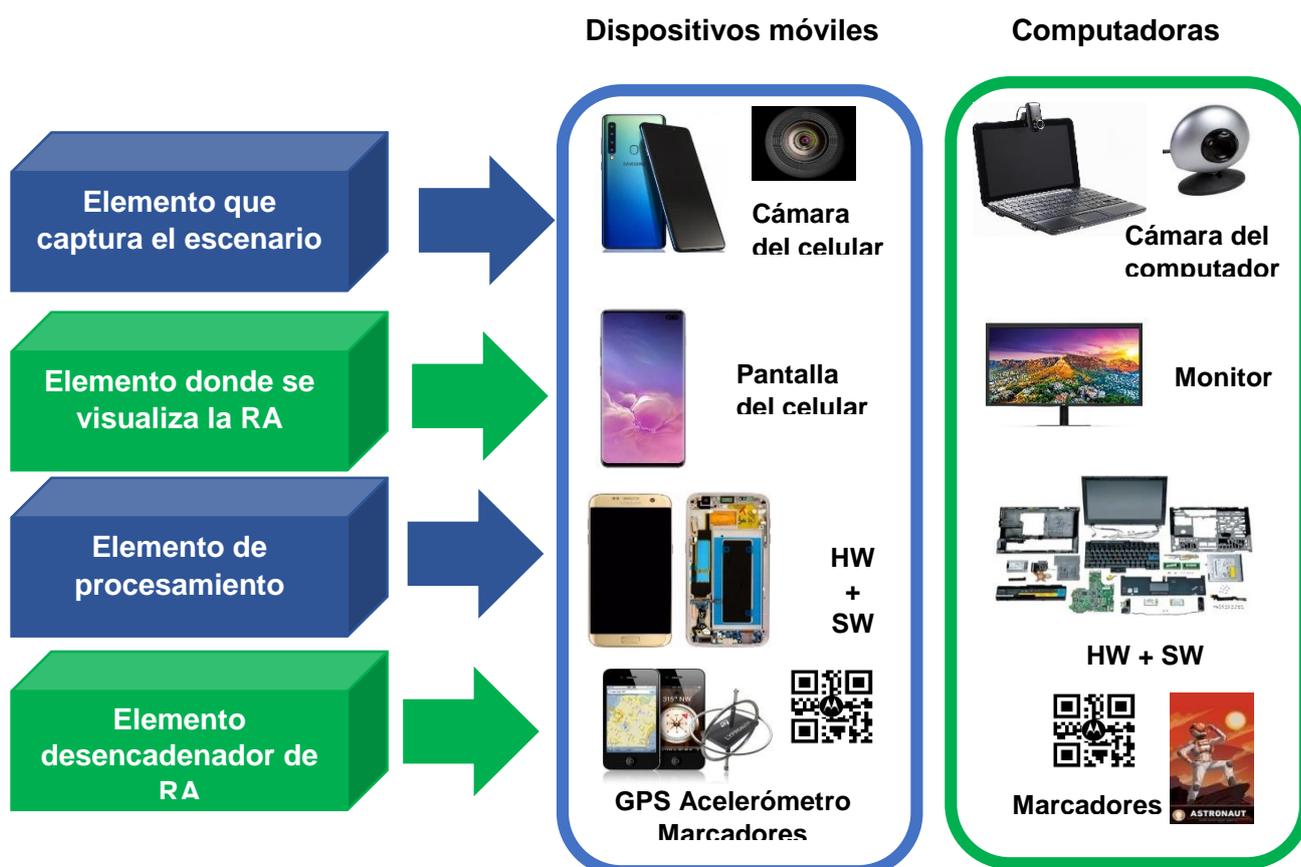


Figura 1.6 Elementos de un SRA [9]

² Un marcador o patrón es una imagen que va a ser procesada mediante realidad aumentada

A continuación, se van a detallar los diferentes elementos que se necesita en un sistema de realidad aumentada:

- En primer lugar, se necesita un elemento que permita capturar la escena, este puede ser la cámara de un dispositivo móvil o de un ordenador.
- Luego, se necesita la pantalla de un dispositivo móvil o de un ordenador para visualizar el escenario real con elementos digitales.
- Se necesita uno o más elementos de procesamiento que trabajen en conjunto para interpretar la escena real, generar los elementos digitales y realizar la unión entre los dos.
- Por último, se necesita un elemento que permita activar la realidad aumentada como, por ejemplo, el GPS del celular, sensores, marcadores, entre otros.

1.3.2.4. Realidad Aumentada y el turismo

Las nuevas tecnologías brindan un valor añadido al lugar que se está visitando, generando nuevas experiencias para el turista. Una de estas nuevas tecnologías es la realidad aumentada, que permite generar aplicaciones que aporten información del lugar que se está visitando [10].

Según el propósito de la aplicación de realidad aumentada para el turismo, se puede clasificar en:

- **Orientadas al patrimonio [10]**

Este tipo de aplicaciones permiten la reconstrucción de un monumento o cualquier pieza arquitectónica, pero se necesita el uso de dispositivos especiales, lo que le hace menos accesible para el turista.

- **Orientadas al guiado y a la información [10]**

Son aplicaciones que brindan información en tiempo real del lugar que se está visitando y se puede acceder por medio de un dispositivo móvil, lo que le hace más accesible que en el caso anterior.

Algunos ejemplos de este tipo de aplicaciones son Wikitude³, Layar⁴, entre otras.

³ www.wikitude.com

⁴ www.layar.com

1.3.2.5. Ventajas de la Realidad Aumentada en el turismo

La inclusión de la realidad aumentada en el turismo presenta grandes beneficios y se están fortaleciendo en diferentes países. Algunas de estas ventajas son las siguientes [10]:

- Permite mejorar la experiencia del turista.
- Sistema interactivo y fácil de utilizar.
- Permite al usuario escoger que tipo de información requiere según el lugar que se encuentre visitando.
- Ayuda a fomentar el turismo, brindando un elemento diferenciador con respecto a otros lugares.
- Permite obtener información del lugar que se está visitando a través de un dispositivo inteligente que lleva consigo cualquier turista al momento de realizar un viaje.
- Facilita la orientación del turista al moverse en una ciudad.

1.3.3. HERRAMIENTAS PARA EL DESARROLLO DE LA APLICACIÓN

Para el desarrollo del presente proyecto de titulación se utilizarán tres herramientas: Vuforia para la parte de realidad aumentada, Unity para el desarrollo de la aplicación y Vegas Pro para la edición de audio.

1.3.3.1. Vuforia

Vuforia Engine es una plataforma de software para crear aplicaciones en realidad aumentada. La plataforma permite el desarrollo de aplicaciones AR⁵ en Android, IOS y UWP⁶ [11] [12].

Las aplicaciones desarrolladas mediante Vuforia permiten [13]:

- Reconocer texto, image targets, model target, entre otros.
- El Target reconocido no se pierda fácilmente cuando la cámara se haya movido.
- Reconocimiento rápido de targets.
- Reconocimiento de varios targets al mismo tiempo.

⁵ Augmented Reality

⁶ Universal Windows Platform

1.3.3.1.1. Arquitectura de Vuforia

La arquitectura de Vuforia está compuesta de los siguientes elementos:

- Cámara: La cámara asegura que la imagen sea captada y sea pasada al Tracker.
- Bloque Tracker: El tracker contiene todos los algoritmos necesarios para detectar las imágenes de referencia o Targets y buscar coincidencias con la base de datos.

Los elementos del bloque tracker están constituidos por la siguiente información:

- Tipo: Unknown_Type, Image_target, multi_target, entre otros [14].
- Identificador: Nombre único que lo identifica dentro de la base de datos, con un tamaño máximo de 64 caracteres [14].
- Información de Estado: Esta información se encuentra en el objeto de estado, los posibles estados que puede tomar un target son DETECTED (El target fue detectado, estado transitorio), TRACKED (El Target está siendo rastreado y está funcionando normalmente), EXTENDED TRACKER (El Target está fuera de vista, demasiado cerca o lejos para ser rastreado directamente), LIMITED (El Target está siendo rastreado con baja precisión) [15].
- Datos de Usuario: Datos genéricos definidos por usuario asociado con el rastreo.

Vuforia soporta dos tipos de bases de datos [16]:

- Device Databases: Es una base de datos local que se almacena en el dispositivo junto con la aplicación.
- Cloud Databases: Los Image Targets se encuentran almacenados en una base de datos en la nube, puede contener hasta 1 millón de Image Targets.

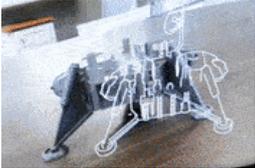
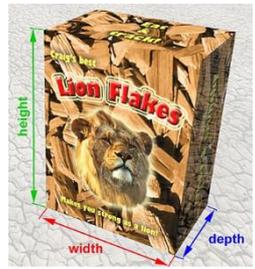
Vuforia permite reconocer una variedad de imágenes y modelos como:

- Model Target.
- Image Target.

- Multi Targets.
- Cylinder Targets.
- Vumarks.

La definición, ejemplos y prerequisites de cada uno de ellos se puede encontrar en la Tabla 1.1.

Tabla 1.1 Targets de reconocimiento de Vuforia [17]

Targets	Definición	Ejemplos	Prerrequisitos	
Model Target	Permite que las aplicaciones puedan reconocer objetos particulares del mundo real en función de la forma del objeto.	Equipos, juguetes, autos, etc.	Obtener el modelo 3D del objeto.	
			El objeto debe ser geoméricamente rígido.	
			No debe tener superficies brillantes.	
Image Target	Son imágenes que vuforia puede reconocer, se basa en las características naturales que posee la imagen.	Empaque de un producto e imágenes ricas en detalles.	Las imágenes pueden ser JPG o PNG.	
			Las imágenes pueden ser RGB o en blanco y negro.	
			El tamaño debe ser menos de 2MB.	
Multi Targets	Son múltiples imágenes que están ubicadas en una forma geométrica definida. Todas las caras de un Multi-Target se pueden rastrear al mismo tiempo.	Marketing, juegos y visualización de productos.	Las imágenes pueden ser JPG o PNG.	
			Las imágenes pueden ser RGB o en blanco y negro.	
			El tamaño debe ser menos de 2MB.	
			Los objetivos deben estar bajo iluminación.	
Cylinder Targets	Permiten detectar imágenes en formas cilíndricas y cónicas. También permite detectar las caras planas del cilindro.	Envases de refrescos, café, etc.	Definir la longitud lateral y diámetro del objetivo.	
			Agregar las imágenes que se desea detectar.	
			Las imágenes pueden ser JPG o PNG.	
			Las imágenes pueden ser RGB o en blanco y negro.	
			El tamaño debe ser menos de 2MB.	
Los objetivos deben estar bajo iluminación.				

Vumarks	Son la evolución de los códigos de barra, son personalizables por lo que se puede tener un Vumark único para cada objeto.	Mercado empresarial, poner instrucciones 3D.	Las imágenes pueden ser JPG o PNG.	
			Las imágenes pueden ser RGB o en blanco y negro.	
			Los objetivos deben estar bajo iluminación.	

Para el desarrollo del presente proyecto se va a utilizar Image Targets que corresponderán a las imágenes y pinturas presentes en el santuario.

- Renderizador de video de fondo: Se procesa la información que se encuentra en el objeto de estado.
- Código de Aplicación: Por cada imagen reconocida, el objeto de estado se actualiza y se llama al método de procesamiento de la aplicación [18].

1.3.3.2. Unity

Unity es una herramienta de desarrollo multiplataforma que permite crear videojuegos y aplicaciones. Además, permite trabajar con diferentes lenguajes de programación [19].

Unity presenta seis ventanas tal como se muestra en la Figura 1.7.



Figura 1.7 Ventana principal de Unity

A continuación, se definirá cada una de las ventanas que está conformada la ventana principal de Unity:

- **Ventana del proyecto (Project Window)**

Esta ventana permite gestionar los Assets⁷ que están incluidos en el proyecto. En la parte izquierda muestra la estructura de las carpetas en forma de jerarquía.

- **Ventana de escena (Scene View)**

Es una ventana interactiva que permite posicionar los diferentes elementos que va a estar constituido el proyecto.

- **Ventana de jerarquía (Hierarchy Window)**

Esta ventana permite visualizar la jerarquía o parentesco que tienen los GameObjects⁸ dentro del proyecto.

- **Ventana del inspector (Inspector Window)**

Esta ventana permite ver y modificar las propiedades de cada uno de los GameObjects presentes en el proyecto.

- **Barra de herramientas (Toolbar)**

La barra de herramientas está constituida por cinco controles básicos que están relacionados con las ventanas del editor de Unity.

- **Ventana de juego (Game View)**

Esta ventana es renderizada por las cámaras presentes en el proyecto. Esta ventana permite realizar pruebas del juego o aplicación.

1.3.3.3. Vegas Pro

Vegas Pro o Sony Vegas como se lo denominaba hace tiempo atrás, es una herramienta de edición de audio y video no lineal [20].

Características

- Proporciona un entorno de audio multipista.
- Importa y exporta en una amplia gama de formatos.

⁷ Es un archivo de audio, imagen, texto, etc. creado dentro o fuera de Unity que pertenece al proyecto.

⁸ Un Game Object es cualquier objeto presente en el proyecto

- Selector de capas (profundidad gráficos).
- Variedad de efectos de audio.
- Software pagado.

Vegas Pro es un software pagado, pero se va a trabajar con la versión de prueba.

1.3.4. PORTAL CAUTIVO

1.3.4.1. Definición

El portal cautivo es una herramienta de seguridad utilizada por redes inalámbricas para autenticación o presentar información como políticas de uso para poder acceder a los recursos de la red [21] [22] [23].

1.3.4.2. Tipos de Portales Cautivos

Los portales cautivos se clasifican por software y por hardware. En la Figura 1.8 se puede ver la clasificación y ejemplos de cada uno de ellos [22].

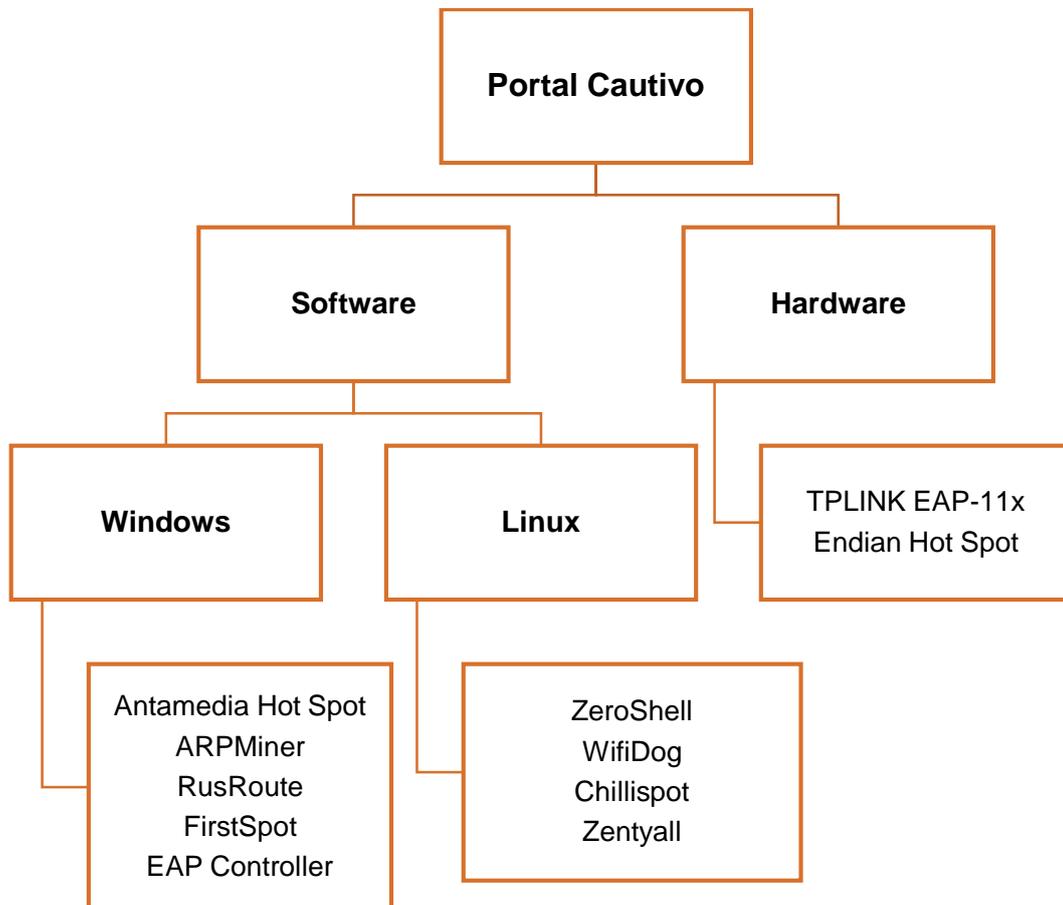


Figura 1.8 Tipos de Portales Cautivos [24] [25] [26] [27] [28]

- **Portales Cautivos por Software**

Los portales cautivos por software son programas que se instalan en una computadora, la cual se conecta con el Access Point y la salida de Internet.

Existe una variedad de portales cautivos por software para Linux y Windows, tal como se muestra en la Figura 1.8.

- **Portales Cautivos por Hardware**

Son dispositivos que vienen integrados con la funcionalidad de un portal cautivo, no se necesitan de dispositivos adicionales.

1.3.4.3. Ventajas

Las ventajas de utilizar portales cautivos son las siguientes [23] [22]:

- Independiente de la plataforma.
- No se requiere configuración en el lado del cliente.
- Se puede utilizar cualquier navegador para la presentación del portal cautivo.
- Permite dar un límite de tiempo para el uso de los recursos de la red a los usuarios que deseen conectarse.

2. METODOLOGÍA

Este capítulo consta de tres secciones: la situación actual del santuario, la fase de diseño y la fase de implementación. La situación actual del santuario, detalla las pinturas e imágenes que están presentes en el santuario. La fase de diseño, la cual consta del análisis de requerimientos y diagramas UML de comportamiento. La fase de implementación, donde se realiza la codificación y configuración del AP y servidor.

En este capítulo se utiliza el término de escena debido a que Unity se desarrolla en base a escenas y de esta manera fue creada la aplicación. Los módulos y las escenas están relacionados de la siguiente manera:

- Módulo de Realidad Aumentada con la escena EscenaRA.
- Módulo de Interacciones con el Usuario con las escenas: Menu, MenuRA y EscenaDeCarga.
- Módulo de Audio con la escena Información.
- Módulo de Ayuda con la escena Ayuda.

2.1. SITUACIÓN ACTUAL

El templo de la parroquia Santiago, ubicado en la provincia de Bolívar, fue declarada Santuario Diocesano del Señor de la Salud el 4 de Julio de 1976 en la ciudad Episcopal de Guaranda [29]. El Santuario presenta en su interior 20 estatuas y 39 pinturas, tal como se indica en el Anexo A y B respectivamente.

El Santuario del Señor de la Salud, por falta de recursos económicos no cuentan con un guía turístico para dar a conocer datos importantes sobre el origen del mismo y el Santo al que representa, lo que hace de la visita una experiencia limitada por la falta de automatización para brindar información, a todos aquellos que hacen paso por el templo religioso de la parroquia Santiago. El templo tampoco cuenta con leyendas para identificar a las estatuas, ni información sobre las pinturas que reposan en ella, provocando que los visitantes no cuenten con información relevante de las obras, ni de la historia de la misma.

2.2. FASE DE DISEÑO

2.2.1. FASE DE PLANEACIÓN DE REQUERIMIENTOS

Para determinar los requerimientos del prototipo, se realizó una entrevista tal como se adjunta en el Anexo C, esta entrevista fue dirigida al presidente, vocal y secretario del GAD-PR de Santiago y al Párroco del Santuario del Señor de la Salud.

La entrevista está conformada por siete preguntas las cuales tiene como objetivo confirmar las necesidades que presenta el santuario e identificar los requerimientos para el desarrollo del prototipo.

2.2.1.1. Requerimientos Funcionales

Los requerimientos funcionales describen lo que el prototipo debe realizar, estos dependerán del tipo de la aplicación y a quién va dirigida [30].

Los requerimientos funcionales se obtuvieron en base a la entrevista y al alcance del proyecto, mismo que son los siguientes:

- El prototipo permitirá reconocer al menos 10 imágenes, para brindar información de su historia, milagros u oración dependiendo si es una pintura o una estatua.
- El prototipo permitirá acceder a información adicional como la historia de la iglesia y la biografía del pintor Ivo Mora.
- El prototipo tendrá la opción de reproducir los audios correspondientes a la información visualizada de la imagen reconocida.
- El prototipo permitirá ver el nombre de la imagen en tiempo real al enfocar la misma con la cámara del dispositivo móvil.
- El prototipo mostrará información de ayuda relacionada con el funcionamiento de la aplicación.

2.2.1.2. Requerimientos No Funcionales

Los requerimientos no funcionales describen el rendimiento, facilidad de uso, presupuestos, tiempo de entrega, entre otros [30].

Los requerimientos no funcionales del prototipo son los siguientes:

- El prototipo deberá funcionar para dispositivos Android.
- La arquitectura del prototipo debe ser autónoma.
- El tipo de patrón de imágenes que se utilizará para el reconocimiento de pinturas e imágenes dentro del santuario, se las realizará mediante las características naturales de la misma, es decir se utilizará Image Targets.
- La base de datos será del tipo Device Database y será desarrollada mediante Vuuforia.
- La aplicación será desarrollada en Unity.

- El prototipo no dependerá de la conexión a Internet para su funcionamiento.
- El usuario podrá acceder a la página web donde está el APK de la aplicación mediante un portal cautivo del tipo hardware.

2.2.2. HISTORIAS DE USUARIO

Las historias de usuario son utilizadas para describir las necesidades del usuario para el desarrollo de un producto en software [31].

2.2.2.1. Formato

El formato de las historias de usuario se encuentra en la Tabla 2.1 y el detalle de cada uno de los campos se muestra a continuación:

- Fecha: Este campo permite especificar la fecha en la que fue realizada la historia de usuario.
- Número: Identificación de las historias de usuario.
- Tipo: Este campo identifica el tipo de historia de usuario. Se han identificado dos:
 - Normal: Historia de usuario realizada en la fase de planeación de requerimientos.
 - Extraordinaria: Historia de usuario realizada en base al formulario de evaluación del prototipo en la fase de Taller de Diseño de RAD.
- Prioridad: Este campo indica el nivel de importancia de la historia de usuario, este campo puede tomar el valor de alta media o baja.
- Nombre Historia: Este campo lleva el nombre descriptivo de la historia de usuario.
- Descripción: Describe la historia de usuario.

Tabla 2.1 Formato de las Historias de Usuario [7]

HISTORIAS DE USUARIO	
Nombre Historia:	Fecha:
	Número:
Prioridad:	Tipo:
Descripción	

2.2.2.2. Detalle de Historias de Usuario

Las historias de usuario en base a los requerimientos enumerados anteriormente son los siguientes:

Tabla 2.2 Historia de Usuario: Reconocimiento de imágenes

HISTORIAS DE USUARIO	
Nombre Historia: Reconocimiento de imágenes	Fecha: 18/10/2019 Número: 1
Prioridad: Alta	Tipo: Normal
Descripción	
El prototipo permitirá reconocer al menos 10 imágenes, para brindar información de su historia, milagros u oración dependiendo si es una pintura o una estatua.	

Tabla 2.3 Historia de Usuario: Información Adicional

HISTORIAS DE USUARIO	
Nombre Historia: Información adicional	Fecha: 18/10/2019 Número: 2
Prioridad: Media	Tipo: Normal
Descripción	
El prototipo permitirá acceder a información adicional como la historia de la iglesia y la biografía del pintor Ivo Mora.	

Tabla 2.4 Historia de Usuario: Audio

HISTORIAS DE USUARIO	
Nombre Historia: Audio	Fecha: 18/10/2019 Número: 3
Prioridad: Alta	Tipo: Normal
Descripción	
El prototipo tendrá la opción de reproducir los audios correspondientes a la información visualizada de la imagen reconocida.	

Tabla 2.5 Historia de Usuario: Visualización en tiempo real

HISTORIAS DE USUARIO		
Nombre Historia: Visualización en tiempo real		Fecha: 18/10/2019 Número: 4
Prioridad:	Alta	Tipo: Normal
Descripción		
El prototipo permitirá ver el nombre de la imagen en tiempo real al enfocar la misma con la cámara del dispositivo móvil.		

Tabla 2.6 Historia de Usuario: Instrucciones de la aplicación

HISTORIAS DE USUARIO		
Nombre Historia: Instrucciones de la aplicación		Fecha: 18/10/2019 Número: 5
Prioridad:	Media	Tipo: Normal
Descripción		
El prototipo mostrará información de ayuda relacionada con el funcionamiento de la aplicación.		

2.2.3. CASOS DE USO

El diagrama de casos de uso permite representar la forma en la que el actor o actores utilizan el sistema en desarrollo [32].

2.2.3.1. Descripción de los Casos de Uso

Para este caso en específico se tendrá un solo actor denominado usuario, el cual podrá activar la cámara de su dispositivo, ver información de pinturas y estatuas, entre otras, tal como se muestra en la Figura 2.1.

En la Tabla 2.7 se muestra el formato para la descripción de los casos de uso y el detalle de cada uno de los campos se muestra a continuación:

- Nombre: Nombre del caso de uso.
- Descripción: Breve descripción del caso de uso.

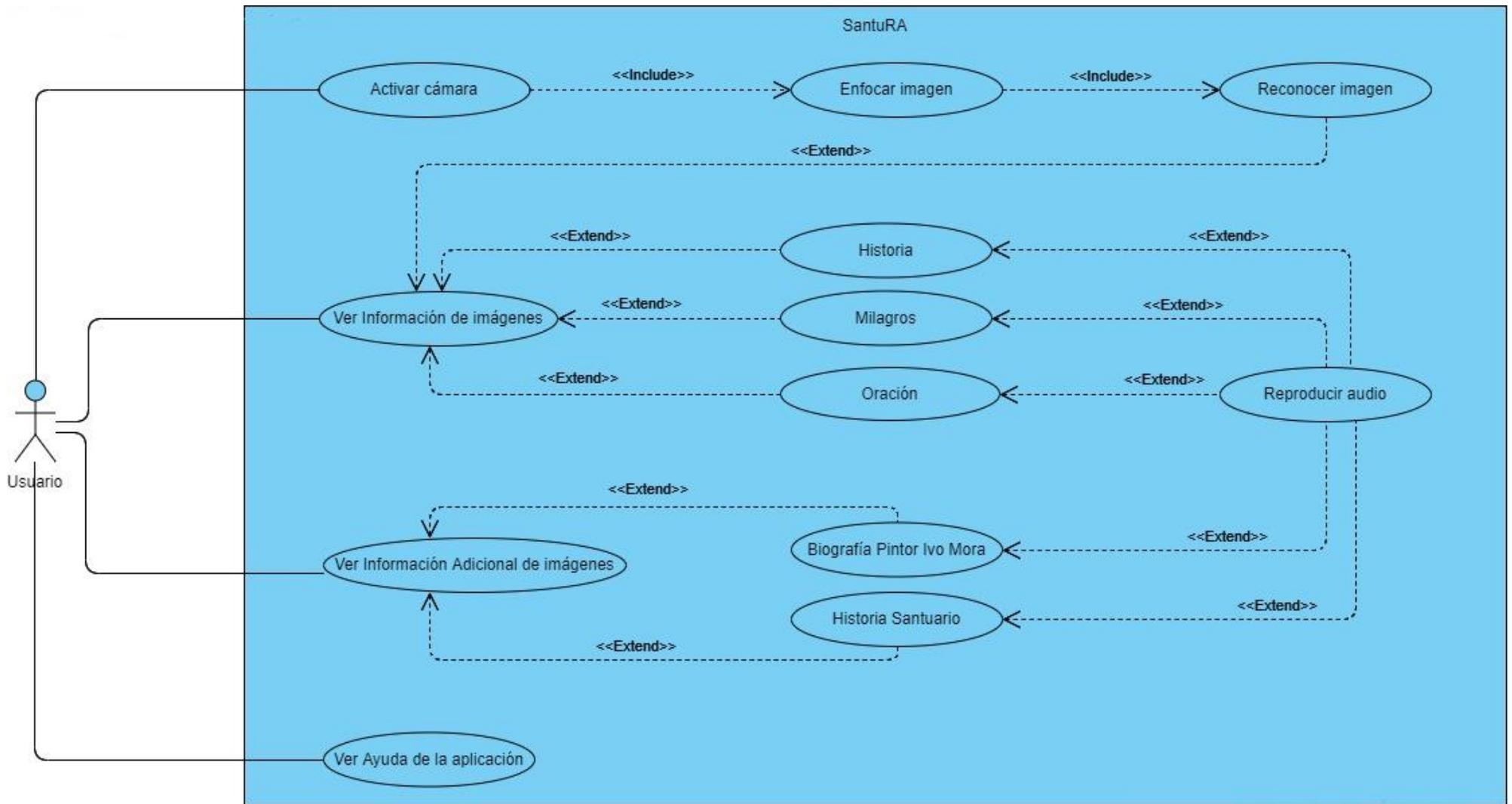


Figura 2.1 Diagrama de casos de uso

- Actor: Elemento externo que interactúa con el sistema.
- Precondición: Es la condición o condiciones en la que debe estar el sistema para poder iniciar con el caso de uso.
- Activación: Evento que origina el caso de uso.
- Postcondición: Es la condición o condiciones en la que se queda el sistema una vez que el caso de uso haya finalizado con éxito.

Tabla 2.7 Formato de la Tabla de descripción de casos de uso

CASOS DE USO	
Nombre	
Descripción	
Actor	
Precondiciones	
Activación	
Postcondición	

La descripción del diagrama de casos de uso está organizada en doce tablas (Tabla 2.8– 2.19) con el formato que se muestra en la Tabla 2.7. A continuación, se presentan las tablas que describen a los casos de uso:

- Activar cámara.
- Enfocar imagen.
- Reconocer imagen.
- Ver información de imágenes.
- Ver Historia.
- Ver Milagros.
- Ver Oración.

- Reproducir audio.
- Ver información adicional de imágenes.
- Biografía Pintor Ivo Mora.
- Historia del Santuario.
- Ver ayuda de la aplicación.

Tabla 2.8 Caso de Uso: Activar cámara

CASO DE USO	
Nombre	Activar cámara
Descripción	Activa la cámara del dispositivo móvil donde fue instalada la aplicación.
Actor	Usuario
Precondiciones	Instalar aplicación en el dispositivo móvil.
Activación	Click en Iniciar en el menú principal.
Postcondición	Enfocar una imagen dentro del santuario.

Tabla 2.9 Caso de Uso: Enfocar imagen

CASO DE USO	
Nombre	Enfocar imagen
Descripción	Se deberá enfocar una imagen dentro del santuario para obtener la información requerida.
Actor	Usuario
Precondiciones	La cámara debe estar activa.
Activación	Cuando la cámara esté activada.
Postcondición	Reconocer la imagen.

Tabla 2.10 Caso de Uso: Reconocer imagen

CASO DE USO	
Nombre	Reconocer imagen
Descripción	La imagen después de ser enfocada podrá ser reconocida.
Actor	Usuario
Precondiciones	La imagen debe estar enfocada. La imagen debe estar en la base de datos de Vuforia.
Activación	Enfocar imagen.
Postcondición	Se habilita el botón de información y se muestra el nombre de la imagen reconocida.

Tabla 2.11 Caso de Uso: Ver Información de imágenes

CASO DE USO	
Nombre	Ver Información de imágenes
Descripción	Permite visualizar la información de la imagen reconocida.
Actor	Usuario
Precondiciones	El botón información debe estar habilitado.
Activación	Presionar el botón información.
Postcondición	Se abrirá un menú dónde podrá escoger el tipo de información que desea visualizar.

Tabla 2.12 Caso de Uso: Ver Historia

CASO DE USO	
Nombre	Ver Historia
Descripción	Permite visualizar la historia de la pintura o escultura que haya sido reconocida.
Actor	Usuario
Precondiciones	
Activación	Presionar el botón historia.
Postcondición	Se abrirá una escena con la historia de la pintura o escultura reconocida.

Tabla 2.13 Caso de Uso: Ver Milagros

CASO DE USO	
Nombre	Ver Milagros
Descripción	Permite visualizar los milagros de la escultura que haya sido reconocida.
Actor	Usuario
Precondiciones	
Activación	Presionar el botón milagros.
Postcondición	Se abrirá una escena con los milagros de la escultura reconocida.

Tabla 2.14 Caso de Uso: Ver Oración

CASO DE USO	
Nombre	Ver Oración
Descripción	Permite visualizar la oración de la escultura que haya sido reconocida.
Actor	Usuario
Precondiciones	
Activación	Presionar el botón oración.
Postcondición	Se abrirá una escena con la oración de la escultura reconocida.

Tabla 2.15 Caso de Uso: Reproducir audio

CASO DE USO	
Nombre	Reproducir audio
Descripción	Permite reproducir el audio correspondiente de la información visualizada.
Actor	Usuario
Precondiciones	
Activación	Presionar el botón play.
Postcondición	Se reproducirá el audio correspondiente.

Tabla 2.16 Caso de Uso: Ver información adicional de imágenes

CASO DE USO	
Nombre	Ver Información adicional
Descripción	Permite ver la información adicional que proporciona la aplicación como es la historia del santuario y la biografía del pintor Ivo Mora.
Actor	Usuario
Precondiciones	Instalar la aplicación en el dispositivo móvil
Activación	Presionar el botón información adicional.
Postcondición	Se abrirá un menú donde podrá escoger la información que desea visualizar.

Tabla 2.17 Caso de Uso: Biografía Pintor Ivo Mora

CASO DE USO	
Nombre	Biografía Pintor Ivo Mora
Descripción	Permite ver la biografía del pintor Ivo Mora.
Actor	Usuario
Precondiciones	
Activación	Presionar el botón Pintor Ivo Mora.
Postcondición	Se abrirá una escena con la biografía del pintor Ivo Mora.

Tabla 2.18 Caso de Uso: Historia del santuario

CASO DE USO	
Nombre	Historia del Santuario
Descripción	Permite ver la historia del santuario.
Actor	Usuario
Precondiciones	
Activación	Presionar el botón Historia Santuario.
Postcondición	Se abrirá una escena con la historia del santuario.

Tabla 2.19 Caso de Uso: Ver Ayuda

CASO DE USO	
Nombre	Ver Ayuda
Descripción	Permite ver las instrucciones para el uso de la aplicación.
Actor	Usuario
Precondiciones	Instalar la aplicación en el dispositivo móvil.
Activación	Presionar el botón ayuda.
Postcondición	Se abrirá una escena con la información del uso de la aplicación.

2.2.4. DIAGRAMAS DE SECUENCIA

El diagrama de secuencia es un diagrama que permite identificar el comportamiento de un sistema [33].

Los diagramas de secuencia correspondientes al prototipo están realizados en base a las diferentes escenas que está conformado el mismo, tal como se muestra en la Figura 2.2, Figura 2.4 y Figura 2.3.

El prototipo está conformado por seis escenas las cuales se detalla a continuación:

- Menú: Escena que tendrá el menú principal de la aplicación, esta es la primera escena que podrá visualizar el usuario.
- Ayuda: Escena en donde se mostrarán las instrucciones necesarias para utilizar la aplicación.
- EscenaRA (Escena Realidad Aumentada): Escena que activa la cámara del dispositivo móvil para enfocar a la estatua o pintura que se desea conocer más información.
- MenuRA (Menú Realidad Aumentada): Escena que tendrá el menú originado después de identificar la imagen.
- Información: Escena que mostrará la información dependiendo de lo que el usuario haya escogido en la escena MenuRA.
- EscenaDeCarga: Escena que tendrá un contador y al finalizar se visualizará la escena EscenaRA

La Figura 2.2 muestra la interacción entre el usuario y las escenas Menú y Ayuda. La escena Menú se activa cuando se inicia la aplicación y la escena Ayuda se muestra cuando el usuario presiona el botón de instrucciones en la escena Menú.

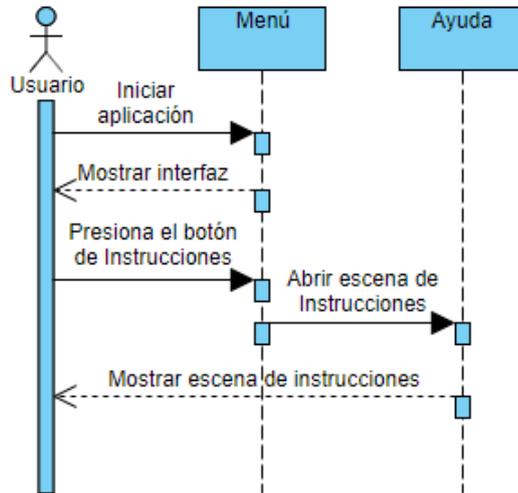


Figura 2.2 Diagrama de Secuencia para Escena de Menú y Ayuda

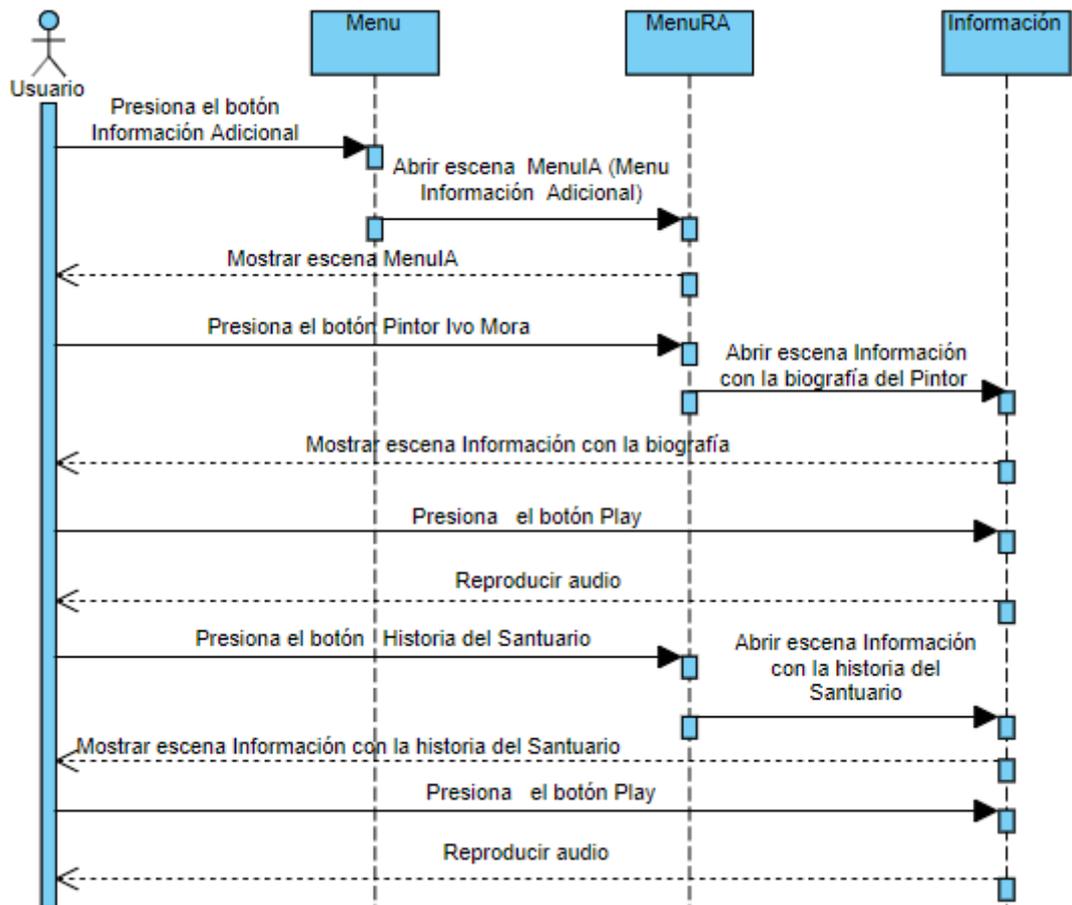


Figura 2.3 Diagrama de Secuencia para Escena de Menú, MenuRA e Información

La Figura 2.3 indica la interacción entre el usuario y las escenas Menú, MenuIA, e Información. En la escena Menú, el usuario presiona el botón Información Adicional el cual abre la escena MenuIA. La escena MenuIA permite acceder a la información del pintor y de la historia del Santuario, para lo cual presionará el botón correspondiente la información que desea visualizar. Luego de presionar el botón se abrirá la escena Información con los datos correspondientes y tendrá la opción de escuchar el audio correspondiente a esa información, presionando el botón play que se encuentra en esta escena.

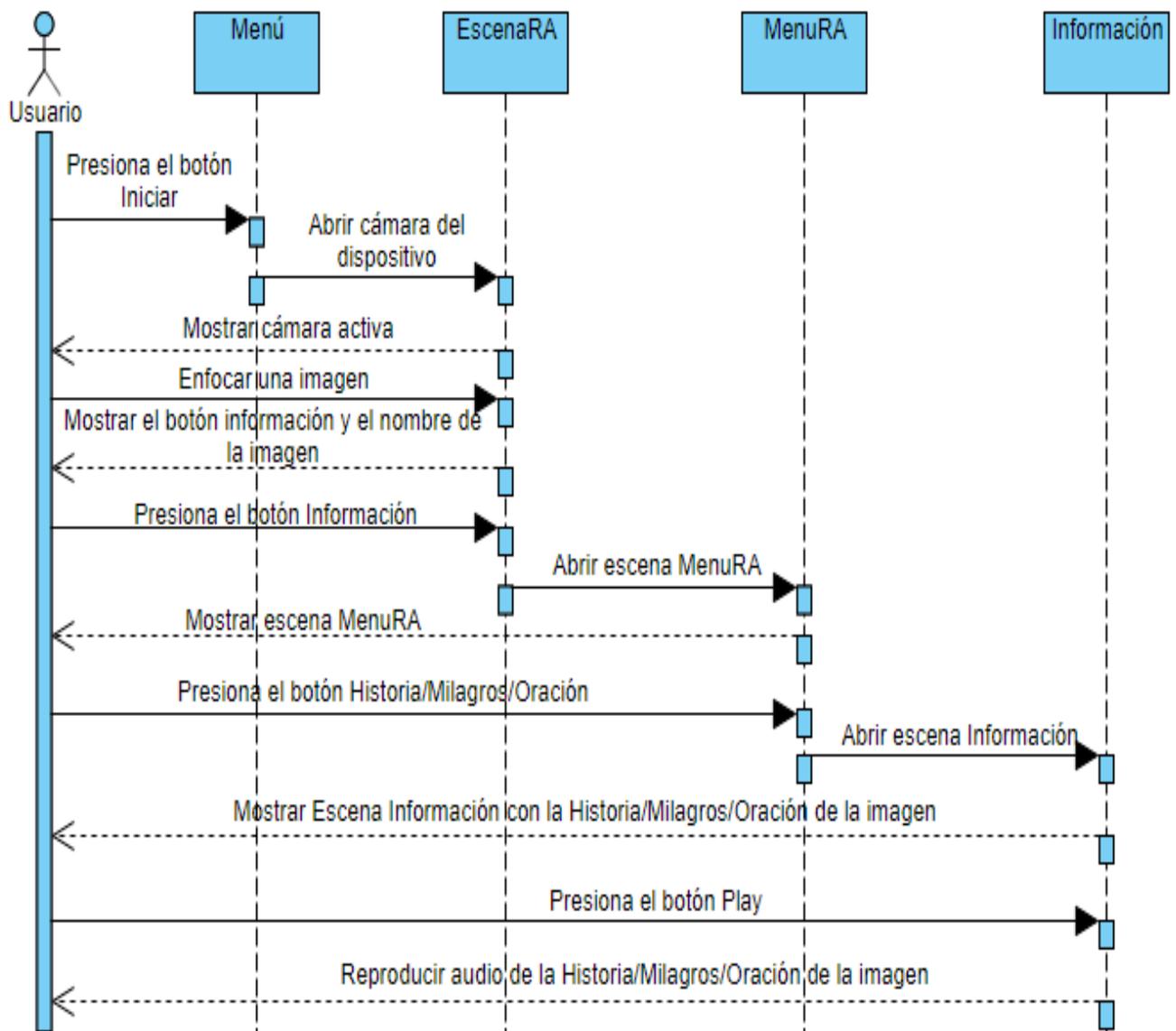


Figura 2.4 Diagrama de Secuencia para Escena de Menú, EscenaRA, MenuRA e Información

La Figura 2.4 muestra la interacción entre el usuario y las escenas Menú, EscenaRA, MenuRA e Información. La escena EscenaRA activa la cámara del dispositivo móvil cuando el usuario presiona el botón Iniciar de la escena Menú, una vez activada la cámara, el usuario debe enfocar una imagen y se mostrará el botón de información y el nombre de la imagen que ha enfocado. Al presionar el botón de información se abrirá la escena MenuRA, en esta escena el usuario puede presionar tres botones (Historia, Milagros u oración) que corresponden al tipo de información que desea visualizar de la imagen reconocida. La escena Información, se mostrará cuando se haya presionado el botón Milagros/Historia/Oración, la cual tendrá la información que el usuario solicitó; además podrá reproducir un audio que corresponderá a la información que está visualizando en ese momento, haciendo click en el botón play de la escena.

2.2.5. SKETCHES DE INTERFACES DE USUARIO

Los sketches de interfaces de usuario del prototipo están conformados por seis escenas:

- Menú.
- EscenaRA.
- MenuRA.
- Información.
- Ayuda.
- EscenaDeCarga.

2.2.5.1. Escena “Menu”

Esta es la escena principal cuando se ha instalado la aplicación (Figura 2.5). Esta escena está conformada por cuatro botones:

- Botón Iniciar: Este botón permite acceder al Módulo de Realidad aumentada, es decir abrir la escena “EscenaRA”.
- Botón Información Adicional: Este botón permite abrir la escena “MenuRA” (Menú Realidad Aumentada).
- Botón Ayuda: Este botón permite abrir la escena de Ayuda.
- Botón Salir: Este botón permite salir de la aplicación.



Figura 2.5 Escena Menú

2.2.5.2. Escena “EscenaRA”

Esta escena se abre al dar click en el botón Iniciar en el menú principal, la cual activa la cámara del dispositivo móvil. Esta escena está conformada por dos botones y un Text, tal como se muestra en la Figura 2.6:

- Botón Información: Este botón permite acceder a la Escena MenuRA.
- Botón Regresar: Este botón permite regresar a la Escena Menu.
- Text: Este elemento permitirá visualizar el nombre de la imagen que ha sido reconocida por la aplicación.

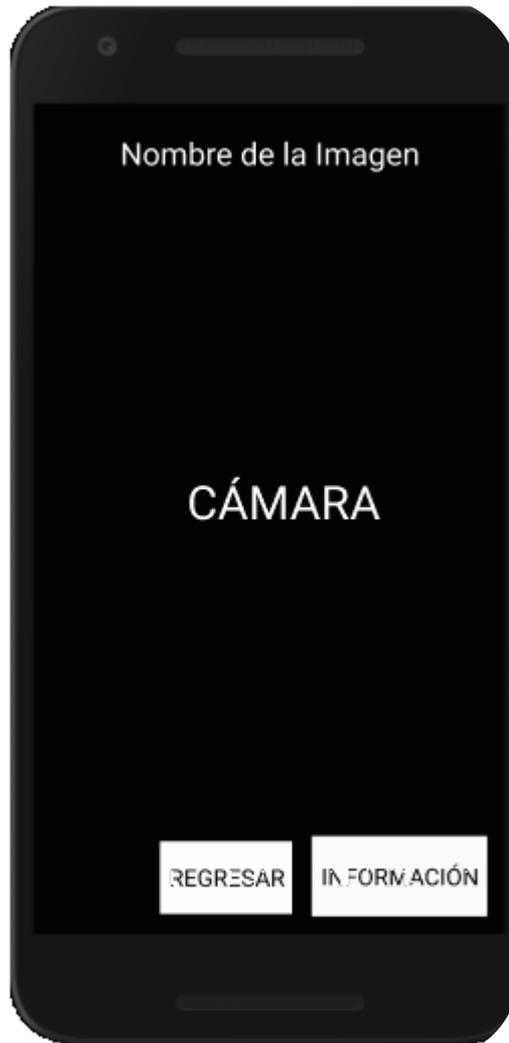


Figura 2.6 Escena EscenaRA

2.2.5.3. Escena “MenuRA”

Esta escena será utilizada para mostrar el menú de Información Adicional y el menú para escoger el tipo de información que el usuario desee visualizar de la imagen reconocida, para ello se ocultarán botones y se cambiarán el nombre mediante código.

Esta escena se abrirá al dar click en el botón Información en la escena EscenaRA o al dar click en el botón Información Adicional de la escena Menu. Esta escena está conformada por los siguientes elementos (Ver Figura 2.7):

- Text: Este Text permitirá visualizar el nombre de la imagen o el título “Información adicional” si se presionó el botón información o el botón Información Adicional respectivamente.

- Botón 1: Este botón permitirá abrir la escena información con la historia de la imagen que ha sido reconocida o con la biografía del pintor Ivo Mora.
- Botón 2: Este botón permitirá abrir la escena información con los milagros de la imagen que ha sido reconocida o con la historia de la iglesia y para pinturas el botón estará oculto.
- Botón Oración: Este botón permitirá abrir la escena información con la oración de la escultura que ha sido reconocida, para pinturas el botón estará oculto
- Botón Regresar: Este botón permite regresar a la escena EscenaRA o a la escena Menu.

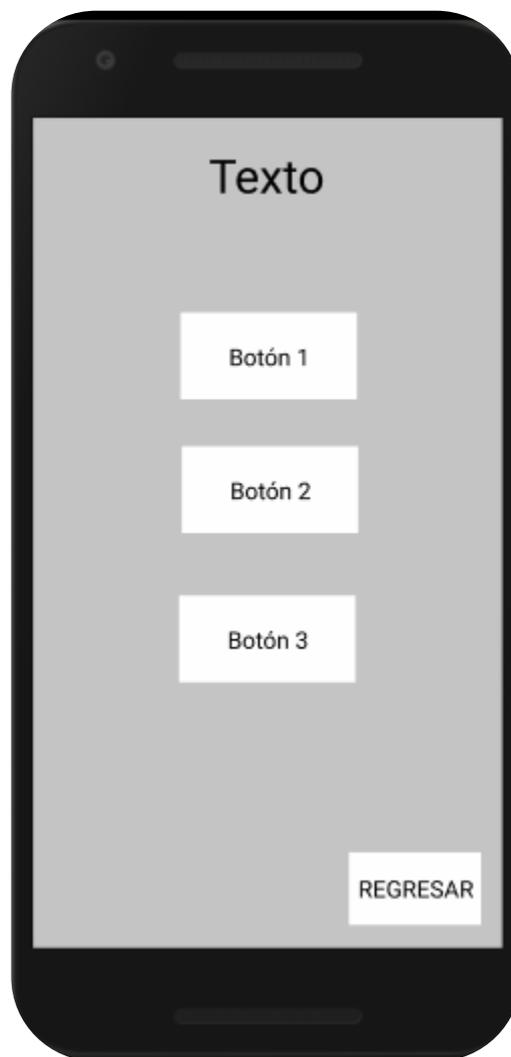


Figura 2.7 Escena MenuRA

2.2.5.4. Escena “Información”

Esta escena se abre al dar click en los botones de Botón 1, Botón 2 o Botón 3 de la Escena MenuRA que corresponden a los botones Historia, Milagros y Oración si la escena se abrió al dar click en el botón Información de la escena EscenaRA, o a los botones Biografía Ivo Mora e Historia de la Iglesia si la escena se abrió al dar click en el botón Información Adicional en la escena Menu.

Esta escena está conformada por ocho elementos, tal como se visualiza en la Figura 2.8 y son los siguientes:

- Text (Título): Este Text permitirá ver el título de la información que se va a visualizar, es decir si se abrió desde la escena EscenaRA tomará el nombre de la imagen reconocida en el módulo de realidad aumentada o la biografía de Ivo Mora y la historia de la iglesia si se escogió la opción de información adicional en la escena Menu.
- Text (Subtítulo): Este Text permite visualizar el tipo de información que el usuario escogió en la escena MenuRA (Historia, Milagros, Oración, Biografía).
- Text (Referencia): Este Test permite visualizar las fuentes de donde se obtuvo la información.
- Text (Información): Este text permite incluir la información correspondiente que el usuario escogió.
- Text (Tiempo): Este text permite visualizar el tiempo de duración del audio.
- Botón Play: Este botón permite iniciar la reproducción del audio correspondiente a la información visualizada.
- Botón Pausa: Este botón permite pausar la reproducción del audio correspondiente a la información visualizada.
- Botón Stop: Este botón permite detener la reproducción del audio correspondiente a la información visualizada.
- Botón Regresar: Este botón permite regresar a la escena anterior, es decir la escena MenuRA.
- Slider: Este elemento permitirá avanzar o retroceder el audio cuando así lo requiera el usuario.



Figura 2.8 Escena Información

2.2.5.5. Escena “Ayuda”

Esta escena se abre al dar click en el botón Ayuda del menú principal, la cual está conformada por los siguientes elementos (Ver Figura 2.9).

- Text: Este text permite visualizar información sobre la aplicación.
- Text: Este text permite visualizar texto adicional.
- Botón Regresar: Este botón permite regresar a la Escena Menú.
- Botón Video: Permite visualizar un video que mostrará el funcionamiento de la aplicación.

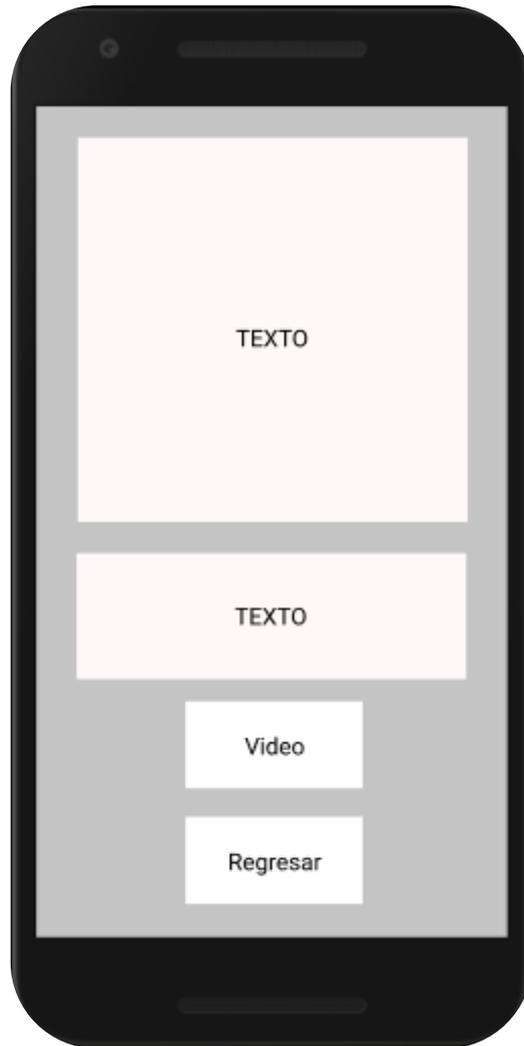


Figura 2.9 Escena Ayuda

2.2.5.6. Escena “EscenaDeCarga”

Esta escena se abre al dar click en el botón Iniciar de la escena Menú y al terminar de cargarse abrirá la escena “EscenaRA”.

Esta escena está conformada por los siguientes elementos (Ver Figura 2.10);

- Text: Permite visualizar el porcentaje de carga.
- Image: Permite incluir una imagen.

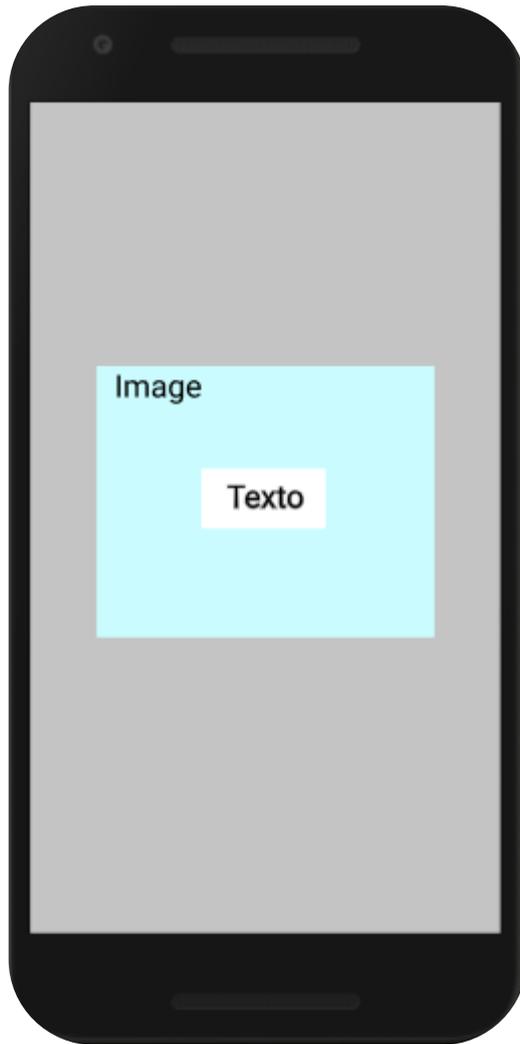


Figura 2.10 Escena EscenaDeCarga

2.2.6. DIRECCIONAMIENTO IP

Para el direccionamiento IP se utilizará una dirección IP de clase C, debido a que la Iglesia tiene una capacidad máxima de 200 personas, por lo que en el escenario crítico existirán 200 posibles usuarios de la red. En la Tabla 2.20 se indica la forma en la que se distribuyó las direcciones IP para los equipos.

Tabla 2.20 Direccionamiento IP para la red de descarga

EQUIPO	DIRECCIÓN IP	MÁSCARA
Access Point	192.168.0.1	255.255.255.0
Servidor	192.168.0.2	255.255.255.0
Administración	192.168.0.3	255.255.255.0
DHCP - POOL	192.168.0.4 – 192.168.0.254	255.255.255.0

En la Figura 2.11, se indica la arquitectura para la descarga del APK de la aplicación, la cual está constituida del servidor de descarga, el cual tiene los servicios de DNS, DHCP y APACHE, Access Point y los dispositivos que desean conectarse al servidor, con sus respectivas direcciones IP.

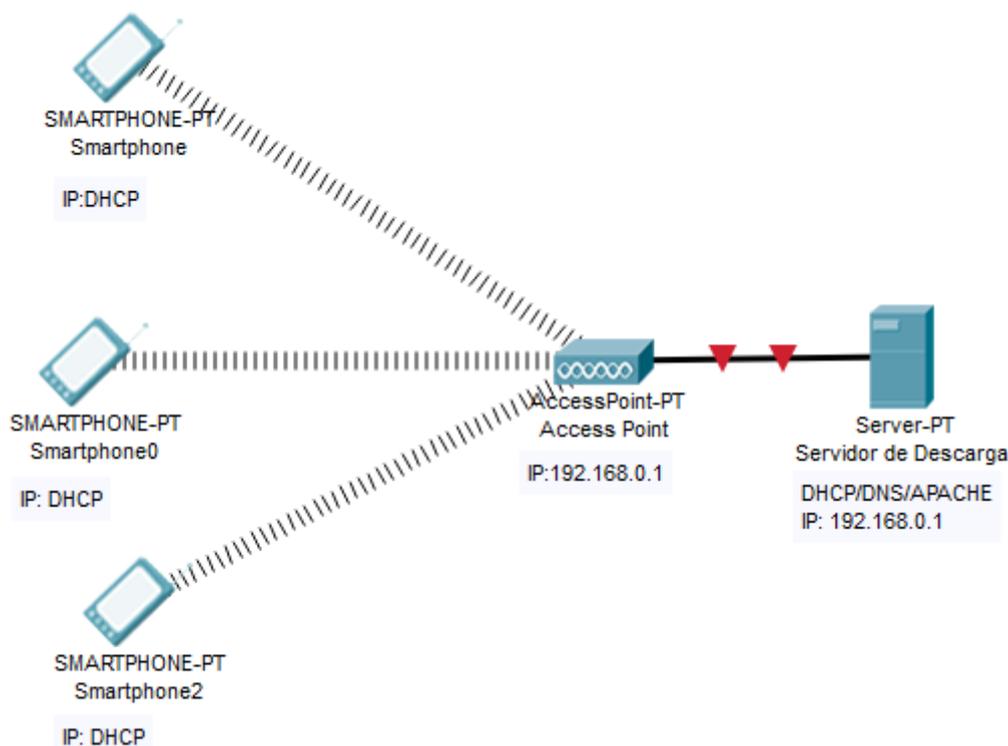


Figura 2.11 Arquitectura para la descarga de la aplicación

2.2.7. ELECCIÓN DEL ACCESS POINT

Para la elección del AP se tomó en cuenta la facilidad de tener un portal cautivo por hardware, que sea de una marca reconocida y de precio asequible, es por ello que se optó por un Access Point de la marca TP-Link.

TP-Link fue fundado en 1996, es un proveedor global de dispositivos y accesorios de red, con una variedad de productos para redes domésticas y para redes de negocios; además ofrece un software gratuito denominado Omada Controller.

Omada Controller permite monitorear y administrar cientos de EAP'S en uno o varios sitios, además la personalización del portal cautivo, límite de tasa (asignación de ancho de banda para cada SSID), programación de reinicio, balance de carga (limitar el número máximo de usuarios por EAP), actualización remota, entre otros [28].

Los AP's que proporcionan un portal cautivo y son compatibles con el software gratuito Omada Controller son todos los modelos EAP-XXX tal como se muestra en la Tabla 2.21:

Tabla 2.21 Tabla comparativo de Access Points

MODELO	MIMO	VELOCIDAD		PoE	ANTENAS		Wi-Fi STANDARDS	PRECIO (USD)
		2,4 GHz	5 GHz		2,4 GHz	5 GHz		
EAP245	si	450 Mbps	1300 Mbps	802.3at	3 x 3,5 dBi	3 x 4 dBi	IEEE 802.11 a/b/g/n/ac	125,13
EAP225	si	450 Mbps	867 Mbps	802.3af Pasive PoE	3 x 4 dBi	2 x 5 dBi	IEEE 802.11 a/b/g/n/ac	79,99
EAP115	no	300 Mbps	-	802.3af	2 x 3 dBi	-	IEEE 802.11 b/g/n	33,60
EAP110	no	300 Mbps	-	Passive PoE	2 x 3 dBi	-	IEEE 802.11 b/g/n	Bajo Pedido

La red inalámbrica que se va a diseñar no necesita de una velocidad mayor a los 100Mbps, o que tenga dos bandas de operación, ya que únicamente será para descargar el APK de la aplicación la cual no pesará más de 200Kb, por lo cual funcionará de manera óptima el AP EAP-110 o el AP EAP-115, ya que los EAP 225 y 245 presentan mayores velocidades, pero sería sobredimensionar la red que se desea implementar. El Access Point EAP-110 se lo puede adquirir únicamente bajo pedido por lo que se optó por el EAP-115.

2.2.8. PARÁMETROS DE CONFIGURACIÓN DEL ACCESS POINT

Los parámetros de configuración del Access Point EAP-115 están divididas en cuatro, la configuración básica, configuración IP, radio y balance de carga [34].

2.2.8.1. Configuración Básica

La configuración básica (Figura 2.12) tiene un solo parámetro que corresponde al nombre del Access Point dentro de la red.

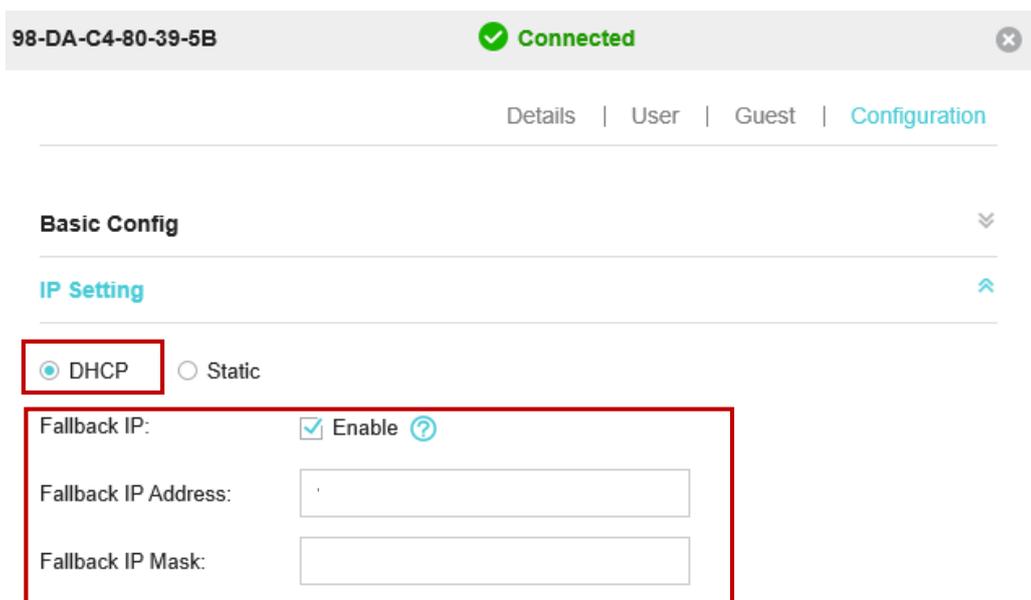
The screenshot shows a web interface for an Access Point. At the top, there is a status bar with the ID '98-DA-C4-80-39-5B', a green 'Connected' indicator, and a close button. Below this are navigation tabs: 'Details', 'User', 'Guest', and 'Configuration'. The 'Basic Config' section is active and highlighted in blue. It contains a single text input field labeled 'Name' with the value '98-DA-C4-80-39-5B' entered. Below the input field is a blue 'Apply' button.

Figura 2.12 Configuración del nombre del Access Point

El nombre del Access Point se va a dejar el nombre por defecto, ya que no se va a incluir ningún AP adicional y será el único dentro de la red.

2.2.8.2. Configuración IP

En la configuración IP se tiene dos opciones estático o dinámico, tal como se muestra en la Figura 2.13 y la Figura 2.14.



The screenshot shows the configuration page for an Access Point with ID 98-DA-C4-80-39-5B, which is currently connected. The 'Configuration' tab is active. Under the 'IP Setting' section, the 'DHCP' option is selected and highlighted with a red box. Below it, the 'Fallback IP' section is also highlighted with a red box and contains the following fields: 'Fallback IP' (checked 'Enable'), 'Fallback IP Address' (empty text box), and 'Fallback IP Mask' (empty text box).

Figura 2.13 Configuración IP dinámica del Access Point



The screenshot shows the configuration page for an Access Point with the 'Static' option selected and highlighted with a red box. The fields for static configuration are: 'IP Address' (empty text box), 'IP Mask' (empty text box), 'Gateway' (empty text box), 'Preferred DNS server' (empty text box, labeled '(Optional)'), and 'Alternate DNS server' (empty text box, labeled '(Optional)'). An 'Apply' button is located at the bottom.

Figura 2.14 Configuración IP estática del Access Point

- La opción **Dinámica**, permite habilitar el ítem **Fallback IP**, esta opción permite habilitar una IP de reserva cuando el AP no ha obtenido por diez segundos una IP del servidor DHCP. El AP trabajará con esa IP de reserva, sin embargo, seguirá intentando obtener una IP del servidor hasta que tenga éxito.
- La opción **Estática**, tiene que ser configurada la dirección IP manualmente y todos los parámetros relacionados para el AP.

En ambas opciones se debe asegurar que las direcciones IP configuradas, no sean utilizadas por ningún otro dispositivo dentro de la red.

Se va a utilizar la opción estática para lo cual se va a configurar como dirección IP la 192.168.0.1 con máscara 255.255.255.0 y el servidor DNS 192.168.0.2.

2.2.8.3. Ajustes de Radio

Los ajustes de radio controlan el comportamiento de la radio en el AP y la interacción con el medio físico, es decir cómo y qué tipo de señal emite el AP. Esta configuración está conformada por el estatus, modo, ancho del canal, canal y potencia de transmisión, tal como se muestra en la Figura 2.15.

The screenshot displays the configuration interface for an Access Point. At the top, the device ID is 98-DA-C4-80-39-5B, and it is connected. The 'Configuration' tab is active. The 'Radio' section is expanded, showing two frequency bands: 2.4GHz (selected) and 5GHz. The configuration for 2.4GHz includes:

- Status: Enable
- Mode: [Dropdown menu]
- Channel Width: [Dropdown menu]
- Channel: [Dropdown menu]
- Tx Power: [Dropdown menu]

An 'Apply' button is located at the bottom of the configuration area.

Figura 2.15 Ajustes de Radio del Access Point

Los ajustes de Radio son para la banda de 2.4 GHz y 5 GHz, en este caso el AP EAP-115 solo tiene la banda de 2.4 GHz, por lo que se mostrará la configuración para la banda de 2.4GHz.

- Modo: En esta opción se escoge el modo IEEE 802.11 que utiliza la radio. Para la banda de 2.4 GHz se tienen las siguientes opciones, tal como se muestra en la Figura 2.16.



Figura 2.16 Configuración del Modo del Access Point

- ❖ 802.11 b/g/n mixed: Todos los usuarios 802.11b, 802.11g y 802.11n que operan en la banda de 2.4GHz pueden conectarse al AP.
 - ❖ 80211 b/g mixed: Todos los usuarios 802.11b y 802.11g, que operan en la banda de 2.4GHz pueden conectarse al AP.
 - ❖ 802.11 n only: Solo usuarios 802.11n que operan en la banda de 2.4GHz pueden conectarse al AP.
- Ancho del canal: Esta opción permite escoger el ancho del canal para la banda de 2.4GHz, tal como se muestra en la Figura 2.17 y son las siguientes:

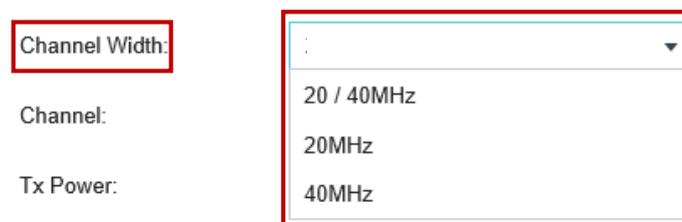


Figura 2.17 Configuración del ancho del canal del Access Point

- ❖ 20/40 MHz: Esta es la opción recomendada por el manual, ya que permite que el AP decida el ancho de banda a utilizar en función del entorno.
- ❖ 20 MHz: Este es un ancho de canal menor, el cual puede ser utilizado para aumentar el número de canales disponibles dentro del ancho de banda. Evitar interferencias entre canales o mayor potencia de transmisión para enlaces a larga distancia.
- ❖ 40 MH: Este es un ancho de banda mayor, el cual es utilizado para obtener mayores tasas de transferencia.

- Canal: Esta opción permite seleccionar el canal utilizado por el AP, tal como se muestra en la Figura 2.18.

Channel:

Tx Power:

Apply

Load Balance

WLANS

Auto

1 / 2412MHz

2 / 2417MHz

3 / 2422MHz

4 / 2427MHz

5 / 2432MHz

6 / 2437MHz

Figura 2.18 Configuración del canal del Access Point

Para obtener niveles óptimos de rendimiento, se debe seleccionar el modo automático para que el AP escoja el mejor canal.

- Potencia de transmisión: Esta opción especifica el valor de la potencia de transmisión, teniendo las opciones de alta, media, baja y personalizada, tal como se muestra en la Figura 2.19.

Tx Power:

Apply

Load Balance

Low

Medium

High

Custom

Figura 2.19 Configuración de la potencia de transmisión

2.2.8.4. Balance de carga

Esta opción permite limitar el número máximo de usuarios que se pueden conectar al Access Point de manera simultánea (Ver Figura 2.20).

98-DA-C4-80-39-5B
✔ Connected
✕

[Details](#) | [User](#) | [Guest](#) | [Configuration](#)

Basic Config ⌵

IP Setting ⌵

Radio ⌵

Load Balance ⌶

2.4GHz

5GHz

Max Associated Clients: Enable

(1-255)

RSSI Threshold: Enable ?

(-95-0 dBm)

Apply

Figura 2.20 Configuración del balance de carga del Access Point

2.2.9. PARÁMETROS DE CONFIGURACIÓN DEL PORTAL CAUTIVO

En este proyecto se va a utilizar un portal cautivo por hardware que viene incorporado en el Access Point TP-Link EAP-115, mismo que se lo configurará mediante el software gratuito que proporciona la marca, denominada Omada Controller.

La configuración del portal cautivo está dividida en tres partes, la Configuración Básica, la Configuración de la Página de Login y configuración de anuncios.

2.2.9.1. Configuración Básica del Portal Cautivo

La configuración básica está conformada por siete parámetros (Nombre del Portal, SSID, tipo de autenticación, tiempo de expiración de autenticación, redirección HTTPS, redirección, URL), tal como se muestra en la Figura 2.21.

Basic Info

Portal Name:	<input type="text"/>
SSID:	<input type="text" value="- Please Select -"/>
Authentication Type:	<input type="text" value="No Authentication"/>
Authentication Timeout:	<input type="text" value="1 Hour"/>
	<input type="checkbox"/> Daily Limit ?
HTTPS Redirect:	<input type="checkbox"/> Enable ?
Redirect:	<input checked="" type="checkbox"/> Enable ?
Redirect URL:	<input type="text"/>

Figura 2.21 Configuración básica del portal cautivo

A continuación, se detallan cada uno de los parámetros de la configuración básica:

- Nombre del Portal: Es un nombre identificativo en el caso que se tenga más de 2 SSID's.
- SSID: Seleccionar el SSID en el caso que se tenga más de uno.
- Tipo de autenticación: En el tipo de autenticación se tiene las siguientes opciones (Figura 2.22):
 - Sin autenticación: Todos los usuarios podrán acceder a los recursos sin ningún tipo de autenticación. Los usuarios de igual forma visualizará la página de inicio de sesión preestablecida.
 - Password simple: Todos los usuarios que deseen conectarse a la red deberán, usarán esta contraseña única para la autenticación.
 - Usuario local: Los usuarios deberán ingresar un usuario y contraseña que son creadas en la opción de User Management.
 - SMS: Los usuarios obtienen códigos de verificación utilizando sus celulares a través de mensajes de texto. Estos códigos deben ser ingresados para pasar la autenticación.

- Facebook: Los usuarios que se conecten a la red, serán redirigidos a la página de Facebook, una vez autenticados podrán acceder a los recursos de la red.
- External RADIUS Server.
- External Portal Server.

En este caso será sin autenticación ya que simplemente servirá para redireccionar a la página de descarga.

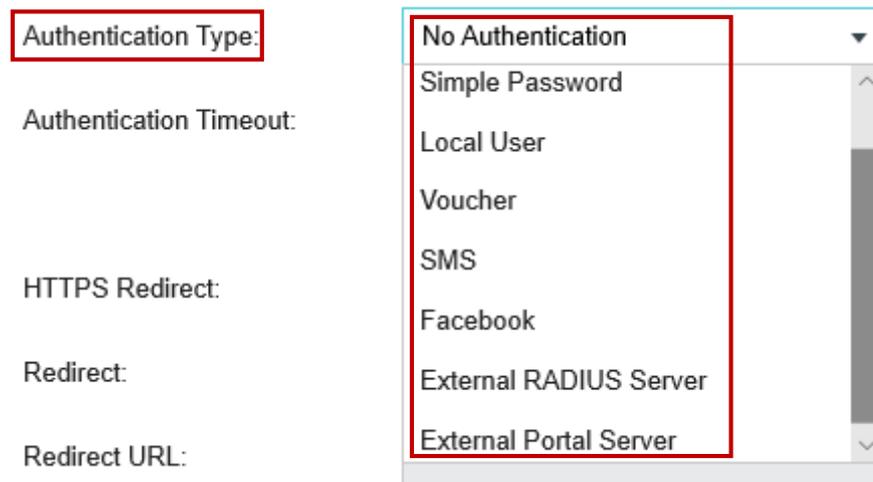


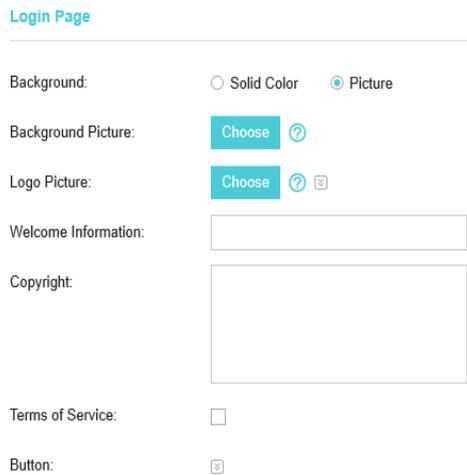
Figura 2.22 Tipo de autenticación del portal cautivo

- Authentication Timeout: El tiempo de expiración de autenticación, es un tiempo en el que el usuario va a poder permanecer conectado y después de tendrá que volver a autenticarse. Se escogió el tiempo de una hora ya que es el tiempo mínimo que proporciona el portal cautivo y será tiempo suficiente para que se descargue el APK de la aplicación.
- Redirect: Habilitar el Redirect, permite redireccionar al usuario a una página web en específico.
- Redirect URL: Es la dirección URL de la página web a la que va a ser redireccionado.

2.2.9.2. Configuración de la Página de Login

Después de haber identificado los parámetros de la configuración básica del portal cautivo que ofrece el AP TP-Link, se procede a identificar los parámetros para la personalización de la página de inicio de sesión (Ver Figura 2.23).

En la página de inicio de sesión se puede editar el fondo, el logotipo, los derechos de autor, términos de servicio, según lo requiera.



Login Page

Background: Solid Color Picture

Background Picture: ?

Logo Picture: ? ?

Welcome Information:

Copyright:

Terms of Service:

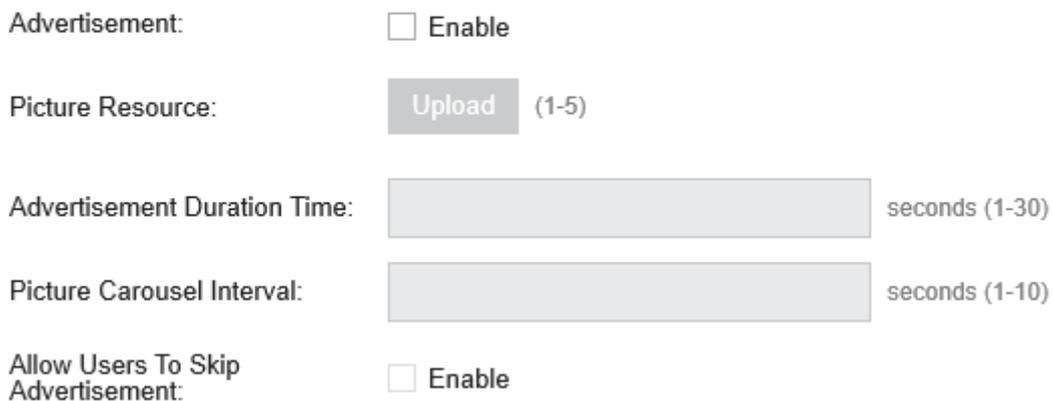
Button:

Figura 2.23 Configuración de la página de login

2.2.9.3. Configuración de anuncios

Esta sección permite habilitar los anuncios publicitarios, si así lo requiere. Se puede mostrar una imagen, el tiempo de duración del anuncio y otros parámetros de configuración, tal como se muestra en la Figura 2.24.

Advertisement



Advertisement: Enable

Picture Resource: (1-5)

Advertisement Duration Time: seconds (1-30)

Picture Carousel Interval: seconds (1-10)

Allow Users To Skip Advertisement: Enable

Figura 2.24 Configuración de anuncios publicitarios

2.2.10. PARÁMETROS DE CONFIGURACIÓN DEL SERVIDOR DE DESCARGA

El servidor de descarga del APK de la aplicación está conformado por tres servicios, el servicio DNS para resolución de nombres, el servicio de DHCP para asignación dinámica de IP's y el servicio WEB en dónde se podrá acceder al APK de la aplicación. Todos estos servicios, serán configurados en Ubuntu Server.

2.2.10.1. Servicio DNS

El servicio de DNS permite asociar una dirección IP a un nombre. Este servicio se utilizará para resolver el nombre de la página WEB que se va a implementar.

En el desarrollo de este proyecto se va a utilizar el servidor de nombres BIND. El archivo de configuración para este servicio es el */etc/bind/named.conf.local*, (este archivo sirve para añadir una zona⁹ DNS). Los registros de este archivo de configuración son los siguientes:

- Archivo de zona directa */etc/bind/db.local*.
- Archivo de zona inversa */etc/bind/db.192*.

El nombre del dominio que se va a utilizar es *santura.com* y la página web tendrá como URL *www.santura.com*.

2.2.10.2. Servicio DHCP

El servicio DHCP es empleado para que los usuarios puedan obtener su dirección IP, máscara de red y características del DNS. Para configurar este servicio se debe editar el archivo */etc/dhcp/dhcpd.conf*, con los valores del DHCP de la Tabla 2.20.

2.2.10.3. Servicio Web

Para el servicio WEB, se va a utilizar **Apache**. Apache es un servidor WEB de código abierto desarrollado por **Apache Software Foundation**. Los archivos de configuración para este servicio, son los siguientes:

- Archivo */etc/apache2/mods-enabled/dir.conf*, especifica el archivo por defecto que se va a entregar si no se indica un recurso.
- Archivo */etc/apache2/apache2.conf*, especifica el nombre del servidor.

⁹ Una zona es la parte de un dominio que es administrada por una organización.

2.3. FASE DE IMPLEMENTACIÓN

En esta sección se realizará la recopilación de información sobre el santo de la parroquia y biografía del autor de las pinturas del Santuario, esto se realizó a través de entrevistas a párrocos y al Señor Ivo Mora. Se realizará la edición de audio de la información que proporcionará la aplicación sobre las estatuas y pinturas. Por último, se implementarán las escenas que conforman la aplicación, además de la codificación e implementación de cada uno de los módulos que tiene el sistema.

2.3.1. ENTREVISTAS

Las entrevistas tienen como objetivo obtener información sobre el Señor de la Salud (Santo representativo de la parroquia) y el pintor Ivo Mora (autor de las pinturas del Santuario), esta información es la que se desplegará en la aplicación, es por ello que se realizó dos entrevistas para el desarrollo del presente trabajo de titulación. La primera entrevista, tal como se muestra en el Anexo D, está dirigida a tres Párrocos que hayan ejercido la cura pastoral en la parroquia Santiago, esta entrevista permitirá obtener información sobre la historia, oración y milagros atribuidos al Señor de la Salud. La segunda entrevista, tal como se muestra en el Anexo E, está dirigida al pintor Ivo Mora autor de las pinturas que se encuentran en el Santuario, esta entrevista tiene como objetivo recopilar información sobre la vida del pintor y poder escribir una pequeña biografía que se mostrará en la aplicación.

2.3.2. GRABACIÓN Y EDICIÓN DE AUDIO

Los audios que forman parte de la aplicación, corresponden a la información de cada santo, pintura, iglesia y biografía del pintor, los cuales fueron realizados y editados mediante el software Vegas Pro.

La grabación se realizó insertando una pista de audio, para ello se da click en el área de lista de pistas y se seleccionará la opción **Insertar pista de audio**, tal como se muestra en la Figura 2.25.

Una vez que se haya creado la pista se da click en el botón Grabar para iniciar la grabación y al finalizar se presiona el botón de detener (Ver Figura 2.26) el que automáticamente abrirá el cuadro de diálogo de Archivos grabados y se guarda un archivo con extensión **.sfk**¹⁰.

En la pista que se grabó se visualiza espacios sin audio al inicio y al final de la grabación, estos espacios deben ser eliminados, para ello se selecciona el audio, se ubica la línea de

¹⁰ Es un archivo de datos inicial creado por Vegas Pro, que almacena la información gráfica de la forma de onda [35].

tiempo en el lugar donde se va a realizar el corte y se presiona la tecla S del teclado. Por último, se selecciona la parte del audio que se va a eliminar y se presiona la tecla Suprimir (Ver Figura 2.29 y Figura 2.28).

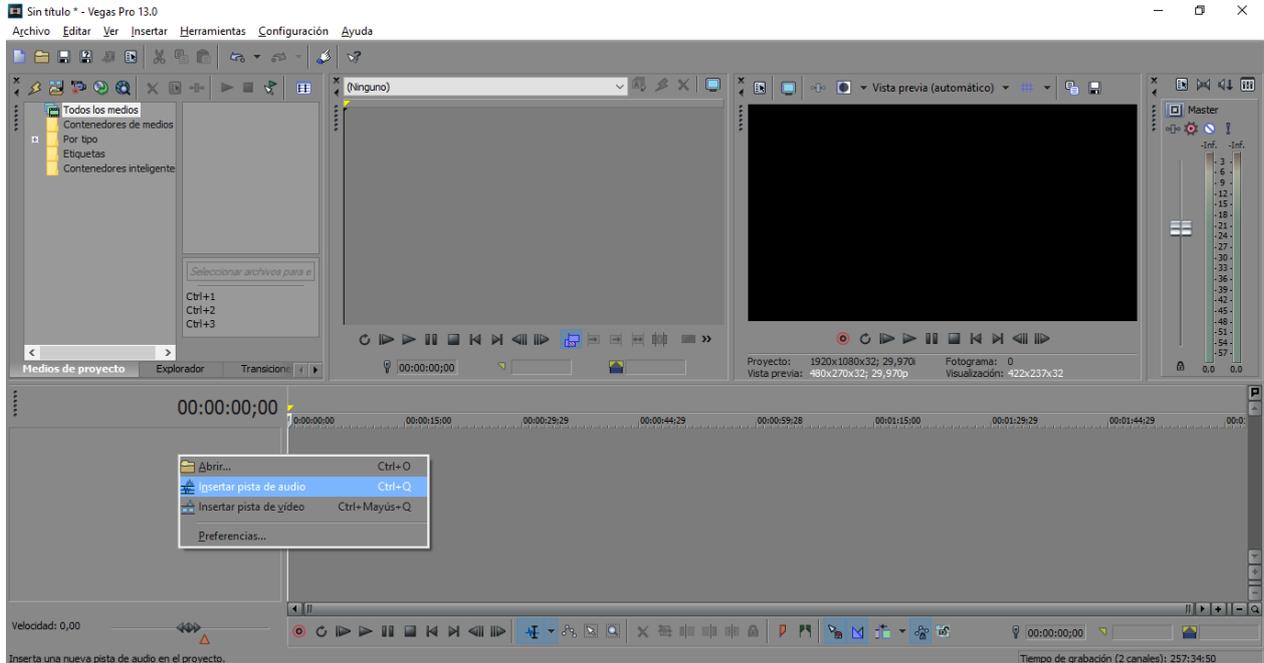
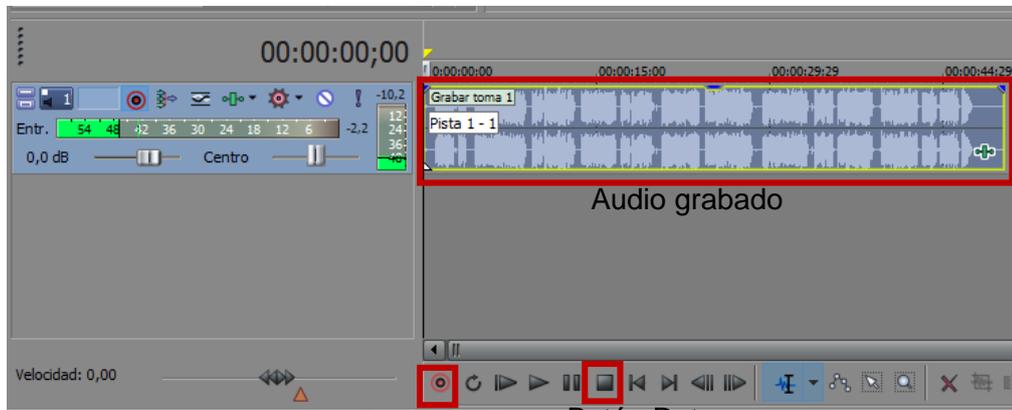


Figura 2.25 Insertar pista de audio



Botón Grabar **Botón Detener**

Figura 2.26 Grabar el audio

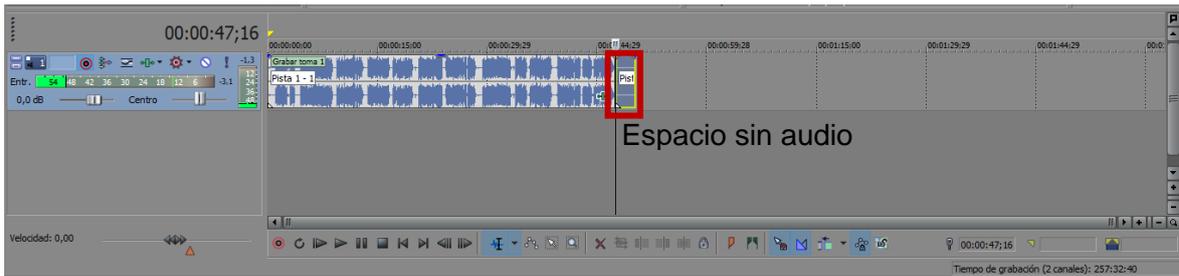


Figura 2.29 Cortar el audio

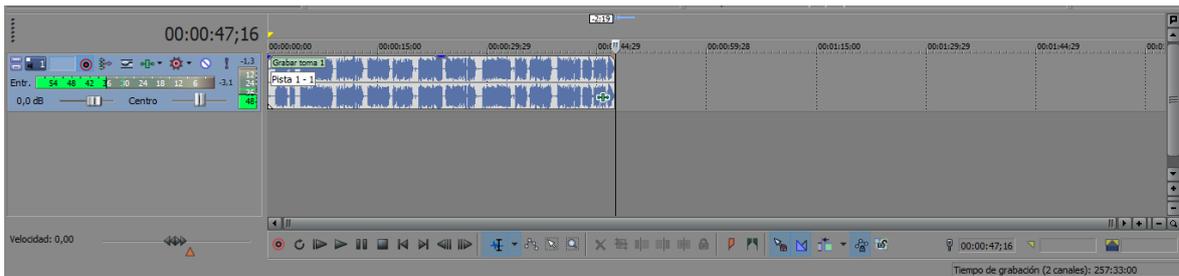


Figura 2.28 Pista después de cortar los espacios sin audio.

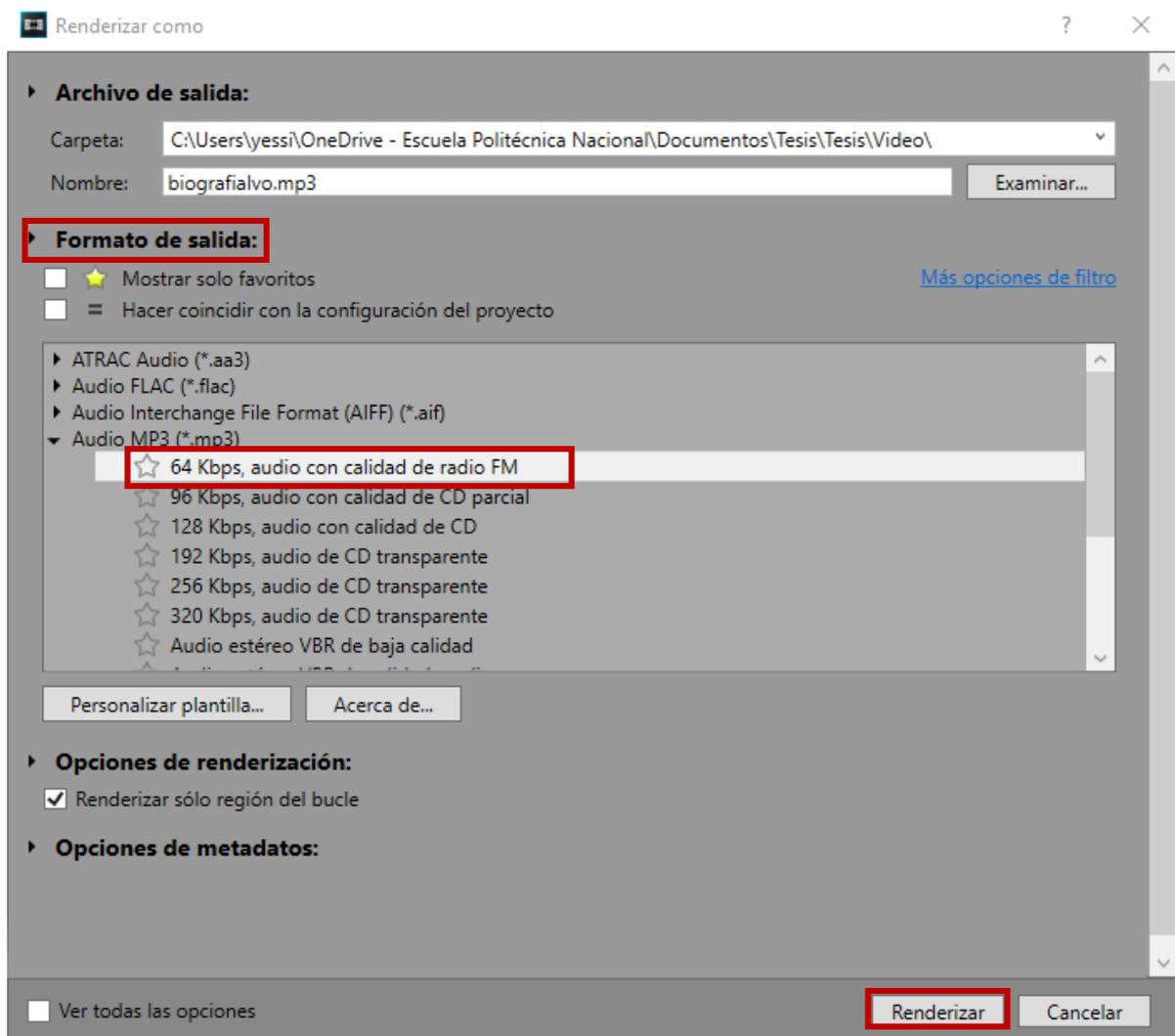


Figura 2.27 Ventana para renderizar la pista

Luego, se procede a renderizar el audio presionando el botón Renderizar en la barra de herramientas, se abrirá la ventana y se escogerá el formato. En este caso se escoge el formato **MP3 a 64Kbps, audio con calidad FM**, con el cual se tiene una calidad aceptable y con un tamaño de archivo menor, como se muestra en la Figura 2.27

Por último, se da click en Renderizar y en unos segundos aparece el archivo en la ruta establecida en la Ventana Renderizar. Este procedimiento se realizó para los cuarenta y dos audios que forman parte de la aplicación.

2.3.3. BASE DE DATOS EN VUFORIA

La base de datos tiene los marcadores que identificará la aplicación. Las imágenes utilizadas como marcadores corresponden a las estatuas y pinturas del santuario. Las imágenes fueron seleccionadas en base a la pregunta tres (**¿Cuáles son las pinturas y estatuas que necesariamente deben estar en la aplicación?**) de la entrevista del Anexo C, en la cual se obtuvo las imágenes más importantes que obligatoriamente debían estar en la aplicación, tal como se muestra en la Tabla 2.22, las cuales son:

- Señor de la Salud.
- San Pedro.
- San Francisco de Asís.
- Patrón Santiago.
- Señor de la Misericordia.
- San José.
- San Isidro Labrador.
- Arcángel San Miguel.
- Sagrado Corazón de Jesús.
- San Vicente Ferrer.
- Jesús Divino Médico.
- Cristo Redentor.

Tabla 2.22 Lista de imágenes y pinturas

	NOMBRE	Padre Carlos Solarte	Luis Alberto Cuji Macas	Euclídes Ramírez Espinoza	Freddy Malatay Caibe	Total
ESTATUAS	Señor de la Salud	1	1	1	1	4
	San Pedro	1	1	1	1	4
	San Francisco de Asís	1	1	1	1	4
	Patrón Santiago	1	1	1	1	4
	Señor de la Misericordia	1	0	1	1	3
	San José	0	1	1	1	3
	San Isidro Labrador	1	0	1	1	3
	Arcángel San Miguel	0	1	0	1	2
	Sagrado Corazón de Jesús	0	0	1	1	2
	San Vicente Ferrer	0	0	1	1	2
	Jesús Crucificado	0	0	1	1	2
	Divino Niño Jesús	1	1	0	0	2
	Niño Jesús de Praga	0	0	0	1	1
	Inmaculada Concepción	0	0	0	1	1
	Santa Teresita	0	0	0	1	1
	Santa Marianita	0	0	0	1	1
	María Auxiliadora	1	0	0	0	1
Virgen del Rosario	0	0	1	0	1	
PINTURAS	Jesús Divino Médico	1	1	1	1	4
	Cristo Redentor	1	1	1	0	3

Las imágenes fueron tomadas mediante la cámara del celular, se encuentran en el Anexo A, y se utilizó Vuforia para crear la base de datos.

El proceso para realizar la base de datos en Vuforia es el siguiente:

- Ingresar a <https://developer.vuforia.com/vui/auth/login>, este es el portal de desarrolladores de Vuforia, tal como se muestra en la Figura 2.30.

Previamente, se debe crear un usuario y proceder a realizar el respectivo login.

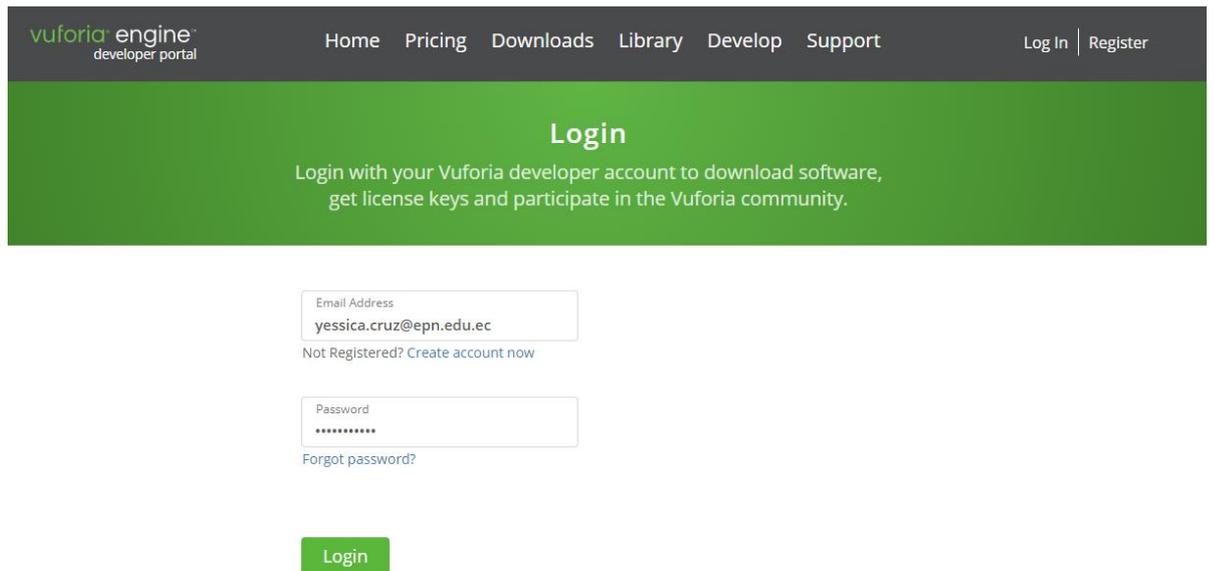


Figura 2.30 Portal de desarrolladores de Vuforia

- Una vez que se haya ingresado al portal existen dos opciones el License Manager en donde se encuentra la licencia que será utilizada al momento de configurar Vuforia en el proyecto y Target Manager en donde se va a desarrollar la base de datos, en este caso se va a seleccionar la opción Target Manager y en **Add Database**, tal como se muestra en la Figura 2.31.

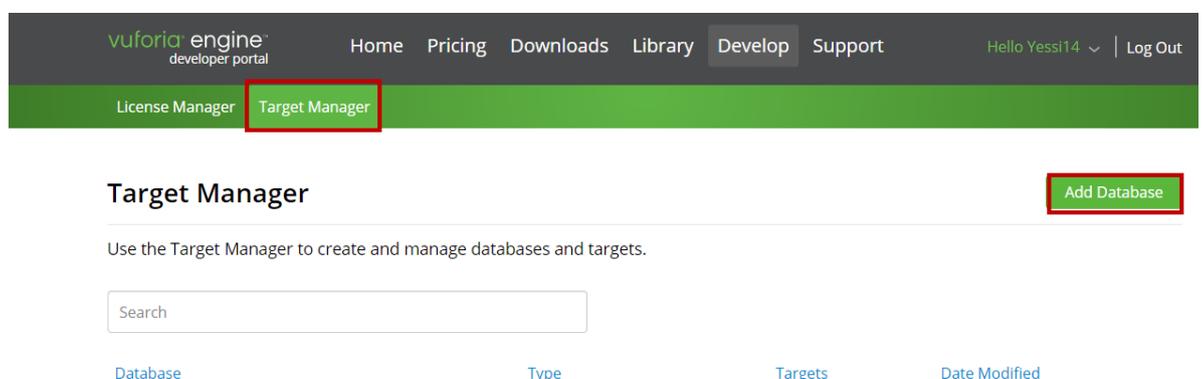


Figura 2.31 Ventana Target Manager

- En la ventana para crear la base de datos se realizó lo siguiente, ver Figura 2.32:
 1. Ingresar el nombre de la base de datos, para el presente proyecto de titulación, la base de datos se nombró como **BDtesis**.
 2. Escoger el tipo de base de datos. En esta opción existen tres posibilidades **Device**, **Cloud** y **Vumark**, en este caso se escogió **Device Databases**, ya

que son bases de datos locales que se almacenan en el dispositivo, además permite un reconocimiento rápido de los marcadores.

3. Crear la base de datos.

Create Database

Database Name *
BDtesis

Type:

Device
 Cloud
 VuMark

Cancel Create

Figura 2.32 Ventana Crear base de datos

La base de datos ya creada se podrá visualizar en la Ventana del **Target Manager** tal como se muestra en la Figura 2.33.

License Manager Target Manager

Target Manager

Add Database

Use the Target Manager to create and manage databases and targets.

Search

Database	Type	Targets	Date Modified
BDtesis	Device	0	Sep 06, 2019

Figura 2.33 Ventana Target Manager con la base de datos creada

- Una vez creada la base de datos, se incluyeron las imágenes que son los patrones que identifica la aplicación, para lo cual se dirige a la base de datos creada y después en **Add Target** (Ver Figura 2.34).

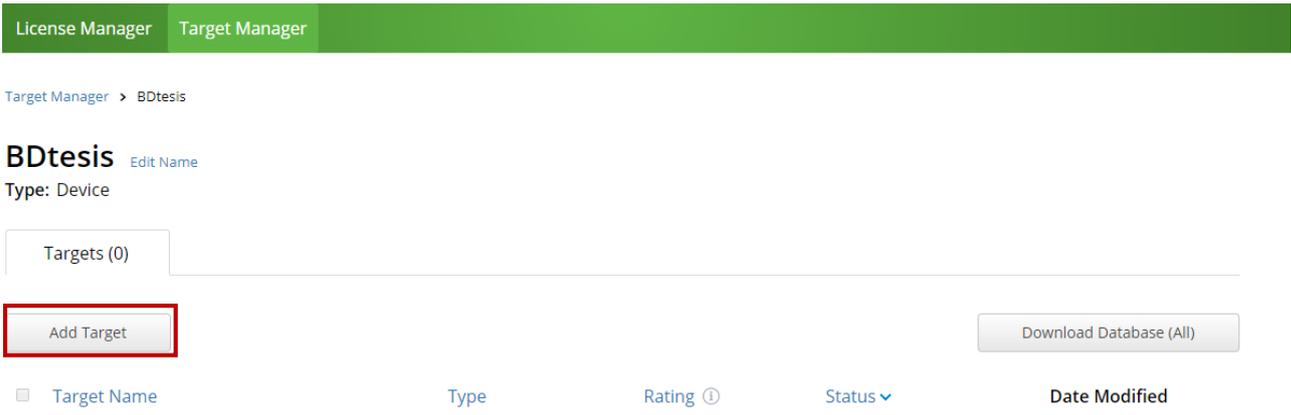


Figura 2.34 Ventana de la base de datos

Al dar click en **Add Target** se abre una ventana en la cual se escoge el tipo, en este caso es **Single Image**, ya que se utilizarán las fotos de las imágenes que están en el santuario. Luego se adjunta el archivo de la imagen la cual debe ser en formato JPG o PNG con un tamaño máximo de 2Mb. Después se debe ingresar el ancho del marcador, el nombre del marcador, y dar click en **Add**. Esto se puede visualizar en la Figura 2.35.

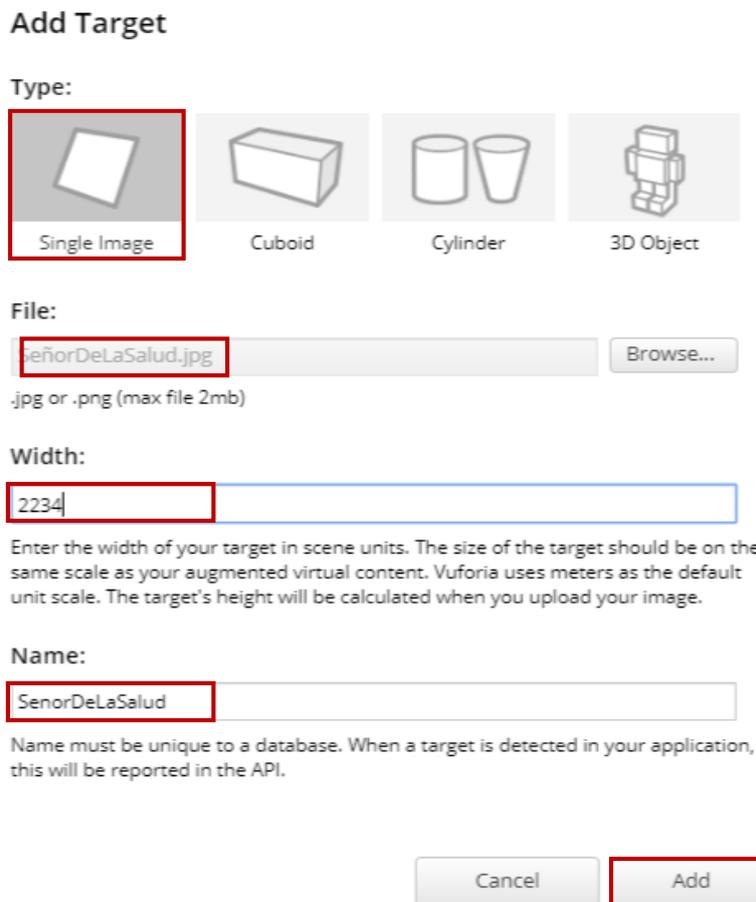


Figura 2.35 Ventana Add Target

En la Figura 2.36 se visualiza la ventana de la base de datos con el target creado, ahí se distinguen cuatro campos importantes: el nombre del Target, el tipo, el rating en estrellas y el estado del marcador. Esta información también se la puede visualizar en la Figura 2.37 que corresponde a la ventana del target, además de datos adicionales como el ID, fecha de modificación y fecha en la que fue añadido el target.

El rating de estrellas depende de la calidad del objetivo y puede variar de 0 a 5 estrellas, Vuforia Engine solo puede detectar un objetivo que tenga al menos una estrella.

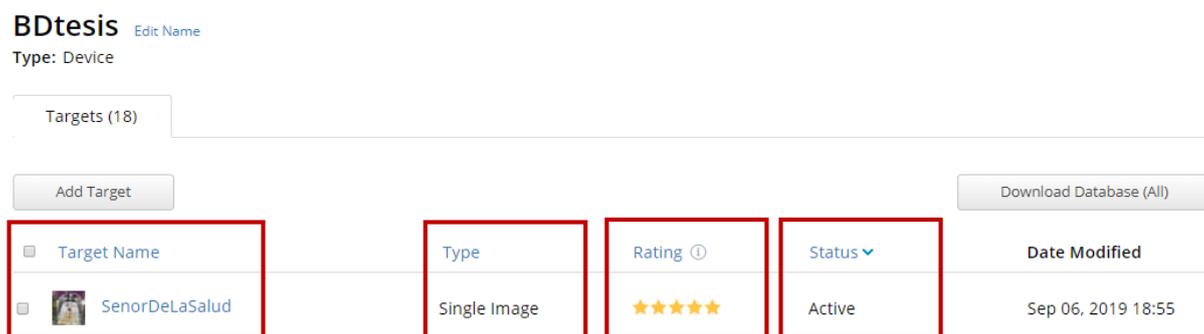


Figura 2.36 Ventana de la base de datos con el marcador creado

SenorDeLaSalud

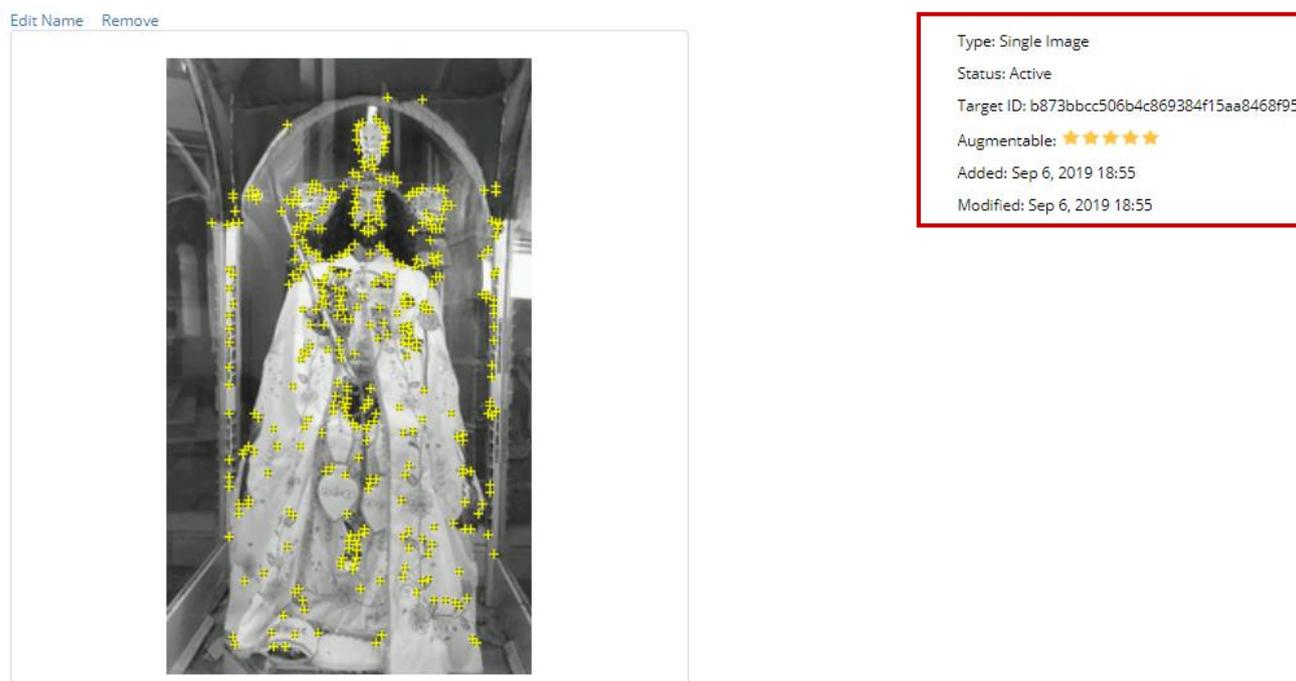


Figura 2.37 Ventana con la información del marcador

- Una vez incluidas todas las imágenes que formarán parte de la base de datos se procede a descargar dando click en el botón **Download Database (All)** donde aparecerá una nueva ventana, tal como se muestra en la Figura 2.38, que permite visualizar el número de marcadores activos que tiene la base y seleccionar la plataforma de desarrollo, en este caso se selecciona Unity Editor.

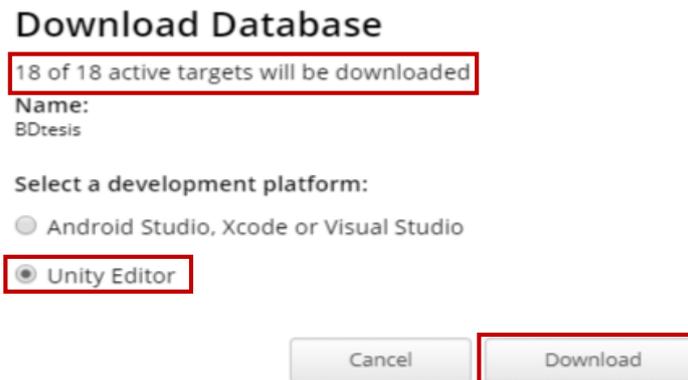


Figura 2.38 Ventana de descarga de la base de datos

El archivo correspondiente a la base de datos tiene extensión **.unitypackage**.

Por último, se incluyó la base de datos, para lo cual el proyecto debe estar abierto, dar click sobre el archivo y automáticamente se exportará en el proyecto.

2.3.4. CONFIGURACIÓN DE LA PLATAFORMA DE DESARROLLO

La plataforma de desarrollo utilizada es Unity, esta permite realizar aplicaciones para dispositivos Android, para lo cual en la ventana **Preferences** se debe incluir el SDK de Android y el JDK de Java, tal como se muestra en la Figura 2.39.

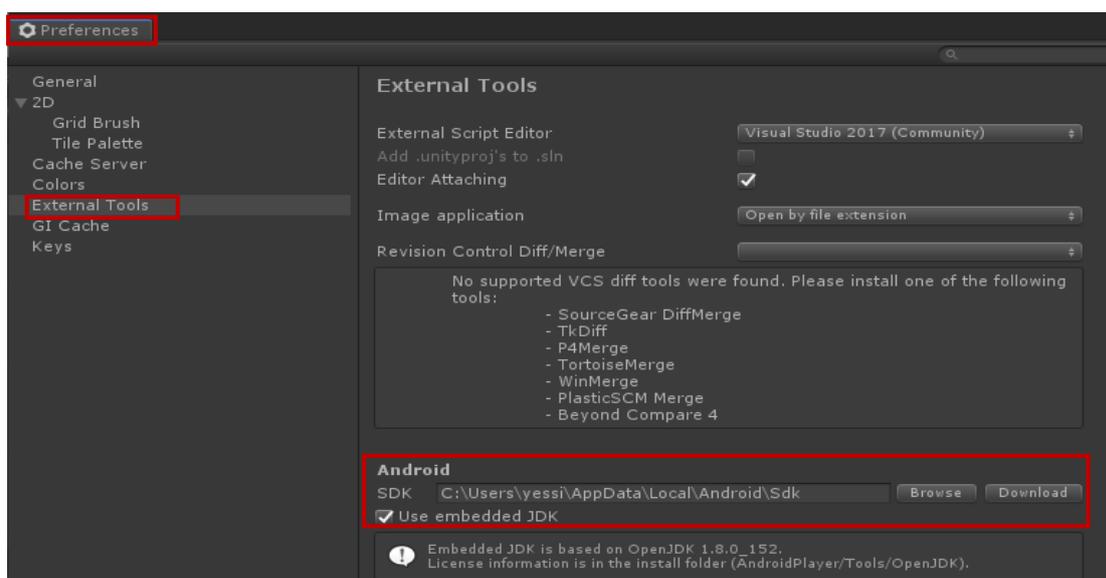


Figura 2.39 Ventana Preferences de Unity

2.3.5. CONFIGURACIÓN PARA LA CONSTRUCCIÓN DE LA APLICACIÓN

Para iniciar el desarrollo de la aplicación se realizaron dos configuraciones que afectan la forma en la que se construye la aplicación, una se realiza en el Player Settings y la otra en el Build Settings dentro de Unity.

2.3.5.1. Build Settings

La ventana Build Settings de Unity permite elegir la plataforma, ajustes de compilación y para compilar la aplicación.

En la ventana Build Settings se escoge la plataforma en la que se va a desarrollar la aplicación, por defecto viene activado la opción **PC, MAC & Linux Standalone**, tal como se muestra en la Figura 2.40, en este caso para el desarrollo de este proyecto se escogió la opción **Android** y se procedió a cambiar la plataforma presionando el botón **Switch Platform**. Este paso se lo puede realizar al inicio o al finalizar el proyecto, pero es recomendable hacerlo al inicio ya que tomará más tiempo cambiar de plataforma a medida que se avance en el proyecto y se tenga más elementos dentro del mismo.

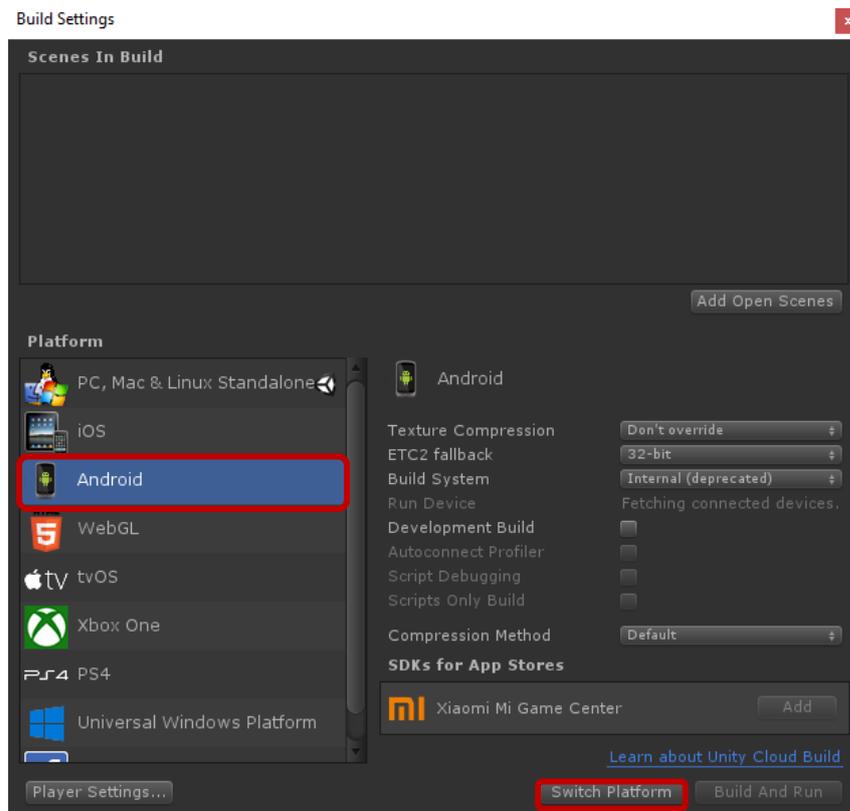


Figura 2.40 Ventana Build Settings

2.3.5.2. Player Settings

Esta ventana de Unity permite configurar el ícono, resolución, imagen de bienvenida, configuraciones XR, otras configuraciones y configuraciones de publicación,.

En esta ventana primero se ingresa el **Company Name** que permite identificar el nombre de la empresa, después el nombre del producto éste está asociado al icono de la aplicación en el escritorio del dispositivo cuando se ha instalado la aplicación. Luego, la versión y por último el ícono de la aplicación, tal como se muestra en la Figura 2.41.

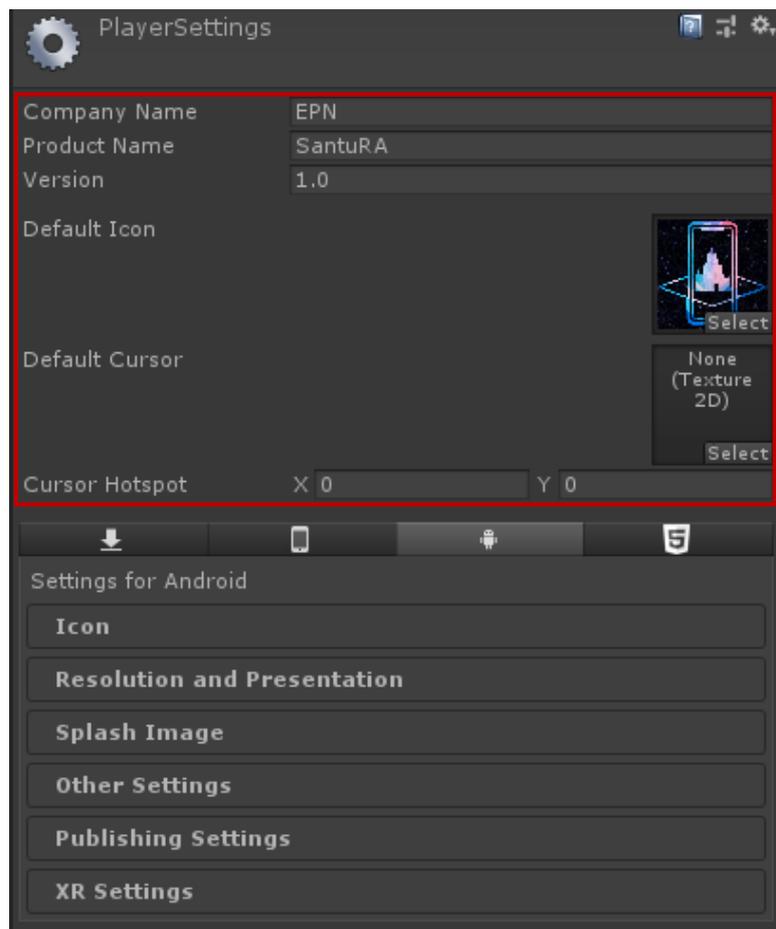


Figura 2.41 Ventana Player Settings

La opción **Splash Screen**, permite incluir una imagen que será visualizada cuando la aplicación se inicie, para habilitarla se marcó la opción **Show Splash Screen**, se adjuntó la imagen en Logos y se estableció el tiempo de duración de la imagen de presentación, esta configuración se la puede visualizar en la Figura 2.42.

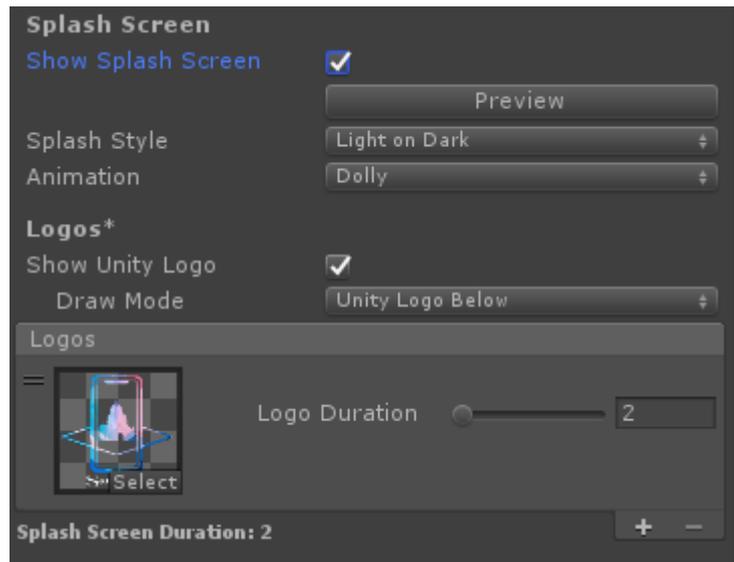


Figura 2.42 Configuración Splash Screen

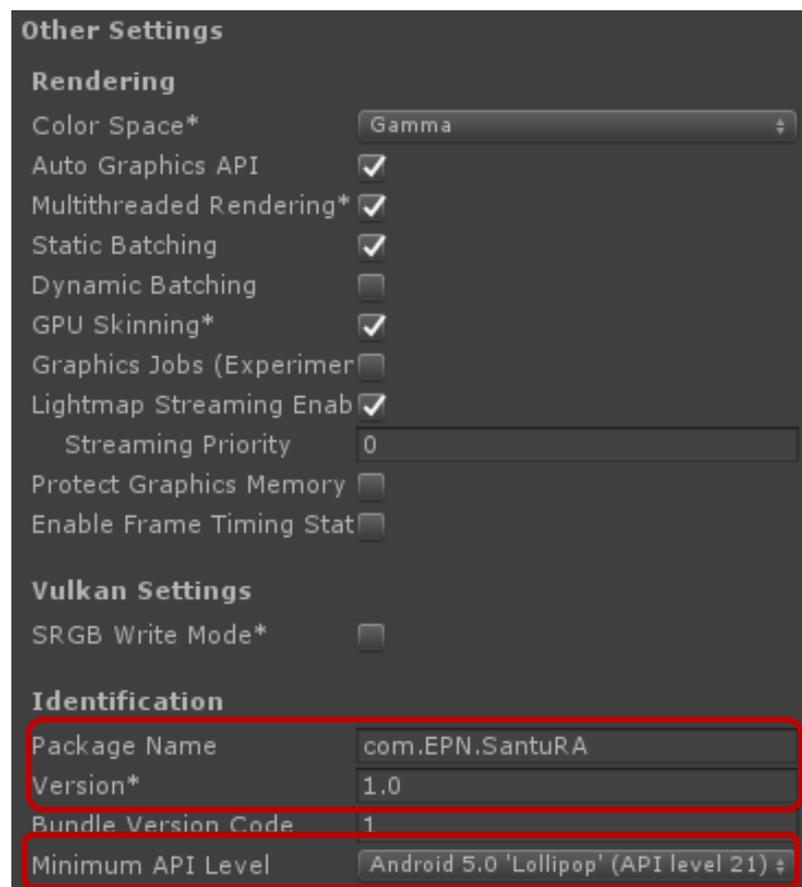


Figura 2.43 Configuración Other Settings

En **Other Settings** también se debe especificar el nombre del producto, de la compañía y la versión de la aplicación para que no se produzca errores al momento de compilar el proyecto, esto se puede visualizar en la Figura 2.43. Además, se estableció el **Mínimum API Level** que es la versión mínima de Android que se requiere para instalar la aplicación, este valor se obtuvo en base a dos elementos que se utilizó y no funcionan en versiones menores a la que se establece en el manual de Unity. Primero es Vuforia la cual requiere una versión mínima a la 4.1.x+ y el componente Video Player la 5.0+, por lo tanto, la versión mínima de Android a la que se podría instalar la aplicación es la 5.0+.

Por último, en la Figura 2.44 se indica la configuración de **XR Settings**, en la cual se habilitó el Kit de desarrollo de software de Vuforia para utilizarlo en el desarrollo de la aplicación.



Figura 2.44 Configuración XR Settings

2.3.6. DESARROLLO DE ESCENAS

En esta sección se implementaron las escenas que forman parte de la aplicación. La aplicación está conformada por seis escenas y son las siguientes:

- Menu.
- EscenaRA.
- MenuRA.
- Información.
- Ayuda.
- EscenaDeCarga.

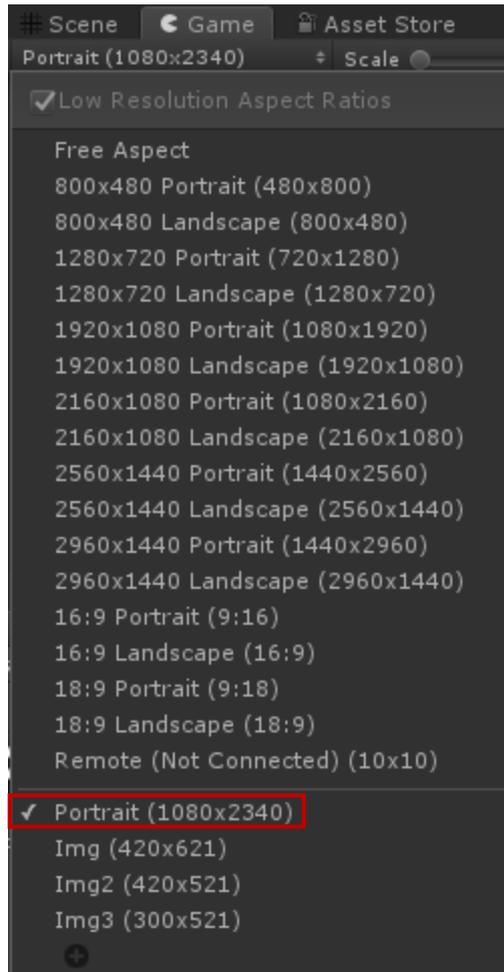


Figura 2.45 Ventana Game

Para el desarrollo de cada una de las escenas que forman parte de la aplicación, se escogió una resolución de 1080x2340 para que se adapte de mejor manera a mayores resoluciones, esto se configura en la ventana Game de Unity tal como se muestra en la Figura 2.45.

En todas las escenas, primero se incluye un CANVAS que es un elemento de Unity en donde deben estar todos los elementos UI¹¹. Este elemento tiene que ser configurado para que se pueda adaptar a cualquier tamaño de pantalla, para ello se dirige a la ventana Inspector de Unity, en la opción **Canvas Scaler**, tal como se muestra en la Figura 2.46. En esta opción se encuentra el parámetro **UI Scale Mode** el cual se debe establecer con **Scale with Screen Size** y en **Reference Resolución** se pone el valor que se colocó en la ventana Game.

¹¹ UI, User Interface

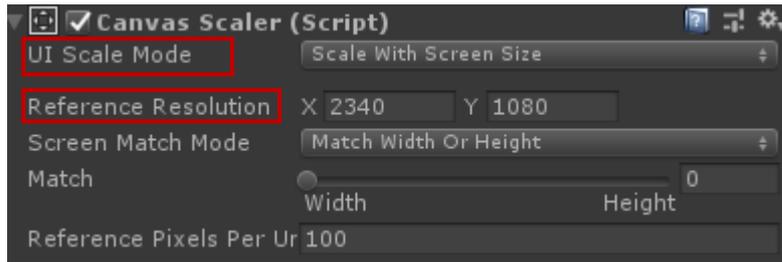


Figura 2.46 Configuración del Canvas

Luego, se incluye un Panel en el cual se adjunta el fondo, para ello se dirige a la ventana Inspector del Panel en **Source Image** se arrastra o se selecciona la imagen que se desea como fondo, tal como se muestra en la Figura 2.47.

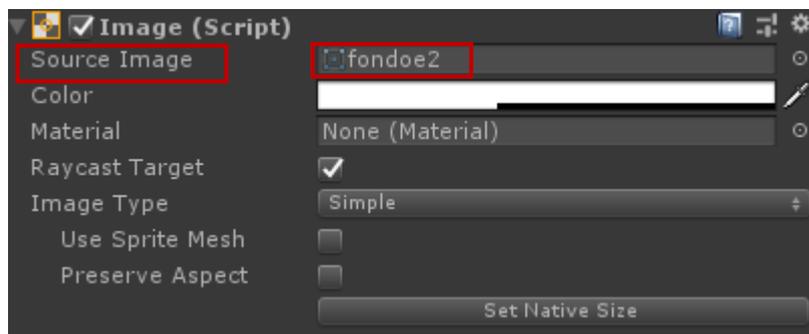


Figura 2.47 Ventana Inspector del Panel

Por último, se incluyen los botones, text o images necesarios para la escena. Todos estos elementos tienen una jerarquía, tal como se muestra en la Figura 2.48. Esta jerarquía permite ubicar el elemento que contiene al otro y es necesario que todos estén dentro del CANVAS. Es importante respetar la jerarquía ya que si los botones no estuvieran dentro del Panel no se visualizarán ya que se encontrarían debajo del mismo.

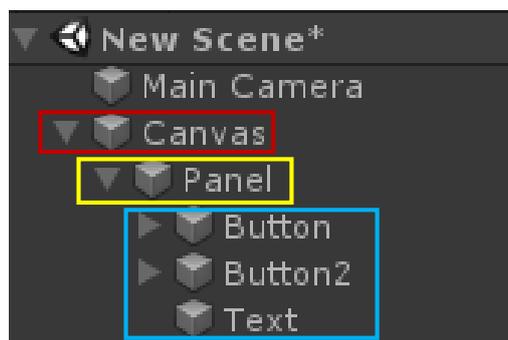


Figura 2.48 Nivel de jerarquía de los elementos UI del Canvas

Como ejemplo, se muestra la escena Información que es la que tiene mayor número de elementos y la escena EscenaRA que es en donde se incluirán los targets que identificará la aplicación. Las imágenes de la ventana de jerarquía y de juego de las escenas restantes se encuentran en el Anexo F.

2.3.1.1 Escena Información

Después de incluir el canvas y el panel con la configuración indicada en la Figura 2.46 y Figura 2.47 se procede a incluir los elementos descritos en la fase de diseño, quedando la ventana de jerarquía de la escena tal como se muestra en la Figura 2.50 y en la Figura 2.49 se visualiza como se ven los elementos en la ventana Game de la escena.

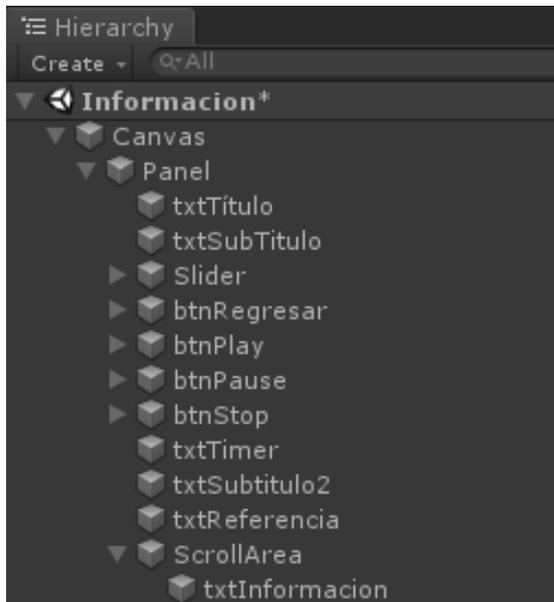


Figura 2.50 Ventana Jerarquía de la escena Información

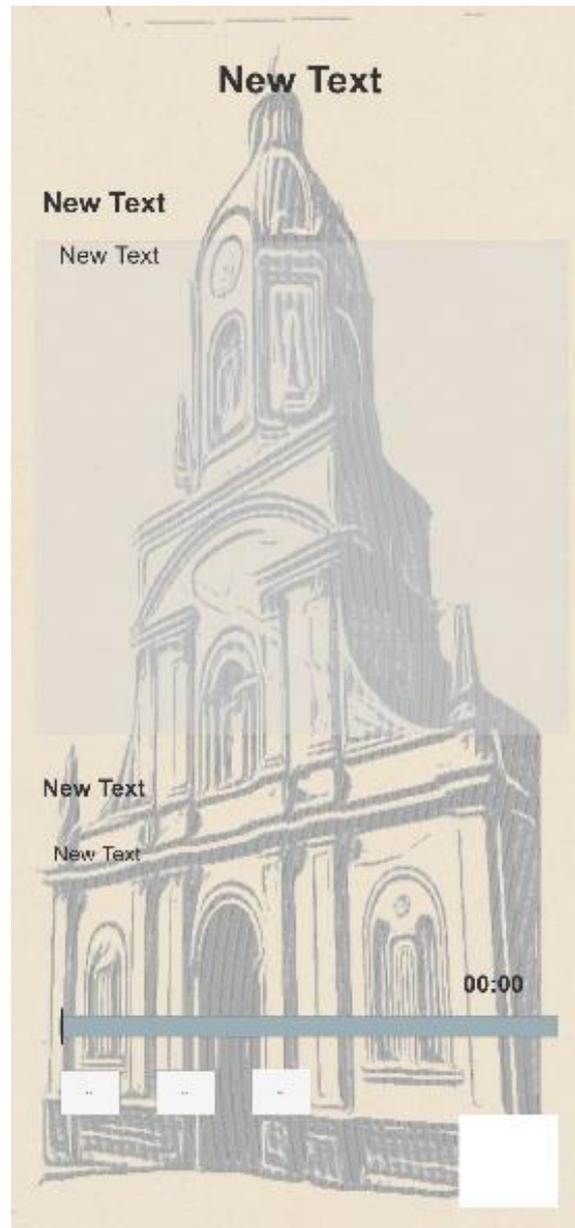


Figura 2.49 Ventana Game de la escena Información

Los elementos Text de la escena se configuraron en la ventana Inspector en la opción Text, ahí se modificó el tamaño de fuente, el tipo de letra, alineación del texto y el color, tal como se muestra en la Figura 2.51.

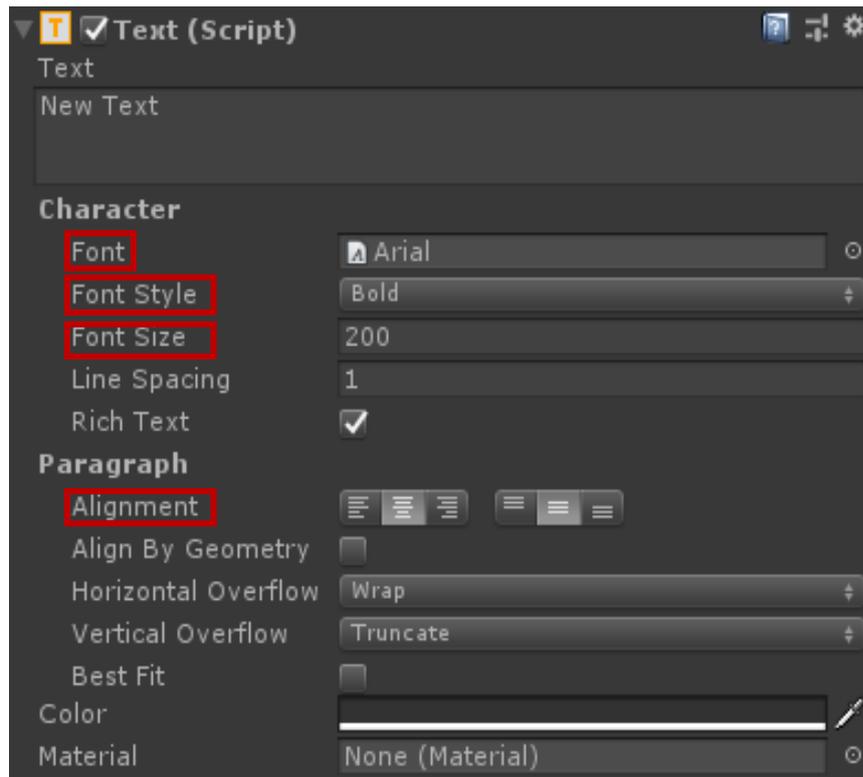


Figura 2.51 Ventana Inspector del elemento Text

El Text `txtInformación` va a contener información que sobrepasará el espacio del elemento text, por lo cual se va a incluir un panel. A este panel en la ventana Inspector se va a añadir el componente **Scroll Rect.**, el cual es utilizado cuando el contenido sobrepasa el espacio donde va a ser mostrado y proporciona la funcionalidad para desplazarse hacia este contenido. También se añadió el componente **Mask** para que solo el contenido dentro del **Scroll Rect** sea visible, tal como se muestra en la Figura 2.52.

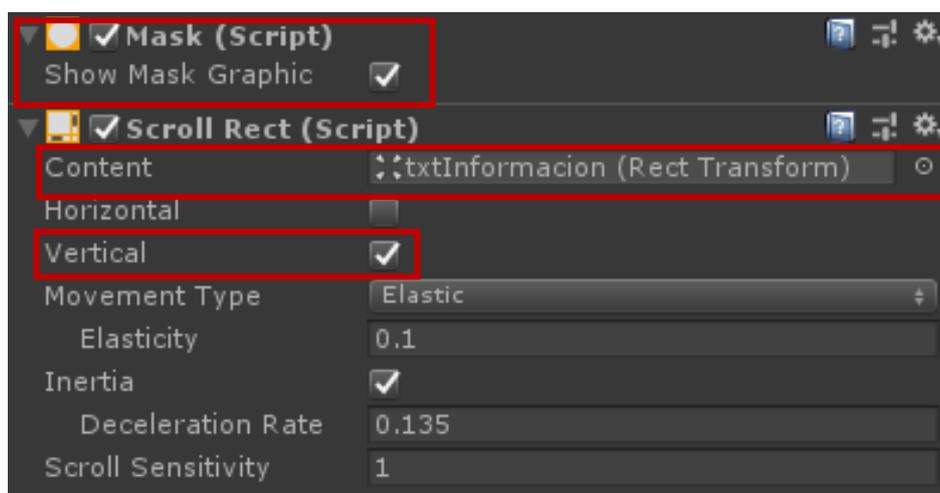


Figura 2.52 Configuración del panel para el elemento txtInformación

El elemento slider está compuesto por los siguientes elementos: el Background, Fill Area y Handle Slide Area, a su vez el Fill Area tiene el elemento Fill y el Handle Slide Area tiene el elemento Handle (Ver Figura 2.54 y 2.55).

- Background: Es una imagen UI que representa el área completa que el **handle** puede atravesar.
- Fill Area: Es un Game Object vacío, cuyo principal objetivo es que su elemento **Fill** esté correctamente alineado. Fill es una imagen UI que se extiende basado en el valor del slider.
- Handle Slide Area: Es un Game Object vacío, cuyo principal objetivo es que su elemento **handle**, esté correctamente posicionado y alineado. **Handle** es una imagen UI que representa el área interactiva del slider.

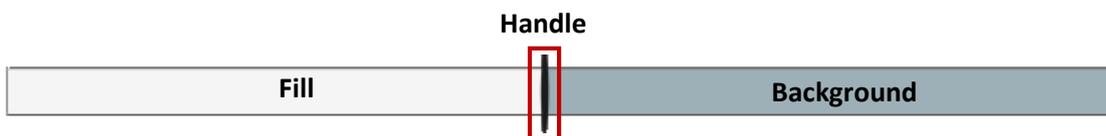


Figura 2.53 Elementos del Slider

Del slider se cambió el color del Background, Fill y Handle en el inspector de cada uno de ellos.



Figura 2.54 Elementos del Slider (Ventana de Jerarquía)

Por último, se añadieron los botones, estos botones tendrán una imagen que será incluida mediante código en el script InfoController.

2.3.1.2 Escena EscenaRA

En la escena se incluye en la ventana de jerarquía el elemento **AR Camera** (Ver Figura 2.55) que permitirá habilitar la funcionalidad de realidad aumentada y se debe eliminar la cámara que viene por defecto al momento de crear la escena. Después, en la Figura 2.56 se muestra la ventana Inspector del elemento **AR Camera** en dónde está la configuración de Vuforia, para lo cual se da click en **Open Vuforia Configuration**.

En la configuración de Vuforia se debe incluir el **App License Key** que se obtiene en el portal de desarrollo de Vuforia y automáticamente aparecerá la base de datos que se importó en la sección 2.3.3, esta configuración se puede visualizar en la Figura 2.57.

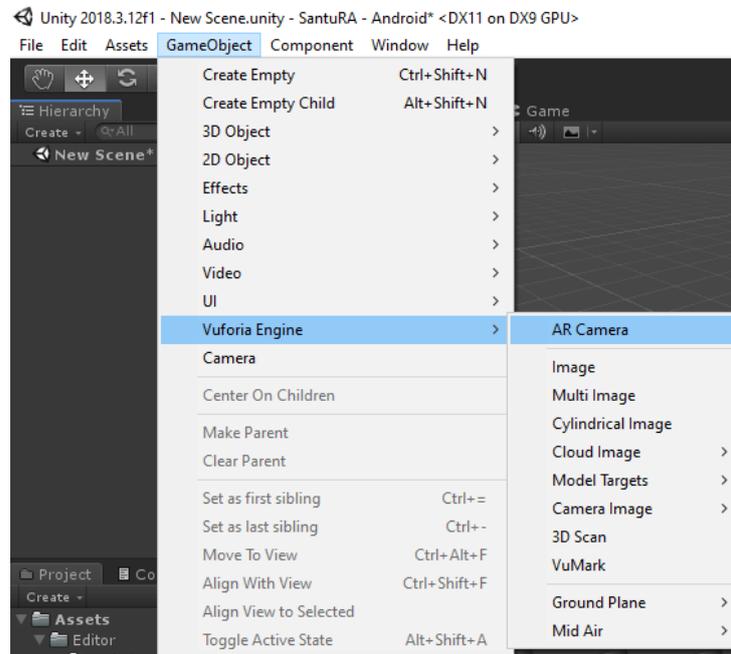


Figura 2.55 Incluir la AR Camera para la escena EscenaRA

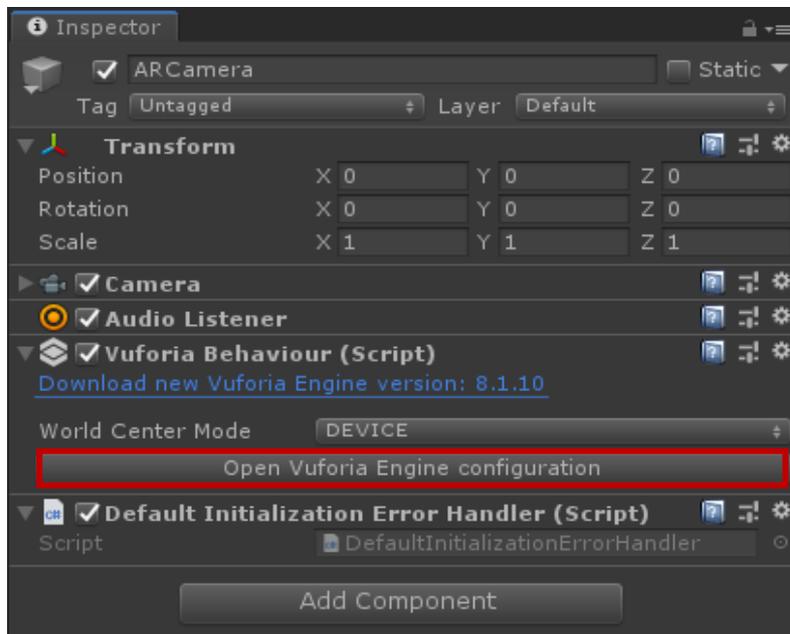


Figura 2.56 Ventana Inspector de AR Camera

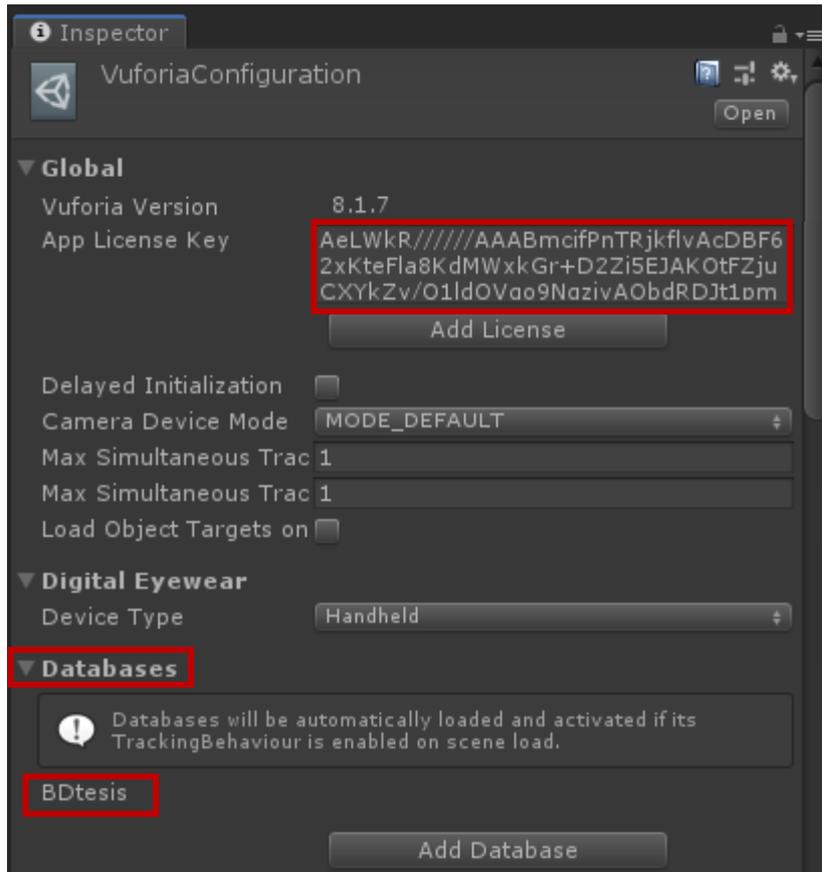


Figura 2.57 Ventana Inspector de la configuración Vuforia

Los pasos realizados para obtener la licencia de desarrollo de Vuforia, fueron los siguientes:

- Ingresar a <https://developer.vuforia.com/vui/auth/login> realizar el respectivo login, una vez que haya ingresado se dirige a **License Manager** y se da click en **Get Development Key**, tal como se muestra en la Figura 2.58.

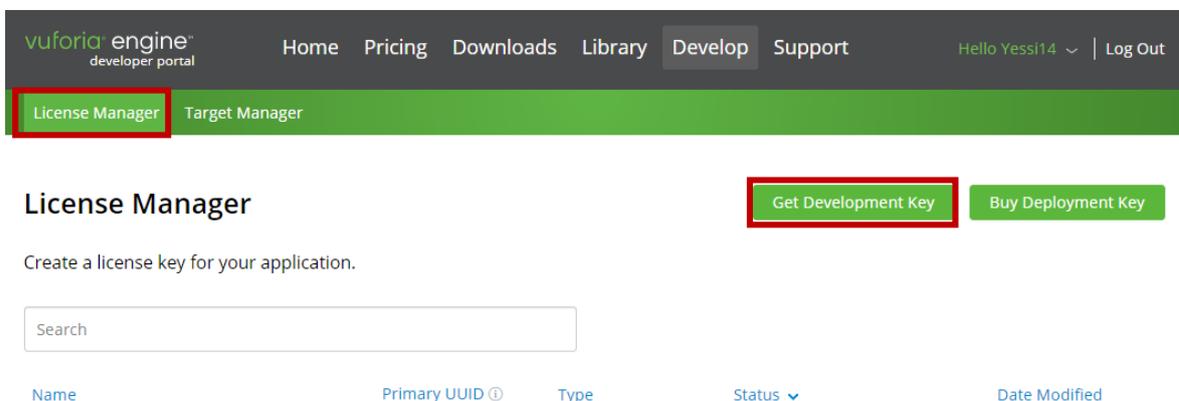


Figura 2.58 Ventana License Manager

- En esta nueva ventana, ver Figura 2.60, se ingresa el nombre de la licencia, se aceptan los términos y se da click en Confirmar, en la Figura 2.59 se muestra la ventana License Manager en donde está la licencia creada con el nombre indicado.

[Back To License Manager](#)

Add a free Development License Key

You can change this later

License Key

Develop
 Price: No Charge
 Reco Usage: 1,000 per month
 Cloud Targets: 1,000
 VuMark Templates: 1 Active
 VuMarks: 100

By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#).

Cancel

Confirm

Figura 2.60 Ventana para crear la licencia

vuforia engine developer portal
Home Pricing Downloads Library **Develop** Support
Hello Yessi14 | Log Out

License Manager
Target Manager

License Manager Get Development Key Buy Deployment Key

Create a license key for your application.

Name	Primary UUID [Ⓞ]	Type	Status [▼]	Date Modified
Proyectos	N/A	Develop	Active	Apr 23, 2019

Figura 2.59 Ventana License Manager con la licencia creada

- Por último, en la Figura 2.61 se encuentra la License Key, esta se copia para utilizarla en el campo **App License Key** de la configuración de Vuforia en Unity.

License Manager > Proyectos

Proyectos [Edit Name](#) [Delete License Key](#)

License Key Usage

Please copy the license key below into your app

```
AeLWkR////////AAABmcoifPnTRjklvAcDBF62xKteFla8KdMWxkGr+D2Z15EJAKOtFZjuCXYkZv/Q1ldOVqo9NqzivAObdRDJt1pmX1y6A/wSP1IKLFPZf0h+F1hBrSXnlrrHLseabbvyGOYytWz9aGBHgwY9/ShPb7JVK+yHNGl3VtY11xdmvaJ1U6jxzuG4lwIZohD51vpL15n3PV1xu7EF3NTkwsG006Vco0XJ4K5vi4IQwMgAwkV80W+79TngMtlmj6aavHG+Yi2vMvDI2rCjdPo0FiJGob1e39pkaJw2Ya9m43r+6iD9qr3Ktd4gLWwge0NoaFwseY4m4Ybb2P3+NySvqAB5PkLL6A3wAh9YxDHaNbtEKB/h+kX2S
```

Plan Type: Develop

Status: Active

Created: Apr 23, 2019 20:28

License UUID: 7244fd4a048f454ca5ba702dc81eaacc

Figura 2.61 Ventana License Key

Una vez configurado el AR Camera con la licencia de Vuforia, se procedió a incluir los Image Target, para lo cual se dará click derecho sobre el AR Camera, seleccionar Vuforia Engine y escoger Image tal como se muestra en la Figura 2.62.

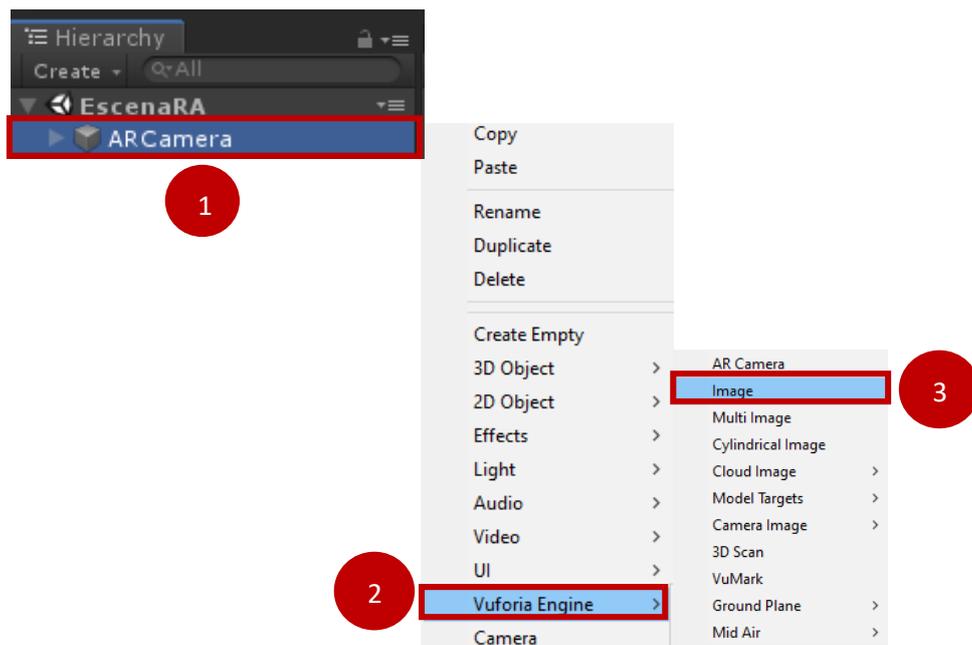


Figura 2.62 Insertar un Image Target en Unity

En la ventana inspector del Image que se añadió dentro de AR Camera se escoge el Image target que se desea adjuntar de la base de datos que se creó en Vuforia, tal como se muestra en la Figura 2.63.

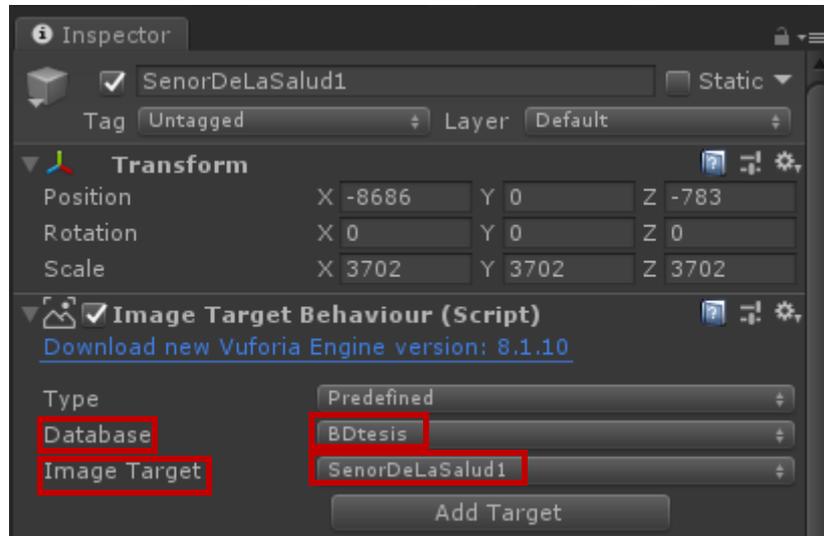


Figura 2.63 Ventana Inspector del Image

A continuación, se crearon tantos Image dentro de AR Camera como Image Targets se tenga en la base de datos que se creó en Vuforia, visualizándose en la Figura 2.64 la Ventana Escena con los Image Targets creados.

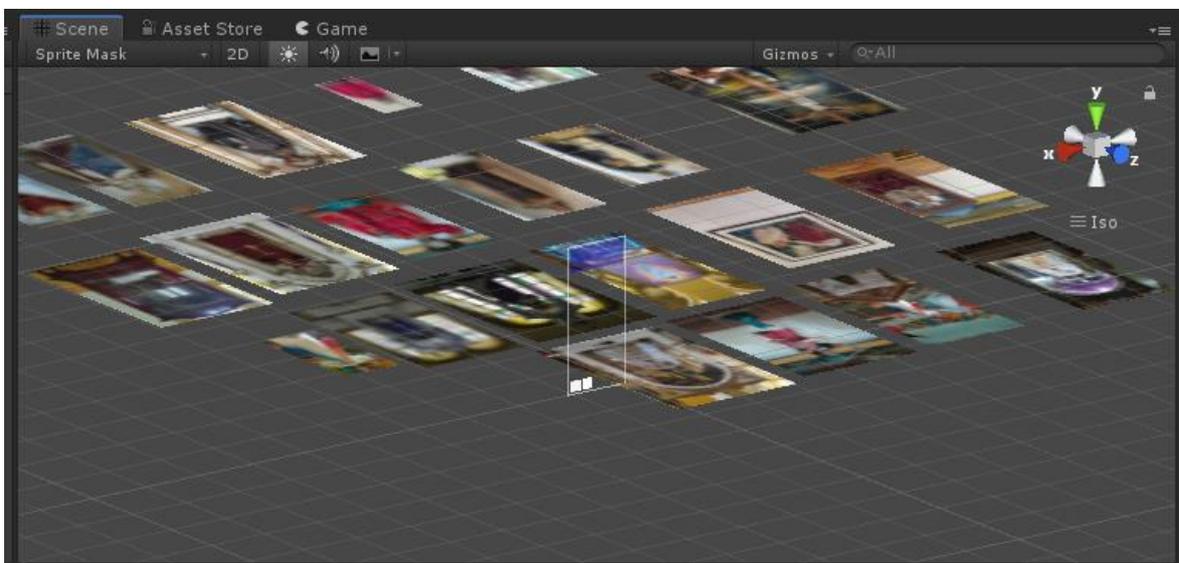


Figura 2.64 Image Targets de la escena EscenaRA

Por último, se incluyó el Canvas que tendrá los elementos que se detalló en la fase de diseño quedando la ventana de Jerarquía como se muestra en la Figura 2.65 y en la Figura 2.66 se visualizará la escena creada en la ventana Game.

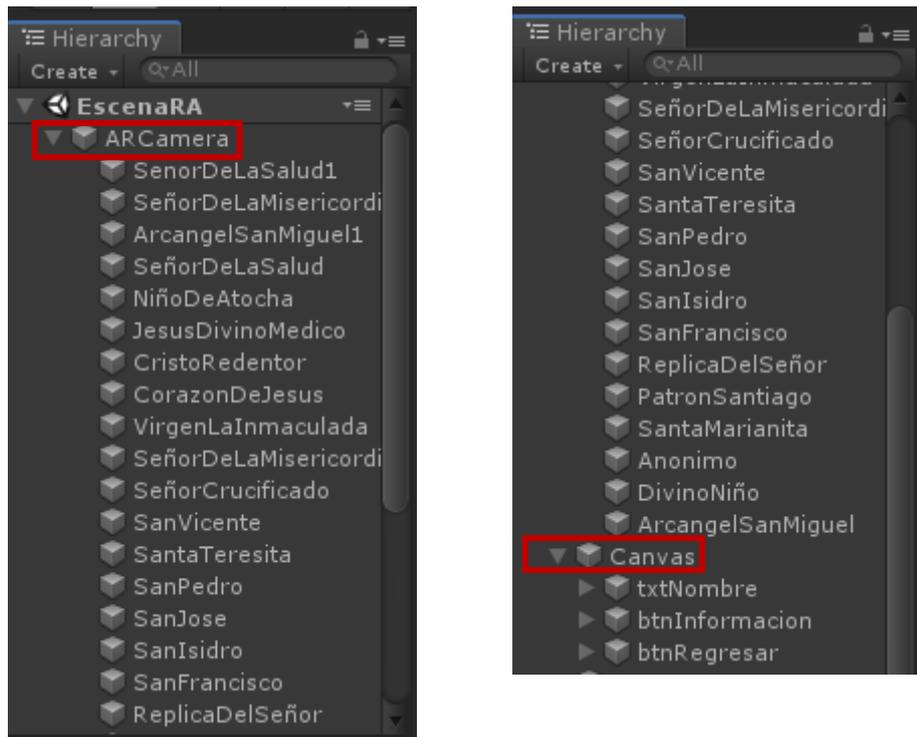


Figura 2.65 Ventana Jerarquía de la escena EscenaRA



Figura 2.66 Ventana Game de la escena EscenaRA

2.3.7. CODIFICACIÓN

En esta sección se presenta el código de cada uno de los scripts que se utilizaron para el funcionamiento de la aplicación

2.3.7.1. Script Menu Controller

Este script está relacionado con la escena Menú, se crea en el Inspector del Canvas en AddComponent y está asociado a la interacción con los botones, es por ello que está formado por cuatro funciones más la función **Start()**.

En el Código 2.1, se observa la declaración de variables, una del tipo **String** la cual será utilizada en el script GuardarEstado que guarda la opción escogida y las cuatro variables del tipo **Button** que permiten incluir las imágenes asociadas a estos botones. Después, se define la función **Start()** en donde se buscarán los componentes que se desean inicializar mediante la función **GameObject.Find("Nombre").GetComponent<Tipo>()** y se cambia la imagen del botón mediante la propiedad **Image.Sprite**, para cargar la imagen se utiliza la función **Resources.Load<Sprite>(string Path)**. Por último, en el Código 2.2 se definen las cuatro funciones correspondientes a los cuatro botones, cada uno de estos abrirá la escena correspondiente, para ello se utilizó la función **SceneManager.LoadSceneAsync(int indice)** la cual permite cargar la escena que está como parámetro de entrada de la función. El índice de la escena se encuentra en la ventana Build Settings tal como se muestra en la Figura 2.67.

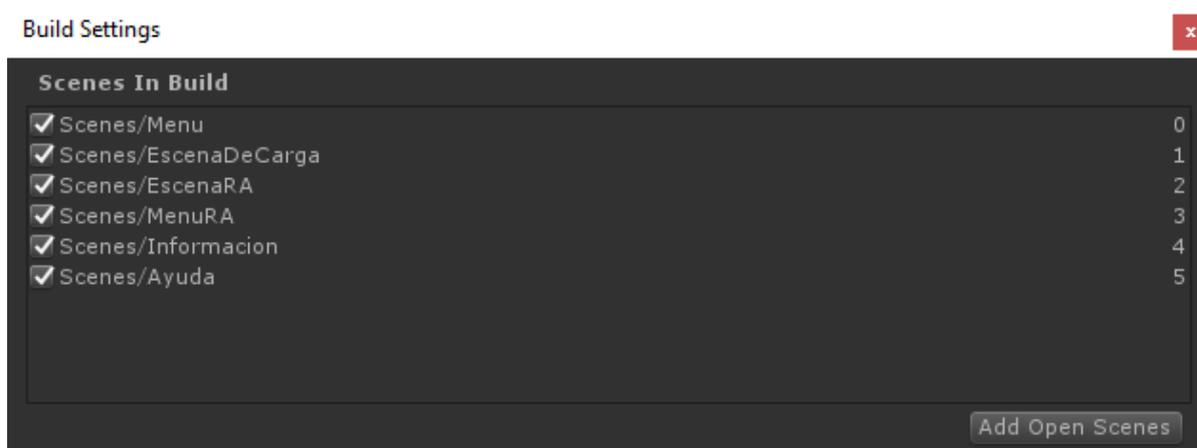


Figura 2.67 Ventana Build Settings con las escenas de la aplicación

Para asociar la función al botón se debe hacer click en el botón en Unity y en la ventana Inspector dirigirse a **onClick()**, dar click en “+” para añadir una función. En **None(Object)** se arrastra el Canvas ya que es el elemento que contiene el script, y en **No Function** se selecciona el script y la función que se desea utilizar para ese botón, esto se puede visualizar en la Figura 2.68.

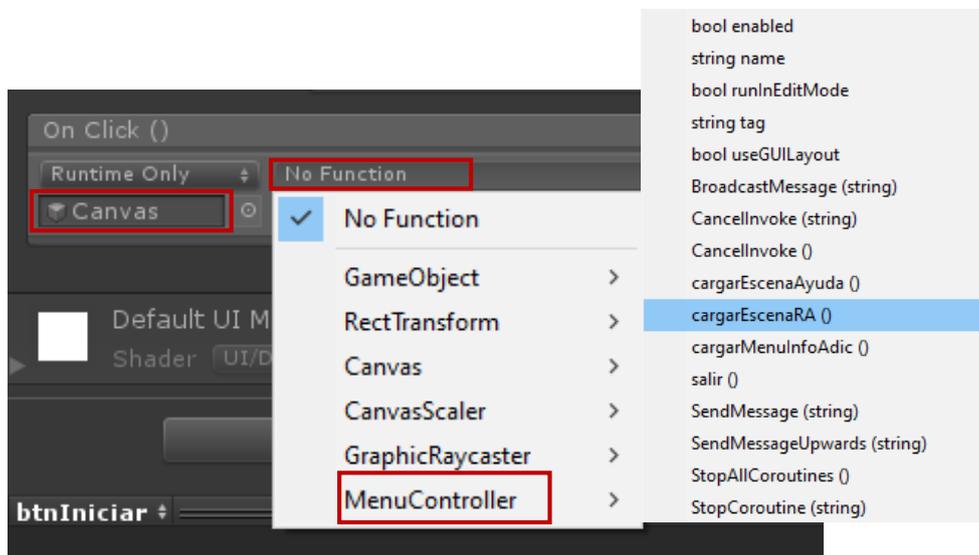


Figura 2.68 Añadir la función cargarEscenaRA() al botón Iniciar

```
//Declaración de variables
public static string opcionM;
Button btnIniciar, btnAyuda, btnIA, btnSalir;

//Inicialización de variables e incluir las imagenes a los botones
private void Start()
{
    btnIniciar = GameObject.Find("btnIniciar").GetComponent<Button>();
    btnAyuda = GameObject.Find("btnAyuda").GetComponent<Button>();
    btnIA = GameObject.Find("btnInfoAdc").GetComponent<Button>();
    btnSalir = GameObject.Find("btnSalir").GetComponent<Button>();
    btnIniciar.image.sprite = Resources.Load<Sprite>("Sprites/iniciarB");
    btnAyuda.image.sprite = Resources.Load<Sprite>("Sprites/Ayuda");
    btnIA.image.sprite = Resources.Load<Sprite>("Sprites/InfAdic");
    btnSalir.image.sprite = Resources.Load<Sprite>("Sprites/Salir");
}
```

Código 2.1 Función Start() del script Menu Controller

```
//Funciones para cambiar de escena y para salir de la aplicación
public void cargarEscenaRA()
{
    SceneManager.LoadSceneAsync(1);
    opcionM = "RA";
}

public void cargarMenuInfoAdic()
{
    SceneManager.LoadSceneAsync(3);
    opcionM = "IA";
}

public void salir()
{
    Application.Quit();
}
```

Código 2.2 Funciones para cambiar de escena del script MenuController

2.3.7.2. Script LoadScene

Este script está relacionado con la escena EscenaDeCarga se crea en el Inspector del Canvas de esta escena en AddComponent. Este script permite cambiar el Fill de la imagen para que mientras vaya cargando vaya apareciendo la imagen y el contador que muestra el avance de la carga.

El script está formado por dos funciones de evento¹². En el Código 2.3 se visualiza la función **Start()** en la cual se buscan los componentes que se desean inicializar, en este caso la imagen y el texto. En el Código 2.4 está la función **Update()** que es utilizada para que se actualice el valor del porcenText que tiene el porcentaje de carga y se incremente el valor del Fill de la imagen utilizando la propiedad **fillAmount**. Una vez que se haya cargado por completo aparecerá la palabra Santura y se abrirá la escena EscenaRA.

```
//Inicialización de variables e
//incluir la imagen a la cual se modificará la propiedad FillAmount
private void Start()
{
    imagen = GameObject.Find("ImgCarga").GetComponent<Image>();
    imagen.sprite = Resources.Load<Sprite>("Sprites/circulo-hueco");
    porcenText = GameObject.Find("txtPorcentaje").GetComponent<Text>();
}
```

Código 2.3 Función Start() del script LoadScene

```
void Update()
{
    if(val1<100)
    {
        //Contador
        val1 += 50 * Time.deltaTime;
        porcenText.text = ((int)val1).ToString() + "%";
    }
    else
    {
        //Al llegar a 100% mostrará Santura y carga la escena EscenaRA
        porcenText.text = "SantuRA";
        SceneManager.LoadScene(2);
    }
    //Modifica el valor de fillAmount en base al contador
    imagen.fillAmount = val1 / 100;
}
```

Código 2.4 Función Update() del script LoadScene

El código completo del script LoadScene se encuentra en el Anexo G.

¹² Se denominan funciones de evento debido a que son activadas por Unity en respuesta a los eventos que ocurren durante la escena.

2.3.7.3. Script Scene Controller

Este script está relacionado con la escena EscenaRA. En el Código 2.5 se visualiza dos funciones, la función **Start()** que permite inicializar la variable y cargar la imagen del botón regresar. Después, se tiene la función **Update()** en donde se guarda el nombre del santo que se ha reconocido mediante el script **DefaultTrackableEventHandler**. Por último, en el Código 2.6 se tiene dos funciones correspondientes a los dos botones de la escena, una que permite cargar la escena **MenuRA** y la otra que permite regresar a la escena **Menu**.

```
//Inicialización del botón regresar
void Start()
{
    btnImgBotonR = GameObject.Find("btnRegresar").GetComponent<Button>();
    btnImgBotonR.image.sprite = Resources.Load<Sprite>("Sprites/BtnRegresarA");
}

//Actualizar el nombre guardado de acuerdo al santo detectado
void Update()
{
    nombre = txtRA.text;
}
```

Código 2.5 Función Start() y Update() del script SceneController

```
//Función asociada al botón Información para cargar la siguiente escena
public void cargarEscena()
{
    SceneManager.LoadScene("MenuRA");
}
//Función para regresar al menú principal
public void Regresar()
{
    SceneManager.LoadScene("Menu");
}
```

Código 2.6 Funciones de cambio de escena del script SceneController

El código completo del script SceneController se encuentra en el Anexo G.

2.3.7.4. Script DefaultTrackableEventHandler

El script DefaultTrackableEventHandler es un script de Vuforia al cual se incorporó segmentos de código para obtener el funcionamiento que se desea. Este script está relacionado a cada uno de los Image Target que se tiene en la escena, el cual es utilizado para ejecutar código durante estados específicos de eventos de seguimiento.

En este script se aumentó el siguiente código:

- Primero se declaró dos variables una del tipo **Text** y otra del tipo **Button**, las cuales corresponden al Text de la escena en dónde se visualizará el nombre de la imagen y al botón que permitirá cargar la siguiente escena (Ver Código 2.7).
- Después, se definen dos funciones **MostrarNombre(Text texto, Button btn)** y **OcultarNombre(Text texto, Button btn)** que permitirán visualizar u ocultar el nombre de la imagen y el botón cuando se ha detectado o perdido el patrón respectivamente (Ver Código 2.7).

```
public class DefaultTrackableEventHandler : MonoBehaviour, ITrackableEventHandler
{
    public Text txtNombre;
    public Button boton;

    public void MostrarNombre(Text texto, Button btn)
    {
        texto.gameObject.SetActive(true);
        btn.gameObject.SetActive(true);
        btn.image.sprite = Resources.Load<Sprite>("Sprites/btnInfo");
    }
    public void OcultarNombre(Text texto, Button btn)
    {
        texto.gameObject.SetActive(false);
        btn.gameObject.SetActive(false);
    }
}
```

Código 2.7 Declaración de variables y definición de métodos del script
DefaultTrackableEventHandler

Por último, se implementa las funciones definidas en el paso anterior en las funciones **OnTrackingFound()** y **OnTrackingLost()** que ya vienen en este script. En el Código 2.8 está la función **OnTrackingFound()**, en la cual se utiliza un condicional para determinar qué target fue detectado, para ello se utiliza la propiedad **TrackableName** que devuelve el nombre del Target y así poder asignar el valor del Text con el nombre de la imagen. En el Código 2.9 se visualiza la función **OnTrackingLost()** dentro de la cual se invoca la función **OcultarNombre(Text texto, Button btn)** para que el text y el botón se oculten cuando el target ya no se haya detectado.

```
PUBLIC_METHODS
#region PROTECTED_METHODS
protected virtual void OnTrackingFound()
{
    if (mTrackableBehaviour.TrackableName == "SenorDeLaSalud1" || mTrackableBehaviour.TrackableName == "SenorDeLaSalud")
    {
        txtNombre.text = "Señor de la Salud";
        MostrarNombre(txtNombre, boton);
    }
}
```

Código 2.8 Método OnTrackingFound()

```
protected virtual void OnTrackingLost()
{
    OcultarNombre(txtNombre, boton);
}
```

Código 2.9 Método OnTrackingLost()

El código completo del script DefaultTrackableEventHandler se encuentra en el Anexo G.

2.3.7.5. Script GuardarEstado

Este script es utilizado en la escena Menú para guardar qué botón se hizo click y de esta manera reutilizar las escenas MenuRA e Información para mostrar diferente tipo de información.

En el script se declaran cuatro variables una del tipo **GuardarEstado** que será utilizada en la función **Awake()** y tres variables del tipo **String** que serán utilizadas en la función **Update()**. En el Código 2.10 está definida la función **Awake()**, dentro de ella se utiliza la función **DontDestroyOnLoad()** para preservar le objeto que contiene este script y de esta manera no se destruya al momento de cambiar de escena. En el Código 2.11 está definida la función **Update()** se guarda el botón que fue presionado en la escena Menu, MenuRA y el nombre de la imagen en la escena EscenaRA.

```
//Se invoca antes del Start, permite que no se destruya el objeto actual
//y esté presente en otras escenas sin duplicarse
private void Awake()
{
    //Condicional para destruir el objeto gEstado duplicado
    if (gEstado == null)
    {
        gEstado = this;
        DontDestroyOnLoad(gameObject);
    }
    else if (gEstado != this)
    {
        Destroy(gameObject);
    }
}
```

Código 2.10 Función Awake() del script GuardarEstado

```

void Update()
{
    //Guarda el nombre del santo que se ha detectado y se requiere la información
    nombreRA = SceneController.nombre;
    //Guarda la opción escogida de la escena escena MebuRA
    opcEscogida = MenuRAcontroller.nombreBoton;
    //Guarda la opción escogida en la escena Menu
    opcEscogida2 = MenuController.opcionM;
}

```

Código 2.11 Función Update() del script GuardarEstado

2.3.7.6. Script MenuRAcontroller

Este script está relacionado con la escena MenuRA, el cual permitirá cargar las imágenes y los text de los botones, el título de la escena y la funcionalidad de los botones dependiendo si se presionó el botón Información Adicional de la escena Menu o el botón Información de la escena EscenaRA.

El script está conformado por cinco funciones, la función **Start()** y las cuatro funciones correspondientes a los botones de la escena. En el Código 2.12 está definida la función **Start()**, en esta función se evalúa mediante un condicional que botón fue presionado para abrir la escena MenuRA en base al valor de la variable **opcEscogida2** del script GuardarEstado.

```

void Start()
{
    txtTitulo = GameObject.Find("Titulo").GetComponent<Text>();
    if (GuardarEstado.opcEscogida2=="IA")
    {
        txtTitulo.text = "INFORMACIÓN ADICIONAL";
        btn1.gameObject.SetActive(true);
        btn2.gameObject.SetActive(true);
        btn3.gameObject.SetActive(false);
        txtBtn1.text = "Biografía Ivo Mora";
        btn1.image.sprite = Resources.Load<Sprite>("Sprites/Biografia");
        txtBtn2.text = "Historia Iglesia";
        btn2.image.sprite = Resources.Load<Sprite>("Sprites/HistoriaI");
    }
    else
    {
        txtTitulo.text = GuardarEstado.nombreRA;
        if (txtTitulo.text == "Cristo Redentor" ||
            txtTitulo.text == "Jesús Divino Médico" || txtTitulo.text == "Señor Crucificado")
        {
            btn1.gameObject.SetActive(false);
            btn3.gameObject.SetActive(false);
            txtBtn2.text = "Historia";
            btn2.image.sprite = Resources.Load<Sprite>("Sprites/historia");
        }
    }
}

```

Código 2.12 Función Start() del script MenuRAcontroller

Tres de las cuatro funciones restantes tienen como objetivo abrir la escena de Información y asignar el valor de la variable **nombreBoton** que permitirá determinar el tipo de información que se cargará en la escena Información. En el Código 2.13 se muestra un ejemplo de este tipo de funciones.

```
public void cambiarEscenaH()
{
    if(GuardarEstado.opcEscogida2=="IA")
    {
        nombreBoton = "biografia";
    }
    else
    {
        nombreBoton = "Historia";
    }
    SceneManager.LoadScene("Informacion");
}
```

Código 2.13 Función cambiarEscenaH() del script MenuRAcontroller

Por último, en el Código 2.14 está definida la función **Regresar()** que permite retornar a la escena anterior desde donde fue presionado el botón que cargó la escena MenuRA, para ello se implementó un condicional.

```
public void Regresar()
{
    if(GuardarEstado.opcEscogida2 == "IA")
    {
        SceneManager.LoadSceneAsync(0);
    }
    else
    {
        SceneManager.LoadSceneAsync(2);
    }
}
```

Código 2.14 Función Regresar() del script
MenuRAcontroller

El código completo del script MenuRAcontroller se encuentra en el Anexo G.

2.3.7.7. Script InfoController

El Script InfoController está relacionado con la escena Información, este script permite cargar la información que el usuario escogió en escenas anteriores y da la funcionalidad a los botones.

En el Código 2.15 se muestra un segmento del código de la función **Start()** en donde se determina si se muestra la información adicional o la historia, milagros u oración de la imagen reconocida; todo esto depende del valor de la variable **opcEscogida** del script **GuardarEstado**.

```

void Start()
{
    //Imágenes de los botones
    btnRegresar.image.sprite = Resources.Load<Sprite>("Sprites/btnRegresarN");
    btnPlay.image.sprite = Resources.Load<Sprite>("Sprites/btnPlay");
    btnPause.image.sprite = Resources.Load<Sprite>("Sprites/btnPause");
    btnStop.image.sprite = Resources.Load<Sprite>("Sprites/btnStop");

    //inicialización de variables
    player = GetComponent<AudioSource>();
    player.clip = clips[index];
    timer = GameObject.Find("txtTimer").GetComponent<Text>();
    sliderMusica = GameObject.Find("Slider").GetComponent<Slider>();
    sliderMusica.minValue = 0;

    //Evalúa el valor de la variable opcEscogida del Script GuardarEstado
    if (GuardarEstado.opcEscogida == "biografía" || GuardarEstado.opcEscogida == "iglesia")
    {
        if(GuardarEstado.opcEscogida=="biografía")...
        else...
    }
    else
    {
        //texto
        titulo.text = GuardarEstado.nombreRA;
        subTitulo.text = GuardarEstado.opcEscogida;
        subTitulo2.text = "Referencias";

        if (titulo.text == "Señor de la Salud" || titulo.text == "Réplica del Señor de la Salud")
    }
}

```

Código 2.15 Función Start() del script InfoController

En el Código 2.16 se visualiza dos funciones, la función **Update()** que invoca a la función **actualizarTimer** que tiene como parámetros de entrada un **AudioSource** que contiene el audio que se va a reproducir y un **Text** donde se visualiza el tiempo que se reproduce el audio.

```

void Update()
{
    ActualizarTimer(player, timer);
}

public void ActualizarTimer(AudioSource auxPlayer, Text auxTimer)
{
    int minutos = (int)auxPlayer.time / 60; //obtiene los minutos del audio
    int segundos = (int)auxPlayer.time % 60; //obtiene los segundos del audio
    //setear el slider con el valor en el que se esta reproduciendo la canción
    sliderMusica.value = auxPlayer.time;
    auxTimer.text = minutos.ToString("00") + ":" + segundos.ToString("00");
}

```

Código 2.16 Función Update y ActualizarTimer del script InfoController

En el Código 2.17 se visualiza la función **ChangedSlider** que permite poner como valor máximo la duración del audio que se va a reproducir, la cual es utilizada en la función **Play()**.

```
//Método para que el valor máximo del slider sea igual a la duración de la pista
public void ChangedSlider(Slider aux, float max)
{
    aux.minValue = 0;
    aux.maxValue = (int)max;
    aux.value = 0; //valor por defecto
}
```

Código 2.17 Función ChangeSlider del script InfoController

En el Código 2.18 se visualiza parte del código de la función **Play()**, esta función reproduce el audio correspondiente a la información visualizada en la escena Información.

```
public void Play()
{
    if (titulo.text == "Señor de la Salud" || titulo.text == "Réplica del Señor de la Salud")
    {
        if (subTitulo.text == "Historia")
        {
            player.clip = clips[17];
            ChangedSlider(sliderMusica, player.clip.length);
            player.Play();
        }
        else if (subTitulo.text == "Milagros")
        {
            player.clip = clips[27];
            ChangedSlider(sliderMusica, player.clip.length);
            player.Play();
        }
        else
        {
            player.clip = clips[29];
            ChangedSlider(sliderMusica, player.clip.length);
            player.Play();
        }
    }
}
```

Código 2.18 Función Play() del script InfoController

En el Código 2.19 se visualizan cuatro funciones, dos corresponden a los botones de detener y pausa que serán utilizadas para el audio que se va a reproducir en la escena. La función **movePosition()** que permite adelantar o retroceder el audio mediante el Slider y la función **Regresar()** que permite volver a la escena MenuRA.

```

public void Stop()
{
    player.Stop();
}

public void Pause()
{
    player.Pause();
}

//Metodo para mover la posición del slider
public void movePosition()
{
    player.time = sliderMusica.value;
}

public void Regresar()
{
    SceneManager.LoadScene("MenuRA");
}

```

Código 2.19 Funciones Stop, Pause, movePosition y Regresar del script InfoController

El Código 2.20 y 2.21 se visualizan dos funciones, la función **Lectura** que permite leer el archivo que tiene la información de cada una de las imágenes que se encuentra en la base de datos de Vuforia; y la función **Buscar** que permite encontrar la información correspondiente a una imagen específica utilizando el parámetro de entrada **nombre**.

```

public string Lectura(int var, StringReader leer)
{
    string cadena = "";
    //si es la primera vez que ingresa a leer
    //se salta a la siguiente que contiene la información
    //ya que está situado en el título
    if (var == 1)
    {
        linea = leer.ReadLine();
        cadena = linea;
        linea = leer.ReadLine();
    }
    //caso contrario sigue con la lectura
    else
    {
        cadena = linea;
        linea = leer.ReadLine();
    }
    return cadena;
}

```

Código 2.20 Función Lectura del script InfoController

```

public void Buscar(string nombre, string var)
{
    StringReader leer = new StringReader(archivo.text);
    int n = 1, f = 1;
    linea = leer.ReadLine();
    string cadena = "", refer = "";
    try
    {
        while (true)
        {
            //evalua si la línea corresponde al santo que se está buscando
            if (linea == nombre)
            {
                //si se cumple la condición se lee la siguiente línea
                linea = leer.ReadLine();
                //se evalúa en base al segundo parámetro de entrada de la función el tipo
                // de información que se desea imprimir
                switch (var)
                {
                    case "Historia":
                        //El punto de parada es Milagros
                        while (linea != "Milagros")
                        {
                            if (linea != "Referencia")
                            {
                                if (f == 1)
                                {
                                    cadena = cadena + "\n" + Lectura(n, leer);
                                    n++;
                                }
                            }
                            else
                                break;
                        }
                    }
                }
            }
        }
    }
}

```

Código 2.21 Función Buscar del script Info Controller

El código completo del script InfoController se encuentra en el Anexo G.

2.3.7.8. Script HelpController

El script HelpController está relacionado con la escena Ayuda, específicamente con el canvas principal, esta escena como se mencionó en la fase de diseño tiene dos canvas. El primer canvas permite mostrar la información inicial y el otro muestra un video. Este script consta de tres funciones, la función **Start**, **HabilitarCanvas** y **Regresar**.

En el Código 2.22 está definida la función **Start()** permite inicializar las variables e incluir el texto que se mostrará en los dos **Text**. En el Código 2.23 está definida la función **Regresar()** la cual está asociada al botón **btnRegresar** que permite cargar la escena Menu. Por último, está la función **HabilitarCanvas()**, la cual deshabilita el canvas actual mediante la propiedad **enable** y habilita el canvas denominado **canvasVideo** que contiene los elementos necesarios para reproducir el video.

```

//Inicialización de variables, imágenes de botones y e información de los Text
void Start()
{
    btnVideo = GameObject.Find("btnVideo").GetComponent<Button>();
    btnVideo.image.sprite = Resources.Load<Sprite>("Sprites/Video");
    btnRegresar = GameObject.Find("btnRegresar").GetComponent<Button>();
    btnRegresar.image.sprite = Resources.Load<Sprite>("Sprites/regresar");
    txtContenido = GameObject.Find("txtContenido").GetComponent<Text>();
    txtContenido.text = "SantuRA, te invita a disfrutar de una nueva experiencia " +
        "en realidad aumntada, utilizando tu celular podrás obtener información " +
        "sobre estatuas (historia, milagros y oración) y pinturas (historia) que " +
        "reposan en uno de los santuarios más visitados en la provincia Bolívar.";
    txtVideo= GameObject.Find("txtSecundario").GetComponent<Text>();
    txtVideo.text = "Conoce más sobre SantuRA en el siguiente video: ";
    canvasV.SetActive(false);
}

```

Código 2.22 Función Start() del script HelpController

```

//Método que permite regresar a la escena anterior
public void Regresar()
{
    SceneManager.LoadSceneAsync(0);
}
//Activa el canvas que contiene el video
public void HabilitarCanvas()
{
    canvas.enabled = false;
    canvasV.SetActive(true);
    canvasVideo.enabled = true;
}

```

Código 2.23 Función Regresar y HabilitarCanvas del script HelpController

El código completo del script HelpController se encuentra en el Anexo G.

2.3.7.9. Script VideoController

El script VideoController está relacionado con la escena Ayuda, específicamente con el **canvasVideo**. Este script presenta dos funciones, la función **cargarEscena** y **Update**, las definiciones de estas funciones se encuentran en el Código 2.24 y Código 2.25 respectivamente. La función **Update()** evalúa si el canvas está activo y habilitado para preparar el video, asignar la textura, reproducir el video y el audio, una vez que haya finalizado el video automáticamente regresará al canvas principal invocando a la función **cargarEscena()**. La función **cargarEscena()** habilita el canvas principal y deshabilita el **canvasVideo**, esta función también es invocada al momento de presionar el botón regresar presente en el **canvasVideo**.

```

public void cargarEscena()
{
    //Habilita el canvas principal
    canvas.enabled = true;
    //Deshabilita el canvas donde se encuentra el video
    canvasVideo.enabled = false;
    GOcanvas.SetActive(false);
    videoPlayer.Stop();
    i = 0;
}

```

Código 2.24 Función cargarEscena() del script VideoController

```

void Update()
{
    //Condional para que se deshabilite el canvasVideo
    if (canvasVideo.isActiveAndEnabled&&!--i)
    {
        //Prepara los recursos necesarios para la reproducción
        videoPlayer.Prepare();
        image.texture = videoPlayer.texture;
        //Reproduce el video
        videoPlayer.Play();
        //Reproduce el sonido
        audioSource.Play();
        i++;
    }
    else
    {
        cargarEscena();
    }
}

```

Código 2.25 Función Update() del script VideoController

El código completo del script VideoController se encuentra en el Anexo G.

2.3.8. CONFIGURACIÓN MÓDULO DE DESCARGA

En esta sección se realiza la configuración del Access Point, portal cautivo y del servidor de descarga con los parámetros analizados en la fase de diseño.

2.3.8.1. Configuración del servidor

El servidor consta de tres servicios, el servicio DNS, DHCP y WEB, la configuración se lo realizará en base a lo descrito en la fase de diseño.

2.3.8.1.1. Servicio DNS

Primero se instala Bind mediante el comando **sudo apt-get install bind9**, luego se procede a editar el archivo de configuración **named.conf.local** que está ubicado en **/etc/bind**. En este archivo se incluye la zona directa e inversa utilizando **santura.com** como dominio, tal como se muestra en la Figura 2.69.

```
GNU nano 2.5.3 File: /etc/bind/named.conf.local
zone "santura.com" {
type master;
file "/etc/bind/db.santura";
};

zone "0.168.192.in-addr.arpa"{
type master;
file "/etc/bind/db.192";
};
```

Figura 2.69 Archivo named.conf.local

Se crean los archivos db.santura y db.192, tomando como base los archivos db.local y db.0 respectivamente, los cuales especificaran la asociación entre nombres y direcciones IP. En este caso se va a incluir el nombre www con dirección IP 192.168.0.2 que corresponderá a la página web, tal como se muestra en la Figura 2.70, se realiza el mismo procedimiento para el archivo db.192 para realizar las consultas inversas, esto se puede visualizar en la Figura 2.72. Después en la Figura 2.71 y Figura 2.73 se muestra el resultado de comprobar que el archivo no contenga errores mediante el comando **named-checkzone** para cada uno de los archivos.

```
GNU nano 2.5.3 File: db.santura
;
;BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA santura.com. root.santura.com. (
1403199 ; Serial
604800 ; Refresh
86400 ; Retry
2419200 ; Expire
604800 ) ; Negative Cache TTL

@ IN NS santura.com.
@ IN A 192.168.0.2
www IN A 192.168.0.2
```

Figura 2.70 Archivo db.santura

```
root@Santura:/etc/bind# named-checkzone santura.com /etc/bind/db.santura
zone santura.com/IN: loaded serial 1403199
OK
```

Figura 2.71 Resultado del comando named-checkzone para el archivo db.santura

```
GNU nano 2.5.3 File: db.192
;
; BIND reverse data file for local loopback interface
;
$TTL 604800
@ IN SOA santura.com. root.santura.com. (
    1 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS santura.com.
2 IN PTR santura.com.
2 IN PTR www.santura.com.
```

Figura 2.72 Archivo db.192

```
root@Santura:/etc/bind# named-checkzone 0.168.192.in-addr.arpa /etc/bind/db.192
zone 0.168.192.in-addr.arpa/IN: loaded serial 1
OK
```

Figura 2.73 Resultado del comando named-checkzone para el archivo db.192

Por último, se reinició el DNS mediante el comando `/etc/init.d/bind9 restart`.

2.3.8.1.2. Servicio DHCP

Se instaló el servicio DHCP mediante el comando `apt-get install isc-dhcp-server`, después se edita el archivo `dhcpd.conf`, este se encuentra en `/etc/dhcp`. En la Figura 2.74 se encuentra la primera parte del archivo en donde se llenaron los siguientes campos:

- **option domain-name:** Indica el nombre de dominio.
- **option domain-name-servers:** Establece las direcciones IP de los servidores DNS.
- **default-lease-time:** Establece el tiempo por defecto que se mantiene la asignación de direcciones.
- **max-lease-time:** Establece el tiempo máximo que se mantiene la asignación de direcciones.

```
GNU nano 2.5.3 File: /etc/dhcp/dhcpd.conf
#
# Sample configuration file for ISC dhcpd for Debian
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
#
# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks...
option domain-name "santura.com";
option domain-name-servers 192.168.0.2;

default-lease-time 18000;
max-lease-time 28800;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

[ Read 118 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^V Prev Page
^X Exit ^R Read File ^M Replace ^U Uncut Text ^T To Spell ^G Go To Line ^U Next Page
```

Figura 2.74 Archivo dhcpd.conf (PARTE I)

En la Figura 2.75 se visualiza la segunda parte del archivo en donde se agrega el rango de direcciones que se utilizarán para asignar a los dispositivos que se conecten al servidor.

```
GNU nano 2.5.3 File: /etc/dhcp/dhcpd.conf
# pool {
#   allow members of "foo";
#   range 10.17.224.10 10.17.224.250;
# }
# pool {
#   deny members of "foo";
#   range 10.0.29.10 10.0.29.230;
# }
#}

subnet 192.168.0.0 netmask 255.255.255.0 {
option domain-name "santura.com";
option broadcast-address 192.168.0.255;
option routers 192.168.0.1;
range 192.168.0.4 192.168.0.254;
}

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^V Prev Page
^X Exit ^R Read File ^M Replace ^U Uncut Text ^T To Spell ^G Go To Line ^U Next Page
```

Figura 2.75 Archivo dhcpd.conf (PARTE II)

2.3.8.1.3. Servicio WEB

Para instalar Apache se utilizó el comando **apt-get install apache2**, después se estableció el nombre del servidor modificando la línea setenta del archivo **apache2.conf** que se encuentra en **/etc/apache/apache2.conf**, tal como se muestra en la Figura 2.76.

```
GNU nano 2.5.3 File: /etc/apache2/apache2.conf Modified
# Global configuration
#
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE! If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the Mutex documentation (available
# at <URL:http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
#ServerRoot "/etc/apache2"
#ServerName www.santura.com
#
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
Mutex file:${APACHE_LOCK_DIR} default
#
# PidFile: The file in which the server should record its process
# identification number when it starts.
# This needs to be set in /etc/apache2/envvars
#
PidFile ${APACHE_PID_FILE}
#
# Timeout: The number of seconds before receives and sends time out.
#
^G Get Help ^O Write Out ^U Where Is ^K Cut Text ^J Justify ^C Cur Pos ^V Prev Page
^X Exit ^R Read File ^N Replace ^U Uncut Text ^I To Spell ^G Go To Line ^U Next Page
```

Figura 2.76 Archivo apache2.conf

A continuación, en la Figura 2.77 se visualiza el archivo **dir.conf** en donde se establece el archivo por defecto que se entrega, este archivo se encuentra en **/etc/apache2/mods-enabled**.

```
GNU nano 2.5.3 File: /etc/apache2/mods-enabled/dir.conf
<IfModule mod_dir.c>
  DirectoryIndex index.html index.htm
</IfModule>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Figura 2.77 Archivo dir.conf

En la ruta `/var/www/html` se encuentra el archivo `index.html`, en esta ubicación se procede a crear dos directorios, uno para guardar el archivo con extensión `.css` y otro para las imágenes tal como se muestra en la Figura 2.78, estos serán utilizados en el archivo `index.html`.

```
root@Santura:/var/www/html# ls
css  download  imagenes  index.html
```

Figura 2.78 Directorios creados en `/var/www/html`

El Código 2.26 corresponde al archivo `estilos.css`, el cual fue creado en el directorio `css` y permite describir el estilo del archivo `index.html`. En este archivo se definió seis selectores (`body`, `header`, `img`, `a`, `footer`, `button`) que corresponden a las etiquetas que hay en el archivo. En cada una de ellas dentro de las llaves se modifica los valores de las propiedades, tal como se muestra en la Figura 2.79.



Figura 2.79 Regla CSS

Las propiedades que se utilizaron para realizar el archivo son las siguientes:

- `color`: Color del texto.
- `font-size`: Tamaño del texto.
- `text-align`: Alineación del texto (`center`, `justify`, `left`, `right`).
- `font-weight`: Negrita para el texto.
- `border`: Cuatro bordes, se puede especificar el color, el grosor y estilo del borde.
- `border-radius`: Esquinas redondeadas.
- `padding`: Cuatro márgenes internos.
- `width`: Ancho.
- `height`: Alto.

- margin: Cuatro márgenes exteriores.

```

GNU nano 2.5.3      File: /var/www/html/css/estilos.css
body{
  background:#e6e6ff ;
  color: Black;
}
header{
  background: Black;
  color: White;
  padding: 1px 0px;
  text-align: center;
  font-size: 18px
}
img{
  width: 300px;
  height: auto;
  margin: auto;
  display: block;
  padding: 10px;
  border: 2px solid #ccc;
}
a{
  font-size: 20px;
}
footer{
  background: Black;
  color: White;
  padding: 1px 0px;
  text-align: center;
}
center a button{
  border-radius: 12px;
  background-color: #000066;
  color: #fff;
  padding: 12px 28px;
  font-weight: bold;
}
^G Get Help      ^O Write Out    ^U Where Is     ^K Cut Text     ^J Justify     ^C Cur Pos     ^V Prev Page
^X Exit          ^R Read File    ^_ Replace      ^U Uncut Text   ^I To Spell    ^_ Go To Line  ^U Next Page

```

Código 2.26 Archivo estilos.css

En el Código 2.27 se visualiza el código del archivo **index.html** se indica el contenido que va a tener la página web, la cual se divide en dos partes la cabecera (**<head> </head>**) y el cuerpo (**<body> </body>**). En la cabecera contiene información sobre el documento esto no se visualiza en el navegador, en este caso se incluyó etiquetas para mostrar las tildes y la ñ, para referenciar el archivo estilos.css, el título y el ícono. En el cuerpo va el contenido del documento, con sus respectivas etiquetas.

```

GNU nano 2.5.3 File: /var/www/html/index.html
<!-- Etiqueta para definir la raíz del documento -->
<html>
  <head>
    <!-- Etiqueta para poder visualizar la ñ y las letras con tile -->
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <!-- Etiqueta para mostrar el título de la página en la pestaña de navegación-->
    <title>
      SANTURA
    </title>
    <!-- Etiqueta para incluir el icono de la página, mostrándolo en
    el encabezado de la pestaña -->
    <link rel="shortcut icon" type="image/x-icon" href="imagenes/IconoIglesia1.ico">
    <!-- Etiqueta para hacer referencia al archivo estilos.css -->
    <link rel="stylesheet" type="text/css" href="css/estilos.css">
  </head>
  <body>
    <!-- Etiqueta para definir el encabezado del documento-->
    <header>
      <p>Santuario del Señor de la Salud</p>
    </header>
    <h1>
      SantuRA
    </h1>
    <p>Santura es una aplicación desarrollada en Adroid para conocer la
      historia, milagros y oraciones de los santos que se
      encuentran en el Santuario.</p>
    <p>Para descargar la aplicación dar click en el botón Descargar.</p>
    <!-- Etiqueta para definir saltos de línea -->
    <br>
    <!-- Etiqueta para adjuntar una imagen al documento -->
    
    <br>
    <!-- Etiqueta para poner en el centro el enlace de descarga-->
    <center>
      <!-- Etiqueta que permite incluir un enlace el cual permitirá descargar el $
      de la aplicación SantuRA -->
      <a href="download/hola" download="SantuRA" >
        <button type="button">
          Descargar
        </button>
      </a>
    </center>
    <br>
    <!-- Etiqueta para el pie de página -->
    <footer>
      <p>Santiago de Bolívar</p>
    </footer>
  </body>

```

[^]G Get Help [^]O Write Out [^]W Where Is [^]K Cut Text [^]J Justify [^]C Cur Pos [^]Y Prev Page
[^]X Exit [^]R Read File [^]U Replace [^]U Uncut Text [^]T To Spell [^]_ Go To Line [^]V Next Page

Código 2.27 Archivo index.html

2.3.8.2. Configuración del Access Point

Como se mencionó en la fase de diseño, se utilizará el software gratuito que proporciona TP-Link para la configuración del Access Point.

Una vez que se haya instalado el software Omada Controller se procede adoptar el AP y aparecerá como activo, luego se dará click en el botón editar para iniciar la configuración del AP, tal como se muestra en la Figura 2.80.

AP Name	MAC Address	IP Address	Status	Model	Hardware Version	Firmware Version	Channel	Client Number	Download	Upload	Action
98-DA-C4-80-39-5B	98-DA-C4-80-39-5B	192.168.0.254	Connected	EAP115(US)	4.0	3.3.0 Build 20190301 Rel. 37101	1(2.4G)	0	0 Bytes	0 Bytes	[Icons]

Figura 2.80 AP activo en Omada Controller

Primero, se configura el nombre del AP para poder identificarlo, en caso de que se incluyan más AP's dentro de la red esto se puede visualizar en la Figura 2.81.

Basic Config

Name:

Figura 2.81 Configuración básica AP

Luego, en la Figura 2.82 se visualiza la configuración IP utilizando los valores que se detalló en la etapa de diseño para la IP, máscara y servidor DNS.

IP Setting

DHCP Static

IP Address:

IP Mask:

Gateway:

Preferred DNS server: (Optional)

Alternate DNS server: (Optional)

Figura 2.82 Configuración IP del AP

Después, en la Figura 2.83 se visualiza la configuración de radio, en dónde se establece el modo, ancho de canal, canal y potencia de transmisión.

Radio ⤴

2.4GHz 5GHz

Status: Enable

Mode: 802.11b/g/n mixed ▼

Channel Width: 20 / 40MHz ▼

Channel: Auto ▼

Tx Power: Medium ▼

Apply

Figura 2.83 Configuración de Radio del AP

La Figura 2.84 corresponde a la configuración de balance de carga que restringe el número de usuarios que pueden conectarse al AP y por consecuencia al servidor este valor se fijó en 200 basándose en el criterio de la sección 2.2.6.

Load Balance ⤴

2.4GHz 5GHz

Max Associated Clients: Enable

(1-255)

RSSI Threshold: Enable ?

(-95-0 dBm)

Apply

Figura 2.84 Configuración de balance de carga

Por último, en la Figura 2.85 se configuró el SSID accediendo a Wireless Settings, en donde se establece el nombre SSID, seguridad y password.

Add SSID
✕

Basic Info ⤴

SSID Name:

Band: 2.4GHz 5GHz

Guest Network: Enable ?

Security Mode:

Wireless Password: 🗑

Advanced Settings ⤵

Apply

Figura 2.85 Configuración SSID

2.3.8.3. Configuración del Portal Cautivo

En la fase de diseño se estableció un portal cautivo por hardware que proporciona el AP TP-Link, para ello se accede a Wireless Control y se inicia la configuración. En la configuración se establece el nombre del portal y se relaciona con el SSID que se creó en la configuración del AP. Después se establece el tipo de autenticación, en este caso no es necesario y por último se habilita la opción de redirección y la URI de redirección que será la página web en donde se podrá descargar el APK de la aplicación, tal como se muestra en la Figura 2.86.

Basic Info

Portal Name:

SSID:

Authentication Type:

Authentication Timeout:

Daily Limit ?

HTTPS Redirect: Enable ?

Redirect: Enable ?

Redirect URL:

Figura 2.86 Configuración portal cautivo

3. RESULTADOS Y DISCUSIÓN

En el presente capítulo, se mostrará las pruebas de los diferentes prototipos que se desarrollaron en la fase de Taller de Diseño para obtener el prototipo final que se implementó. Las pruebas correspondientes de los módulos de realidad aumentada, interacción con el usuario, audio, ayuda y descarga del prototipo final.

3.1. EVALUACIÓN DE LOS PROTOTIPOS

Como se mencionó en el marco teórico, la metodología RAD se basa en la creación de prototipos que permite descubrir las necesidades de los usuarios. Estos prototipos fueron evaluados por tres personas que forman parte del GAD-PR de Santiago y el Padre de la parroquia, utilizando como formato la Tabla 3.1, cuyos campos son los siguientes:

- Nombre del Observador: Es la persona encargada de mostrar el prototipo y responsable de llenar la tabla con la experiencia del usuario.
- Fecha: Es la fecha en la que se realizó la prueba del prototipo.
- Nombre del Proyecto.
- Ubicación: En dónde se utilizará la aplicación.
- Nombre del programa.
- Nombre del Usuario.
- Cargo.
- Reacción del usuario: En la reacción del usuario se puede tener tres posibles opciones: Mala (El usuario no está convencido de la funcionalidad del prototipo, se requieren modificaciones), Buena (El usuario está conforme, pero requiere modificaciones) y Muy Buena (No se requiere modificaciones).
- Sugerencia del usuario: Son las modificaciones que el usuario desea realizar sobre el prototipo.
- Firma.
- Plan de revisión: Fecha en la cual se probará el siguiente prototipo con las modificaciones realizadas.

Tabla 3.1 Formato para la evaluación de los prototipos [7]

FORMULARIO DE EVALUACIÓN DEL PROTOTIPO				
Nombre del observador:			Fecha:	
Nombre del proyecto:			Empresa o ubicación:	
Nombre del programa:			Número de prototipo:	
	Usuario1	Usuario 2	Usuario 3	Usuario 4
Nombre del usuario				
Cargo				
Reacción del usuario				
Sugerencia del usuario				
Firma				
Plan de revisión				

3.1.1. PROTOTIPO NÚMERO UNO

En este prototipo se evaluó el funcionamiento de las escenas que conforman la aplicación.

Estas escenas son las siguientes:

- Escena Menu: Escena principal en donde puede dirigirse a la experiencia de realidad aumentada, información adicional, ayuda o salir de la aplicación (Figura 3.1).
- Escena Ayuda: Escena en donde se indica el funcionamiento de la aplicación (Figura 3.2).
- Escena EscenaRA: Escena que activa la cámara del dispositivo móvil y en donde podrá enfocar la imagen para obtener información adicional presionando el botón Información (Figura 3.3).
- Escena MenuRA: Escena que permitirá escoger la información que desea visualizar de la imagen reconocida, el número de botones dependerá si es una pintura o una estatua (Figura 3.4 y Figura 3.5).
- Escena Información: Escena con la información escogida en el MenuRA, con la posibilidad de poder escuchar en vez de leer (Figura 3.6).

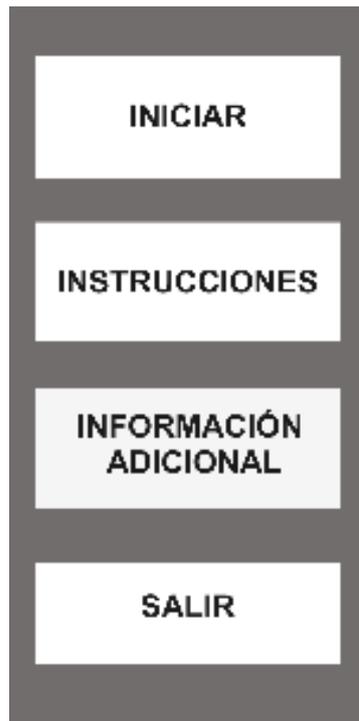


Figura 3.1 Escena Menu (Prototipo 1)

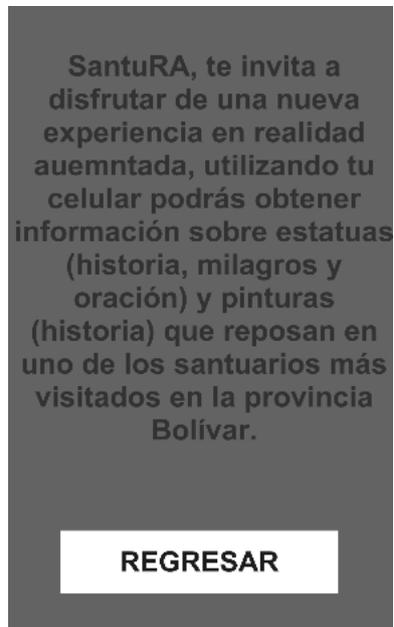


Figura 3.2 Escena Ayuda (Prototipo 1)



Figura 3.3 EscenaRA (Prototipo 1)



Figura 3.4 MenuRA de una pintura (Prototipo 1)



Figura 3.5 MenuRA de una escultura (Prototipo 1)

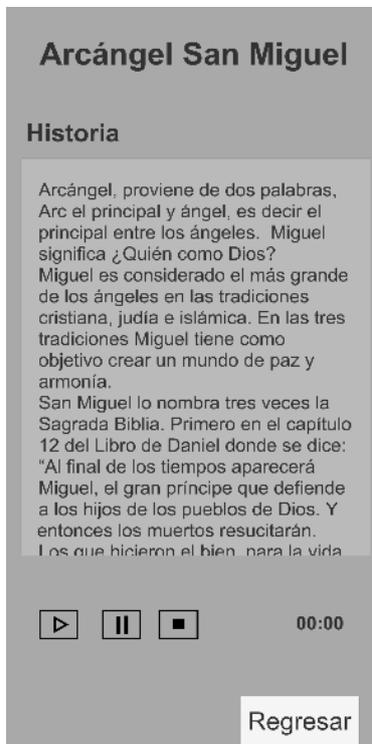


Figura 3.8 Escena

Información con la historia de una escultura (Prototipo 1)



Figura 3.7 Escena

Información con los milagros de una escultura (Prototipo 1)

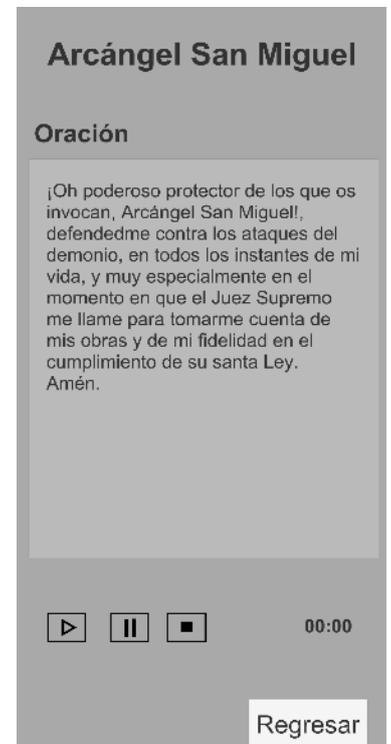


Figura 3.6 Escena

Información con la oración de una escultura (Prototipo 1)

Tabla 3.2 Evaluación del prototipo número uno

FORMULARIO DE EVALUACIÓN DEL PROTOTIPO					
Nombre del observador:		Yessica Cruz		Fecha:	17/01/2020
Nombre del proyecto:			Empresa o ubicación:		
Aplicación en Realidad Aumentada			Santuario del Señor de la Salud		
Nombre del programa:			Número de prototipo		
SantuRA			1		
	Usuario1	Usuario 2	Usuario 3	Usuario 4	
Nombre del usuario	Señor Luis Cuji Macas	Ingeniero Freddy Malatay Caibe	Licenciado Euclides Ramírez	Padre Carlos Solarte	
Cargo	Presidente del GAD-PR de Santiago	Secretario del GAD-PR de Santiago	Vocal del GAD-PR de Santiago	Párroco del Santuario del Señor de la Salud	
Reacción del usuario	Muy Buena	Buena	Buena	Muy Buena	
Sugerencia del usuario	Sin Comentarios	Se debe incorporar un slider que permita ubicar en cualquier posición del audio	Mostrar las referencias de la información visualizada	Sin Comentarios	
Firma					
Plan de revisión		30/01/2020	30/01/2020		

La Tabla 3.2 corresponde a la evaluación del prototipo con el personal del GAD y el párroco el día 17 de enero del 2020. Los datos obtenidos de la evaluación, son la aprobación por parte de dos de los usuarios y dos retroalimentaciones para mejorar el funcionamiento del prototipo cuya próxima evaluación se fijo en la fecha 30 de enero de 2020.

3.1.2. PROTOTIPO NÚMERO DOS

En la evaluación del prototipo número uno, se obtuvo dos correcciones correspondientes a la escena Información y son los siguientes:

- Incorporar un slider que les permita ubicar en cualquier posición del audio.
- Incorporar un Text para mostrar las referencias de la información visualizada.

El nuevo prototipo incluyó los nuevos cambios descritos en la Tabla 3.2 tal como se muestra en la Figura 3.9.

En la escena Información correspondiente al prototipo número dos, se incorporó el slider para el audio y un text para las referencias de la información visualizada.

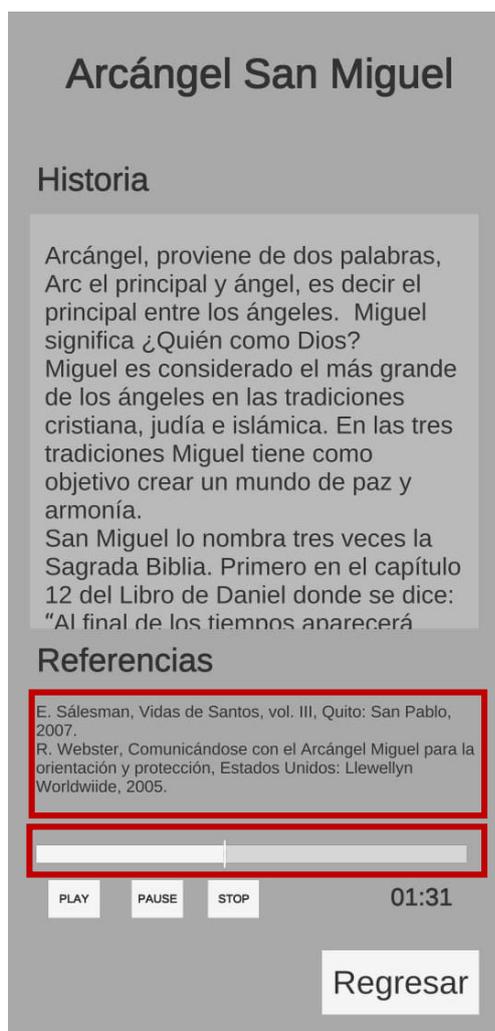


Figura 3.9 Escena Información (Prototipo 2)

La Tabla 3.3 corresponde a la evaluación del prototipo con dos personas del GAD quienes comentaron los cambios del prototipo anterior y se evaluaron el día 30 de enero del 2020. Los datos obtenidos de la evaluación, son dos retroalimentaciones por parte del personal del GAD para mejorar el funcionamiento del prototipo cuya próxima evaluación se fijó en la fecha 12 de febrero de 2020.

Tabla 3.3 Evaluación del prototipo número dos

FORMULARIO DE EVALUACIÓN DEL PROTOTIPO					
Nombre del observador:		Yessica Cruz		Fecha:	30/01/2020
Nombre del proyecto:			Empresa o ubicación:		
Aplicación en Realidad Aumentada			Santuario del Señor de la Salud		
Nombre del programa:			Número de prototipo:		
SantuRA			2		
	Usuario1	Usuario 2	Usuario 3	Usuario 4	
Nombre del usuario	Señor Luis Cuji Macas	Ingeniero Freddy Malatay Caibe	Licenciado Euclides Ramírez	Padre Carlos Solarte	
Cargo	Presidente del GAD-PR de Santiado	Secretario del GAD-PR de Santiago	Vocal del GAD-PR de Santiago	Párroco del Santuario del Señor de la Salud	
Reacción del usuario	Buena	Buena	Muy Buena	Muy Buena	
Sugerencia del usuario	Sin Comentarios	El botón de información debería aparecer cuando la imagen es reconocida	El nombre de la imagen reconocida debe ir en un cuadro con fondo para que pueda ser visualizada de mejor manera.	Sin Comentarios	
Firma					
Plan de revisión		12/02/2020	12/02/2020		

3.1.3. PROTOTIPO NÚMERO TRES

En la evaluación del prototipo número dos, se obtuvo dos correcciones correspondientes a la escena EscenaRA y son los siguientes:

- El botón Información debe aparecer cuando la imagen es reconocida.
- El nombre de la imagen reconocida debe ir en un cuadro con fondo para mejor visualización.

El nuevo prototipo incluirá los nuevos cambios descritos en la Tabla 3.3 tal como se muestra en la Figura 3.10 y Figura 3.11.

En la escena EscenaRA se incorporó un fondo de color azul en el Text para que se pueda visualizar de mejor manera el nombre de la imagen que se está reconociendo, tal como se muestra en la Figura 3.11, además en la Figura 3.10 se visualiza la modificación que se realizó en el código para que el botón de información aparezca únicamente si la imagen es reconocida.



Figura 3.11 Escena EscenaRA con el marcador identificado



Figura 3.10 Escena EscenaRA sin detectar el marcador

La Tabla 3.4 corresponde a la evaluación del prototipo con dos personas del GAD quienes comentaron los cambios del prototipo anterior y se evaluaron el día 12 de febrero del 2020. Los datos obtenidos de la evaluación, es la retroalimentación por parte de una persona del GAD para mejorar el funcionamiento del prototipo cuya próxima evaluación se fijo en la fecha 28 de febrero de 2020.

Tabla 3.4 Evaluación del prototipo número tres

FORMULARIO DE EVALUACIÓN DEL PROTOTIPO					
Nombre del observador:		Yessica Cruz		Fecha:	12/02/2020
Nombre del proyecto:			Empresa o ubicación:		
Aplicación en Realidad Aumentada			Santuario del Señor de la Salud		
Nombre del programa:			Número de prototipo:		
SantuRA			3		
	Usuario1	Usuario 2	Usuario 3	Usuario 4	
Nombre del usuario	Señor Luis Cuji Macas	Ingeniero Freddy Malatay Caibe	Licenciado Euclides Ramírez	Padre Carlos Solarte	
Cargo	Presidente del GAD-PR de Santiago	Secretario del GAD-PR de Santiago	Vocal del GAD-PR de Santiago	Párroco del Santuario del Señor de la Salud	
Reacción del usuario	Muy Buena	Buena	Muy Buena	Muy Buena	
Sugerencia del usuario	Sin Comentarios	En la escena de ayuda incorporar un video para mejor comprensión.	Sin Comentarios	Sin Comentarios	
Firma					
Plan de revisión		28/02/2020			

3.1.4. PROTOTIPO NÚMERO CUATRO

En la evaluación del prototipo número tres, se obtuvo una corrección correspondiente a la escena Ayuda. El nuevo prototipo incluirá el nuevo cambio descrito en la Tabla 3.4 tal como se muestra en la Figura 3.13 y Figura 3.12.

En la escena Ayuda se incorporó un botón adicional que mostrará un video para mejorar la comprensión de cómo funciona la aplicación tal como se muestra en la Figura 3.12. En la Figura 3.13 se visualiza el video que es incorporado incluyendo un nuevo canvas dentro del cual se adjuntó un Raw Image que es en dónde se visualizará el video y un botón de regresar para volver a la escena Ayuda original.

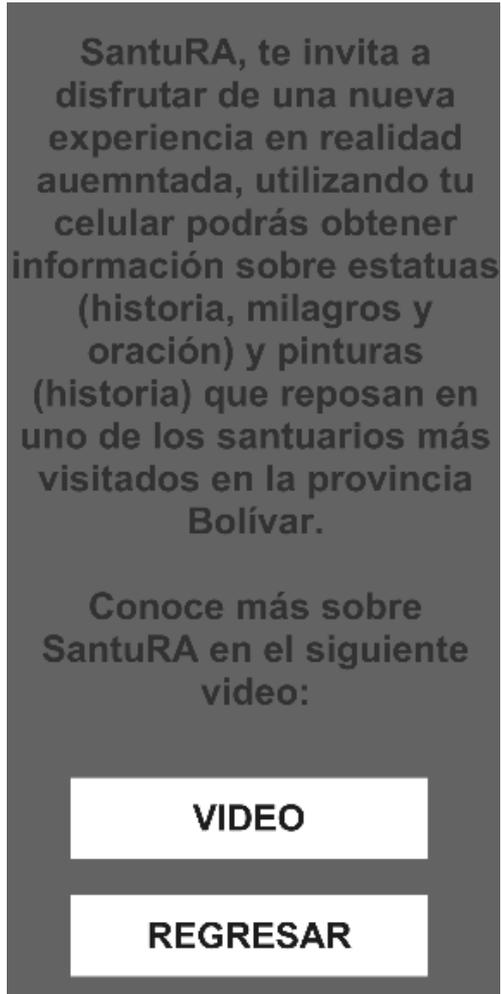


Figura 3.12 Escena Ayuda - Canvas Principal (Prototipo 4)



Figura 3.13 Escena Ayuda - Canvas Video (Prototipo 4)

La Tabla 3.5 corresponde a la evaluación del prototipo con la persona del GAD quien comentó el cambio del prototipo anterior y se evaluó el día 28 de febrero del 2020. Se obtuvo una reacción satisfactoria por parte del último comentario que realizó y no dió ninguna retroalimentación adicional. Por lo cual este es el prototipo final y el que se utilizó para el desarrollo del presente trabajo de titulación con los respectivos cambios que se realizaron a lo largo de la etapa de Taller de Diseño de la metodología RAD.

Tabla 3.5 Evaluación del prototipo número cuatro

FORMULARIO DE EVALUACIÓN DEL PROTOTIPO					
Nombre del observador:		Yessica Cruz		Fecha:	28/02/2020
Nombre del proyecto:			Empresa o ubicación:		
Aplicación en Realidad Aumentada			Santuario del Señor de la Salud		
Nombre del programa:			Número de prototipo:		
SantuRA			4		
	Usuario1	Usuario 2	Usuario 3	Usuario 4	
Nombre del usuario	Señor Luis Cuji Macas	Ingeniero Freddy Malatay Caibe	Licenciado Euclides Ramírez	Padre Carlos Solarte	
Cargo	Presidente del GAD-PR de Santiado	Secretario del GAD-PR de Santiago	Vocal del GAD-PR de Santiago	Párroco del Santuario del Señor de la Salud	
Reacción del usuario	Muy Buena	Muy Buena	Muy Buena	Muy Buena	
Sugerencia del usuario	Sin Comentarios	Sin Comentarios	Sin Comentarios	Sin Comentarios	
Firma					
Plan de revisión					

3.2. PRUEBAS DE LA APLICACIÓN

Las pruebas de la aplicación corresponden al prototipo final que fue aprobado por el GAD-PR de Santiago y el párroco del Santuario. Al prototipo número cuatro se realizó mejoras en la interfaz gráfica, se incluyeron los demás marcadores correspondientes a las imágenes del santuario, audios, entre otros.

Las pruebas de la aplicación SantuRA se las realizaron en dos celulares con diferentes resoluciones. El primer celular es de marca Samsung y el segundo Xiaomi cuyas características se encuentran en la Tabla 3.6 y la Tabla 3.7 respectivamente.

Las pruebas tienen como objetivo evaluar el funcionamiento de cada una de las escenas que conforman la aplicación, las cuales se van a dividir en dos partes con dos formatos diferentes (Funcionamiento, Realidad Aumentada).

El formato de la Tabla 3.8 corresponde a las Interacciones con el Usuario, el funcionamiento de los botones, despliegue de las escenas, cargar las imágenes de los botones y cambio de nombre de los botones. Esto se verificará para todas las escenas.

El formato de la Tabla 3.9 corresponde a la parte de realidad aumentada, es decir verificar que se detecte cada una de las imágenes que debe identificar la aplicación.

Cada una de estas tablas tendrá como resultado dos posibles resultados **correcto** e **incorrecto**.

Tabla 3.6 Características del celular Samsung J5 Prime

Tamaño de la pantalla	5 pulgadas
Resolución	720 x 1280 pixeles
Cámara	13 MP
Procesador	Quad-Core 1.4GHz
Versión de Android	8

Tabla 3.7 Características del celular Xiaomi Redmi Note 8

Tamaño de la pantalla	6.3 pulgadas
Resolución	1080 x 2340 pixeles
Cámara cuádruple	48MP + 8MP +2MP+2MP
Procesador	Snapdragon 665 2GHz
Versión de Android	9

Tabla 3.8 Formato para las pruebas de funcionamiento

Visualización de la escena		
Botones	Imagen	
	Texto	
	Funcionalidad	

Tabla 3.9 Formato para las pruebas de reconocimiento de las imagenes

Imagen		Reconocimiento
Estatuas		
Pinturas		

3.2.1. PRUEBAS DE LA ESCENA MENU

La Tabla 3.10 muestra los resultados de las pruebas realizadas a los cuatro botones que conforman la escena. En los botones se evalúan si se carga la imagen y el texto correcto correspondiente a ese botón y si abre la escena correcta, además de la visualización de la escena.

Tabla 3.10 Resultados de las pruebas de funcionamiento de la escena Menu

PARÁMETRO		VALOR
Visualización de la escena		Incorrecto
Botón Iniciar	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón Ayuda	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón Información Adicional	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón Salir	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto

3.2.2. PRUEBAS DE LA ESCENA AYUDA

La Tabla 3.11 muestra el resultado de las pruebas realizadas a los dos botones y de la reproducción del video.

Tabla 3.11 Resultados de las pruebas de funcionamiento de la escena Ayuda

PARÁMETRO		VALOR
Visualización de la escena		Incorrecto
Botón Video	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón Regresar	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Video	Reproducción	Correcto

3.2.3. PRUEBAS DE LA ESCENA DE CARGA

La Tabla 3.12 muestra el resultado de las pruebas realizadas al text que contiene el timer y a la propiedad Fill de la imagen.

Tabla 3.12 Resultados de las pruebas de funcionamiento de la escena de carga

PARÁMETRO		VALOR
Visualización de la escena		Incorrecto
Text	Visualización del timer	Correcto
	Velocidad	Incorrecto
Image.Fill	Avance del Fill	Incorrecto

3.2.4. PRUEBAS DE LA EscenaRA

La Tabla 3.13 muestra los resultados de las pruebas realizadas a los dos botones y al text de la escena, además en la Tabla 3.14 están las pruebas realizadas a las imágenes que debe reconocer la aplicación.

Tabla 3.13 Resultados de las pruebas de funcionamiento de la EscenaRA

PARÁMETRO		VALOR
Visualización de la escena		Incorrecto
Botón Información	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón Regresar	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Text	Visualización del nombre	Correcto

Tabla 3.14 Resultados de las pruebas de reconocimiento de las imágenes

Imagen		Reconocimiento
Estatuas	Señor de la Salud	Correcto
	San Pedro	Correcto
	San Francisco de Asís	Correcto
	Patrón Santiago	Correcto
	Señor de la Misericordia	Correcto
	San José	Correcto
	San Isidro Labrador	Correcto
	Arcángel San Miguel	Incorrecto
Pinturas	Jesús Divino Médico	Correcto
	Cristo Redentor	Correcto

3.2.5. PRUEBAS DE LA ESCENA MenuRA

La Tabla 3.15 muestra los resultados de las pruebas realizadas a los cuatro botones que conforman la escena, ya que tres de ellos tendrán diferentes nombres, imágenes o estarán ocultos, dependiendo si fue llamado desde el botón Información Adicional de la escena Menu o el botón información de la escena EscenaRA.

Tabla 3.15 Resultados de las pruebas de funcionamiento de la escena MenuRA

PARÁMETRO		VALOR
Visualización de la escena		Incorrecto
Botón 1	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón 2	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón 3	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón Regresar	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Text	Visualización del nombre	Correcto

3.2.6. PRUEBAS DE LA ESCENA INFORMACIÓN

La Tabla 3.16 muestra los resultados de las pruebas realizadas, debido a que en esta escena se visualizará la información que haya seleccionado en la escena MenuRA, y se reproducirá el audio correspondiente.

Tabla 3.16 Resultados de las pruebas de funcionamiento de la escena Información

PARÁMETRO		VALOR
Visualización de la escena		Correcto
Botón Play	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón Pause	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón Stop	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Botón Regresar	Imagen	Correcto
	Texto	Correcto
	Funcionalidad	Correcto
Text	Visualización de la información	Correcto
Slider	Adelantar o retroceder el audio	Correcto

La escena Información presenta un inconveniente ya que se demora alrededor de cinco segundos, esto afecta la experiencia del usuario.

3.3. PRUEBAS DEL MÓDULO DE DESCARGA

El módulo de descarga está compuesto por el servidor y el AP. La primera prueba que se realizó es que aparezca la red Santuario que fue configurada en el Access Point, esto se puede visualizar en la Figura 3.15. Después, de ingresar la clave se pone a prueba el servicio de DHCP para comprobar si asigna o no una dirección IP al dispositivo móvil, tal como se muestra en la Figura 3.14 y se logra conectar al AP. Una vez que se haya conectado a la red, en la Figura 3.16 se observa que el portal cautivo se activó, mostrando la página que se configuró en el AP, se da click en aceptar y se redireccionará a la página www.santura.com en dónde se podrá descargar el APK de la aplicación lo que indica que está funcionando correctamente el servicio WEB (Ver Figura 3.17).



Figura 3.14 IP que fue asignado al dispositivo móvil



Figura 3.15 Visualización de la red Santuario en el dispositivo móvil

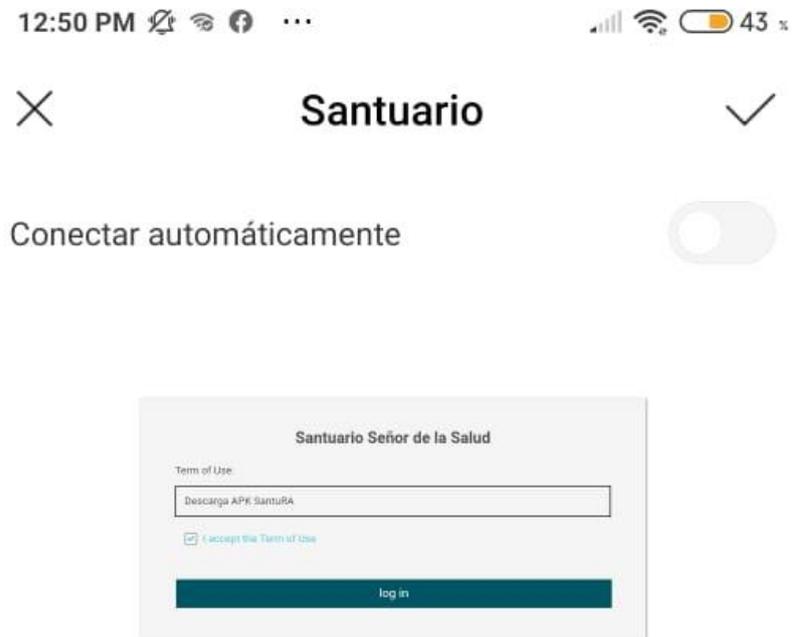


Figura 3.16 Página del portal cautivo



Figura 3.17 Página en dónde se podrá descargar el APK de la aplicación

3.4. CORRECCIÓN DE ERRORES

Las pruebas realizadas en la sección 3.2, arrojaron errores en la parte visual y una en el código de la escena de carga. Estos errores serán corregidos explicando el problema, la causa y la solución de cada uno de ellos.

El primer problema surgió al girar el celular lo que provocó que la aplicación cambie a landscape a pesar de que en el celular esté bloqueada la opción de rotación automática,

además la aplicación fue diseñada para orientación en portrait. La causa de ello fue que en los ajustes para Android (Player Settings), está la sección de Resolución y presentación, ahí se encuentra la opción **Default Orientation** la cual permite definir la orientación de la pantalla para la aplicación. Esta opción se encontraba configurada en *Auto Rotation*. La solución a este problema fue Configurar *Default Orientation* con Portrait que permite que la aplicación se ejecute de forma vertical con el botón de inicio en la parte inferior. Esta configuración se muestra en la Figura 3.18.

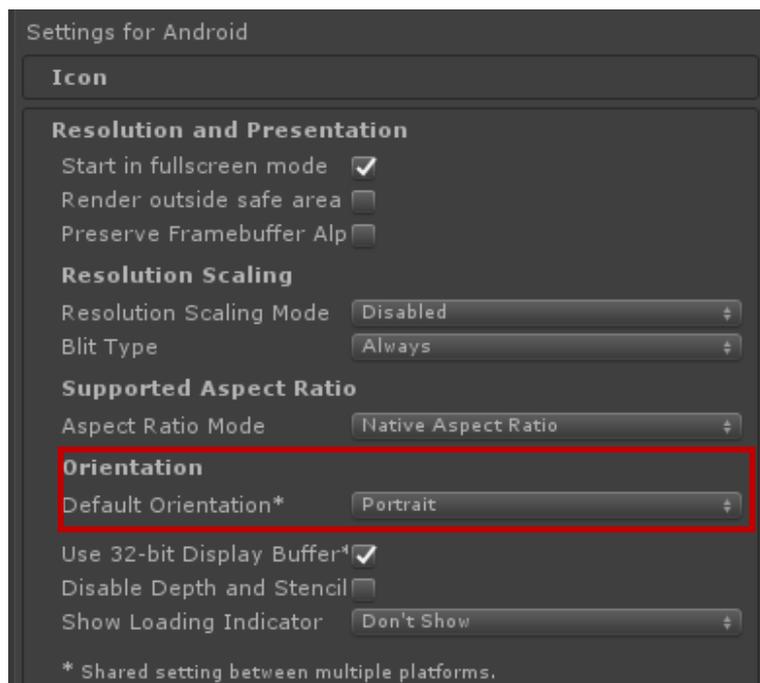


Figura 3.18 Cambio del valor de Orientation en Player Settings

El segundo problema surgió en el celular Samsung cuya resolución es menor en comparación al Xiaomi, esto provoca que los elementos de la escena se muevan con respecto a su posición original o no se muestren algunos de ellos. La causa de este problema es que la propiedad *UI Scale Mode* del Canvas (área donde se encuentran todos los elementos de interfaz de usuario) se encuentra configurado con el valor *Constant Physical Size* (Los elementos de interfaz de usuario mantienen el mismo tamaño sin importar el tamaño de la pantalla). Otro factor que influyó fue los anclajes que presentan cada uno de los elementos de interfaz de usuario, ya que todos ellos se encontraban en el centro del Canvas. Para solventar este problema se debe configurar el *UI Scale Mode* con el valor de *Scale With Screen Size*, que permite que los elementos de interfaz de usuario cambien de tamaño de acuerdo al tamaño de la pantalla y los anclajes se ubicaron de acuerdo a la posición real que debe tener dentro del Canvas. Estas configuraciones se pueden visualizar en la Figura 3.20 y Figura 3.19 respectivamente

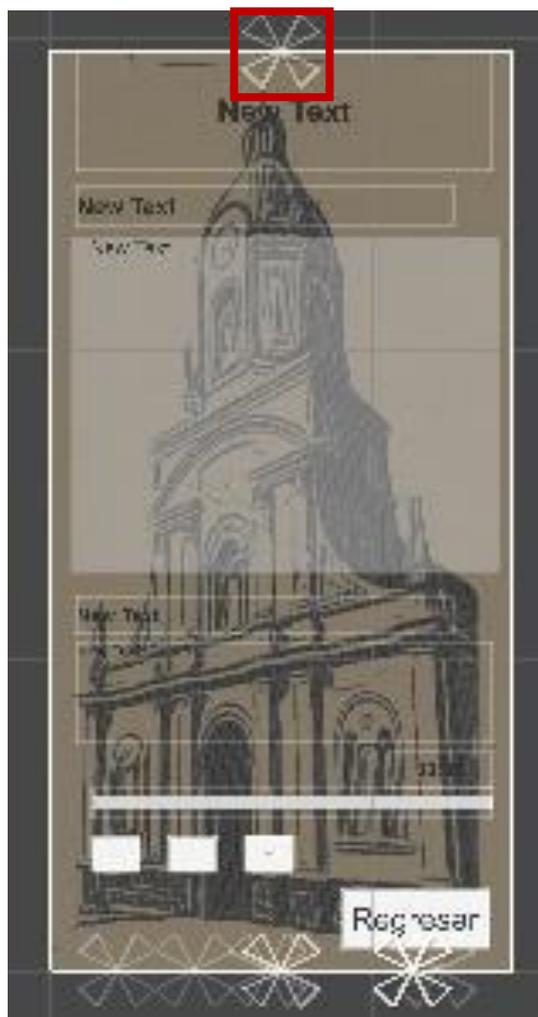


Figura 3.19 Posición de los anclajes de los elementos del canvas

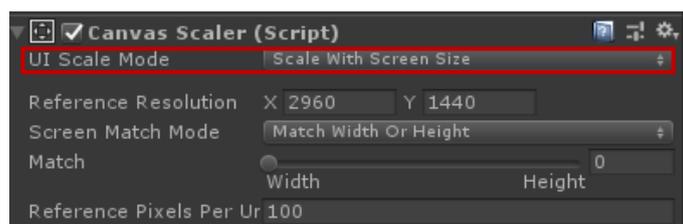


Figura 3.20 Configuración del Canvas

El cuarto problema fue el retardo de cinco segundos al abrir la escena de Información, la causa de ello fue la propiedad **Load Type**. Load Type es el método que utiliza Unity para cargar el audio en tiempo de ejecución, la cual estaba configurada como Descompress On Load. Descompress On Load descomprime el archivo de audio tan pronto es cargado, se utiliza para audios comprimidos pequeños para evitar una sobrecarga de rendimiento de descompresión al instante. En la Figura 3.21 se muestra la solución a este problema, la

cual es configurar el Load Type en Compressed in Memory, el cual mantiene el sonido comprimido en memoria y lo descomprime mientras se está reproduciendo.

SOLUCIÓN

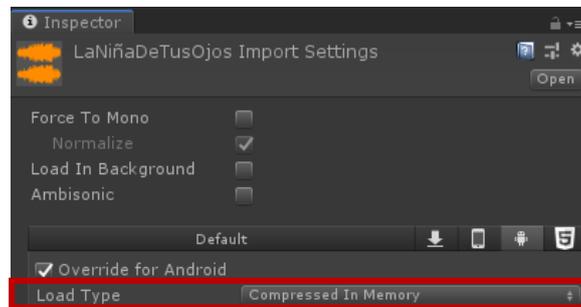


Figura 3.21 Ventana Inspector del audio

Por último, el timer que se utilizó en la escena EscenaDeCarga es muy lento y hace que permanezca alrededor de cuatro segundos antes de que se cargue la siguiente escena. La solución a este problema fue multiplicar por un factor de cincuenta al deltaTime tal como se muestra en el Código 3.1.

```
void Update()
{
    if(vall<100)
    {
        vall += 50 * Time.deltaTime;
        percentText.text = ((int)vall).ToString() + "%";
    }
    else
    {
        percentText.text = "SantuRA";
        SceneManager.LoadScene(2);
    }
    imagen.fillAmount = vall / 100;
}
```

Código 3.1 Modificación del código del Script LoadScene

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- La metodología RAD en la etapa de Taller de Diseño permitió que se apruebe el diseño y funcionamiento de la aplicación en base al análisis de los prototipos, esto permitió que el usuario esté consciente del producto que va a recibir al finalizar la fase de implementación.
- La ventana Game de Unity permite visualizar como se vería la aplicación en el dispositivo móvil, gracias a esto se pudo realizar pruebas para verificar que la aplicación se adapte a diferentes resoluciones modificando el valor de la opción Aspect.
- El desarrollo de la aplicación se la realizó mediante el seguimiento basado en marcadores, esto permitió que la aplicación sea más amigable con el usuario ya que enfoca la imagen que desea conocer y obtiene la información correspondiente a esa imagen.
- Las imágenes que serán utilizadas como marcadores deben ser ricas en detalles, tener buen contraste, no tener patrones repetitivos, formato JPG o PNG y su tamaño debe ser menor a 2 MB. En los formatos JPG las imágenes pueden ser RGB o a escala de grises, pero no CMYK. Esto permite obtener una mayor precisión al momento del reconocimiento, caso contrario resulta más difícil rastrearlos.
- La versión mínima de Android para el uso de la aplicación SantuRA es la 5.0 debido a que el uso de Vuforia se permite desde la versión 4.1.x+, y el componente Video Player 5.0+ por lo tanto, la versión mínima es la 5.0.
- Los nuevos equipos de conectividad permiten tener funciones adicionales, como es el caso del Access Point TP-Link EAP-115, el cual proporciona un portal cautivo que permite la autenticación por medio de redes sociales, mensajes de texto, entre otros.
- El servicio DHCP es importante ya que, al funcionar correctamente al conectarse al AP automáticamente le asignará una dirección IP y de esta manera acceder al servidor en donde se encuentra el APK de la aplicación.

4.2. RECOMENDACIONES

- Seleccionar la plataforma Android en la ventana de Build Settings, antes de desarrollar el proyecto ya que, si se realiza al finalizarlo puede tomar más tiempo de lo normal.
- En la ventana Build Settings en la sección de Scenes in Build se deben incluir todas las escenas que se han creado, caso contrario al momento de ejecutar la aplicación no se visualizará todas aquellas escenas que no se encuentren en esta ventana.
- Las imágenes que se incluyen en Unity al ser un proyecto en 3D la imagen se configura como Texture, por lo que se debe cambiar el Texture Type a Sprite (2D and UI) en la ventana Inspector de la imagen, caso contrario Unity no reconoce la imagen y no puede ser utilizada en el desarrollo del proyecto.
- RAD es una metodología orientada para pequeños proyectos ya que en la fase de Taller de Diseño se crean prototipos de la aplicación, lo que aumenta de forma considerable el tiempo de entrega del proyecto si éste es muy extenso.
- Utilizar las pinturas como marcadores no es una solución recomendable ya que al momento de crear el marcador en Vuforia el rating en estrellas es muy bajo (1 a 2 estrellas) lo que afecta al reconocimiento rápido de los marcadores.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] «Turismo Religioso, una oportunidad para promover el turismo en Ecuador – Ministerio de Turismo». <https://www.turismo.gob.ec/turismo-religioso-una-oportunidad-para-promover-el-turismo-en-ecuador/> (accedido ene. 27, 2020).
- [2] «Festividades del Señor de la Salud en Santiago de Bolívar | El Comercio». <https://www.elcomercio.com/cartas/festividades-senor-salud-santiago-bolivar.html> (accedido ene. 28, 2020).
- [3] H. L. Pombo y A. N. Martín, «Análisis y Desarrollo de Sistemas de Realidad Aumentada», p. 124.
- [4] «QUÉ ES RAD, FRAMEWORK, IDE – CONCEPTOS Y APLICABILIDAD | Scriptcase Blog - Development, Web Design, Sales and Digital Marketing». <https://www.scriptcaseblog.net/es/development-es/que-es-rad-framework-ide-conceptos-y-aplicabilidad/> (accedido ene. 28, 2020).
- [5] W. Hentzen, *The Software Developer's Guide*. Hentzenwerke, 2002.
- [6] M. L. Sanz, *Programación Web en el Entorno Servidor. (MF0492_3)*. Grupo Editorial RA-MA.
- [7] K. E. Kendall, J. E. Kendall, y A. Vidal Romero Elizondo, *Análisis y diseño de sistemas*. México: Prentice Hall, 2011.
- [8] «Creación de Videojuegos en Español: Desarrollo de Componentes - Francisco Jurado, Javier Asbusac, EspaCursos - Google Libros». <https://books.google.com.ec/books?id=u2DABAAAQBAJ&pg=PA227&dq=tom+caudell+realidad+aumentada&hl=es-419&sa=X&ved=0ahUKEwiB86v5z7joAhUQRK0KHbWyCPUQ6AEILzAB#v=onepage&q=tom%20caudell%20realidad%20aumentada&f=false> (accedido mar. 26, 2020).
- [9] F. Telefónica, *Realidad Aumentada: una nueva lente para ver el mundo*. Fundación Telefónica, 2011.
- [10] J. L. L. Olivencia, A. G. Plaza, y C. R. Jiménez, «SISTEMAS DE RECOMENDACIÓN PARA REALIDAD AUMENTADA EN UN SISTEMA INTEGRAL DE GESTIÓN DE DESTINOS», *Rev. ANÁLISIS Tur.*, vol. 0, n.º 14, pp. 69-81, 2012, doi: 10.1234/RAT2011n11.
- [11] «Vuforia Developer Portal |». <https://developer.vuforia.com/> (accedido ene. 28, 2020).
- [12] «Getting Started with Vuforia Engine in Unity». <https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html> (accedido feb. 06, 2020).
- [13] I. L. V. Loor, «JIPIJAPA - MANABÍ - ECUADOR», p. 141.
- [14] «How To Use the Trackable Base Class». <https://library.vuforia.com/content/vuforia-library/en/articles/Solution/How-To-Use-the-Trackable-Base-Class.html> (accedido feb. 28, 2020).
- [15] «Interpreting Tracking State API Results». <https://library.vuforia.com/content/vuforia-library/en/articles/Solution/tracking-state.html> (accedido feb. 28, 2020).
- [16] «Targets». <https://library.vuforia.com/content/vuforia-library/en/articles/Solution/Targets.html> (accedido feb. 28, 2020).
- [17] «Overview». <https://library.vuforia.com/features/overview.html> (accedido feb. 06, 2020).
- [18] «Documento_completo__.compressed.pdf-PDFA.pdf». Accedido: feb. 28, 2020. [En línea]. Disponible en: http://sedici.unlp.edu.ar/bitstream/handle/10915/53250/Documento_completo__.compressed.pdf-PDFA.pdf?sequence=1#page=167.
- [19] «Unity Real-Time Development Platform | 3D, 2D VR & AR Visualizations». <https://unity.com/> (accedido mar. 27, 2020).
- [20] «Video editing | VEGAS». <https://www.vegascreativesoftware.com/us/> (accedido mar. 28, 2020).

- [21] J. J. T. Bacusoy, E. A. M. Lino, M. M. O. Hernández, y Y. B. M. Lino, *SISTEMA DE COMUNICACIÓN INALÁMBRICO CON TECNOLOGÍA MIKROTIK Y SU CONTRIBUCIÓN EN LA TRANSMISIÓN DE VOZ Y DATOS EN EL TERMINAL TERRESTRE DEL CANTÓN JIPIJAPA*. 3Ciencias, 2018.
- [22] D. R. C. Herrera, «DISEÑO E IMPLEMENTACIÓN DE UN PORTAL CAUTIVO QUE PERMITA LA VENTA DE TICKETS DE INTERNET PARA UN HOTSPOT, EMPLEANDO HERRAMIENTAS DE SOFTWARE LIBRE», p. 164.
- [23] «85a0d39adcd0a172a1dadb8205d5ed473875.pdf». Accedido: mar. 28, 2020. [En línea]. Disponible en: <https://pdfs.semanticscholar.org/4ce5/85a0d39adcd0a172a1dadb8205d5ed473875.pdf>.
- [24] «MoralGomezAlbertoR-ETISa2009-10.pdf». Accedido: feb. 07, 2020. [En línea]. Disponible en: https://ddd.uab.cat/pub/trerecpro/2013/hdl_2072_206750/MoralGomezAlbertoR-ETISa2009-10.pdf.
- [25] «Es/2.2/Portal Cautivo - Zentyal Linux Small Business Server». https://wiki.zentyal.org/wiki/Es/2.2/Portal_Cautivo (accedido feb. 07, 2020).
- [26] «RusRoute - Última versión. Descargar gratis», *Downloadastro*. <https://rusroute.es.downloadastro.com> (accedido feb. 07, 2020).
- [27] Antamedia, «Software de HotSpot WiFi», *Antamedia*. <https://www.antamedia.com/es/el-hotspot/> (accedido feb. 07, 2020).
- [28] «EAP Controller | Software Controlador Centralizado | TP-Link Iberia». <https://www.tp-link.com/es/business-networking/management-platform/eap-controller/> (accedido feb. 07, 2020).
- [29] Argüello, Marco, *Monografía de Santiago de Bolívar*, Primera. Editora ABC, 2014.
- [30] «Desarrollo de Sistemas de Información una Metodología Basada en el Modelado - Vicenç Fernández Alarcón - Google Libros». https://books.google.com.ec/books?id=Sqm7jNZS_L0C&pg=PA84&dq=requiremientos+funcionales&hl=es-419&sa=X&ved=0ahUKEwj1bbw-9jnAhW-IXIEHRmVAIlgQ6AEIPTAD#v=onepage&q=requiremientos%20funcionales&f=false (accedido feb. 17, 2020).
- [31] «Desarrollo de Software: Requisitos, Estimaciones y Análisis. 2 Edición - Daniel Ramos Cardozzo - Google Libros». https://books.google.com.ec/books?id=tBaYCwAAQBAJ&printsec=frontcover&dq=que+son+historias+de+usuario+dise%C3%B1o+de+software&hl=es-419&sa=X&ved=0ahUKEwjT2OC8_tjnAhVJl3IEHUIID5EQ6AEIOTAC#v=onepage&q=que%20son%20historias%20de%20usuario%20dise%C3%B1o%20de%20software&f=false (accedido feb. 17, 2020).
- [32] «CasosUsoUML.pdf». Accedido: mar. 12, 2020. [En línea]. Disponible en: http://www.lsi.us.es/~javierj/cursos_ficheros/metricaUML/CasosUsoUML.pdf.
- [33] «diagramassecuencia.pdf». Accedido: mar. 10, 2020. [En línea]. Disponible en: <http://www.javier8a.com/itc/bd1/diagramassecuencia.pdf>.
- [34] «Download for EAP115 | TP-Link Iberia», *TP-Link*. <https://www.tp-link.com/es/support/download/eap115/> (accedido may 07, 2020).
- [35] «63-manual-sony-vegas.pdf». Accedido: sep. 15, 2020. [En línea]. Disponible en: <http://tic.iespm.es/index.php/descargas/send/13-manuales/63-manual-sony-vegas#page=123&zoom=100,0,581>.

ORDEN DE EMPASTADO