



Acta de Correcciones al Proyecto de Grado Ingeniería de Sistemas y Computación

Fecha: Agosto 30 de 2021

Autores: María Camila López Leal, David Hernández Cárdenas

Nombre del Proyecto de Grado: Asistente Informativo Virtual

Director:

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.



Firma de Director(a) del Proyecto de Grado

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero de Sistemas y computación.

Camilo Rocha

Dr. HERNÁN CAMILO ROCHA
Decano de la Facultad de Ingeniería

Gerardo Mauricio Sarría

Dr. GERARDO MAURICIO SARRIA
Director Carrera Ingeniería de Sistemas y Computación

Andrés A Navarro N.

Dr. ANDRÉS NAVARRO NEWBALL
Director(a) Trabajo

Gloria Inés Álvarez

Dra. GLORIA INÉS ÁLVAREZ
Jurado 1

JPG

Mag. JUAN PABLO GARCIA
Jurado 2

Santiago de Cali, Julio 2021.

Señores

Pontificia Universidad Javeriana Cali.

Dr. Gerardo Mauricio Sarria


Director Carrera de Ingeniería de Sistemas y Computación.

Cali.

Cordial Saludo.

Por medio de la presente me permito informarle que los estudiantes de Ingeniería de Sistemas y Computación, Maria Camila López Leal (cod: 8932747) y David Hernández Cárdenas (cod: 8932180) trabajaron bajo mi dirección en el proyecto de grado titulado “Asistente Informativo Virtual” el cual se encuentra finalizado y listo para sustentación.

Atentamente,

A handwritten signature in black ink that reads "Andrés A. Navarro Newball". The signature is written in a cursive style with a horizontal line underneath the name.

Dr. Andrés Adolfo Navarro Newball

Santiago de Cali, Julio 2021.

Señores

Pontificia Universidad Javeriana Cali.

Dr. Gerardo Mauricio Sarria

Director Carrera de Ingeniería de Sistemas y Computación.

Cali.

Cordial Saludo.

Nos permitimos presentar a su consideración el trabajo de grado titulado “Asistente Informativo Virtual” con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el proyecto de grado y posteriormente optar al título de Ingeniero de Sistemas y Computación.

Al firmar aquí, damos fe que entendemos y conocemos las directrices para la presentación de trabajos de grado de la Facultad de Ingeniería aprobadas el 26 de Noviembre de 2009, donde se establecen los plazos y normas para el desarrollo del anteproyecto y del trabajo de grado.

Atentamente,

María Camila López

Maria Camila López Leal
Código: 8932747



David Hernández Cárdenas
Código: 8932180

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería.
Ingeniería de Sistemas y Computación.
Trabajo de Grado.

Asistente Informativo Virtual

Maria Camila López Leal
David Hernández Cárdenas

Director: Dr. Andrés Adolfo Navarro Newball

Julio 2021



Resumen

La humanidad se ve en la necesidad de reconocer la existencia de diferentes sociedades, comprendiendo y respetando su cultura, su doctrina y su manera de vivir. Esto lo lleva a interactuar eminentemente con los demás, permitiéndole crear lazos de afecto, amistad o negocios. Sin duda, el turismo es el principal factor que permite esta interacción. Es así, como se permite abordar esta situación desde el continuo tecnológico, denominándose turismo inteligente y computación cultural. De este modo, en esta investigación se implementó una aplicación para dispositivos móviles que enriquece la visita de locales y extranjeros al barrio San Antonio de la ciudad de Santiago de Cali, Colombia usando conceptos como la realidad aumentada, el procesamiento del lenguaje natural y técnicas como speech-to-text, text-to-speech, skeletal animation y blend shapes. El objetivo principal es guiar e informar acerca del barrio con el fin de reconocerlo como un lugar patrimonial.

Palabras Clave: Realidad Aumentada, Procesamiento del lenguaje natural, Speech-To-Text, Text-To-Speech, Skeletal Animation, Blend Shapes, Turismo Inteligente, Computación cultural.

Abstract

Humans need to recognize the existence of different societies, understanding and respecting their culture, doctrine and way of life. This leads them to interact eminently with others, creating bonds of affection, friendship or business. Undoubtedly, tourism is the main factor that allows this interaction. That being said, this situation can be approached from the technological continuum, calling it intelligent tourism and cultural computing. In order to reach that, in this research were implemented an application for mobile devices to enrich the visit of locals and foreigners to the San Antonio neighborhood in the city of Santiago de Cali, Colombia using concepts such as augmented reality, natural language processing and techniques such as speech-to-text, text-to-speech, skeletal animation and blend shapes. The main objective is to guide and inform about the neighborhood in order to recognize it as a heritage site.

Palabras Clave: Augmented Reality, Natural Language Processing, Speech-To-Text, Text-To-Speech, Skeletal Animation, Blend Shapes, Intelligent tourism, Cultural computing.

Índice general

1. Descripción del Problema	3
1.1. Planteamiento del Problema	3
1.1.1. Formulación	4
1.1.2. Sistematización	4
1.2. Objetivos	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Específicos	4
1.3. Justificación	4
1.4. Alcances y Delimitaciones	5
1.4.1. Alcances	5
1.4.2. Delimitaciones	5
1.4.3. Entregables	5
2. Marco Teórico y Trabajos relacionados	7
2.1. Marco de Referencia	7
2.1.1. Áreas Temáticas	7
2.1.2. Marco Teórico	7
2.1.3. Trabajos Relacionados	19
3. Metodología, Análisis y Diseño	23
3.1. Metodología	23
3.1.1. Tipo de Estudio	25
3.2. Análisis	25
3.2.1. Conocimiento de los sitios escogidos por los habitantes (Encuesta)	26
3.2.2. Metodología de levantamiento de requerimientos	27
3.2.3. Arquetipo de Usuario	29
3.2.4. Levantamiento de requerimientos	30
3.2.5. Requerimientos funcionales	30
3.2.6. Requerimientos no funcionales	32
3.3. Diseño	33
3.3.1. Diagramas UML	33
3.3.2. Modelo de datos	41
3.3.3. Prototipado	43

4. Implementación y Despliegue	50
4.1. Dispositivos y herramientas de desarrollo y despliegue	51
4.2. Implementación	52
4.2.1. Motor de desarrollo	52
4.2.2. Vuforia Augmented Reality Software Development Kit (SDK)	52
4.2.3. Avatares del sistema	54
4.2.4. Reconocimiento del Ambiente	64
4.2.5. Locales del sistema	70
4.2.6. Interacción por voz	71
4.2.7. Procesamiento del lenguaje natural	76
4.2.8. Implementación base de datos no relacional Firebase	81
4.2.9. Máquina de estados finita controlada por eventos	81
4.2.10. Comunicación y servicios REST	82
4.2.11. Manejo de versiones	83
5. Aseguramiento y Control de Calidad	84
5.1. Verificación	85
5.2. Validación	86
5.2.1. Pruebas unitarias (UT):	87
5.2.2. Pruebas de producto (PQT):	89
5.2.3. Pruebas de aceptación del usuario (UAT):	91
5.2.4. Evaluación de interfaz de usuario y usabilidad	92
5.2.5. Evaluación de satisfacción	93
5.2.6. Resultados generales:	99
6. Conclusiones	100
7. Trabajo Futuro	101
Bibliografía	102
8. Anexos	106
8.1. Análisis de lugares escogidos	106
8.2. Encuesta acerca de los 3 lugares escogidos	107
8.3. Requerimientos funcionales	108
8.4. Código fuente del sistema	112
8.5. Requerimientos de prueba	114

Índice de figuras

2.1. Continuo Realidad-Virtualidad. [MK94].	8
2.2. Modelo 3D y Conjunto de mallas	11
2.3. Traslación en tres dimensiones[DPW07]	12
2.4. Rotación de un objeto en Ejes x,y,z [DPW07]	13
2.5. Esqueleto cuerpo humano con animación activa [SELdFG09].	15
2.6. Ejemplos BlendShape objetivo: Neutro, Sonrisa, Desacuerdo y Tristeza [Anj18].	16
2.7. Sensor LiDAR (izquierda) generando nube de puntos (derecha) del Templo Sewu, ubicado en Indonesia [Fir21].	17
3.1. La espiral del pensamiento creativo. [Res07]	23
3.2. Relación de ciclos con fases de la espiral creativa.	25
3.3. Tablero historias de usuario del sistema.	29
3.4. Ciclo 1 Parte 1: Historias de usuario definidas.	31
3.5. Ciclo 2 Parte 1: Historias de usuario definidas.	31
3.6. Ciclo 3 Parte 1: Historias de usuario definidas.	32
3.7. Ciclo 4: Historias de usuario definidas.	32
3.8. Tablero de requerimientos no funcionales.	33
3.9. Planteamiento arquitectural abreviado. Cliente se comunica con el servidor a partir de microservicios controlados por eventos.	34
3.10. Capa de Cliente.	35
3.11. Máquina de estados finita controlada por eventos para el acceso a los componentes de la capa del cliente y los servicios de la capa del servicio.	36
3.12. Eventos y comandos de interacción con el avatar.	37
3.13. Capa de Servicio.	38
3.14. Arquitectura de microservicios controlada por eventos. Se definen una Capa de Cliente y una Capa de Servicio. Además, estas capas se comunican por medio de un controlador de eventos que está basado en la ocurrencia de sucesos para la generación de transiciones y peticiones (Ver Figura 3.11).	39
3.15. Diagrama de flujo del Asistente Informativo Virtual.	40
3.16. Diagrama de caso de uso del Asistente Informativo Virtual.	41
3.17. Modelos de datos no relacionales de la sección de comentarios(a) y valoración(b)	42
3.18. Interfaces de las secciones definidas en el levantamiento de requerimientos (Ver Sección 3.2.4)	44
3.19. Capturas de Pantalla de algunos estados en INFORA	45
3.20. Elementos UI para AR Tomado de: [ARK](a) y [ARC](b).	46
3.21. Interfaz propia del Asistente Informativo virtual para la guía del menú de realidad aumentada.	46

3.22. Modificaciones a interfaz del Menú de lugares en el Barrio San Antonio.	47
3.23. Modificaciones finales a interfaces.	48
3.24. Captura de pantalla probando el sistema desde el editor de Unity, verificando el área segura para el desarrollo de interfaces en diferentes dispositivos tanto Android como iOS.	49
4.1. Modelo de comunicación por capas de Vuforia Engine.	53
4.2. Disposición jerárquica de los huesos de la mano derecha del avatar.	55
4.3. API de caracterización.	56
4.4. Christopher: Avatar Iglesia San Antonio.	57
4.5. Alonso: Avatar Imprenta La Linterna.	57
4.6. Aria: Avatar Tertuliadero La Colina.	57
4.7. Mapeo corporal avatar 3D.	58
4.8. Mascara tren inferior (izquierda) y mascara tren superior (derecha).	59
4.9. Máquina de estados de la capa del cuerpo completo.	60
4.10. Máquina de estados de la capa de la parte superior del cuerpo.	60
4.11. Componente Skinned Mesh Renderer - Deformaciones faciales y corporales totalmente parametrizables (Ver Sección 2.1.2.5).	61
4.12. Visema letra l.	62
4.13. Visema letra u.	62
4.14. Visema letra f.	62
4.15. Visema letras a-e.	62
4.16. Visema letras o.	62
4.17. Visema letra t.	62
4.18. Configuración parámetro de activación de cada uno de los visemas de las Figuras 4.12 a la 4.17	63
4.19. Planteamiento capa a capa de algoritmos posibles de Vuforia Fusion.	64
4.20. Mapeo sensor LiDAR con aplicación Vuforia Area Target Creator - Iglesia San Antonio.	65
4.21. Mapeo sensor LiDAR con aplicación Vuforia Area Target Creator - Domo San Antonio.	66
4.22. Modelo 3D Área Iglesia San Antonio.	67
4.23. Modelo 3D Área Domo San Antonio.	67
4.24. Imagen objetivo Tertuliadero La Colina.	68
4.25. Imagen objetivo Imprenta La Linterna.	68
4.26. Indicador de superficie diseñado y usado en el prototipo.	69
4.27. Tabla de textos configuraciones regionales: Español, Inglés y Francés.	70
4.28. Componente interfaz de usuario de elección de idioma.	71
4.29. Componente de interfaz de usuario del menú de realidad aumentada para el acceso al micrófono.	75
4.30. Ejemplo de llaves y valores para preguntas de tipo contextual (menciona a la Iglesia de San Antonio) y no contextual (se refiere a cualquier tema).	78

4.31. Ejemplo de entidades de aprendizaje automático (marcas azules) definidas por el equipo.	79
5.1. Modelo de verificación y validación adaptado.	84
5.2. Tablero de seguimiento en Wrike.	85
5.3. Distribución de tipos de pruebas en los ciclos de desarrollo del Asistente Informativo Virtual.	87
5.4. Prueba satisfactoria del despliegue de la cámara de realidad aumentada.	88
5.5. Toma de pantalla del Unity Profiler mostrando un promedio de cota inferior de 17.35 ms de tiempo de respuesta de CPU.	90
5.6. Toma de pantalla del Unity Profiler mostrando un promedio de cota superior de 25.23 ms de tiempo de respuesta de CPU.	90
5.7. Elementos en un test de usabilidad Tomada de: [Kat]	94
5.8. Imágenes recolectadas en la proceso de evaluación con usuarios finales, en el domo de la Iglesia(a) e Imprenta La Linterna(b).	98
8.1. Comentarios y calificación de visitantes del Tertuliadero la Colina.	106
8.2. Comentarios y calificación de visitantes de la Imprenta la Linterna.	107
8.3. Distancia encontrada entre los tres lugares escogidos.	107
8.4. Porcentaje de visitas Iglesia de San Antonio Cali.	108
8.5. Porcentaje de conocimiento acerca de la Imprenta La Linterna.	108
8.6. Porcentaje de visitas Tertuliadero La Colina.	108
8.7. Ciclo 1 Parte 1.	109
8.8. Ciclo 1 Parte 2.	109
8.9. Ciclo 2 Parte 1.	110
8.10. Ciclo 2 Parte 2.	110
8.11. Ciclo 3 Parte 1.	111
8.12. Ciclo 3 Parte 2.	111
8.13. Ciclo 4.	112

Introducción

La ciudad de Santiago de Cali, Colombia en pro de resaltar como distrito especial, está apostándole a incrementar deporte, cultura, turismo y mercado empresarial. Por ejemplo, hacia el oeste de la ciudad se encuentra uno de los barrios más típicos de la cultura caleña y que hace parte del plan de desarrollo, San Antonio. Su creación y consolidación está estrechamente ligada a la construcción de la capilla de San Antonio (templo representativo del barrio) realizada hace casi 300 años, más exactamente en el año 1746. Hostales, academias de baile, restaurantes, teatros y monumentos, son algunas de las tantas cosas que han surgido con el paso de los años y que hoy representan la idiosincrasia de los ciudadanos caleños.

Estos establecimientos públicos y privados a pesar del tiempo, conservan y denotan una tradicionalidad que se evidencia por su arquitectura representativa, su entorno urbano y la particular afabilidad de sus dueños. Entre sus calles se llevan a cabo manifestaciones claras de arte y cultura, una muestra de esto es la realización de *La Calle del Arte*, en donde se cierran varias cuadras del barrio, para adecuarlas como escenario para la presentación de obras de teatro, grupos musicales, exposiciones de arte, artesanías y mercado de pulgas cada septiembre. Sin lugar a dudas, el presentar a la comunidad toda la historia, las costumbres y las tradiciones del barrio es un factor clave para el reconocimiento del barrio como patrimonio no sólo local, sino también nacional e internacional representativo de la región del Valle del Cauca, Colombia.

Sin embargo actualmente, los métodos utilizados para brindar información respecto a los diferentes eventos y lugares en el barrio evidencian algunas limitaciones. Por ejemplo, parte de la información está en manos de particulares e instituciones privadas; los medios públicos de difusión como carteles, no son claros y son escasos; y, no existe información bilingüe que sea inclusiva con visitantes internacionales.

Por ello, se propone implementar una manera de compartir y dar a conocer el lugar por medio de vías diferentes e impactantes a través de la tecnología, en este caso, con el uso de la realidad aumentada y el procesamiento del lenguaje natural.

Descripción del Problema

1.1. Planteamiento del Problema

El patrimonio arquitectónico, antropológico, urbanístico y social del barrio San Antonio es uno de los pilares fundamentales para que locales y turistas se enamoren cada vez más de la ciudad de Santiago de Cali, Colombia. Sin duda alguna, toda la historia que caracteriza al barrio debería ser conocida por todos aquellos que se desenvuelven entre sus calles a diario, ya que es un icono representativo de la región. Reconocer algunas de las casas con mayor antigüedad, tener presentes eventos y ferias, diferenciar los monumentos y el distinguir la colina donde se encuentra ubicado, serían circunstancias para el aprovechamiento del sector.

A pesar de ello, y tras realizar una salida de campo, se evidenció que la manera en que se informa acerca del barrio alrededor de sus lugares y eventos, tiene limitaciones. Por ejemplo, no existe una estrategia para generar interés respecto a estos; la información puede llegar desactualizada; los carteles informativos no son claros o están incompletos. En contraste, Boboc et. al [BGDT17] proponen un sistema basado en realidad aumentada que muestra una antigua iglesia, hoy en día inexistente, en el lugar en que esta debió estar ubicada. Así, se presenta información a los visitantes de manera visual e interactiva, resaltando la importancia del lugar y su historia.

Lo anterior, sugiere que el continuo virtual y sus tecnologías, brindan un gran potencial informativo que podría complementar las visitas al barrio San Antonio. Por ello, Milgram y Kishino [MK94] definieron El Continuo de la Virtualidad como el uso de diferentes tipos de objetos (también llamados actores virtuales) con los que se interactúa en situaciones particulares en las que se incluya la realidad y la virtualidad. Esto permite la creación de ambientes virtuales en donde los comportamientos de los usuarios van de la mano con las vivencias reales del ambiente en el que se encuentren.

El desarrollo de tecnología apoyada por el continuo virtual que busque superar algunas de las limitaciones comunicativas encontradas durante una salida de campo al barrio San Antonio trae retos relacionados con la manera de identificar y estructurar la información relevante e implementar una solución eficiente y usable. Lo anterior, lleva a proponer la pregunta de investigación presentada en la formulación.

1.1.1. Formulación

¿Cómo desarrollar el prototipo de un asistente informativo virtual basado en realidad aumentada que brinde información a locales y visitantes del barrio San Antonio?

1.1.2. Sistematización

- ¿Cómo se estructurará la información a obtener acerca del barrio San Antonio?
- ¿Cómo se integrará una arquitectura con la información obtenida?
- ¿Cómo se implementará una aplicación soportada por la arquitectura propuesta?
- ¿Cómo se evaluará la experiencia de los usuarios con el sistema?

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar el prototipo de un asistente informativo virtual que haga uso de realidad aumentada para brindar información a locales y visitantes del barrio San Antonio.

1.2.2. Objetivos Específicos

- Identificar, analizar y estructurar la información del barrio San Antonio que será usada en el asistente informativo virtual.
- Desarrollar una arquitectura para el asistente informativo virtual que permita integrar la información obtenida.
- Implementar la aplicación del asistente informativo virtual basándose en la arquitectura propuesta.
- Evaluar el control de la calidad y la experiencia de usuario del sistema implementado.

1.3. Justificación

La creación del asistente informativo virtual lleva consigo el estudio, el análisis y el reconocimiento del entorno en el que será implementado, en este caso se llevara a cabo en el barrio San Antonio.

San Antonio es un barrio que resalta por su historia y arquitectura tradicional, un lugar significativo para la ciudad, donde sus habitantes y visitantes son quienes al compartir cultura, tradición y gastronomía recíprocamente preservan su identidad y reconocimiento. Es así como el barrio es parte del patrimonio cultural caleño, razón por la cual es importante preservar y potenciar su rol generando valor y mayor conocimiento, no solo a través del contenido de la información proporcionada,

sino también, a través de la forma de transmitirla para llegar a los sentidos. Estas formas pueden ser visuales y auditivas y apoyadas por tecnologías como la realidad aumentada, lo que al mismo conlleva a uno retos de desarrollo y usabilidad que contribuyen al área de investigación del grupo Destino.

Adicionalmente, es fundamental para un lugar patrimonial generar un valor diferenciador al resto [Val20], es así como desarrollar un asistente informativo virtual que haga uso de tecnologías como realidad aumentada, aporta significativamente a conseguirlo. Esto permite dar a conocer el barrio San Antonio desde una experiencia enriquecida por la interacción humano computador, produciendo identidad y propiedad al barrio desde un punto de vista tecnológico.

1.4. Alcances y Delimitaciones

1.4.1. Alcances

- Se diseñará un sistema informativo audiovisual a partir de realidad aumentada basado en 3 lugares dentro del barrio San Antonio.
- Se elegirán al menos 3 tipos de lugares entre:
 - Hostales.
 - Restaurantes.
 - Monumentos.
 - Teatros.

1.4.2. Delimitaciones

- El sistema se limitará al espacio público abarcado por los 3 lugares escogidos del barrio San Antonio.
- Para la información suministrada de los lugares escogidos, se llegará a acuerdos con las entidades públicas y privadas que podrían hacer uso del sistema.
- El sistema a desarrollar estará disponible solamente para el sistema operativo móvil Android.
- Se realizan recorridos y pruebas utilizando sistemas virtuales y visitas esporádicas.

1.4.3. Entregables

Al finalizar el proyecto se tendrán los siguientes entregables:

- Un documento de tesis.
- Un prototipo de aplicación móvil que hace uso de técnicas de realidad aumentada.
- Un manual de usuario.

- Documentación, registro y evaluación de la experiencia de usuario con los usuarios del sistema.

Por ende, dado que el documento es uno de los principales entregables del proceso de investigación, es importante tener en cuenta que en él se explicará cómo fue llevado a cabo el Asistente Informativo Virtual. Exponiendo como primer paso el estado del arte y los conceptos principales relacionados con el proyecto. Después, se continúa con un proceso de análisis para determinar qué requerimientos y características se deberían cumplir para satisfacer los objetivos planteados, además cuál fue la metodología para hacerlo.

Seguido a esto, se muestra la fase de diseño, donde se describen los elementos tales como prototipo y diagramas UML, que representan el planteamiento del análisis realizado. Posteriormente, se expone el capítulo de implementación del aplicativo donde se documentan cuales fueron las herramientas y estrategias utilizadas y su razón de elección para lograr integrar lo que se especificó en el análisis y diseño.

Por último, se detalla el capítulo de aseguramiento y control de calidad, donde se muestra la metodología seguida y los tipos de prueba utilizados, para evaluar el correcto funcionamiento del sistema verificando que cumple con todo lo especificado.

Marco Teórico y Trabajos relacionados

2.1. Marco de Referencia

De acuerdo al sistema de clasificación de computación de la ACM, las áreas temáticas que abarca el proyecto son:

2.1.1. Áreas Temáticas

- **Computing methodologies** - Computer graphics - Graphics systems and interfaces - Mixed / augmented reality
- **Human-centered computing** - Ubiquitous and mobile computing - Ubiquitous and mobile devices - Mobile devices
- **Human-centered computing** - Human computer interaction (HCI) - HCI design and evaluation methods
- **Social and professional topics** - User characteristics - Cultural characteristics

2.1.2. Marco Teórico

En la siguiente sección se especifican los conceptos teóricos y técnicos relacionados con la implementación y despliegue del proyecto (Ver **Sección 4**).

2.1.2.1. Realidad Aumentada

La realidad aumentada (RA) permite a los usuarios interactuar en el mundo real con objetos y agentes de origen virtual, creados a partir de técnicas de computación gráfica y desplegados por medio de dispositivos que los aumentan al mundo real.

Los observadores experimentan un tipo de inmersión parcial donde la extensión del mundo real permite nuevas interacciones con objetos generados por computador. Estas interacciones son efectuadas sin sacar completamente a los usuarios del espacio real en el que se encuentren.

En sus inicios, el termino fue adoptado desde diferentes definiciones. [MTUK94] la definió como “Aumentar las reacciones naturales que percibe un observador por medio de señales simuladas”, en

otras palabras la RA era considerada bajo un concepto de revelación de contenido que no era accesible o experimental en el contexto actual del observador. Sin embargo, el término fue incurriendo en cambios, incluso se consideró una variación de la realidad virtual (RV), definida como la interacción y exploración de ambientes o mundos virtuales a través de una experiencia de inmersión total gracias al uso de periféricos de inmersión como por ejemplo gafas y cascos. Fue así como se consideró a la RA como el uso de cascos con viseras transparentes que desplegaban objetos de origen virtual.

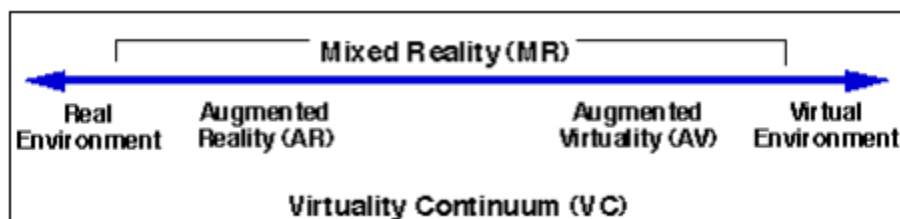


Figura 2.1: Continuo Realidad-Virtualidad. [MK94].

A partir de lo anterior, la definición de ambas tecnologías fue contemplada por la creación de una línea de realidades denominada “Continuo Realidad-Virtualidad” (Ver **Figura 2.1**). Este continuo separó el mundo real del virtual, considerando las tecnologías internas como realidad mixta (RM). En ellas, se encuentra la RA como una categoría de inmersión parcial reconocida por su gran flexibilidad y utilidad en sistemas tales como: médicos, ingenieriles, arquitectónicos y en la industria de los video juegos. Algunas de las aplicaciones existentes en algunas de las áreas de estudio son:

- **Educación:** Ej: “TeachAR: An Interactive Augmented Reality Tool for Teaching Basic English to Non-Native Children” [DPD⁺16]
- **Medicina:** Ej: “Augmented Reality System for Aiding Mild Alzheimer Patients and Caregivers” [KEA⁺18]

2.1.2.2. Computación Móvil

La tecnología de la computación móvil está basada en el nacimiento de las redes inalámbricas quienes posibilitaron la transmisión de datos, voz y vídeo por medio de ondas electromagnéticas. Este hecho permite dejar a un lado los medios físicos y/o la infraestructura cableada, creando dispositivos portátiles con acceso a la red.

La principal característica de la computación móvil se basa en la portabilidad de los dispositivos y el centrarse en tecnología de uso personal. De este modo, el usuario logra una interacción constante y de fácil acceso con sistemas informáticos de carácter ubicuo. Según Mark Weiser [Mar93] hacia el año 1993 las principales cualidades de estos medios ubicuos son:

- Constante interacción con los usuarios.

- Alta compatibilidad con el medio.
- Parecer invisibles para los usuarios.

A pesar de que estos dispositivos mantienen las características mencionadas, con el paso del tiempo se han especificado desde diferentes áreas de la investigación. Particularmente, la computación móvil es presentada a partir de sistemas de geolocalización, libros electrónicos, celulares inteligentes, computadores portátiles y consolas de juego.

2.1.2.3. Interacción Humano-Computador

La Interacción Humano-Computador es el estudio de cómo la tecnología permite la resolución de tareas y trabajos a través del uso y la comunicación humana con medios tecnológicos. Esta disciplina está dividida en dos grandes ámbitos. Inicialmente se conoce desde un enfoque académico donde se estudia la manera en que las personas se comunican con la tecnología con el fin de conseguir lo que quieren [III15].

Para ello, se tiene al diseño como segundo aspecto clave. Su principal razón de ser se centra en cómo diseñar sistemas que cumplan con las finalidades para los que son propuestos, envolviendo aspectos personales como la psicología y la economía y tecnológicos como los dispositivos y los lenguajes de programación que serán usados.

La definición de tres palabras han determinado el diseño de todo sistema basado en la interacción humano-computador [III15] :

- **Usabilidad:** Llevar a cabo los objetivos para los que son creados los sistemas por medio de la interactividad con estética, diversión y satisfacción.
- **Utilidad:** El software se desarrolla con un fin claro resolviendo un problema y brindando un factor de productividad a los usuarios.
- **Uso:** Toda creación está enfocada en el uso que se le dará, los sistemas deben crearse atractivos y aceptables para los fines que se especifican y para las organizaciones o usuarios a las que van dirigidos.

Con el fin de determinar el cumplimiento de estas cualidades, se han propuesto métodos de valoración y pruebas con los que se evalúa la interacción del usuario final con el producto de software.

Entre estos métodos se pueden encontrar algunas como [III15]:

- **Métodos de indagación**
 - Entrevistas personales.
 - Cuestionarios.

- Observaciones de campo.
- **Métodos de inspección**
 - Heurística por principios de usabilidad.
 - Recorridos de usabilidad (prototipado, reuniones).
- **Métodos de pruebas**
 - Métodos de descubrimiento.
 - Retroalimentación en voz alta.
 - Categorización con tarjetas.

2.1.2.4. Computación Cultural

El termino de computación cultural nace como un campo de la interacción humano-computador, donde se cubre el gran impacto e influencia que tiene el desarrollo tecnológico para el apoyo y la innovación cultural. La creación de software enfocado al ámbito cultural está estrechamente asociada a la necesidad de llevar a los propios y visitantes información e historia del entorno cultural en el que se encuentran, por medio de soluciones que impacten y que llamen la atención de los usuarios [tDJ18].

Es así, como el termino es usado por la variedad de tecnologías emergentes que traen consigo innovación y soluciones computacionales dinámicas que impactan y hacen más interactivos los sistemas, como por ejemplo la Realidad Aumentada [tDJ18].

2.1.2.5. 3D Computer Graphics

Diseño y modelado en tercera dimensión

Los procesos de diseño y modelado en tercera dimensión se refieren a un conjunto de acciones utilizadas con el fin de crear representaciones matemáticas de cualquier superficie y/o objeto en tres dimensiones, tales como personas, animales, flores, herramientas, ciudades enteras y prácticamente cualquier elemento que pueda ser imaginado o que simplemente ya exista, por lo que se encuentran diversas técnicas encargadas de especificar matemáticamente las propiedades de la forma y apariencia del modelo que se quiere construir [PS09], de tal manera, que este pueda ser plasmado en una conjunto o colección de polígonos llamados mallas y ser dispuestos en un espacio tridimensional para posteriormente ser almacenado y utilizado; los modelos una vez diseñados y creados pueden utilizarse como componentes que a su vez hagan parte de modelos más complejos [Dav18] (Ver **figura 2.2**).

Los modelos 3D son ampliamente utilizados en diversas áreas como lo son Medicina, Arquitectura, Diseño, Computación gráfica, entre otros... por lo que son cada vez más las herramientas especializadas en la construcción de estos modelos, en la actualidad algunos ejemplos de las más

utilizadas y conocidas son Blender, MakeHuman, FreeCAD y SketchUp, las cuales permiten crear modelos solos, simples o complejos, hasta crear conjuntos de modelos que simulen superficies mucho grandes.

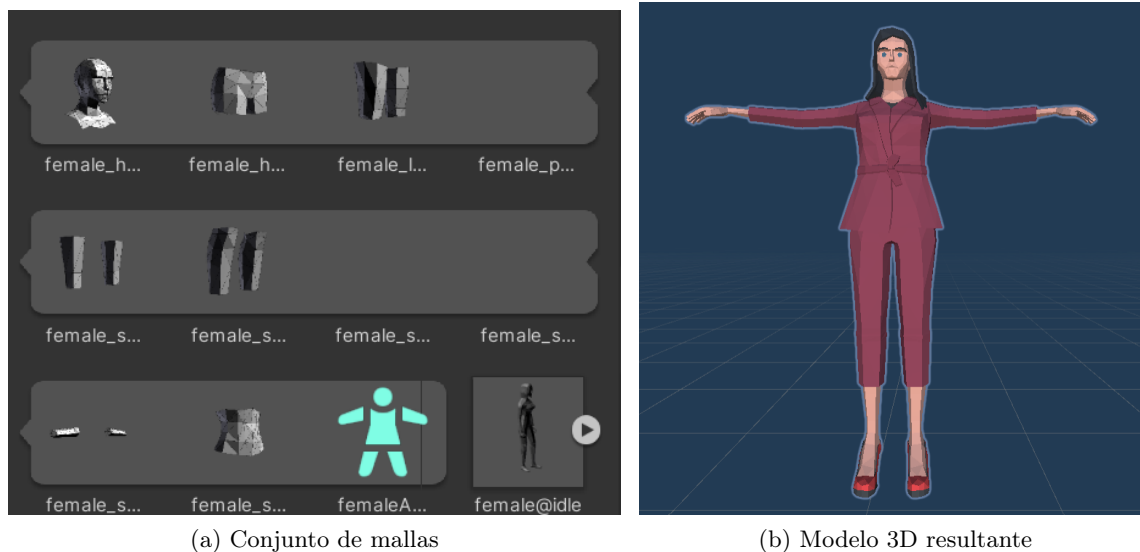


Figura 2.2: Modelo 3D y Conjunto de mallas

Transformaciones de modelos en tercera dimensión

Las transformaciones en modelos y objetos 3D permiten modificar características primitivas propias del elemento con el objetivo de obtener un aspecto o movimiento distinto a los valores que tenga este en dicho momento. [Dav18] Por lo que con estas transformaciones es posible cambiar el tamaño del modelo por medio de su anchura y altura, además de poder cambiar su escala en el plano y así poder verse de mayor o menor tamaño conservando proporcionalmente las medidas que tenga predeterminadas en el momento, como también se puede rotar el objeto entorno a un eje dentro del plano, inclinarse en su posición, distorsionar su figura y cambiar su perspectiva para alterar la impresión de profundidad.

Los métodos utilizados para realizar transformaciones geométricas en tres dimensiones, se derivan o amplían a partir de los métodos ya existentes en dos dimensiones, en donde se agregan consideraciones nuevas para el eje Z tomando como referencia un plano con coordenadas XYZ. En la mayoría de los casos, su aplicación es sencilla, sin embargo en algunos casos esta derivación de 2D a 3D no es tan obvia [DPW07]. Un ejemplo a resaltar es cuando se habla de rotaciones en dos dimensiones, se consideran solo sobre ejes perpendiculares en un plano XY, pero, en el espacio tridimensional, se puede seleccionar cualquier orientación espacial para el eje de rotación, este tipo de complejidades se tratan actualmente con paquetes gráficos dentro de las herramientas de diseño

y modelado en 3D mencionadas anteriormente, las cuales facilitan el uso de transformaciones para los usuarios y así un flujo más practico a la hora de crear contenido tridimensional.

Traslación de modelos tridimensionales

La traslación de modelos tridimensionales brinda la posibilidad de modificar la posición de los objetos para ser colocados punto por punto en otra ubicación dentro del plano, es decir, pasar de una posición $\mathbf{P} = (x, y, z)$ a una posición $\mathbf{P}' = (x', y', z')$ sumandole las distancias de traslación t_x, t_y, t_z a las coordenadas cartesianas de \mathbf{P} [DPW07]:

$$x' = x + t_x, \quad y' = y + t_y, \quad z' = z + t_z \quad (2.1)$$

de esta manera se logran trasladar los puntos del modelo en un espacio tridimensional (2.1). Para realizar estas traslaciones en conjuntos de puntos se suele hacer uso de operaciones de traslación representadas en forma de matriz, en el que \mathbf{P} y \mathbf{P}' se representan en coordenadas con matrices de columnas de 4 elementos y el operador de traslación \mathbf{T} (2.2).

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.2)$$

El objeto se traslada transformando cada uno de los puntos dentro de las coordenadas definidas en el modelo y posteriormente lo reconstruye en una nueva ubicación(Ver **figura 2.3**).

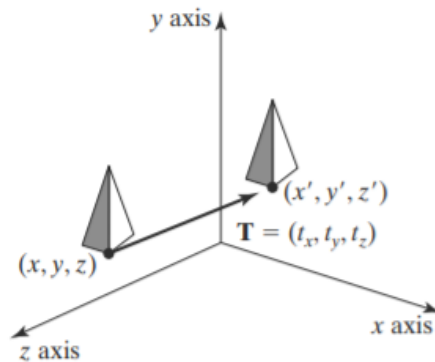


Figura 2.3: Traslación en tres dimensiones[DPW07]

Rotación de modelos tridimensionales

Cuando se necesita lograr que un objeto gire, se hace uso de transformaciones de rotación las cuales permiten establecer la orientación de un objeto, girando un ángulo sobre un eje específico y

en algunos casos, cuando se requieren rotaciones sobre ejes más complejos se hace en conjunto con transformaciones de traslación (2.1.2.5), en este sentido la rotación de un objeto permite girar el mismo sobre cualquier eje del espacio tridimensional en el que se encuentre [Dav18], siendo útil por ejemplo para modificar la dirección hacia donde mira un avatar humanoide.

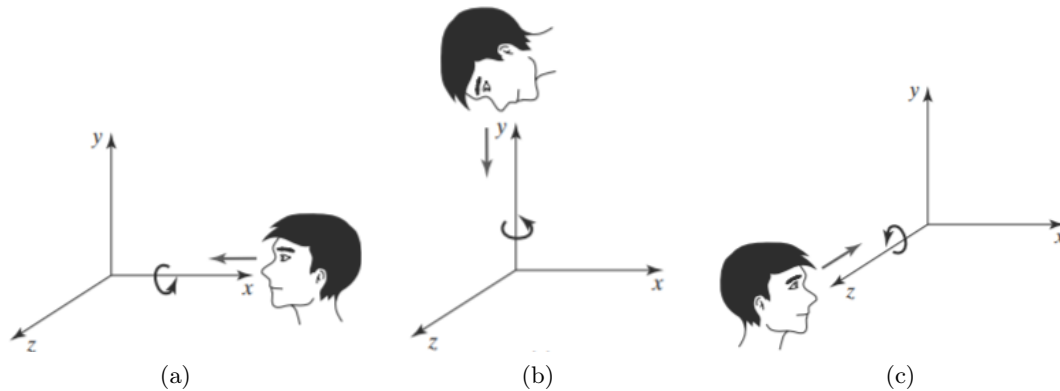


Figura 2.4: Rotación de un objeto en Ejes x,y,z [DPW07]

Dentro de la computación gráfica y en general dentro de la matemática necesaria para generar la traslación de todos los puntos que componen un modelo, existen distintas formas y métodos para realizarlo, es común que las herramientas existentes para manejo de gráficos 3D en un computador, hagan uso de representaciones de Cuaterniones como alternativa a la mayoría de métodos, ya que este ha demostrado ser más eficiente para generar una rotación sobre un eje seleccionado arbitrariamente.

Los Cuaterniones son una extensión de los números complejos en dos dimensiones, los cuales son útiles en una serie de procedimientos gráficos en los computadores, estos requieren menos espacio de almacenamiento que las matrices 4x4, y se encuentra más sencillo usarlos para escribir procedimientos en las secuencias de transformaciones cuando a rotaciones se refiere, lo cual es importante cuando se crean animaciones puesto que estas requieren de secuencias de movimientos más complejas en un objeto [DPW07].

Escala de modelos tridimensionales

Las transformaciones de escala en objetos tridimensionales y en general, son comúnmente utilizadas para cambiar el tamaño de los objetos y que estos sean más grandes o más pequeños. La expresión matricial para realizar este proceso es similar a la utilizada en dos dimensiones, básicamente es una escala de la misma incluyendo la coordenada Z en la matriz de transformación. La transformación de la escala tridimensional dados unos parámetros s_x , s_y y s_z para un punto puede

representarse como (2.3)

$$P' = S.P \quad (2.3)$$

De esta manera, las coordenadas del punto $P = (x, y, z)$ se multiplican con el valor de s_i asociado y toman el tamaño correspondiente como se observa en (2.4)

$$x' = x.s_x \quad y' = y.s_y \quad z' = z.z \quad (2.4)$$

Esta ecuación permitirá entonces la escalabilidad de los puntos pertenecientes al modelo, sin embargo, en algunos casos es necesario escalar un objeto uniformemente para que tenga un tamaño proporcional al inicial, para lograrlo es necesario que $s_x = s_y = s_z$ y así generar una escalabilidad uniforme [DPW07].

Animación esquelética de modelos tridimensionales

La animación corporal de modelos tridimensionales con capacidades motrices se puede llevar a cabo por medio de técnicas como animación esquelética o en inglés, Skeletal Animation - Rigging.

En este proceso, todo el modelo se estructura en un esqueleto, que es una combinación de huesos constituidos jerárquicamente. El esqueleto tiene un hueso para cada parte móvil del humanoide: un hueso para el hombro, uno para el brazo, uno para el antebrazo, uno para la palma de la mano, y algunos más para los dedos. Cada hueso necesita saber a qué hueso padre se conecta, para así rotarlo y a su vez, rotar todos sus hijos [SELdFG09].

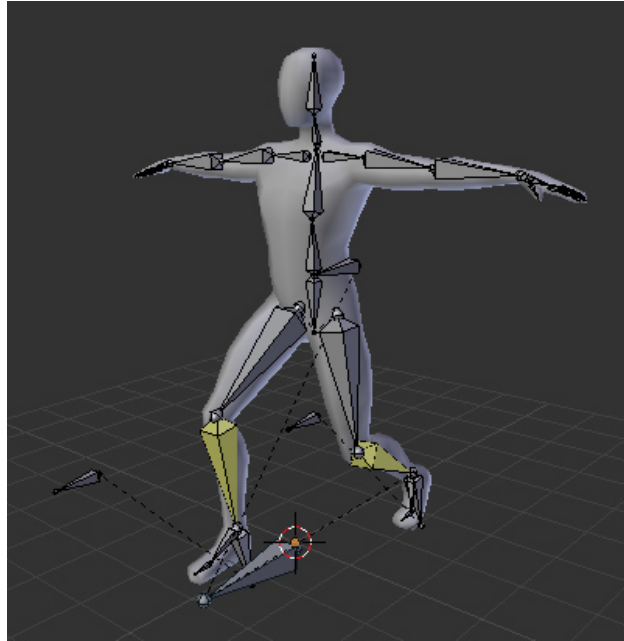


Figura 2.5: Esqueleto cuerpo humano con animación activa [SELdFG09].

Blend Shapes de modelos tridimensionales

Los Blend Shapes son una parametrización aproximada de cualquier tipo de expresión facial de modelos tridimensionales. Se representan como una suma lineal ponderada de la combinación de gestos faciales objetivo tal como se ven en la **Figura 2.6**. Estos ejemplifican expresiones faciales específicas o acciones musculares faciales. Esta suma está dada por la especificación de pesos para cada fotograma de la animación, lo que permite crear infinidad de animaciones con base en el cambio matricial de una animación inicial (gesto facial objetivo) [Anj18].

$$\begin{bmatrix} fx_0 \\ fy_0 \\ fz_0 \\ \vdots \\ \vdots \\ \vdots \\ fx_n \\ fy_n \\ fz_n \end{bmatrix} = \begin{bmatrix} x & x & \dots & x \\ y & y & \dots & y \\ z & z & \dots & z \\ x & x & \dots & x \\ y & y & \dots & y \\ | & | & \dots & | \\ \mathbf{b}_0 & \mathbf{b}_1 & \dots & \mathbf{b}_n \\ | & | & \dots & | \\ | & | & \dots & | \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} \quad (2.5)$$

De esta manera, matemáticamente la representación de una animación facial está denotada como un vector columna \mathbf{f} que contiene todas las coordenadas de los vértices del modelo en un orden arbitrario (como xxx yyy zzz, o alternativamente xyz xyz xyz tal como se evidencia en la expresión 2.5), pero consistente en todas las posibles combinaciones.

$$\mathbf{f} = \sum_{k=0}^n w_k \mathbf{b}_k \quad (2.6)$$

Así mismo, por medio de la ecuación 2.6 se reconoce el hecho de que el resultado facial \mathbf{f} es una sumatoria directamente proporcional al peso w_k que se le asigne a cada gesto facial objetivo \mathbf{b}_k . Con esto, se dice que \mathbf{b}_0 es el BlendShape objetivo que representa un rostro neutral. Por ello, la técnica es considerada como la añadidura de vectores que modifican un resultado matricial inicial que se evidencia en el cambio facial del rostro animado [Anj18].



Figura 2.6: Ejemplos BlendShape objetivo: Neutro, Sonrisa, Desacuerdo y Tristeza [Anj18].

2.1.2.6. Sensores de reconocimiento del ambiente

En la actualidad, tecnologías como la realidad aumentada están apoyadas directamente en el uso de hardware, que permite por ejemplo, el preprocesamiento del ambiente. Para esto, existen dispositivos como sensores de reconocimiento del ambiente. Entre ellos se encuentran aquellos con tecnología de tiempo de vuelo. Estos calculan la distancia midiendo el tiempo que tarda un pulso de luz en viajar desde la fuente hasta el objetivo observado y luego hasta el detector (normalmente situado junto a la fuente). En algún sentido pueden ser considerados como sensores de radar que se basan en la luz [Kha08].

El tiempo de vuelo es directamente proporcional a la velocidad de la luz (en un medio determinado -espacio, aire o agua- y ajustado por la densidad y temperatura del medio). Los sensores de distancia basados en tiempo de vuelo se centran en reconocer el ambiente a partir de diferentes medios. En el caso de los dispositivos móviles con sistema operativo Android tienen la capacidad de reconocer el ambiente a partir de sensores de tiempo de vuelo que funcionan basados en la detección de luz y creando mapas de profundidad, estos se denominan ToF. Sin embargo, aquellos con sistema operativo iOS (últimas versiones) hacen uso de un láser que se denomina sensor de detección y alcance de luz LiDAR, el cual construye una nube de puntos que puede ser fácilmente reconocible a través de un computador [RM15].

Sistemas de imágenes tridimensionales

Los sistemas de imágenes tridimensionales basados en sensores, utilizan láseres tipo LiDAR para medir el alcance de los puntos de los objetos dentro de su campo de visión, así como el rumbo (ángulos) hacia esos puntos dentro del marco de coordenadas del instrumento. Estos sistemas suelen emitir una nube de puntos (o nubes de puntos) con coordenadas tridimensionales (normalmente X, Y, Z, ángulo y rango) asociadas a cada punto. Además, muchos de estos sistemas también emiten la intensidad de la señal devuelta desde cada punto medido en los objetos, así como como la información de color rojo, verde y azul (RGB) de cada punto [Fir21].



Figura 2.7: Sensor LiDAR (izquierda) generando nube de puntos (derecha) del Templo Sewu, ubicado en Indonesia [Fir21].

2.1.2.7. Algoritmos de reconocimiento del ambiente

En tecnologías como la realidad aumentada, el entorno juega un papel de vital importancia. Por esto, con el paso del tiempo se han desarrollado nuevas técnicas con las cuales reconocer los ambientes. Es así como existe el problema de localización y mapeo simultáneos, más conocido en su forma abreviada como SLAM. Este inicialmente aborda el problema de un robot, objeto o persona que navega por un entorno desconocido. Mientras navega por el entorno, el robot busca adquirir un mapa del mismo, y al mismo tiempo desea localizarse a sí mismo utilizando su mapa [Kha08].

El uso de los problemas SLAM está motivado de dos maneras diferentes: El primero está interesado en generar modelos detallados del entorno. Mientras el segundo, busca mantener una sensación precisa de la ubicación de un robot u objeto móvil [Kha08].

Una posibilidad alternativa, es la odometría visual. Esta se basa en el flujo óptico o el seguimiento de características en una secuencia de imágenes generados por una cámara. En la actualidad, esta técnica se ha aplicado a los vehículos de exploración de Marte, Spirit y Opportunity, en su navegación de largo alcance, y se ha comprobado que es muy útil. En particular, el algoritmo basado en la detección y el seguimiento de rasgos utilizando un par de cámaras estereoscópicas proporciona resultados fiables con buena precisión para la estimación de la distancia de conducción, así como el deslizamiento de las ruedas [MCM07].

Incluso, compañías de creación de aplicaciones en realidad aumentada como Vuforia [PTC21b] implementaron sus propios algoritmos de Visual-Inercial de localización y mapeo simultáneos (VISLAM) que combina los beneficios de la odometría visual y la localización y mapeo simultáneos (SLAM). Este invento funciona mejor en entornos con pocas características en comparación con el seguimiento basado en SLAM. Además, proporciona una estimación de la escala del mundo y proporciona robustez para recuperarse cuando el seguimiento se pierde por completo sobre las soluciones únicamente basadas en odometría visual.

2.1.2.8. Procesamiento del Lenguaje Natural

El procesamiento del lenguaje natural o NLP por sus siglas en inglés, es un campo de las ciencias de la computación que se centra en permitir que los sistemas informáticos entiendan el lenguaje de forma "natural", tal cual como lo hacen los humanos. Comúnmente, esto hace referencia a tareas como la comprensión de los sentimientos o intenciones de textos, el reconocimiento del habla y la generación de respuestas a preguntas [Kib13].

Estas tareas proveen capacidades de gran impacto en áreas como la computación cultural, posibilitando la inclusión de tecnologías inteligentes en ambientes variantes como lo pueden ser los lugares culturales tratados en esta investigación.

2.1.3. Trabajos Relacionados

En la siguiente sección se documentan las características principales de diferentes procesos de investigación basados en temáticas similares y construidos con tecnologías en común a las usadas en este proyecto.

Historical Site Reconstruction

La reconstrucción de sitios históricos por medio de realidad aumentada ha sido aplicada por distintas entidades y grupos de interés con la intención de recrear estructuras que posiblemente ya no existan ó hayan cambiado con el tiempo, entre otras razones.

En este proyecto[WCCL18] llevado acabo por académicos, se buscó integrar la historia del desarrollo urbano y los bienes culturales, con el fin de crear una conexión con los recuerdos del patrimonio cultural en la isla de Taiwán.

El objetivo principal de proyecto se llevo acabo construyendo una aplicación piloto, donde se ilustra la antigua estación de ferrocarril de Taitung a partir de modelos 3D; los cuales son desplegados por medio de tarjetas postales utilizadas como imágenes objetivo, se crearon múltiples diseños de tal manera que cada una de ellas recreó la antigua estación con sus modificaciones a lo largo de los años, hasta que dejo de existir.

Algunas de estas aplicaciones cuentan con ciertas limitaciones para llevarse acabo, sin duda al tratarse de patrimonio cultural, debe trabajarse en cooperación con entidades académicas, de industria y gobierno; para obtener datos validos y correctamente suministrados.

Towards Tantalluc

La interacción humano-computador es un factor clave para la fortalecer el éxito en la implementación de sistemas interactivos, ChimúAR y hacia Tantalluc son dos aplicaciones móviles realizadas para el Museo de América en Madrid, España. Donde se hizo uso de la realidad aumentada junto con los vídeo juegos, para alentar la exploración del museo a través de una experiencia interactiva [CIEA15], la cual resalta la cultura Chimú proveniente de Perú.

- Chimú AR

Chimú AR funciona a partir del reconocimiento de imágenes ilustrativas de los artefactos Chimú ubicados en la exhibición, en este caso se seleccionaron tres, donde se despliega información por medio de AR acerca de cada uno de ellos de tal manera que los usuarios conozcan y aprendan acerca de su cultura e historia.

- Hacia Tantalluc

La aplicación Hacia Tantalluc consiste en permitir al usuario caracterizarse como el protagonista

del vídeo juego, que toma lugar en distintas escenas relatando historias a partir de los pasos para encontrar y contruir objetos antiguos del imperio Chimú, y a su vez intenta invitar a quienes se encuentren en el museo a visitar y explorar la exhibición.

El uso de realidad aumentada, junto con computación gráfica y desarrollo de vídeo juegos permite a los desarrolladores crear e inspirar muchas formas a través de las cuales impactar para brindar información de manera interactiva a los visitantes del museo, sin embargo. implementar este sistema implica tratar con limitaciones como el avance de la tecnología, en cuanto a capacidad de almacenamiento de los dispositivos, calidad de imagen y también limitaciones en el ambiente de aplicación, puesto que se presentaban problemas de conectividad en el museo por lo que debió ser realizada de tal forma que no necesitara conexión a internet.

GoFind!

GoFind! pretende ser una aplicación móvil que aloja imágenes históricas que despliega al usuario por medio de interfaces en Realidad Aumentada (RA). Principalmente, GoFind! [LLH18] hace uso de geolocalización, esto le permite asociar a un determinado punto, un conjunto de imágenes que enseñan lo que era el lugar anteriormente. Es así, como su creación está dirigida a historiadores y turistas con gran interés hacia el desarrollo e historia de los lugares que visitan.

Esta aplicación inicialmente no hace uso del sistema de geolocalización del dispositivo móvil, por lo contrario, toma como entrada inicial valores de localización en latitudes del usuario. Con el movimiento del dispositivo móvil, GoFind! notificará aquellos lugares que reconoce en su base de datos, desplegándole al usuario la interfaz gráfica de la aplicación, la que consta de dos opciones. La primera opción le permite al usuario desplegar una imagen del lugar en una pantalla emergente de color transparente, a su vez, la segunda opción proporciona la capacidad de situar la imagen histórica como un objeto virtual aumentado. En ambas opciones, el usuario tiene la posibilidad de navegar entre las diferentes imágenes que el lugar tenga asociadas.

Quito Vision

La ciudad de Quito, ubicada en Ecuador está en busca de convertirse en una ciudad inteligente, por lo cual le está apostando al desarrollo de aplicaciones que fortalezcan áreas como el turismo. Por esta razón, en el centro histórico se desarrolló una aplicación móvil llamada Quito Vision [ASV⁺18]. Esta aplicación tiene como objetivo brindar información acerca de los edificios pertenecientes al lugar, su versión inicial esta disponible para sistema operativo móvil Android 3.0 y anteriores.

Para para proyectar la información recolectada a cerca de los edificios se hizo uso de la realidad aumentada(AR), en esta se adjuntaron también enlaces a sitios web determinados dentro del centro histórico. En el desarrollo de la aplicación decidieron utilizar JavaScript, HTML para la interfaz gráfica, geolocalización para conocer las coordenadas en el camino y reconocimiento de imágenes.

Se presentaron ciertas dificultades para mostrar algunos objetos de realidad aumentada debido a condiciones de fabricación de dispositivos, conexión a internet y el estado del clima; Además se identificaron mayores problemas cuando se utilizaba la aplicación en áreas cerradas.

Magic-Eyes

Magic-Eyes es una aplicación móvil donde se quiere resaltar el parque de Yuanmingyuan, ubicado en china; el parque pertenece a una zona donde a lo largo de la historia se ha sufrido de diversos incendios y saqueos, razón por la que no se ve muy atractivo.

El objetivo se basa en resaltar una ruta no tan recorrida para guiar turistas e interesados a través de las reliquias culturales del parque [XDYY16], se trabajó desde la innovación cultural utilizando una historia basada en hechos reales, conocida por los trabajadores del lugar, y recrearla de manera interactiva para motivar a los usuarios en el recorrido. La aplicación utiliza distintas figuras ubicadas a lo largo del camino, con las cuales se crean escenarios y panoramas animados del parque por medio de realidad aumentada; la historia es guiada por la representación de un guardabosques que narra la historia y los pasos a seguir, también, se despliega un mapa animado en pantalla que se atribuye a la ruta escogida, mostrando la ubicación y sentido hacia donde se dirige el usuario.

El uso de realidad aumentada, evidenció un beneficio en el turismo de la zona seleccionada, según encuestas realizadas a un grupo de personas que utilizaron la aplicación y terminaron el recorrido satisfactoriamente, demostrando que la técnica de innovación y la historia generaron mejor resultados que varios intentos realizados para revivir las zonas menos atractivas del parque, aun así, se encontraron con dificultades para extraer la información acerca de su historia, puesto que con la poca interacción turística que tenía, pocos trabajadores del lugar la conocían.

Riverwalk

La aplicación móvil Chicago 0,0 Riverwalk AR [MGA16] propone una experiencia de realidad aumentada a todos quienes visiten las cercanías del río Chicago. Esta experiencia se basa en desplegar contenido en dos dimensiones (2D) a través del dispositivo móvil de los visitantes. Esto lo hace por medio de imágenes obtenidas a través de recursos como museos. Sin embargo, el sistema no cuenta con contenido propio aparte de la interfaz de usuario.

La naturaleza y arquitectura propia del lugar obligó a los creadores a implementar un enfoque llamado cámara dual. Su objetivo se centraba en estimar la posición y orientación absoluta de la cámara, esto con el fin de obtener un resultado mucho más preciso y, a su vez, una mejor experiencia de usuario. Para lograrlo combinaron información de la cámara de la aplicación junto con el sensor de la cámara del móvil, guiando al usuario a ubicar el dispositivo en el lugar apropiado.

Sin duda el estado del arte le permitió al equipo extraer ideas y ejemplos de proyectos similares, los cuales inspiraron algunas de las funcionalidades finalmente implementadas.

[Historical Site Reconstruction](#) evidencia lo interesante que fue exponer la historias de lugares, a través de imágenes comparativas con las fachadas actuales y antiguas. [Towards Tantalluc](#) expone factores interactivos para llamar la atención del público por medio de realidad aumentada, por lo que dentro del proyecto se reconoció la importancia de establecer acciones para interactuar con el Asistente Informativo.

Los artículos de [GoFind!](#) y [Quito Vision](#) brindan su experiencia con la geolocalización y cómo aporta para ciertas funciones, como por ejemplo, utilizar un mapa para ubicar a los usuarios. Así mismo, [Magic-Eyes](#) y [Riverwalk](#) presentan como la realidad aumentada, puede contribuir a cambiar los comportamientos de la zonas, generando mayor turismo. En definitiva, en fases como el análisis y diseño se tuvieron en cuenta todas estas recomendaciones para darle valor al proyecto.

Metodología, Análisis y Diseño

3.1. Metodología

El profesor Mitchel Resnick investigador en aprendizaje del laboratorio de medios del Instituto Tecnológico de Massachusetts (MIT) es quien dirige un grupo llamado "Lifelong Kindergarten", dedicado a ayudar a los niños de todas las edades a jugar, pensar y experimentar con el diseño y la tecnología. Tras un continuo estudio del aprendizaje con niños, Resnick planteó un enfoque con el que los estudiantes incursionarían en el análisis, implementación y evaluación de proyectos. Para lograrlo, dio vida al espiral infinito especificado en la **Figura 3.1** a continuación:

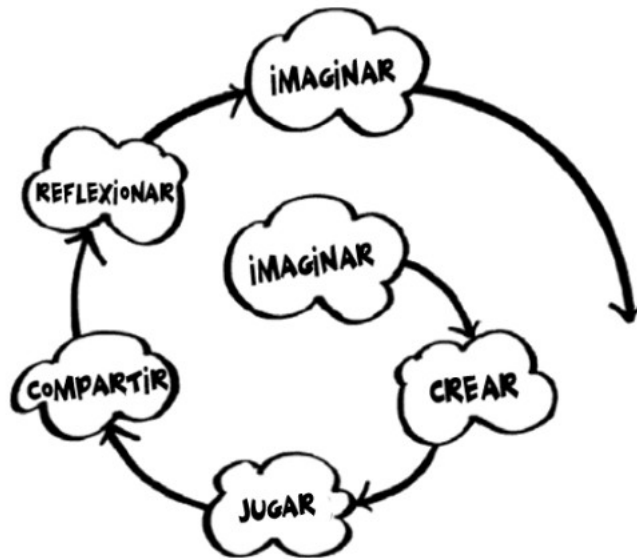


Figura 3.1: La espiral del pensamiento creativo. [Res07]

La proyección del futuro expresada en la espiral denota la capacidad de que todo lo que se imagine puede convertirse en realidad, a través de la repetición de cinco pasos clave:

- **Imaginar:** Visualizar aquello que se desea crear sin límite alguno, desprendiéndose de juicios de valor como los conocimientos previos que atan la capacidad imaginativa.
- **Crear:** Hacer realidad lo planteado o imaginado por medio de composiciones, dibujos o artefactos.

- **Jugar:** Explorar, disfrutar, escuchar, tocar y utilizar todas las creaciones haciendo énfasis en el reconocimiento de que todo es mejorable y que las evoluciones surgirán en cualquier momento.

- **Compartir:** Enseñar o mostrar el proyecto a otras personas teniendo en consideración sus opiniones.

- **Reflexionar:** Llevar a cabo un proceso de retroalimentación donde se realicen los cambios oportunos.

Su ejecución iterativa lleva a la creación de artefactos que promueven un pensamiento investigativo y experimental. Por ello, para la realización de este proyecto de grado se hará uso de La espiral del pensamiento creativo como fuente principal del proceso metodológico, al mismo tiempo que se integrarán a esta investigación las fases del proceso de Ingeniería de Software por medio de la utilización de la metodología ágil de proyectos Scrum, a través de actividades tales como el levantamiento de requerimientos por medio de historias de usuario y el desarrollo incremental basado en ciclos temporales.

En total se planearon cinco ciclos para el proyecto, por lo que las etapas de la espiral creativa se relacionan de la siguiente manera, en la **Figura 3.2** se encuentran descrito el ciclo 0, que es distinto a los demás puesto que es aquí donde se realiza todo el proceso de análisis y diseño; y los ciclos del 1 al 4 que constan de las mismas etapas. Al final de cada ciclo se llevaron a cabo las fases de Compartir y Reflexionar, donde se resumían los objetivos alcanzados y se identificaban aquellas cosas que debían ser enviadas al siguiente ciclo. Por ello después del ciclo 0, se inicia con planeación, donde se organiza el ciclo actual conforme los requerimientos y resultados del ciclo pasado.

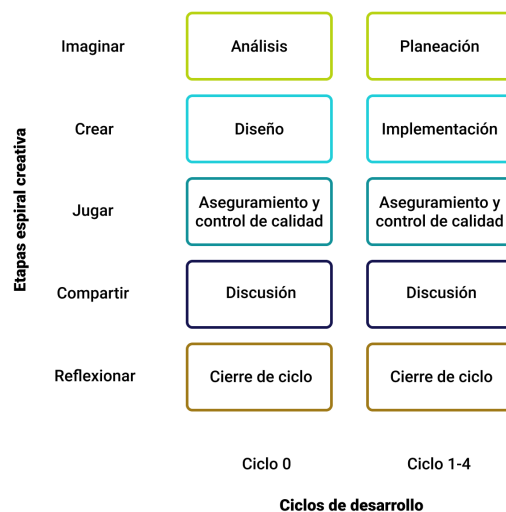


Figura 3.2: Relación de ciclos con fases de la espiral creativa.

3.1.1. Tipo de Estudio

Basados en el constante crecimiento y usabilidad de la Realidad Aumentada (RA) en términos sociales y culturales, se busca desarrollar un prototipo de aplicación que se adapte a las necesidades del medio a la que va dirigida, en este caso, el barrio San Antonio de Cali. Por ello, es imprescindible centrar la investigación bajo un estudio exploratorio, puesto que aunque la Realidad Aumentada es utilizada en muchos aspectos, no es aún una tecnología totalmente explorada y enfocada hacia ámbitos culturales, ya que no es común ver este tipo de aplicación como método para fortalecer turismo en lugares reconocidos por su tradicionalidad y bajo impacto tecnológico.

3.2. Análisis

Con el fin de identificar y definir los lugares a utilizar en el prototipo, se lleva a cabo un proceso de adaptación en el que se obtiene la información necesaria de manera virtual y presencial. Inicialmente se reconoce el barrio San Antonio por medio de herramientas tales como TripAdvisor, Google Maps y Street View (Ver **Anexo8.1**). Cada una de ellas brinda diferentes aspectos a tener en cuenta. Por ejemplo, TripAdvisor bajo su ideal de proporcionar reseñas de contenido relacionado con viajes, posibilita reconocer la opinión de las personas que han visitado el barrio y sus lugares más representativos. Mientras que Google Maps y Street View permiten ubicar geográficamente y distinguir visualmente el barrio de manera muy cercana a la realidad. Es así, como el uso de estas herramientas facilita la elección de los lugares bajo criterios tales como:

- Opiniones y valoración de locales y visitantes.

- Relevancia histórica reseñada en la web.
- Ubicación.

De esta manera, se da la selección de diferentes opciones de lugares en el barrio San Antonio que cumplieran con los criterios mencionados. Posteriormente, se procede con una salida de campo y un acercamiento a los propietarios que permitió evaluar y confirmar de manera presencial la elección realizada. Los lugares elegidos constan de un lugar público y dos lugares privados.

1. **Iglesia de San Antonio:** La Iglesia de San Antonio fue construida en el año 1786 y se convirtió en propiedad pública en 1944. Además, fue declarada monumento nacional desde 1997. Su importancia histórica en la ciudad la ha llevado a obtener las mejores valoraciones de reseñas en línea (4,5 estrellas). En ellas se resalta su tradicionalidad, su arquitectura antigua y ubicación en lo alto de la colina del barrio. Alrededor se encuentran ventas de artesanías, artistas como cuenteros y teatreros, restaurantes y el monumento a la maceta. También, el área cercana a la iglesia ha sido utilizada en varias ocasiones para diferentes eventos. Incluso, algunos turistas la consideran el mejor lugar de la ciudad.
2. **La Linterna:** La Linterna es una de las imprentas tipográficas más antiguas de Colombia. Desde el año 1938 esta imprenta ha estado prestando sus servicios en el barrio San Antonio. Se caracterizan por el uso de tipos móviles para la realización de las impresiones, técnica que hoy en día la tecnología ha reemplazado, por ello, también usan impresión digital. El lugar realizó carteles para promocionar conciertos de bandas tales como Gun's and roses, Metálica y cantantes como Elton John, generando su reconocimiento no solo en la ciudad sino también el país. La Linterna se ha convertido en un icono de arte gráfico en el barrio San Antonio, su taller fue abierto al público con el fin de exponer el proceso de impresión y los resultados del mismo.
3. **Tertuliadero La Colina:** La colina es reconocida como la tienda más antigua de la ciudad de Cali, fue creada hacía el año 1942. Inicialmente fue catalogada como una tienda de barrio y con el paso del tiempo se convirtió en un tertuliadero icono en la ruta turística de la ciudad. Esto se dio gracias a la constante visita de personas que querían pasar un rato agradable en donde disfrutaban del ambiente y la música. El lugar preserva el detalle de una arquitectura colonial al ser una casa de la época, junto con una decoración tradicional y bohemia, además, conserva también la imagen característica desde su creación permitiéndole ser un lugar patrimonio al cual visitar en el barrio San Antonio.

3.2.1. Conocimiento de los sitios escogidos por los habitantes (Encuesta)

Al realizar una encuesta (Ver **Anexo 8.2**) a un grupo de 32 personas se evidencia que San Antonio es considerado uno de los barrios más representativos de la ciudad con un puntaje de aceptación

de 8,93 en una escala del 1 al 10, donde los participantes han tenido experiencias enriquecedoras entre familiares y amigos, de las cuales resaltan su gastronomía, arquitectura y un ambiente acogedor y bohemio.

De los tres lugares escogidos (la Iglesia, Tertuliadero La Colina, Imprenta La Linterna) para ser parte del Asistente Informativo Virtual, solo La Iglesia es conocida por casi el 85% de personas, mientras que La Linterna y La Colina solo la conocen el 7% y 41% de los encuestados respectivamente, lo cual es interesante puesto que estos últimos lugares poseen una historia importante dentro del barrio y la ciudad, que muchos no conocen, algunos de los encuestados resaltaron sus experiencias en el Tertuliadero como "Buena música, cerveza, aborrajados", pero, también se encontraban confusiones como "Es divertido escuchar a los cuenteros con sus magnificas historias al igual que bailar algunas danzas andinas", haciendo referencia a los cuenteros ubicados en la Iglesia.

Con respecto a La Linterna solo 2 de 32 de los encuestados manifestaron conocer acerca de ella diciendo "Conozco los carteles tradicionales que hacen" y "Que imprimen por decir algo, a la antigua. Alguna vez salió en el periódico".

De esta manera, se considera de gran importancia resaltarlos en el aplicativo para que los usuarios turistas y locales conozcan acerca de ellos y sus aspectos más notables. En las **Figuras 8.4, 8.5, 8.6 del Anexo 8.2** se adjuntan los porcentajes de respuestas sobre visitas y conocimiento de los lugares.

3.2.2. Metodología de levantamiento de requerimientos

Para llevar a cabo el levantamiento de requerimientos se tiene en cuenta el contexto creativo y ágil inmerso en el desarrollo del proyecto, por lo que se encontró pertinente hacer uso de historias de usuario y storyboard para la realización de este paso inicial en la etapa de análisis y diseño. Esto fue realizado en la fase de diseño del ciclo 0 (Ver **Figura 3.2**)

Historias de usuario

Las historias de usuario hacen parte de las diversas técnicas para plantear los requerimientos del software, usualmente se hace uso de una representación de notas de papel (stiky notes) para denotar la información que contienen, puesto que estas simbolizan una lluvia de ideas o resumen de conversaciones llevadas a cabo con el cliente o grupo de trabajo.

Estas se escriben en base a las acciones generales que debe cumplir un sistema y no un detalle de las mismas, de esta manera se tiene el concepto de lo que se quiere lograr permitiendo agilidad y rapidez a la hora de necesitar modificaciones, para su uso es importante disminuir la cantidad de texto y es preferible dividir los requerimientos en pequeñas partes, puesto que se enfatiza la comunicación verbal más que la escrita y, además, permite que sea más objetiva cada una de las historias de usuario, sin embargo detalles importantes o necesarios pueden ser anotados directamente en la nota. A la hora de hacer pruebas en función de los requerimientos, las historias de usuario

se vuelven un gran aliado cuando están correctamente escritas, ya que en la fase de pruebas puede usarse directamente como el criterio de aceptación y de esta manera saber cuando se ha validado satisfactoriamente.

Storyboard

A la hora de realizar el proceso de levantamiento de requerimientos, se encuentra la necesidad de buscar un mecanismo para organizar, dividir y dirigir el orden y estado del proyecto en relación a los objetivos que se deben alcanzar en un momento determinado, para esto se encontró efectivo hacer uso de un storyboard, el cual permite mantener un seguimiento continuo al desarrollo del prototipo del asistente informativo virtual.

Un storyboard representa el flujo de trabajo durante todo el periodo de implementación y desarrollo; permite ubicar, en este caso, las historias de usuario resultantes en la fase de análisis, en ciclos o sprints y relacionar cada una con actividades generales del proyecto, además, permite realizar anotaciones sobre el estado individual de cada una de las historias de usuario para estar en una comunicación activa con los desarrolladores, lo cual, brinda rapidez y eficiencia para el control del avance del proyecto.

El uso de historias de usuario y storyboarding ha sido visto en varias metodologías ágiles como scrum y XP puesto que estas se enfocan en una documentación menor a la de metodologías pesadas, y conserva un manejo sencillo para el control de cambios dentro de requerimientos y alcances (Ver **Figura 3.3**).

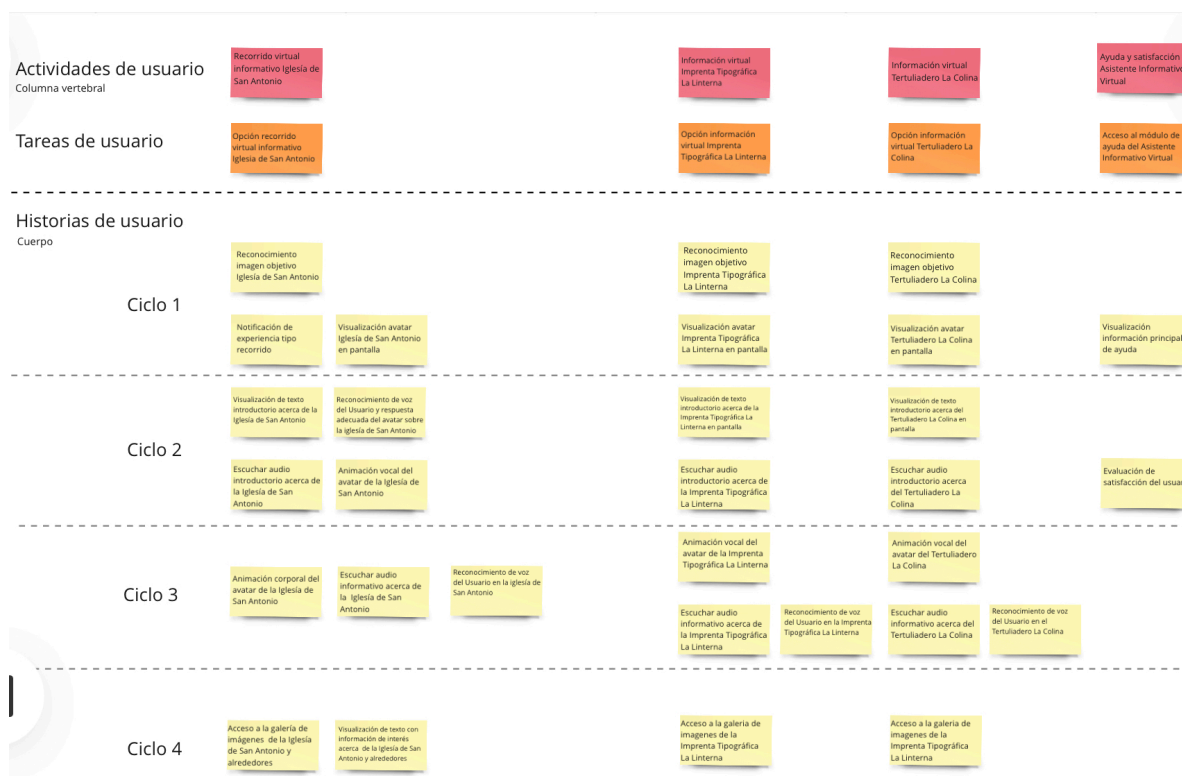


Figura 3.3: Tablero historias de usuario del sistema.

3.2.3. Arquetipo de Usuario

Con el fin de direccionar de manera objetiva el diseño del aplicativo del Asistente Informativo Virtual, se realizó un proceso de análisis para caracterizar aquellos usuarios que utilizarían la aplicación. Además de la información obtenida durante del planteamiento del problema y después de realizar algunas visitas al barrio San Antonio y apoyarse con material accedido por medio de la red, el equipo pudo concluir que los aspectos relevantes o a tener en cuenta dentro del usuario objetivo son:

- Edad: Entre 15-45 años.
- Nivel de adaptabilidad a la tecnología: Medio - Alto.
- Idioma Nativo: Español, Inglés, Francés.
- Intención: Visitar y conocer acerca del barrio San Antonio y algunos de sus lugares representativos.

3.2.4. Levantamiento de requerimientos

Para llevar a cabo la identificación de los requerimientos, el equipo desarrolló una actividad que se dividió en varias sesiones de discusión, donde se ejecutó una lluvia de ideas y aportes de los integrantes que quedó plasmada paralelamente en historias de usuario utilizando la aplicación **Miro** (Ver **Figura 3.3**), posterior a esto se presentó una retroalimentación con el director del proyecto quien especificó correcciones en el planteamiento de cada uno de los requerimientos. Se mejoró en principal medida la redacción de cada una de las historias de usuario y su organización estructurada en ciclos definidos dentro del Storyboard.

Cabe resaltar, que la aplicación tiene como objetivo informar a quienes visitan el barrio San Antonio, acerca de datos generales dando visibilidad a lugares no tan reconocidos actualmente como la Imprenta La Linterna, el Tertuliadero La Colina y también su icono representativo que es la Iglesia, así que, fue de gran importancia separar los requerimientos específicos en cada uno de los lugares a trabajar, para esto se crearon cuatro secciones:

- Recorrido virtual informativo Iglesia de San Antonio.
- Información virtual Imprenta Tipográfica La Linterna.
- Información virtual Tertuliadero La Colina.
- Ayuda y satisfacción Asistente Informativo Virtual.

Teniendo en cuenta lo anterior, se dio una clasificación para los lugares con respecto al tipo de experiencia que se va a brindar, las cuales son experiencia tipo recorrido y experiencia tipo informativa, donde el tipo recorrido abarca dos o más secciones de un lugar, por lo que hay datos para cada una de estas secciones, mientras que la experiencia tipo informativa abarca solo una sección general del lugar y brinda información acerca de la misma.

Ahora bien, los requerimientos se centran en las funcionalidades directamente de la aplicación y su flujo de uso, además, de incluir un apartado de comunicación activa con los usuarios que, permita obtener recomendaciones sobre la usabilidad y experiencia brindada por el Asistente Informativo Virtual. Así mismo, se clasificaron a partir de ciclos de desarrollo en la etapa de implementación.

3.2.5. Requerimientos funcionales

A continuación (Ver **Figuras 3.4 a 3.7**) se anexan algunos de los requerimientos funcionales especificados para la aplicación (Ver requerimientos completos en el **Anexo 8.3**):

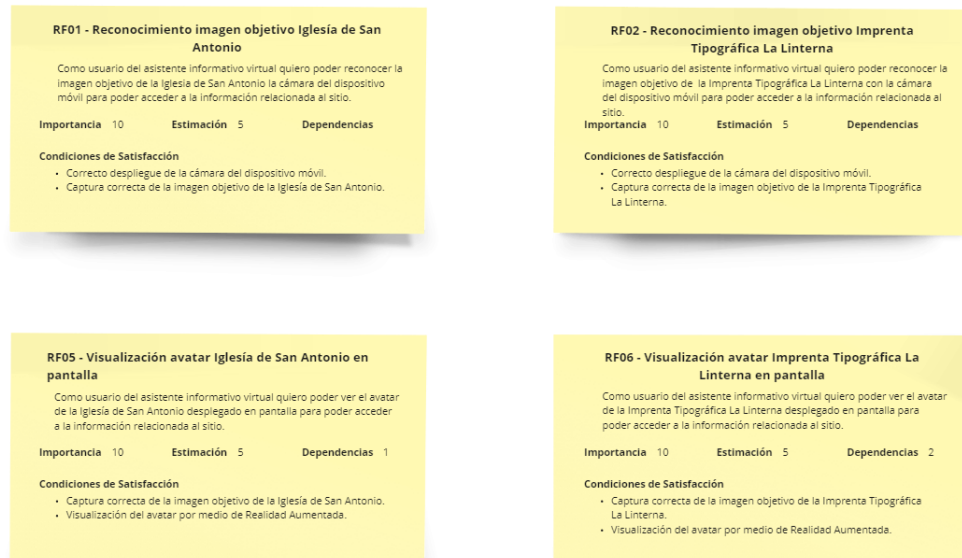


Figura 3.4: Ciclo 1 Parte 1: Historias de usuario definidas.

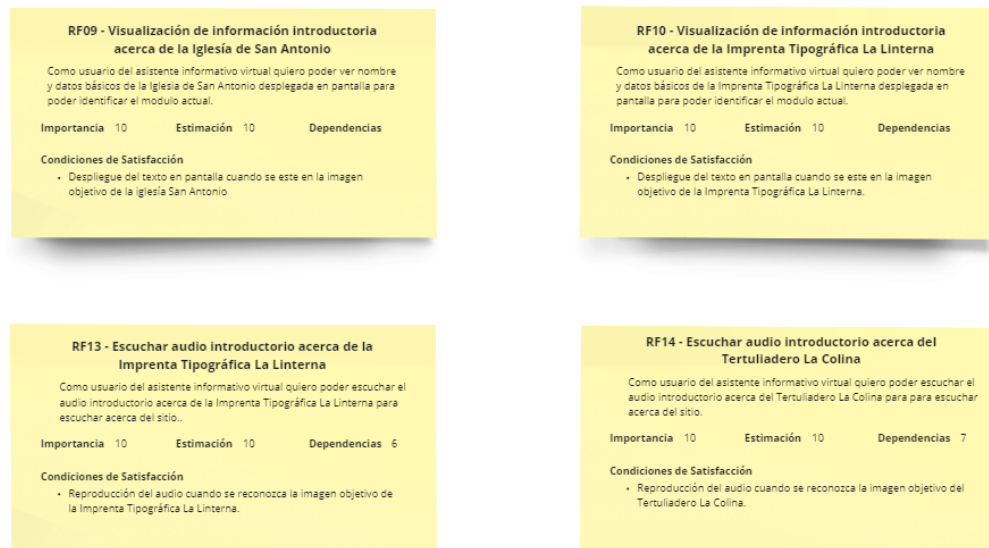


Figura 3.5: Ciclo 2 Parte 1: Historias de usuario definidas.

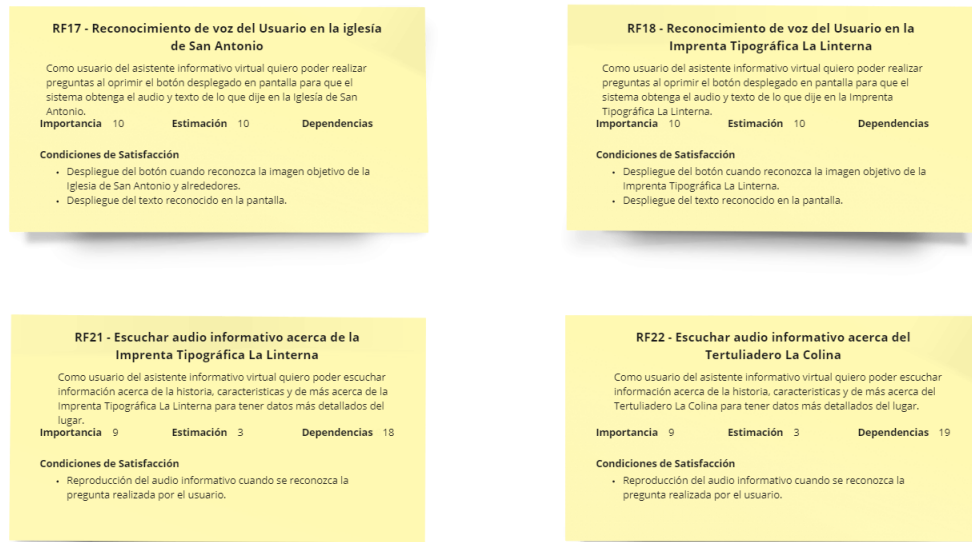


Figura 3.6: Ciclo 3 Parte 1: Historias de usuario definidas.

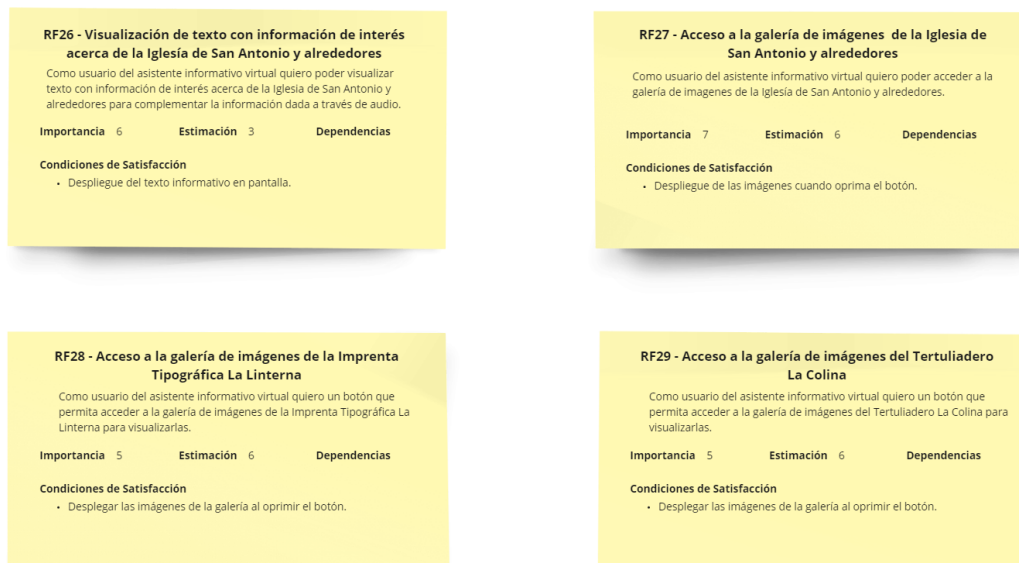


Figura 3.7: Ciclo 4: Historias de usuario definidas.

3.2.6. Requerimientos no funcionales

Los requerimientos no funcionales se organizaron de acuerdo a atributos de calidad tales como Portabilidad, Desempeño y Usabilidad en un storyboard aparte de los requerimientos funcionales,

en la **Figura 3.8** se listan los requerimientos definidos.

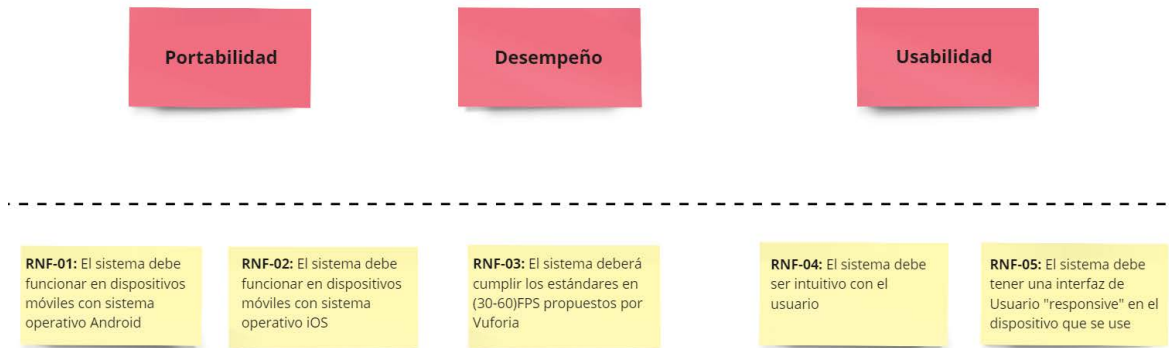


Figura 3.8: Tablero de requerimientos no funcionales.

3.3. Diseño

Para el apartado de diseño se presenta la planificación de la arquitectura del sistema, los modelos de datos planteados y la maquetación de la interfaz gráfica del asistente informativo virtual.

3.3.1. Diagramas UML

Los diagramas UML que se presentan a continuación pertenecen al conjunto de documentación técnica del proceso de investigación en su etapa de diseño.

3.3.1.1. Arquitectura

La arquitectura es una pieza clave en todo proyecto de tecnología. En este caso, se planteó hacer uso de una arquitectura de microservicios controlada por eventos. Este enfoque se eligió principalmente debido a que el sistema contaba con un conjunto de herramientas y servicios que se comunicaban tanto localmente como a través de la red. Por ende, al usar una arquitectura de bajo acoplamiento y alta cohesión le permitió al prototipo hacer uso de memoria y procesamiento local, pero también, hacer uso de hasta 6 diferentes servicios virtualizados en servidores de Microsoft Azure y Firebase.

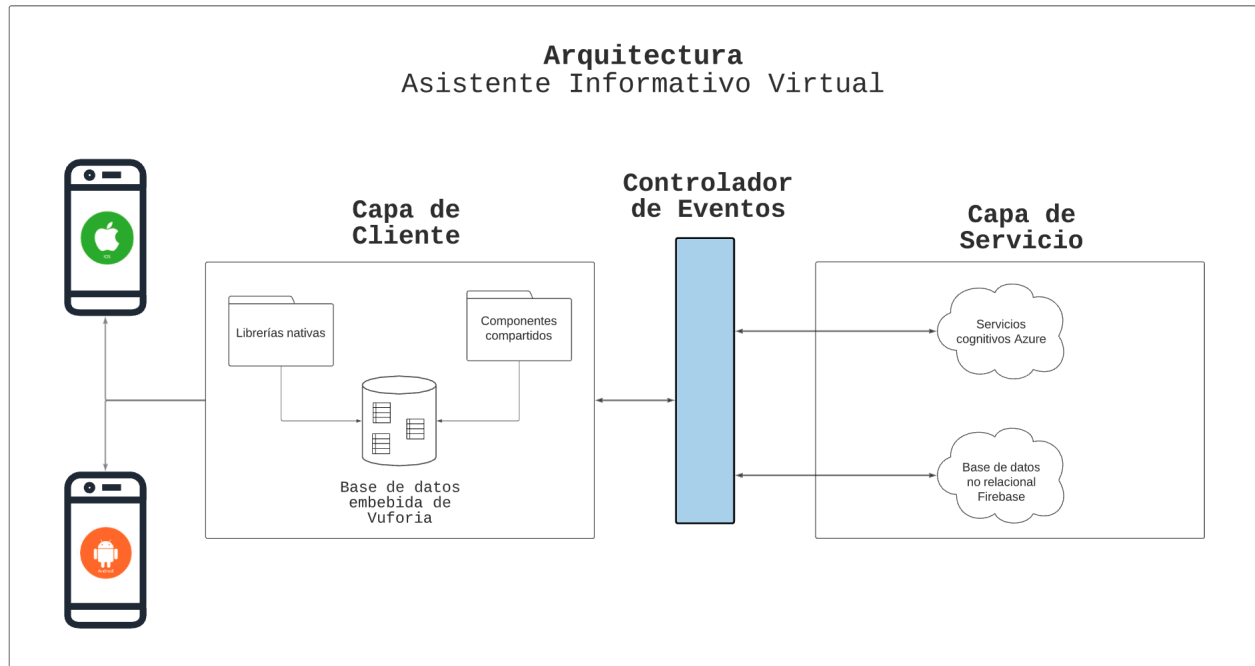


Figura 3.9: Planteamiento arquitectural abreviado. Cliente se comunica con el servidor a partir de microservicios controlados por eventos.

De esta manera, la Capa de Cliente (Ver **Figura 3.10**) se compone por dos grupos de componentes y una base de datos embebida. Al ser el prototipo un desarrollo móvil, se usaron los sistemas operativos Android y iOS para su despliegue, por esto se definió un grupo donde se ubicaron las clases y librerías nativas usadas. Además, dado que el desarrollo se basó en multiplataforma se definió un grupo de componentes que ambos sistemas operativos accedían, entre ellos, la interfaz de usuario, los modelos 3D aumentados y las animaciones de los mismos. Así mismo, de manera embebida se creó una base de datos que alojaba los objetivos aumentables como por ejemplo la malla en 3D de la fachada de la iglesia de San Antonio (Ver **Figura 4.22**).

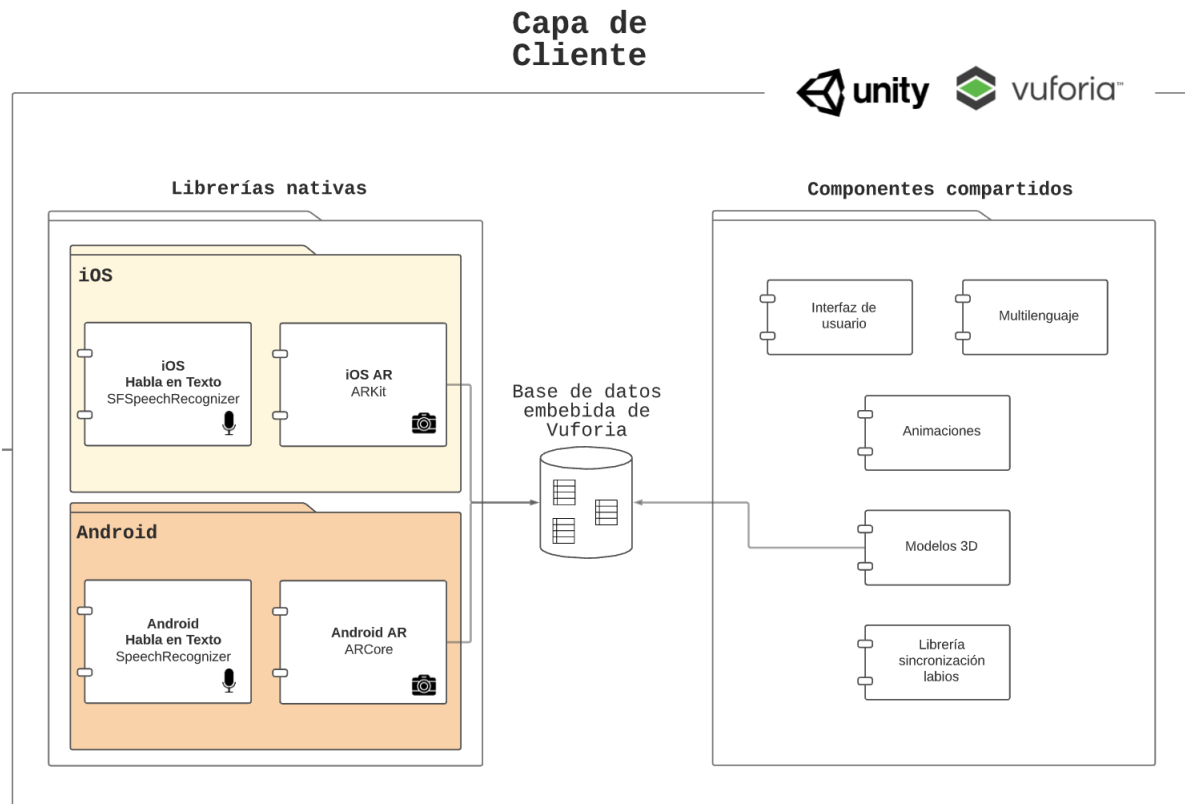


Figura 3.10: Capa de Cliente.

De igual modo, los eventos definidos representaron las conexiones entre las dos capas planteadas para la arquitectura (Capa de Cliente y Capa de Servicio - Ver **Figura 3.9**). En este sentido, el equipo se encargó de diseñar un controlador de eventos local.

Este controlador se basó en el desarrollo de una máquina de estados finita controlada por diferentes eventos, los cuales serían transiciones. En la **Figura 3.11** se evidencia la definición de estados y eventos que permitían acceder a cada una de las capas definidas en el arquitectura del Asistente Informativo Virtual.

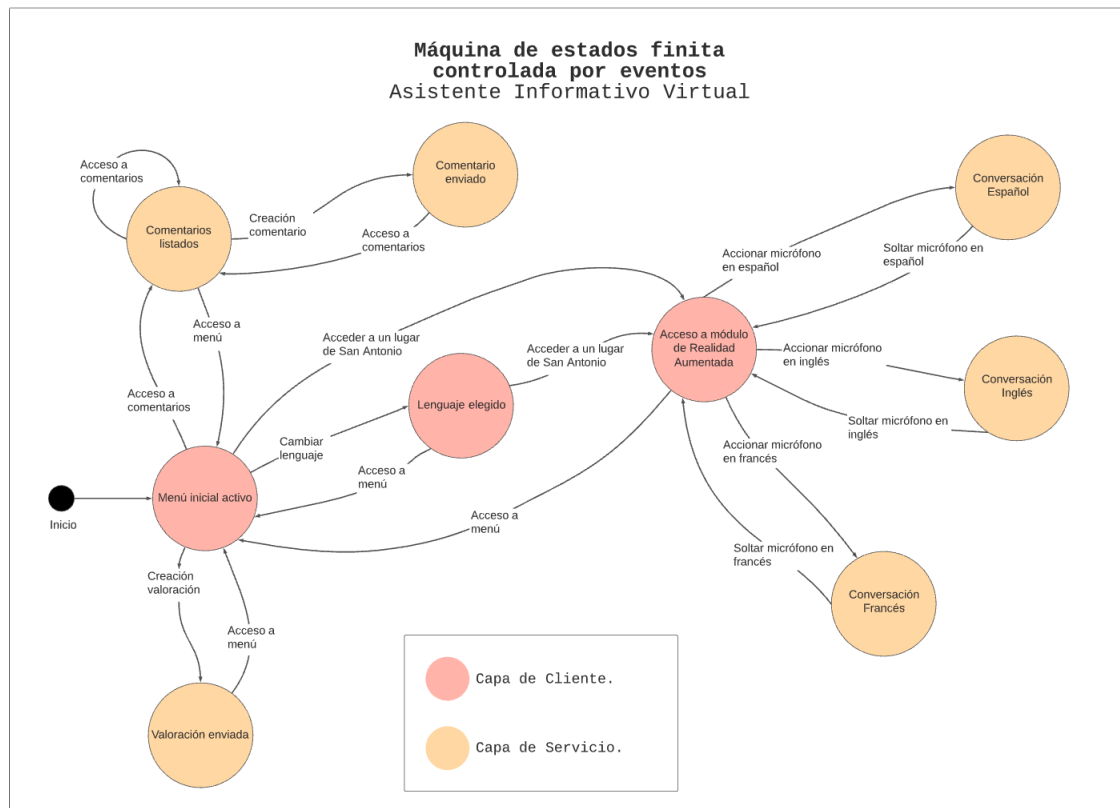


Figura 3.11: Máquina de estados finita controlada por eventos para el acceso a los componentes de la capa del cliente y los servicios de la capa del servicio.

De igual manera, con el fin de especificar la interacción que se lleva a cabo con los avatares, se detallaron los eventos que permiten la interacción a manera de conversación. Además, se tuvieron en cuenta los comandos que permiten que los procesos técnicos sucedan. Esto se evidencia en la **Figura 3.12**

Así mismo, con el fin de tener una ruta de comunicación con los servicios alojados en la red se definió una Capa de Servicio (Ver **Figura 3.13**) donde se diferenció cada uno de los servicios que comprendían el uso de computación en la nube. Para ello, se definieron dos conjuntos de herramientas. El primero se basaba en servicios cognitivos dispuestos por Microsoft Azure para la generación de conversaciones y el entendimiento del lenguaje natural. La segunda era el alojamiento de información a partir de esquemas no relacionales almacenados en el servicio de base de datos en tiempo real de Firebase.

Estas herramientas se virtualizaron en diferentes servidores ubicados lo más cerca posible al lugar propio de la investigación Cali, Colombia. Esto principalmente con el fin asegurar bajas latencias y

validar su alta cohesión.

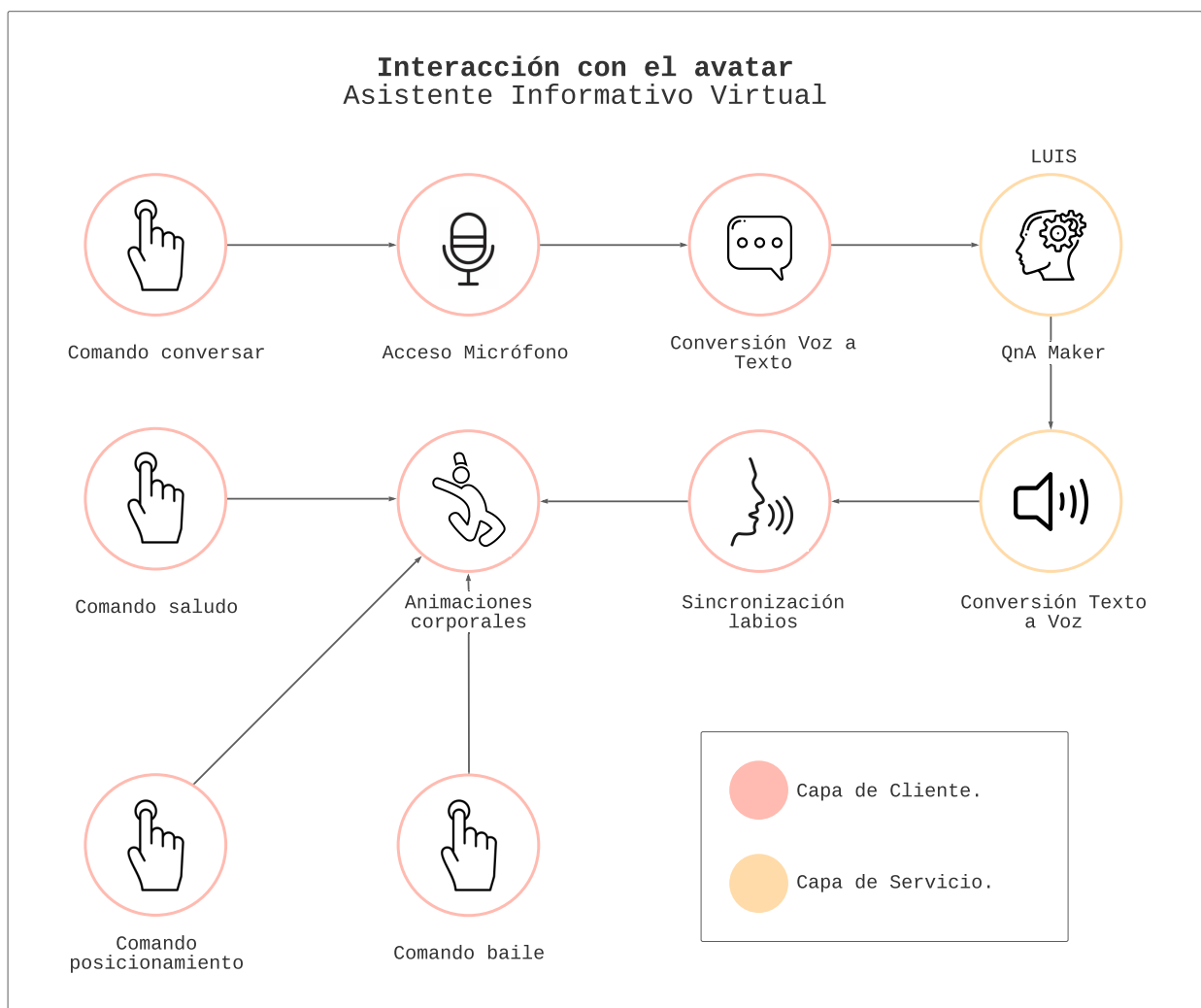


Figura 3.12: Eventos y comandos de interacción con el avatar.

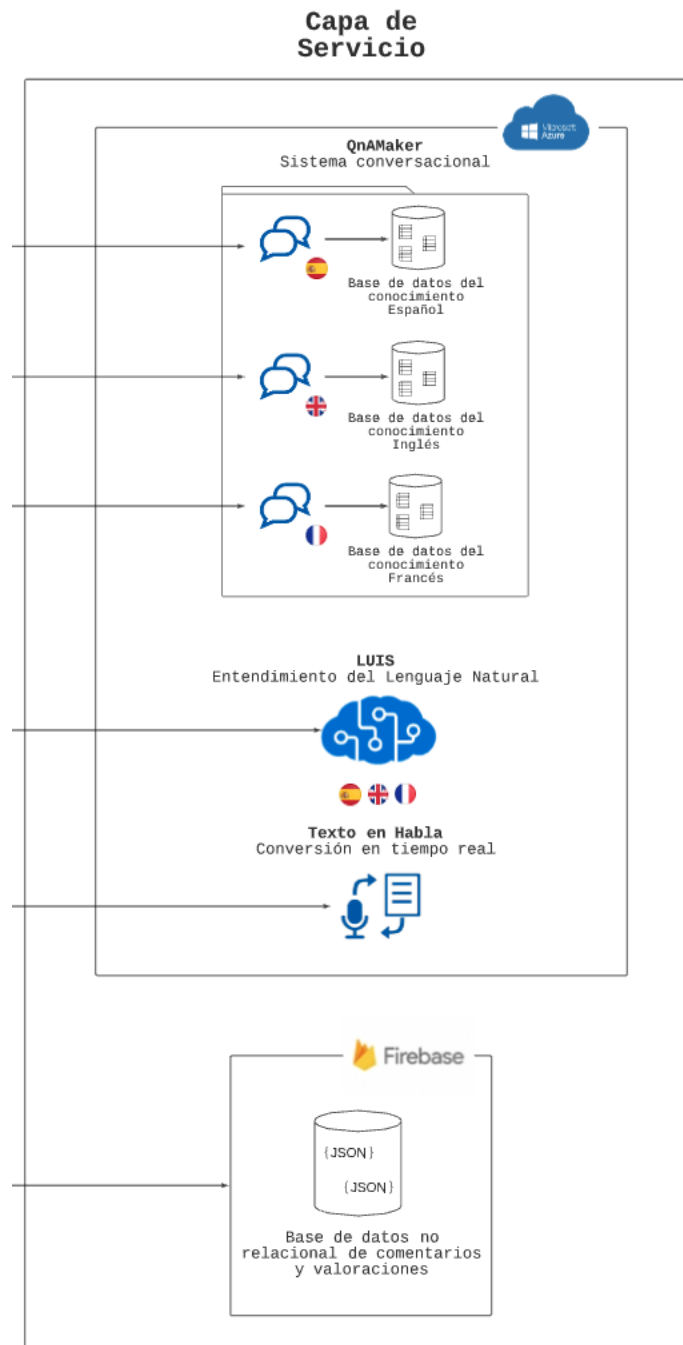


Figura 3.13: Capa de Servicio.

Finalmente, por medio de la **Figura 3.14** se evidencia conjuntamente las capas y el controlador de eventos definidos para la arquitectura del prototipo.

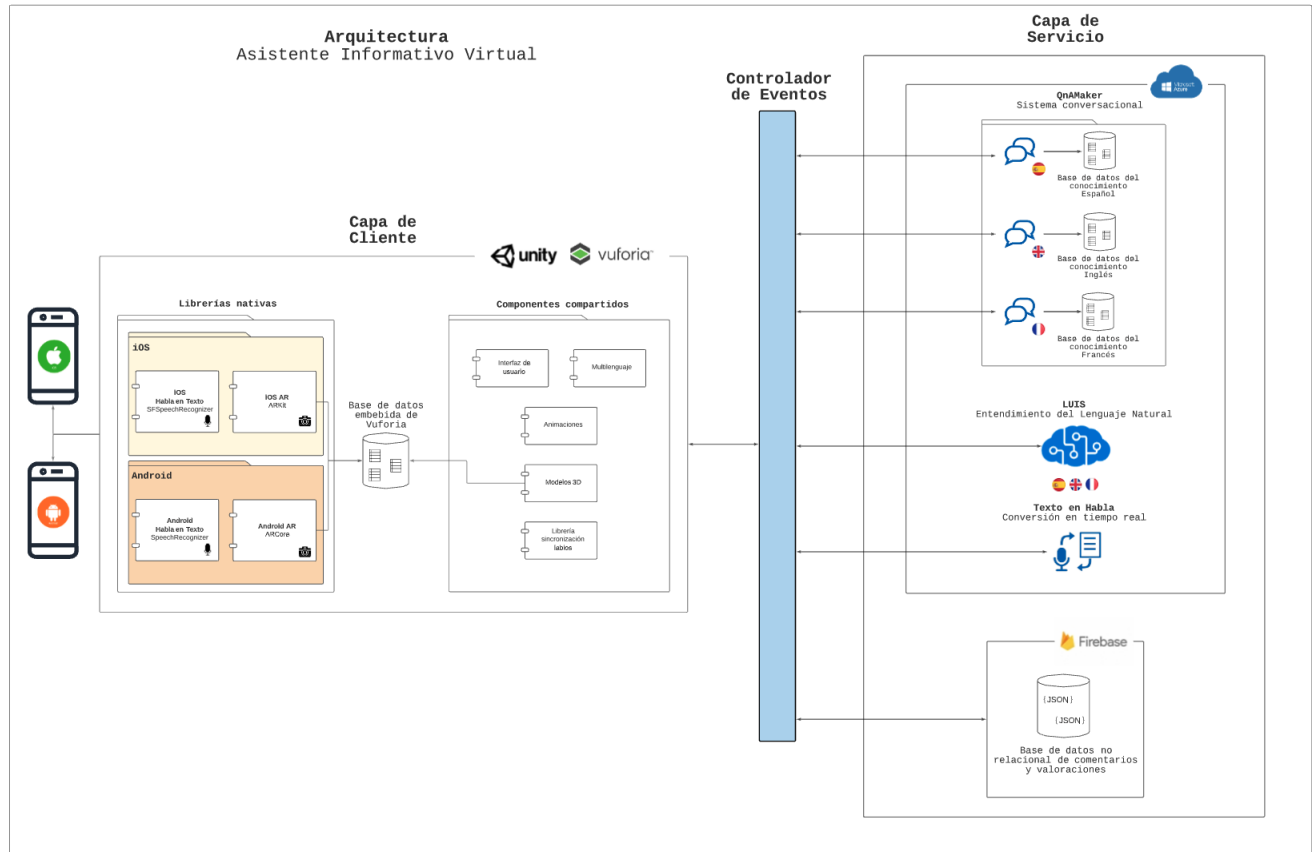


Figura 3.14: Arquitectura de microservicios controlada por eventos. Se definen una Capa de Cliente y una Capa de Servicio. Además, estas capas se comunican por medio de un controlador de eventos que está basado en la ocurrencia de sucesos para la generación de transiciones y peticiones (Ver Figura 3.11).

3.3.1.2. Diagrama de flujo

Tal como se evidencia en la **Figura 3.15**, el diagrama de flujo define los dos principales módulos o menús del prototipo. Es por esto, que está encabezado por una decisión inicial que da acceso a cada uno de ellos. El menú de ayuda y valoración del sistema le provee al usuario la posibilidad de retroalimentar la experiencia en la aplicación, lo que implica el despliegue y el envío de un formulario con información, pasos que están dados por una decisión que también permite acceder al listado de ayudas.

Por otro lado, el menú de realidad aumentada del barrio San Antonio da acceso a las principa-

las funcionalidades del Asistente Informativo Virtual. Principalmente, el sistema posibilita el acceso a los tres lugares elegidos para la investigación, los que se diferencian por métodos informativos de tipo recorrido e informativo.

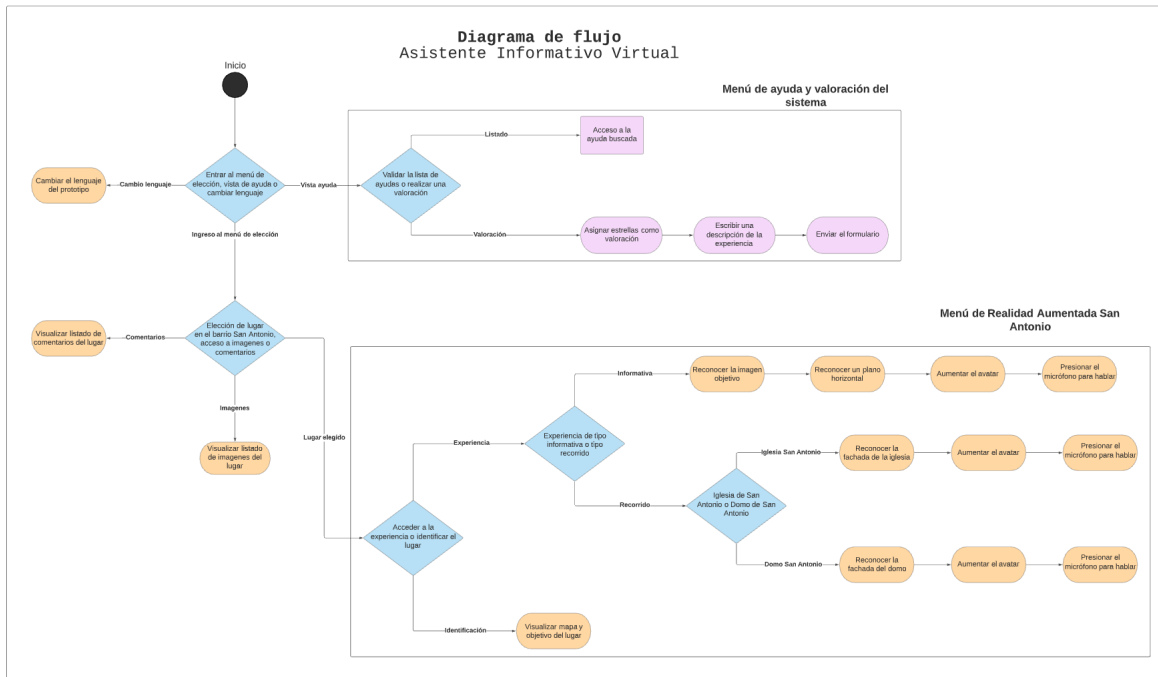


Figura 3.15: Diagrama de flujo del Asistente Informativo Virtual.

3.3.1.3. Diagrama de caso de uso

Siguiendo con una línea de documentación técnica basada en diseño, en la **Figura 3.16** se muestra el diagrama de casos de uso que representa las posibles acciones que tiene el usuario a la hora de interactuar con la aplicación.

Por ello, se definen tres casos de uso. El primero es la visualización de avatares en realidad aumentada. Este caso de uso es el principal puesto que el actor o usuario final se ve en la necesidad de llevar a cabo diferentes actividades previas como la elección de uno de los lugares del barrio San Antonio y el reconocimiento de áreas y planos horizontales. Así mismo, este caso de uso habilita al actor a acceder a funcionalidades clave como el reconocimiento de la voz por medio del acceso al micrófono.

Los siguientes casos de uso, permiten entrar a las secciones de comentarios y valoración del prototipo. Es así como, por ejemplo, el poder comentar en la aplicación incluye el acceder al listado de comentarios creados para cada uno de los lugares. También, el enviar una valoración acerca de la experiencia en el sistema incluye el poder navegar por cada una de las opciones de ayuda que

proporciona el prototipo.

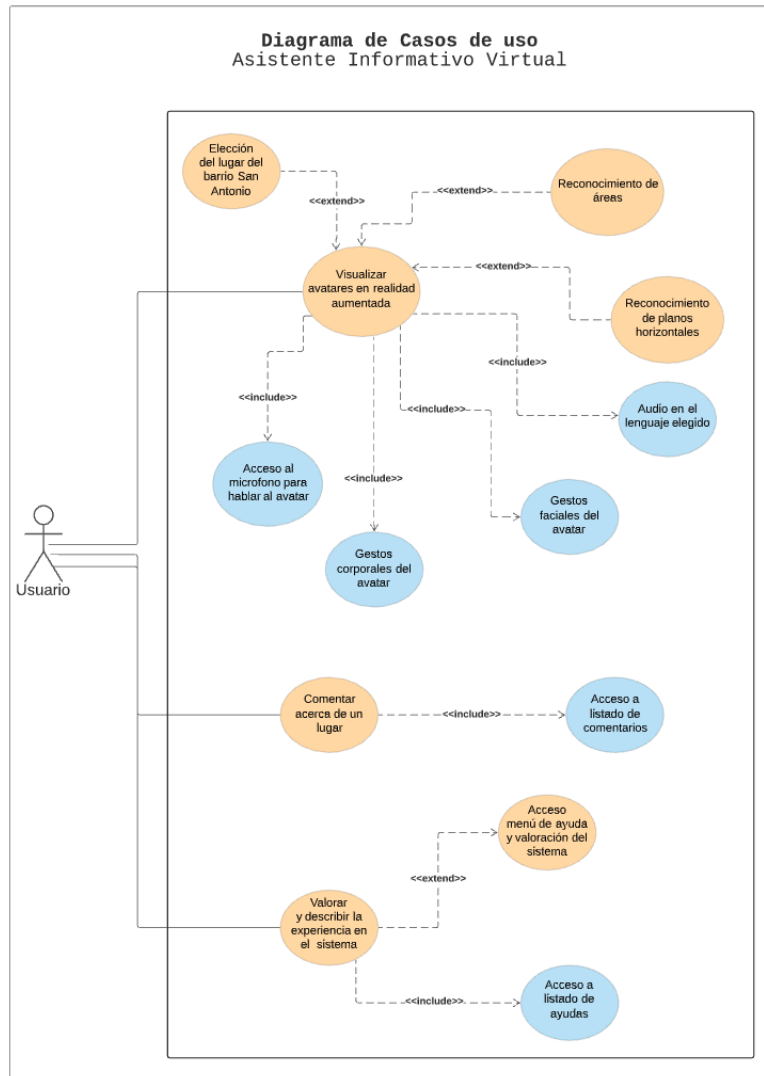


Figura 3.16: Diagrama de caso de uso del Asistente Informativo Virtual.

3.3.2. Modelo de datos

Haciendo énfasis en la arquitectura presentada en la **Figura 3.14** es pertinente tener presente que se hizo uso de tres diferentes tipos de bases de datos en el prototipo.

La primera base de datos funciona de manera embebida como un sistema de recolección de objetivos

reconocibles por la aplicación. Estos objetivos se almacenan en la base de datos como conjuntos de puntos reconocibles por la cámara al momento de desplegar el prototipo. Además, cuenta con un esquema interno en el que recopila información acerca de las imágenes y áreas, tal como un id de reconocimiento, estado, fecha de subida y su nivel para ser aumentable.

El segundo esquema utilizado en el prototipo, se centra en una base de datos del conocimiento. Esta tecnología se implementa para la creación de una capa conversacional con un agente virtual. Su planteamiento se basa en alojar un conjunto de llaves - quienes serian las posibles preguntas de los usuarios - y valores - quienes serian las posibles respuestas de los agentes -. En definitiva, se plantea un diccionario en donde se definen todas las posibles peticiones de los usuarios y las respuestas de los agentes.

El tercer enfoque implementado es el de una base de datos no relacional basada en esquemas. Esta base de datos alojaría los datos generados por los usuarios en secciones del Asistente Informativo Virtual como los comentarios y el menú de valoración. En este sentido, se plantearon dos esquemas con los cuales modelar la información allí manejada. Los comentarios (Ver **Figura 3.17a**) se definieron como un esquema no relacional y Valoración (Ver **Figura 3.17b**)

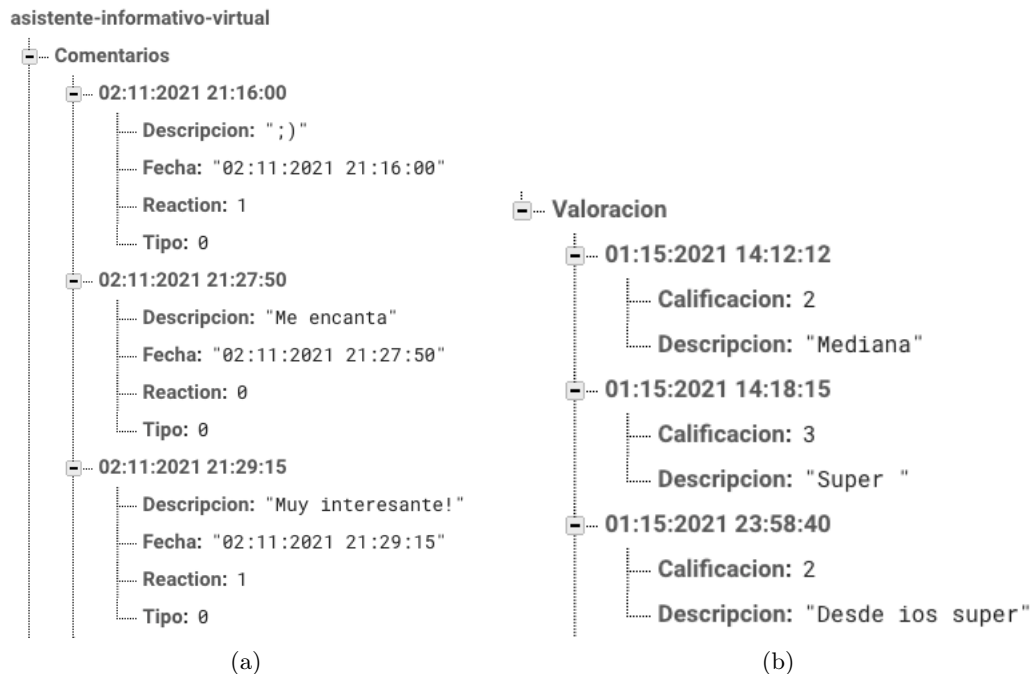


Figura 3.17: Modelos de datos no relacionales de la sección de comentarios(a) y valoración(b)

3.3.3. Prototipado

La interfaz gráfica es la primera impresión que se lleva un usuario al utilizar la aplicación, también, influye directamente en la usabilidad y por lo tanto en la experiencia obtenida. Para caracterizar de manera visual el prototipo del Asistente Informativo Virtual, se llevaron a cabo dos fases, la primera donde se diseña principalmente el flujo completo de la aplicación y cómo este sería articulado con lo expuesto en el [Levantamiento de requerimientos](#) y [Diagramas UML](#), además, de brindar un acercamiento inicial al diseño estético de la aplicación en sí; la segunda fase, se realizó posteriormente a la retroalimentación en conjunto con usuarios, por lo tanto se corrigieron las observaciones encontradas y se dieron detalles finales para iniciar con la implementación.

Como primera medida hay que tener en cuenta que el Asistente Informativo Virtual al ser una aplicación en realidad aumentada, tuvo dos enfoques para el diseño de su interfaz, los cuales son la interfaz de realidad aumentada donde se busca un aspecto mucho más minimalista y la interfaz de la aplicación como tal que brinda información relevante al usuario, partiendo de esto se inicio un proceso de diseño con ayuda de la herramienta Figma. Figma es una herramienta de diseño que permite crear Interfaces y prototipos en general a partir de vectores, por lo tanto es modificable en poco tiempo, lo cual le brindó rapidez al diseño.

El objetivo principal con el diseño de la interfaz del usuario en este proyecto se direccionó en brindar un aspecto que fuera intuitivo, fácil de usar y también brindara una sensación agradable en términos estéticos para los usuarios finales de la aplicación. Para esto se tuvieron en cuenta las heurísticas de usabilidad propuestas por Jakob Nielsen [[Jaka](#)] y las recomendaciones brindadas en las evaluaciones a los usuarios.

En el proceso de reflejar las especificaciones del sistema en la interfaz, se plantearon las vistas a partir de las secciones definidas en el Storyboard durante el [Levantamiento de requerimientos](#). Es así como se definieron cinco interfaces principales, en la [Figura 3.18a](#) se puede observar el menú inicial que permite al usuario ingresar al módulo de ayuda y a los lugares dentro del Barrio San Antonio; en la [Figura 3.18b](#) se observa la interfaz donde se listan las preguntas frecuentes y la valoración de experiencia de los usuarios adecuado principalmente para quejas y reclamos, En la [Figura 3.18c](#) se encuentra el menú que permite ingresar a los lugares junto con un elemento deslizable donde muestra información relevante y fotos correspondientes a los lugares; finalmente en las [Figuras 3.18d y 3.18e](#) se encuentran las interfaces asociadas a los tipo de experiencia recorrido e informativa respectivamente.

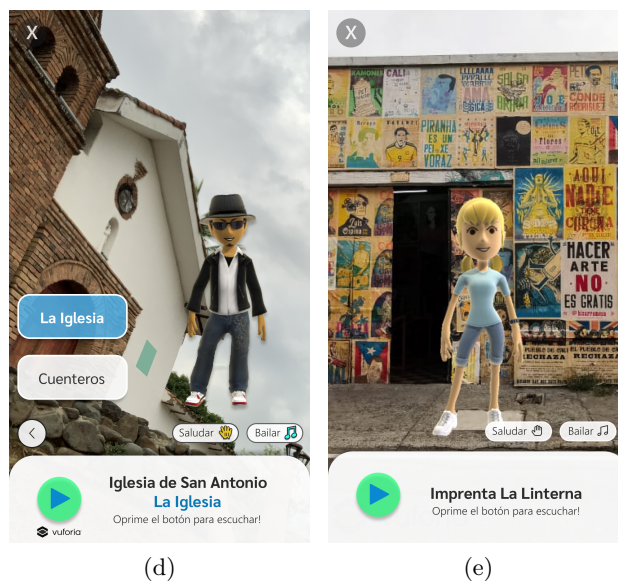
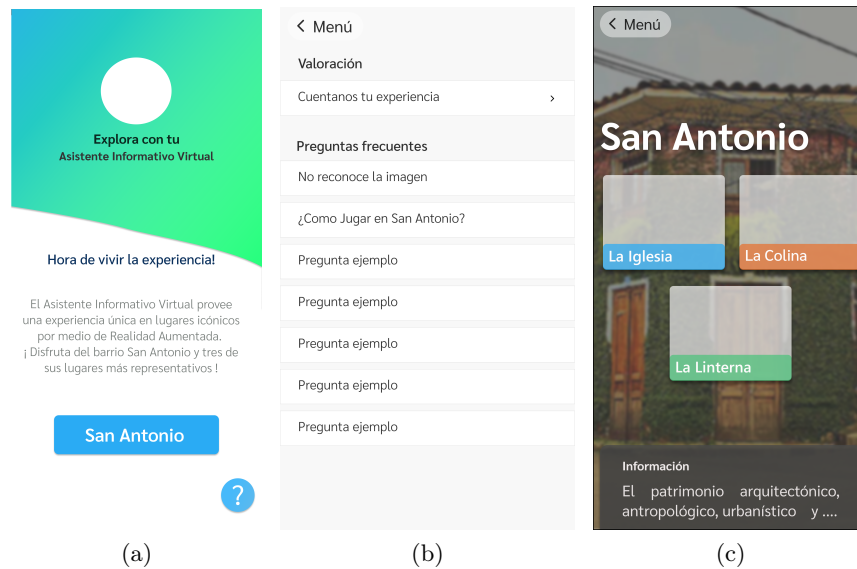


Figura 3.18: Interfaces de las secciones definidas en el levantamiento de requerimientos (Ver **Sección 3.2.4**)

El diseño se abarcó en reuniones con el equipo donde se discutieron los elementos de la interfaz hasta tener un flujo completo, aquí se plantearon las heurísticas de Nielsen y cómo estarían representadas dentro de la aplicación. A continuación se muestran algunas de las heurísticas utilizadas:

Visibilidad del estado del sistema: El diseño debe mantener informados a los usuarios sobre lo que está sucediendo, por lo tanto es necesario reflejar los estados (errores, cambios, aceptación) de la aplicación en la interfaz gráfica, en la **Figura 3.19** se pueden ver algunos de los mensajes de estado diseñados para las peticiones que se realizan. Por ejemplo en **3.19a** y **3.19b** se muestran mensajes de éxito al ser enviado correctamente el Comentario o Valoración, esto le permite al usuario reconocer que efectivamente fue enviado. En **3.19c** aparece un mensaje que se produjo cuando el usuario intento enviar un comentario sin reacción, por lo que se le indica que la coloque para poder continuar

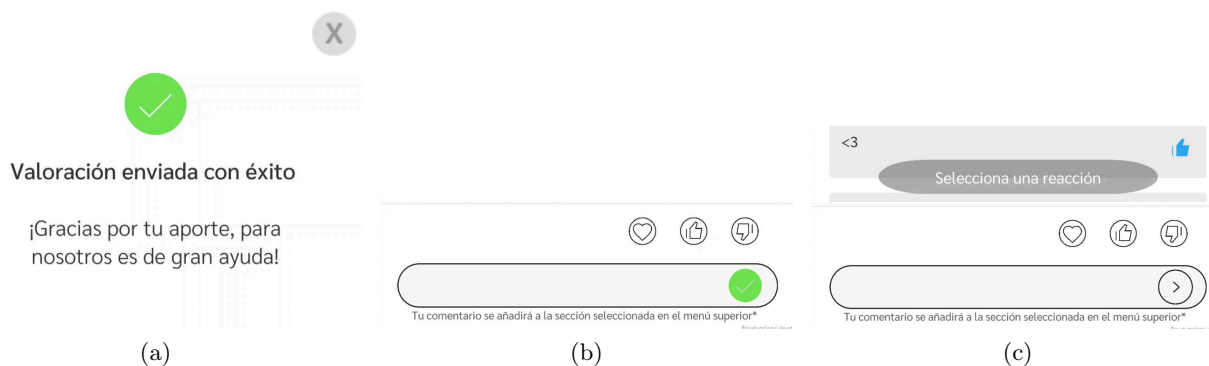


Figura 3.19: Capturas de Pantalla de algunos estados en INFORA

Coherencia y estándares: Se debe mantener la coherencia en cuanto acciones y elementos que se usen comúnmente en la industria, de esta manera los usuarios reconocen el funcionamiento de dicha acción u elemento de manera natural. Por ejemplo en la **Figura 3.21** se visualiza la guía para acceder a las funcionalidades que utilizan realidad aumentada, generalmente instructivos similares son utilizados para este tipo de aplicación, esto lo pudimos comprobar analizando las librerías nativas de ARKit y ARCore (Ver **Sección 4.2.2**), estas ya vienen con elementos de interfaz para AR implementados, en la **Figura 3.20** se puede observar el "ARCoachingView" de ARKit y el "ModelViewer" de ARCore, las librerías proveen estas clases para que sean utilizadas abiertamente por herramientas como Vuforia (Ver **Sección 4.2.2**) y suelen utilizarlas en sus aplicaciones de AR propias como Measurement de iOS. Si bien su principal función es explicar al usuario como ingresar, se vuelven una característica repetitiva en la mayoría de sistemas afines.

La Ley de Jakob establece que las personas pasan la mayor parte de su tiempo utilizando productos digitales distintos a los suyos. Las experiencias de los usuarios con esos otros productos establecen sus expectativas. No mantener la coherencia puede aumentar la carga cognitiva de los usuarios al obligarlos a aprender algo nuevo..

([Jaka])

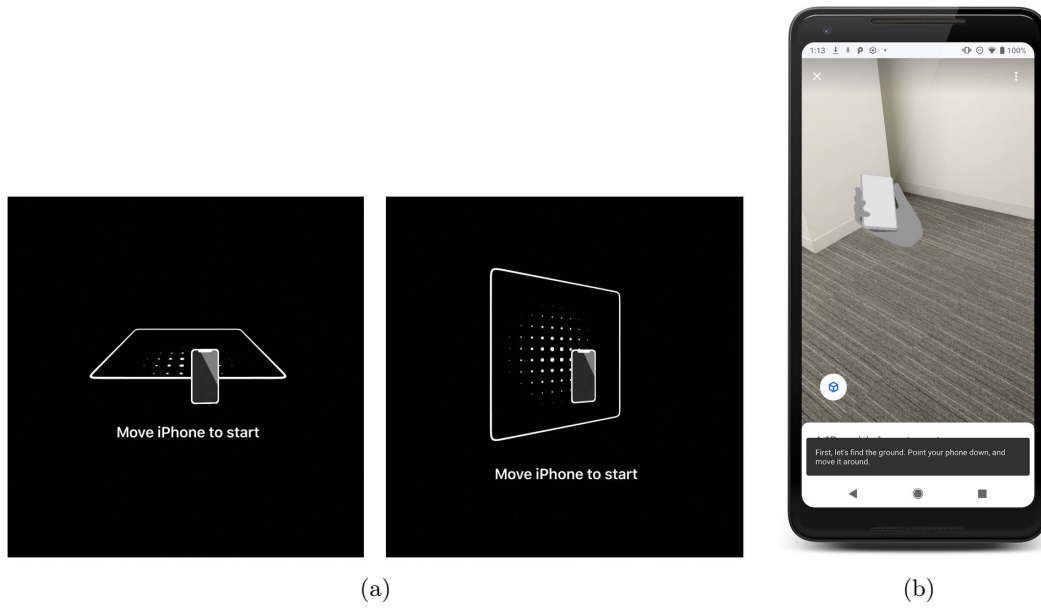


Figura 3.20: Elementos UI para AR Tomado de: [ARK](a) y [ARC](b).

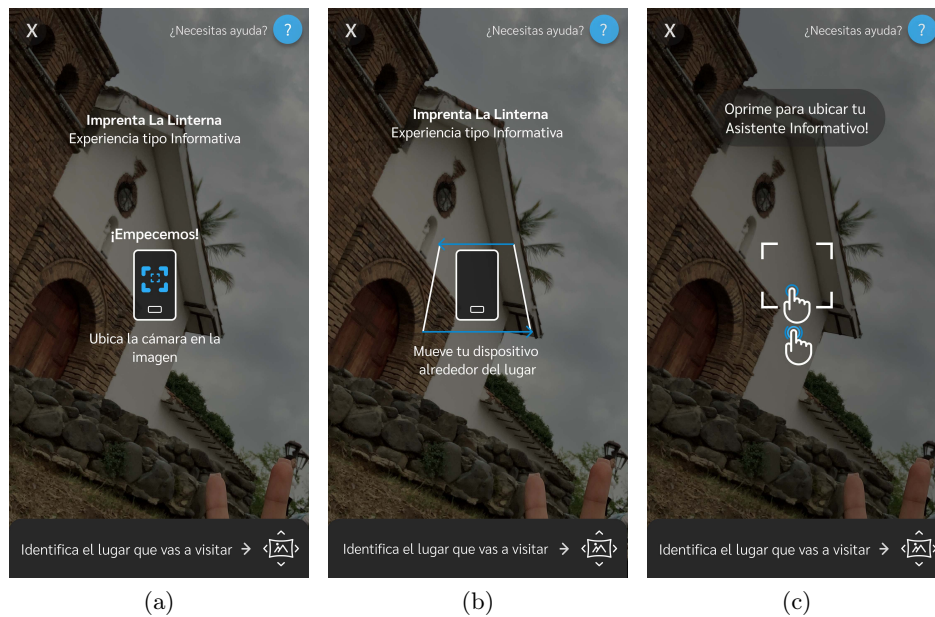


Figura 3.21: Interfaz propia del Asistente Informativo virtual para la guía del menú de realidad aumentada.

Diseño estético y minimalista: Las interfaces no deben tener elementos innecesarios en pantalla, si bien se agregan elementos estéticos es importante que estos no disminuyan la visibilidad a lo principal. En el caso de la interfaz de realidad aumentada se mantuvo estrictamente lo necesario, dándole protagonismo al Asistente y las acciones que se pueden realizar sobre él, como se puede observar en la **Figura 3.18d y 3.18e**. Es así como se logra que el usuario se centre sobre el Asistente Informativo y sus acciones de Saludar, Bailar y Hablar por medio del micrófono.

Ayuda y documentación: Si bien la idea es crear un diseño que no necesite ayuda, si es necesario tener la documentación adecuada en caso que el usuario presente dificultades para realizar una tarea. Por esta razón se implementó un módulo de ayuda **3.18b** que tuviera las preguntas generales que un usuario pudiera necesitar a la hora de utilizar la aplicación, además, se ubico en varios escenarios para que se encontrara este módulo fácil y rápidamente, se puede acceder desde **3.18a** y desde todas las interfaces de guía **3.21a, 3.21b, 3.21c**.

Una vez el equipo llegó a una versión estable del diseño del prototipo como se mencionó anteriormente en Figma, se pasó a una segunda fase donde se realizó una evaluación de la mano de usuarios para validar la viabilidad del mismo. Esta evaluación se encuentra descrita en la sección **5.2.4** más ampliamente. Como resultado en esta prueba se pudo evidenciar que algunos elementos no resultaban agradables para los usuarios, por ejemplo, el menú de lugares dentro del Barrio San Antonio (ver **3.18c**).



Figura 3.22: Modificaciones a interfaz del Menú de lugares en el Barrio San Antonio.

En la **Figura 3.22** se puede evidenciar el cambio realizado, donde se pasa de una interfaz que brinda un aspecto recargado a otra mucho más simple y minimalista. También se añadió la funcionalidad de comentarios de manera general y asociada a cada uno de los lugares después de la evaluación por lo que se añadió una interfaz asociada (Ver **Figura 3.23a**). La gran ventaja de realizar este tipo de pruebas en el ciclo 0 es que se pueden encontrar soluciones que no causen retrocesos en ciclos futuros.

Después de todo este proceso se llegó a un resultado que cumpliera con los objetivos planteados para la interfaz y contara con la aceptación de usuarios finales, en la **Figura 3.23** se encuentran plasmadas las interfaces que cambiaron durante este proceso, sumándose a las ya expuestas en las **Figuras 3.18a**, **3.18b** y **3.22**, para llevar a cabo dichas modificaciones se siguieron nuevamente las heurísticas de Nielsen [Jaka] a través de nuevas reuniones con el equipo de trabajo. En las **Figuras 3.23b** y **3.23c** se hace la diferenciación entre tipo Recorrido y tipo Informativo puesto que la función de reposicionar solo se encuentra dentro de la experiencia Informativa. Vale aclarar que la aplicación no cuenta con inicio de sesión puesto que la idea es brindar un ingreso rápido a las funcionalidades, por lo tanto no se cuenta con dicha interfaz de login.

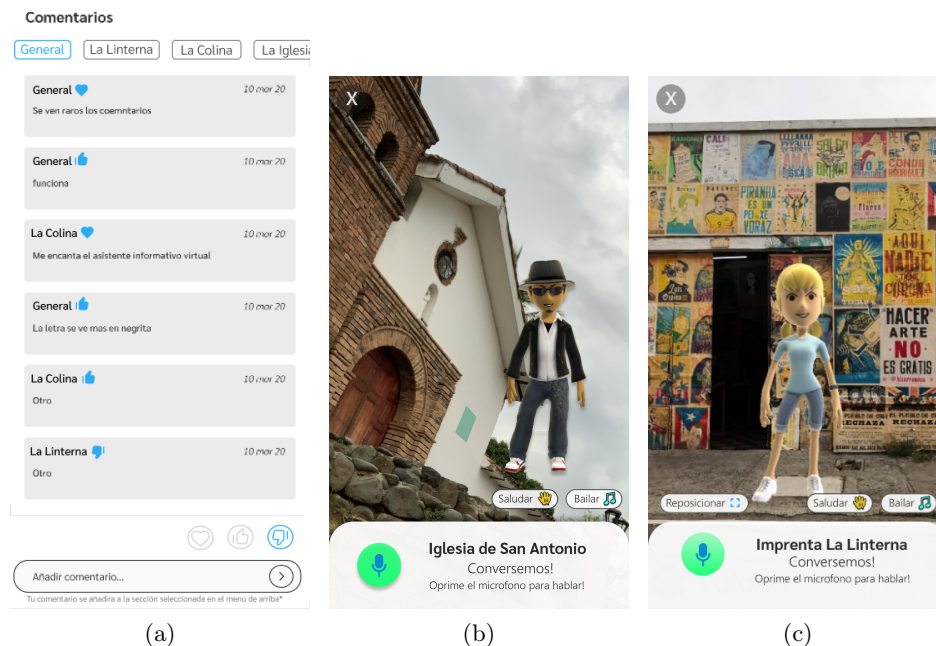


Figura 3.23: Modificaciones finales a interfaces.

Al finalizar las dos fases de diseño de prototipo, se procedió a implementar la interfaz anteriormente mostrada, por medio de la herramienta escogida Unity, que se detallará más adelante en el capítulo de Implementación. Unity cuenta con la posibilidad de añadir paquetes y librerías adicionales, entre ellos el **Unity Device Simulator** (Ver **Figura 3.24**), el cual permitió probar la

interfaz sin necesidad de tener la aplicación instalada en un dispositivo móvil, y además gracias al modelamiento de objetos de interfaz de usuario de Unity se provee la capacidad de brindar adaptabilidad en cualquier característica de tamaño del dispositivo final en el que el usuario instale la aplicación.

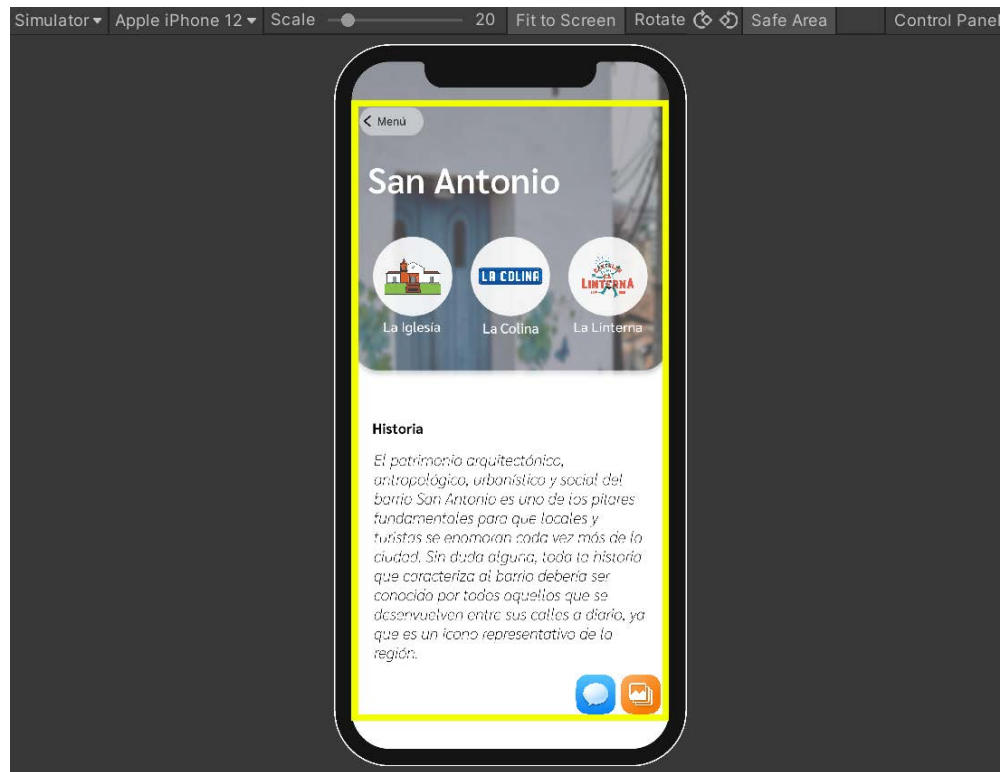


Figura 3.24: Captura de pantalla probando el sistema desde el editor de Unity, verificando el área segura para el desarrollo de interfaces en diferentes dispositivos tanto Android como iOS.

Implementación y Despliegue

Esta sección se centra en detallar la razón de elección y el uso del conjunto de soluciones implementadas para el desarrollo y despliegue del proyecto. Es importante tener en cuenta que el proyecto hizo uso de diferentes herramientas que le permitieron concebirse como un proyecto tecnológico integral basado principalmente en el uso de realidad aumentada y procesamiento del lenguaje natural.

Por ello, con el fin de contextualizar cada uno de los componentes tecnológicos implementados, a continuación se encuentra un listado previo de cada una de las áreas tecnológicas tratadas en este capítulo:

- Realidad Aumentada.
 - Reconocimiento del ambiente.
 - Reconocimiento de imágenes.
 - Reconocimiento de planos.
- Modelado 3D.
- Animación Corporal Modelos 3D.
- Animación Facial Modelos 3D.
- Locales (Multilenguaje).
- Speech To Text.
- Text To Speech.
- Conversación por medio del entendimiento del lenguaje natural.
- Bases de datos no relacionales.
- Máquinas de estado.
- Servicios REST.

4.1. Dispositivos y herramientas de desarrollo y despliegue

Con el fin de desarrollar y desplegar el prototipo en los dos principales sistemas operativos móviles Android y iOS, los equipos usados en ambos procesos fueron los siguientes:

- Equipos de desarrollo:
 - Equipo 1
 - **Referencia:** Asus Vivobook.
 - **Procesador:** Core i5 4210U (2.4GHz).
 - **Memoria:** 16 GB.
 - **Procesador gráfico:** NVIDIA GForce 840M (2GB).
 - Equipo 2
 - **Referencia:** MacBook.
 - **Procesador:** Core i5 4210U (2.4GHz).
 - **Memoria:** 16 GB.
 - **Procesador gráfico:** Intel(R) HD Graphics (1GB).
- Equipos de despliegue:
 - Equipo Móvil 1
 - **Referencia:** Huawei Mate 20.
 - **Procesador:** HiSilicon Kirin 980.
 - **Memoria:** 6 GB.
 - Equipo Móvil 2
 - **Referencia:** iPhone 12.
 - **Procesador:** Apple A14 Bionic.
 - **Memoria:** 6 GB.

Además de permitir el despliegue en ambos sistemas operativos, la elección de los equipos de despliegue se basó en las capacidades en hardware para llevar a cabo tareas de reconocimiento del ambiente, como la creación de una nube de puntos, tal como lo hace iOS (Ver **Sección 2.1.2.6**). Así mismo, se contó con la tecnología en software de librerías de desarrollo de realidad aumentada nativas.

Con el fin de implementar el prototipo funcional, en cada uno de los equipos de desarrollo fue instalado el motor de desarrollo de videojuegos Unity con el cual se realizó la implementación del proyecto, ya que provee compatibilidad con un conjunto amplio de librerías y, además, permite la exportación multiplataforma del sistema haciendo uso de un único código fuente desarrollado en lenguaje de programación C#, en el caso particular de este proyecto. Así mismo, en el dispositivo de desarrollo Apple se hizo uso de la plataforma de desarrollo Xcode con el fin de desplegar el proyecto en dispositivos iOS.

4.2. Implementación

Esta sección reúne la caracterización de cada una de las diferentes herramientas usadas y mencionadas al inicio del presente capítulo (Ver **Capítulo 4**).

4.2.1. Motor de desarrollo

Con el fin de llevar a cabo la implementación del proyecto se decidió analizar la elección del motor de desarrollo que integraría las herramientas que se mencionan en este capítulo (Ver **Capítulo 4**). En este caso, se tuvo como base los dos motores de desarrollo más llamativos y comercializados en la industria.

- **Unity:** Unity 3D es un motor de videojuegos multiplataforma 2D y 3D, que permite exportar proyectos en dispositivos móviles, computadores, dispositivos de realidad virtual y aumentada y consolas de manera efectiva. En él se desarrolla en lenguajes de programación tales como C# y JavaScript.
- **UnReal:** UnReal es un motor de videojuegos que se caracteriza por un alto grado de portabilidad y es una herramienta con grandes capacidades gráficas por lo cual se usa comúnmente en desarrollo de videojuegos de alto nivel. Además, permite exportación multiplataforma.

La elección del motor **Unity** finalmente se basó en la conectividad con cada una de las tecnologías usadas en el proyecto y detalladas en el presente capítulo, además, del conocimiento y manejo previo de la herramienta y el lenguaje de programación elegido. Igualmente, es válido mencionar que se principalmente se eligió **Unity**, dado que soporta el marco de trabajo elegido para el desarrollo de realidad aumentada de Vuforia (Ver **Sección 4.2.2**), mientras que UnReal no lo soporta.

4.2.2. Vuforia Augmented Reality Software Development Kit (SDK)

El SDK de Vuforia es un kit de desarrollo de aplicaciones con tecnología de realidad aumentada para dispositivos móviles como celulares y gafas de realidad aumentada con enfoques de desarrollo nativo y de plataforma cruzada (único código, diferentes sistemas operativos). Además, hace uso de tecnología de visión por computador, lo que permite el reconocimiento de imágenes, objetos 3D y actualmente, áreas en tiempo real.

Este kit provee diferentes características técnicas y tecnológicas en realidad aumentada. Según la **Figura 4.1**, Vuforia Engine está constituido lógicamente por un sistema basado en capas en donde se diferencian los objetos y los ambientes reconocibles en la primera capa. Así mismo, para lograr reconocer a través de la cámara estos objetos y ambientes se usa una tecnología llamada Vuforia Fusion. Esta hace parte de la segunda capa del modelo. En ella está construido un conjunto de algoritmos que utilizan tecnologías nativas e híbridas para el despliegue de la realidad aumentada según las capacidades del dispositivo durante el tiempo de ejecución.

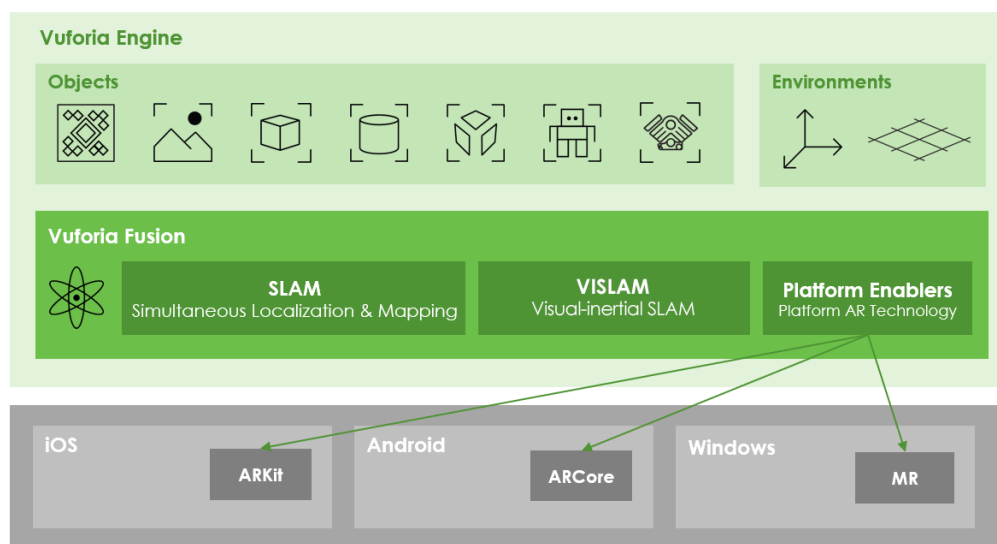


Figura 4.1: Modelo de comunicación por capas de Vuforia Engine.

Por ejemplo, si la aplicación se ejecuta en un Google Pixel 2 XL aprovechará ARCore. Mientras si se ejecuta la aplicación en un dispositivo que no soporta ARCore, este usará Vuforia VISLAM (Ver **Sección 2.1.2.7**) si y solo si el dispositivo cuenta con los sensores requeridos y ha sido calibrado por Vuforia.

El mismo escenario se aplica a los dispositivos iOS. Si el prototipo se despliega en un iPhone X, aprovechará automáticamente ARKit para todas las características de reconocimiento de objetos y del ambiente. Si este mismo prototipo se ejecuta en un dispositivo iOS más antiguo (que no soporta ARKit pero ha sido calibrado por Vuforia) utilizará automáticamente Vuforia VISLAM (Ver **Sección 2.1.2.7**) para las tareas dependientes [PTC21b]. Esto quiere decir que todas las tecnologías de realidad aumentada usadas en esta investigación son accesibles a través de un amplio conjunto de dispositivos con sistemas operativos como Android y iOS [PTC21b]. En definitiva, esta fue la principal razón por la cual se seleccionó esta tecnología en su versión 9.8.11.

Igualmente, es importante tener en cuenta que las tecnologías nativas usadas en los dos sistemas operativos en los que se desarrolló y desplegó el sistema son las siguientes:

- **ARCore:** Plataforma creada por Google con el fin de permitir a dispositivos con sistema operativo Android comunicarse con el ambiente concediendo la capacidad de desarrollar experiencias de realidad aumentada.

- **ARKit:** Kit desarrollado por Apple para proveer servicios de realidad aumentada a todos los usuarios de sus dispositivos con sistemas operativos móviles como iOS y iPadOS.

4.2.3. Avatares del sistema

Dado que el proyecto se centraba en comunicar de manera asertiva y receptible información asociada al barrio San Antonio en Cali y tres de sus lugares más representativos, se realizó un proceso de selección del tipo de modelado 3D a usar para la plataforma. Por ello, principalmente basándose en la idea de una comunicación natural del mensaje, en el proyecto se eligieron avatares de tipo humanoide con casi las mismas capacidades motrices humanas. Esto enmarca las capacidades para realizar movimientos faciales y corporales.

Dicho esto, con el fin de parametrizar la creación de los avatares, en el proyecto se implementa el modelado 3D con el uso de una interfaz de programación de aplicaciones (API) desarrollada en la plataforma Unity y accesible a través de su tienda web [Ale]. Esta permitió la creación de los humanoides especificando cada una de sus características físicas corporales con intervalos numéricos que cambian la cantidad de vértices de cada una de las mallas que componen el conjunto de las partes del cuerpo del humanoide.

Aparte de esto, las prendas que usan los avatares hacen parte de un conjunto de polígonos ya modelados que pueden ser elegidos al momento de su creación. Incluso, la interfaz provee la capacidad de crear los avatares aleatoriamente en tiempo de ejecución. En el siguiente fragmento de código desarrollado en lenguaje de programación C# se instancian cada una de las partes del cuerpo y rostro de los avatares basados en la información entregada al API:

En resumen, la creación de cada avatar se centró en instanciar cada elemento de los avatares a partir de números escalares definidos (Ver código fuente en el **Anexo 8.4**).

4.2.3.1. Animación corporal - Skeletal Animation

La técnica elegida y usada para la animación corporal de los avatares se conoce como Skeletal Animation (Ver **Sección 2.1.2.5**). Principalmente fue conveniente su elección dado que los modelos son diseñados previamente con un conjunto de aristas jerárquicas que representan los huesos del humanoide, además, permite la rotación de cada una de las extremidades como si fuese un esqueleto humano con sus respectivas articulaciones definidas.

En este orden de ideas, es válido tener en cuenta el concepto de pose en T, pose en cruz o en inglés, T-Pose. Este define el estado natural (sin ningún tipo de animación) de los modelos basados en técnicas como Skeletal Animation.

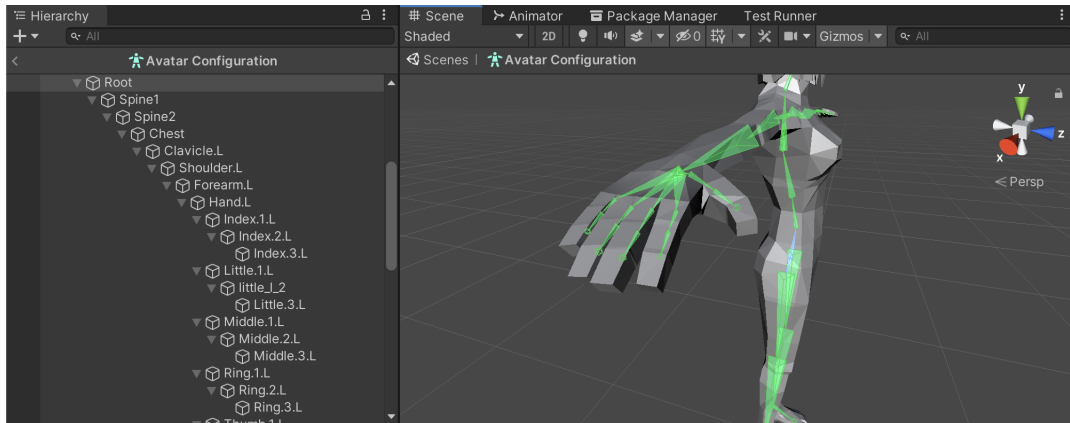


Figura 4.2: Disposición jerárquica de los huesos de la mano derecha del avatar.

Una vez teniendo definida la disposición jerárquica de la **Figura 4.2**, el siguiente paso consistía en la creación de cada uno de los humanoides que comunicarían en las tres zonas elegidas en la investigación. Por ello, se eligieron modelar tres diferentes tipos de avatar. Con el API usado, se parametrizó la creación de cada característica de los avatares. Entre ellas, el género, los tonos de piel y cabello, accesorios como gafas, prendas de vestir como camisetas, pantalones y zapatos. Estas características se configuraron a través de la interfaz gráfica que se evidencia en la **Figura 4.3**.

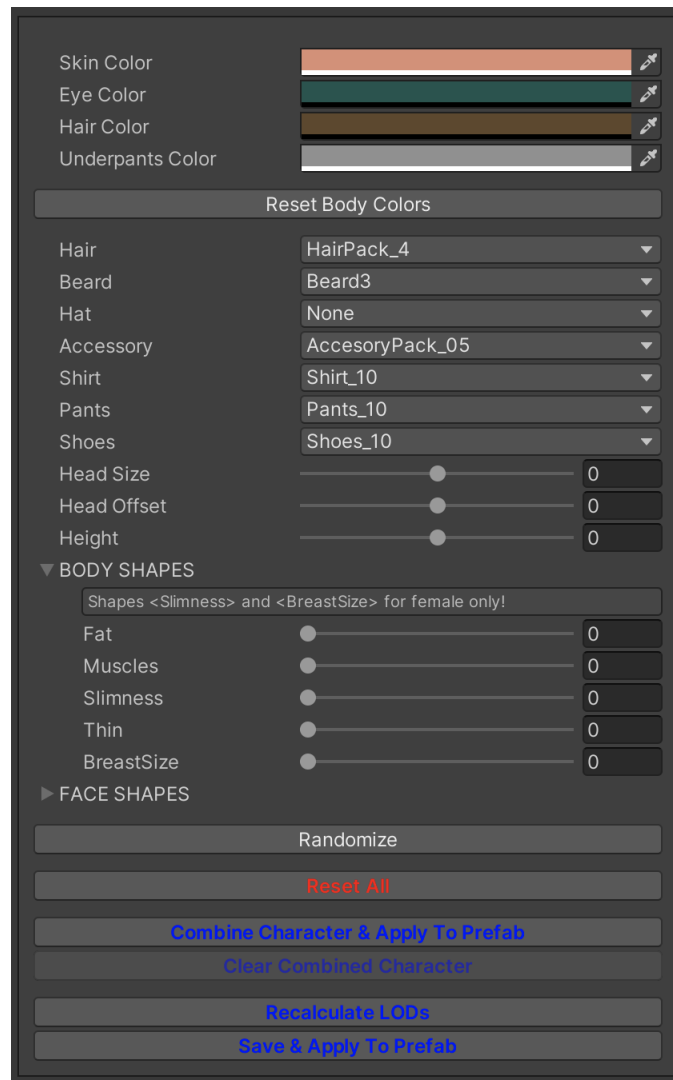


Figura 4.3: API de caracterización.

Igualmente, haciendo énfasis en el contexto al que iba dirigido el sistema, fue oportuno tener presente la diversidad de sus usuarios finales. Por ende, los avatares se construyeron bajo este ideal, consiguiendo modelarlos con la siguientes características:

- **Christopher - Iglesia de San Antonio:** Hombre de tez blanca, cabello rubio y gafas negras. Camiseta azul, pantalón largo y zapatos azules.
- **Alonso - Imprenta La Linterna:** Hombre de tez negra, cabello oscuro y afro. Chaqueta

verde, pantalón corto y sandalias grises.

- **Aria - Tertuliadero La Colina:** Mujer de tez blanca, cabello castaño y gafas moradas. Saco gris, pantalón pescador y botas blancas.

A continuación, en las **Figuras 4.4 a la 4.6** se presentan cada uno de los modelos creados en su pose por defecto, pose en T:



Figura 4.4: Christopher: Avatar Iglesia San Antonio.

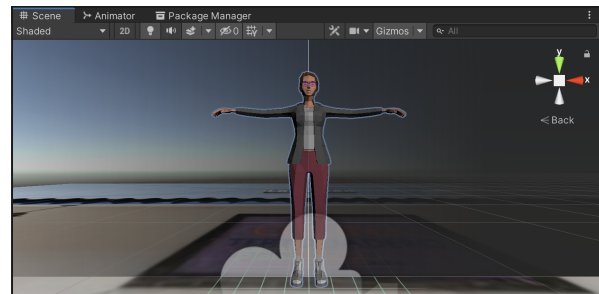
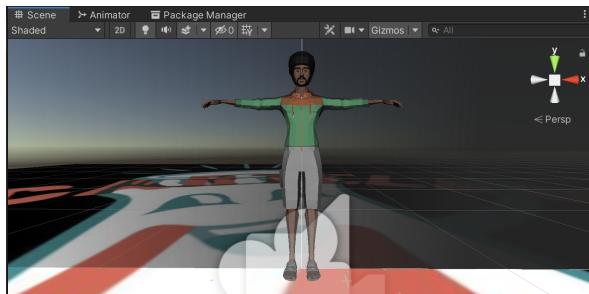


Figura 4.5: Alonso: Avatar Imprenta La Linterna. Figura 4.6: Aria: Avatar Tertuliadero La Colina.

4.2.3.2. Motion Capture Databases

Con el fin de implementar avatares corporalmente modelados como humanos y a su vez animados como tal, se hizo uso de la base de datos de animaciones de la universidad Carnegie Mellon [CMU] y el recurso de animaciones de Adobe, Mixamo [Inc]. Estas bases de datos fueron implementadas por medio de la tecnología **Motion Capture**. Además, tienen más de 3000 diferentes animaciones con usos directamente relacionados con los movimientos humanos. Por consiguiente, fue necesario incorporar las animaciones con los modelos ya existentes (Ver **Figuras 4.4 a la 4.6**) . Para ello, se implementó un mapeo corporal entre los avatares existentes y las animaciones importadas desde

las bases de datos. En la **Figura 4.7** se puede evidenciar la conformación esquelética del avatar con cada uno de los puntos de movimiento clave. En definitiva lo que se obtuvo fue la capacidad de reproducir cualquiera de las animaciones en el cuerpo de cada uno de los avatares.

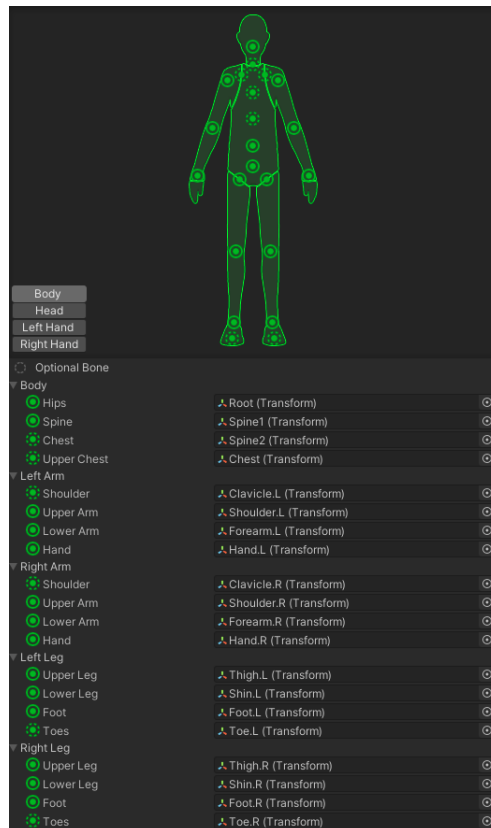
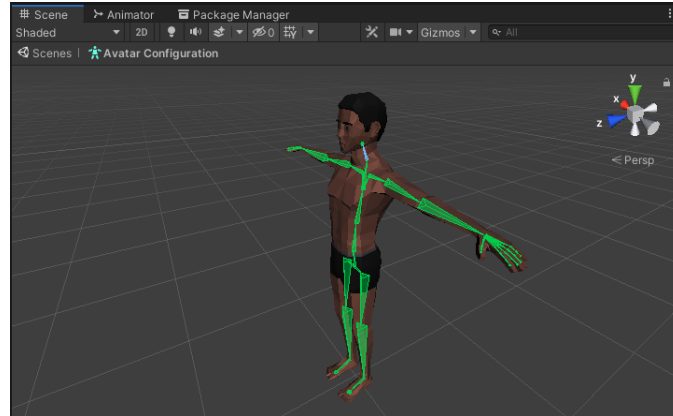


Figura 4.7: Mapeo corporal avatar 3D.

De igual modo, reconociendo la importancia de animar a los avatares según las extremidades involucradas en cada movimiento, el proyecto requirió la implementación de una técnica llamada **enmascaramiento**. Este método descarta algunos de los datos de animación dentro de un clip, lo que permite en definitiva animar zonas específicas del humanoide, en lugar de todo su cuerpo.

A partir de la **Figura 4.8** se puede evidenciar la posibilidad que se tiene para anular zonas inhabilitadas para la reproducción de animaciones. Aquellas partes del cuerpo que aparezcan en rojo no se animarán, mientras las que aparezcan en verde si lo harán. Por eso, se tienen dos principales mascararas, parte superior e inferior de cada humanoide, con el fin de animar el avatar con incluso dos animaciones, cada una llevándose a cabo en las dos mascararas. Por ejemplo, los avatares fueron animados con la capacidad de bailar salsa. Esto requirió que moviera los brazos y la cabeza, pero al mismo tiempo, debió mover las piernas.

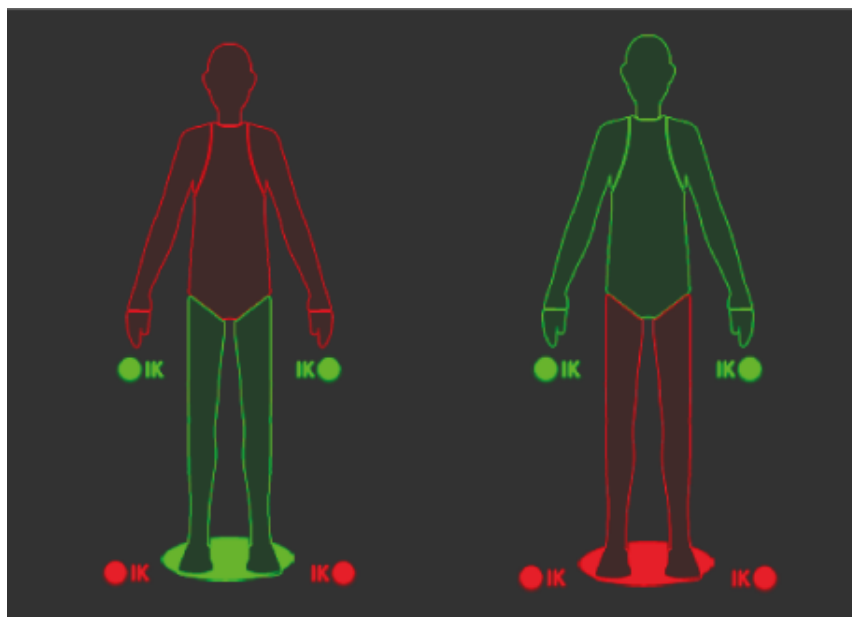


Figura 4.8: Mascara tren inferior (izquierda) y mascara tren superior (derecha). Cada una de ellas permite animar el avatar únicamente en la zona verde de la misma.

Por ende, debido al uso de la técnica de enmascaramiento se implementó un componente de animación en el que se definieron dos capas asociadas a cada una de las mascararas. Dentro de cada una de estas capas se desarrolló una respectiva máquina de estados con la cual controlar cada uno de los movimientos corporales de los avatares a partir de la ejecución de transiciones circulares (pasar de un estado inactivo a un estado de saludo y regresar al estado inactivo automáticamente).

En la **Figuras 4.9 y 4.10** se evidencian las dos máquinas de estado desarrolladas. Estas tienen como estado inicial un estado de inactividad, esto quiere decir que una vez el avatar sea posicionado,

corporalmente se vería inactivo. Sin embargo, la implementación de un conjunto finito de parámetros de activación, permitió la creación de un algoritmo encargado de la elección aleatoria de un parámetro cada 3 segundos. Con esto, los avatares llevaban a cabo una transición de estado cada 3 segundos. Estas transiciones lo hacían verse saludando y moviendo la cabeza y las manos.

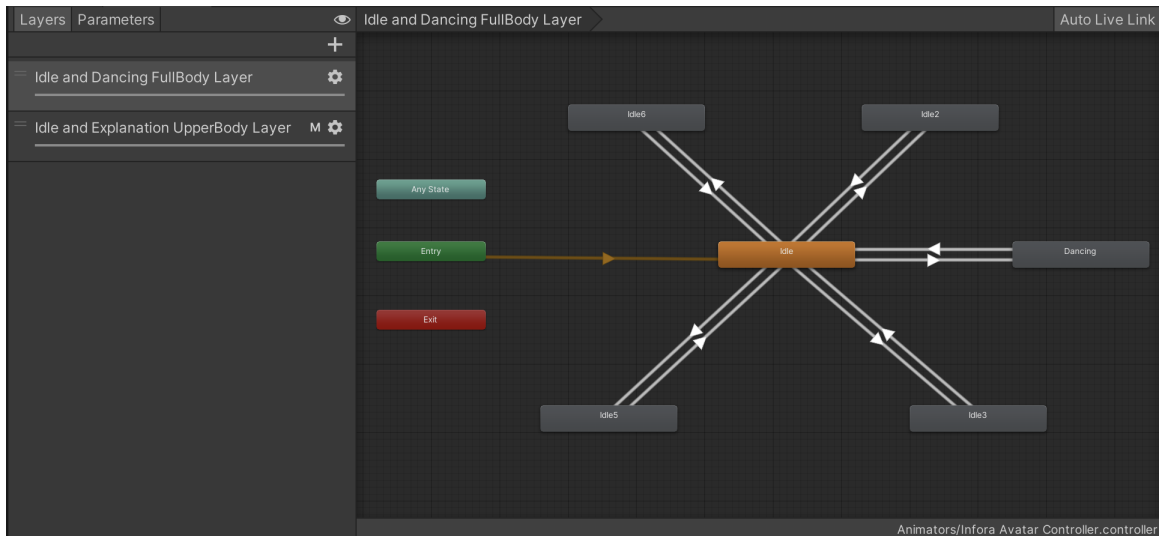


Figura 4.9: Máquina de estados de la capa del cuerpo completo.

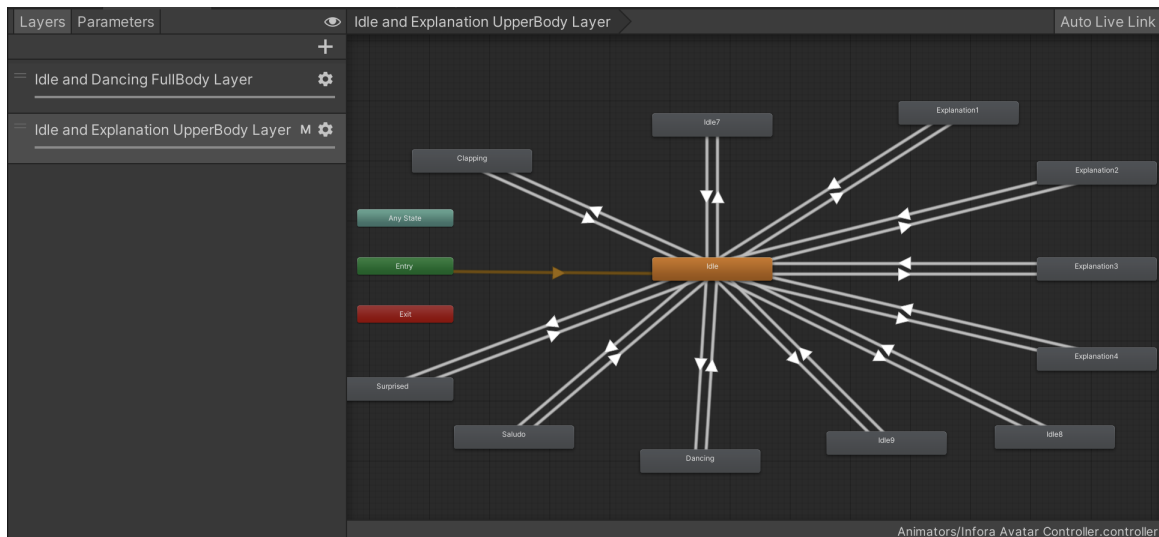


Figura 4.10: Máquina de estados de la capa de la parte superior del cuerpo.

A la izquierda de cada una de estas figuras se evidencian las dos capas definidas y a la derecha la respectiva máquina de estados implementada en cada capa.

4.2.3.3. Animación Facial - Blend Shapes

Las deformaciones faciales basadas en Blend Shapes (Ver **Sección 2.1.2.5**) al ser implementadas junto con técnicas como **Animación corporal - Skeletal Animation**, le permitieron al equipo animar paralelamente mallas faciales y corporales. En este caso al tratarse de avatares con características humanas, fue necesario la unión de tecnologías que desacoplaran las animaciones faciales de la corporales para generar una comunicación asertiva. Por ejemplo, una vez los avatares fueran aumentados, el equipo decidió que debían saludar a los usuarios. Por ende, los humanoides debían hablar y al mismo tiempo, ser animados con un saludo en donde agitara el brazo, implementado la técnica de enmascaramiento que ejemplifica la **Figura 4.8**. En este orden de ideas, a cada uno de los avatares se le implementó un conjunto de 6 diferentes deformaciones faciales que se evidencian en las **Figuras 4.12 a la 4.17**. Esto se llevo a cabo gracias que desde la plataforma de desarrollo Unity se tiene la posibilidad de definir un componente llamado **Skinned Mesh Renderer** que contiene un conjunto de Blend Shapes objetivo creados para una especifica malla.

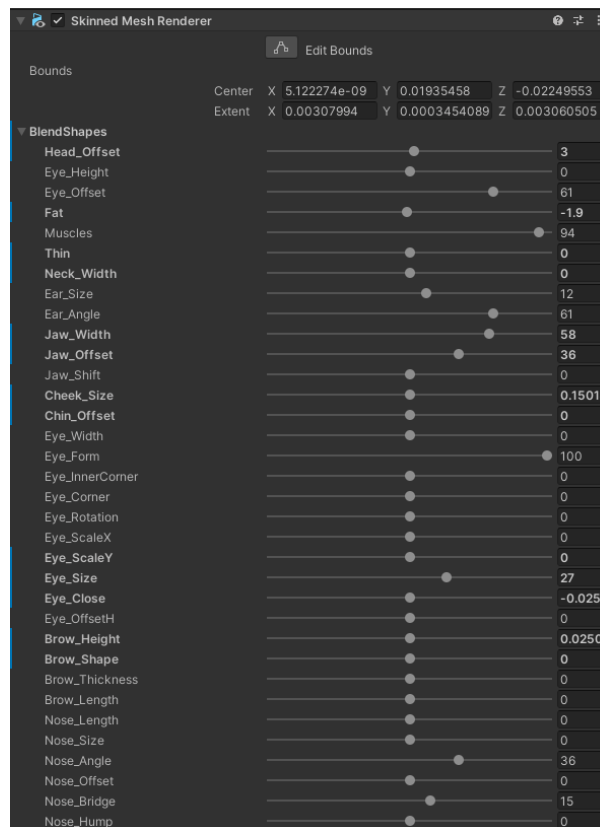


Figura 4.11: Componente Skinned Mesh Renderer - Deformaciones faciales y corporales totalmente parametrizables (Ver **Sección 2.1.2.5**).

A partir de lo explicado en la **Sección 2.1.2.5** los gestos faciales objetivo permiten la creación de nuevas y diferentes animaciones faciales, por consiguiente, el haber usado el componente de la **Figura 4.11**, permitió el desarrollo de un conjunto de animaciones faciales para diferentes tipos de sonidos del lenguaje, los que se denominan fonemas. Así mismo, es importante tener en cuenta que la representación facial de los fonemas se llaman visemas.

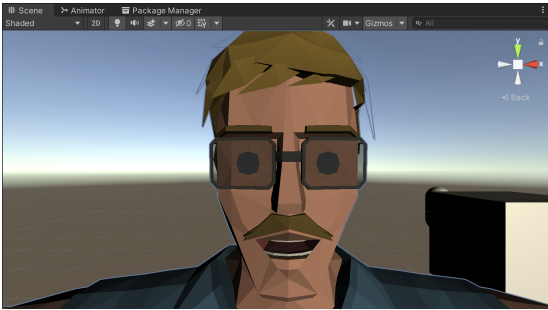


Figura 4.12: Visema letra l.

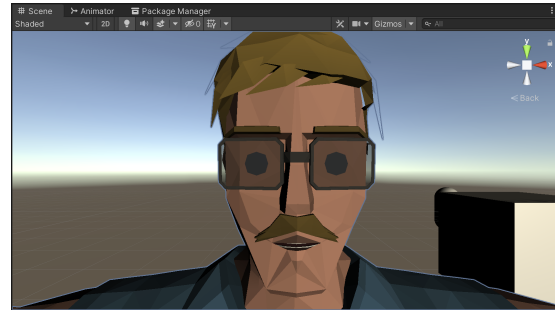


Figura 4.13: Visema letra u.



Figura 4.14: Visema letra f.



Figura 4.15: Visema letras a-e.



Figura 4.16: Visema letras o.

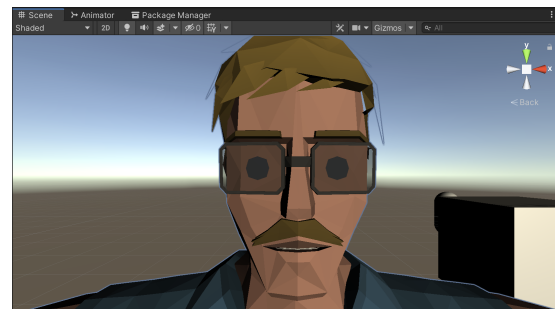


Figura 4.17: Visema letra t.

Conjunto de animaciones faciales o visemas definidos para cada uno de los avatares 3D. El avatar que se tomó como ejemplo para la matriz de figuras es Christopher de la Iglesia de San Antonio (Ver **Figura 4.4**).

4.2.3.4. Sincronización de movimientos faciales

La sincronización de los visemas anteriormente mencionados (Ver **Sección 4.2.3.3**) fue un proceso basado en etapas de reflexión de la investigación (Ver **Metodología 3.1**). Dado que inicialmente se hizo uso de un movimiento facial estático, el equipo reflexionó ante lo desarrollado y planteó alcanzar una experiencia con una mayor capacidad comunicativa e inmersiva, el proceso se centró en definir cada uno de los visemas como un conjunto de vértices que representarían movimientos faciales del avatar y cada movimiento finalmente sería usado como parte de la articulación fonética de los humanoides. La combinación lógica de los visemas se implementó por medio de la librería SALSA Lip Sync [CMS21].

Su elección fue realizada gracias a que provee una aproximación de sincronización de labios, independiente del idioma y para todo tipo de personajes modelados en 2D y 3D. Además, es importante resaltar que se eligió esta suite de desarrollo dado que los archivos de diálogo (formato audio) son procesados en tiempo real, lo que le concidió al proyecto reducir o eliminar el retraso al momento de animar los avatares con cualquier clip de audio. Su implementación se basó en el uso de un procesador de colas (propio de la librería SALSA) que asegura una mejor experiencia de sincronización de sonido y movimiento, disminuyendo la latencia y la pérdida de animaciones.

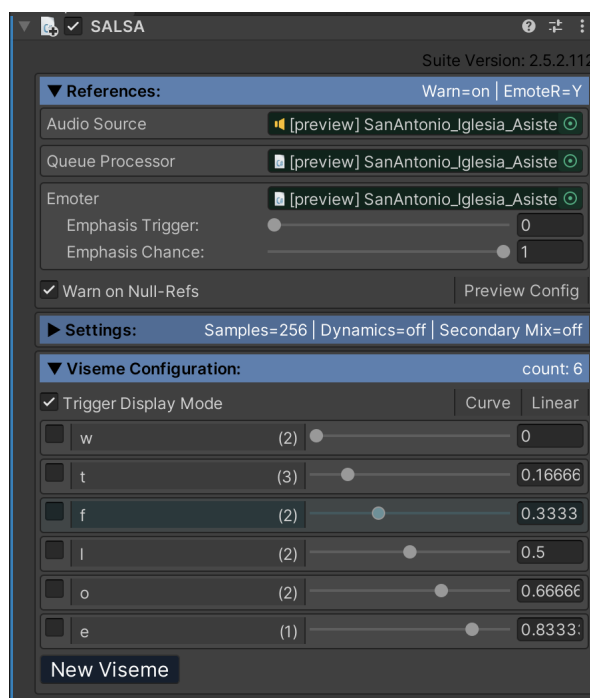


Figura 4.18: Configuración parámetro de activación de cada uno de los visemas de las **Figuras 4.12 a la 4.17**.

Además, esta librería en su versión 2.5.2 provee un funcionamiento basado en la configuración de visemas y emociones definidos por medio de una condición que los acciona y los habilita para ser usados en la animación que se está creando en tiempo de ejecución. Esta condición está dada por la elección de un número decimal dentro de un intervalo de 0 a 1 con hasta 7 cifras significativas tal como se puede ver en la **Figura 4.18**. Este número define la amplitud de la onda de sonido digital en un instante de tiempo en específico, lo que se denomina muestra. Este método fue implementado gracias a que la librería hace uso de la técnica frecuencia de muestreo de ondas sonoras con una capacidad de hasta 44.000 muestras por segundo, denotado como 44KHz, lo que en definitiva le brindó al sistema una aproximación en tiempo de real de mediana fidelidad.

4.2.4. Reconocimiento del Ambiente

En la actualidad, la tecnología de la realidad aumentada está directamente relacionada con la comunicación entre software y hardware. En este sentido, como se mencionó en la **Sección 4.2.2**, Vuforia Fusion permite acceder a tecnologías de visualización basadas en la estimación de posición. A esta se le reconoce por implementar métodos de odometría que permiten el reconocimiento total del ambiente y del usuario (Ver **Sección 2.1.2.7**).

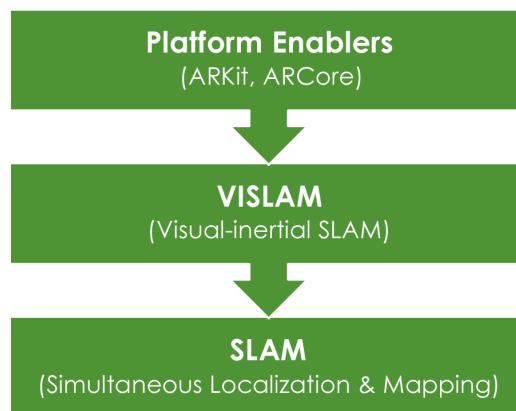


Figura 4.19: Planteamiento capa a capa de algoritmos posibles de Vuforia Fusion.

Lo que se muestra en la **Figura 4.19** es que una experiencia de realidad aumentada creada con Vuforia Engine inicialmente intentará utilizar la tecnología más alta (librerías nativas - ARCore y ARKit -) e irá yendo hacia abajo dependiendo de lo que esté disponible en el dispositivo móvil durante el tiempo de ejecución de la aplicación.

Bajo este ideal, en la etapa de implementación del proyecto se afirmó la decisión de optar por Vuforia Fusion dado que provee la capacidad de conectarse directamente con estas tecnologías y

usarlas a través del marco de trabajo de Vuforia. Además, porque tiene la capacidad de controlar la falta de las librerías en los dispositivos más antiguos a través de planteamiento capa a capa de algoritmos posibles que se evidencia en la **Figura 4.19**.

4.2.4.1. Reconocimiento de áreas

Posterior a la elección del enfoque tecnológico de reconocimiento del ambiente, indagando las capacidades en hardware y software de las tecnologías actuales, en el proyecto se optó por la utilización de hardware infrarrojo, exactamente, **Sensores LiDAR** (Light Detection and Ranging, Ver **Sección 2.1.2.6**).

Esta elección se centró principalmente en temas de usabilidad del sistema, tal cual como el enfoque de cámara dual implementada por uno de los trabajos referenciados en el estado del arte (Ver **Sección 2.1.3**). Además, estos sensores hacen parte del conjunto de medios más usados para el entendimiento de ambientes en la actualidad, dado que son emisores de rayos láser infrarrojo que se reflejan en los objetos con el fin de reconocerlos por medio de la creación de una nube de puntos que los representa, llamada comúnmente en inglés Point Cloud (Ver **Sección 2.1.2.6**).

De esta forma, se construyó una respectiva nube de puntos para una de las zonas elegidas como base de la investigación. Para lograrlo, las posibilidades radicaban en el uso de tecnología de escaneo con láser como Matterport, NavVis o Leica. Sin embargo, esto se implementó a través de un sistema de imágenes tridimensionales (Ver **Sección 2.1.2.6**) propio del marco de trabajo de Vuforia, llamado **Vuforia Area Target Creator**.



Figura 4.20: Mapeo sensor LiDAR con aplicación Vuforia Area Target Creator - Iglesia San Antonio.



Figura 4.21: Mapeo sensor LiDAR con aplicación Vuforia Area Target Creator - Domo San Antonio.

Este sistema de reconocimiento de Vuforia está desarrollado para dispositivos móviles con sensores LiDAR incorporado y ARKit como librería nativa, por ende los modelos fueron escaneados por un dispositivo móvil iPhone 12. La construcción de puntos se realizó por medio de visitas presenciales al barrio San Antonio en Cali, Colombia. Para escanearlos, el equipo a cargo del proyecto realizó la tarea de hacer un barrido minucioso por cada uno de los lugares objetivo. Esto se evidencia en el momento del escaneo a través de las **Figuras 4.20 y 4.21**.

Una vez realizado el proceso de escaneo de los lugares seleccionados se exportaron los modelos desde el dispositivo móvil iOS desde el cual fueron construidos para luego ser importados en el proyecto. Después de cargados, su representación eran mallas u objetos modelados en 3D pero con la especial característica de ser reconocidos y aumentados por cualquier otro dispositivo Android y/o iOS por medio del enfoque explicado en la **Sección 4.2.4**. Finalmente, los resultados de las mallas escaneadas y texturizadas se evidencian en las **Figuras 4.22 y 4.23**.

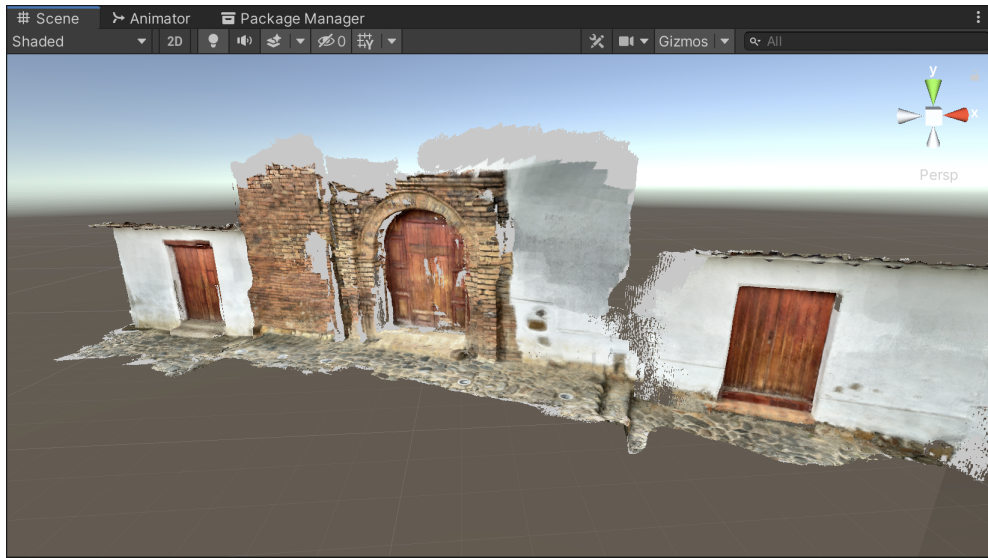


Figura 4.22: Modelo 3D Área Iglesia San Antonio.

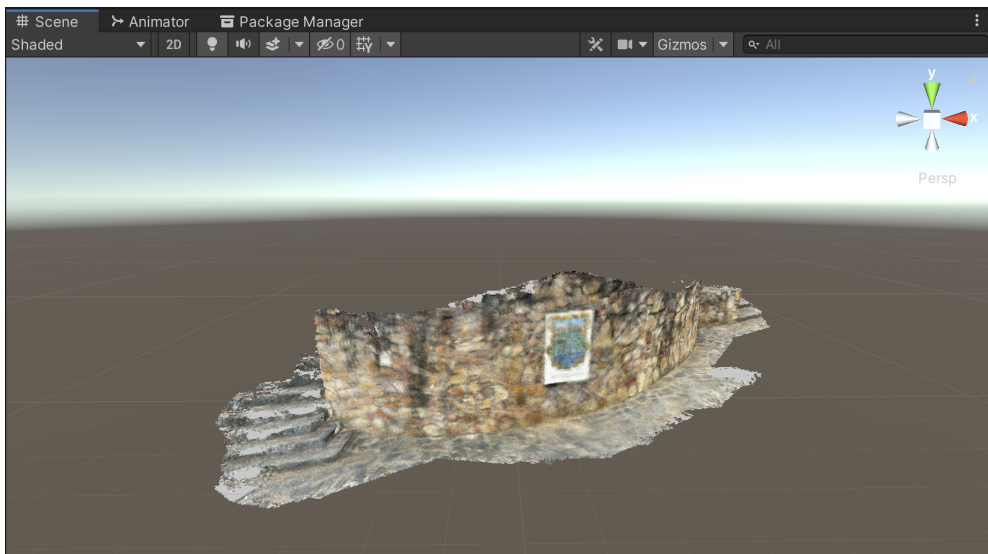


Figura 4.23: Modelo 3D Área Domo San Antonio.

4.2.4.2. Reconocimiento de imágenes

Con el fin de proveer una mejor experiencia de usuario a la hora de acceder a las funcionalidades del menú de realidad aumentada, se optó por la captura de imágenes objetivo.



Figura 4.24: Imagen objetivo Tertuliadero La Colina.



Figura 4.25: Imagen objetivo Imprenta La Linterna.

Estas imágenes objetivo (Ver **Figuras 4.24** y **4.25**) permitieron tener un paso más de reconocimiento en el sistema, puesto que su funcionalidad principal no era aumentar contenido por medio de su reconocimiento, por lo contrario, resaltaban la marca personales del sistema y de los lugares en donde se estuviera llevando a cabo la experiencia. Además, brindaron la capacidad de hacer un proceso previo de escaneo, lo cual facilitó el uso de técnicas más avanzadas como el reconocimiento de planos horizontales mencionada en la **Sección 4.2.4.3**.

4.2.4.3. Reconocimiento de planos

Debido a que el proyecto se basaba en tres puntos claves del barrio San Antonio, para dos de los lugares elegidos: La Linterna y La Colina, se implementó la tecnología de reconocimiento de planos horizontales puesto que los avatares necesitaban ser posicionados de pie sobre una superficie plana. Es pertinente tener presente que esta técnica hace uso igualmente de las librerías nativas ARCore y ARKit mencionadas en la **Sección 4.2.2** y por medio de las técnicas para el reconocimiento del ambiente de la **Sección 4.2.4**. La implementación de esta tecnología se basó en el uso de dos clases propias de Vuforia Engine:

- **SmartTerrain:** Encargada de encontrar, rastrear superficies y ejecutar pruebas de impacto de superficies en el ambiente reconocido por el hardware del dispositivo.

- **PositionalDeviceTracker:** Responsable de la orientación en el espacio de coordenadas del ambiente para el reconocimiento de la posición del dispositivo.

Además, el marco de trabajo le brindó al proyecto la posibilidad de hacer uso de dos conceptos de experiencia de usuario con las cuales acceder a los algoritmos para el reconocimiento de planos y el posicionamiento de objetos sobre ellos.

El primer concepto se basa en la ubicación de un indicador de superficie (Ver **Figura 4.26**), lo que define un estado del reconocimiento de un plano y le deja saber al usuario que el sistema lo ha reconocido y lo está reconociendo de manera satisfactoria y efectiva.

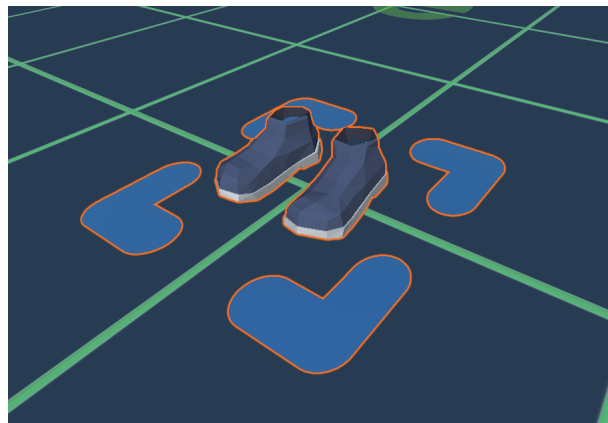


Figura 4.26: Indicador de superficie diseñado y usado en el prototipo.

El segundo concepto es la interacción del usuario con el indicador de superficie aumentado por el sistema. Esta interacción se define como el uso de un toque en la pantalla, lo que se puede llamar un toque de impacto.

Estas nociones le brindaron al proyecto tener control sobre el posicionamiento y direccionamiento de los avatares al ser puestos en el plano horizontal, haciendo uso de técnicas de rotación de objetos en tiempo real como las mencionadas en la **Sección 2.1.2.5**.

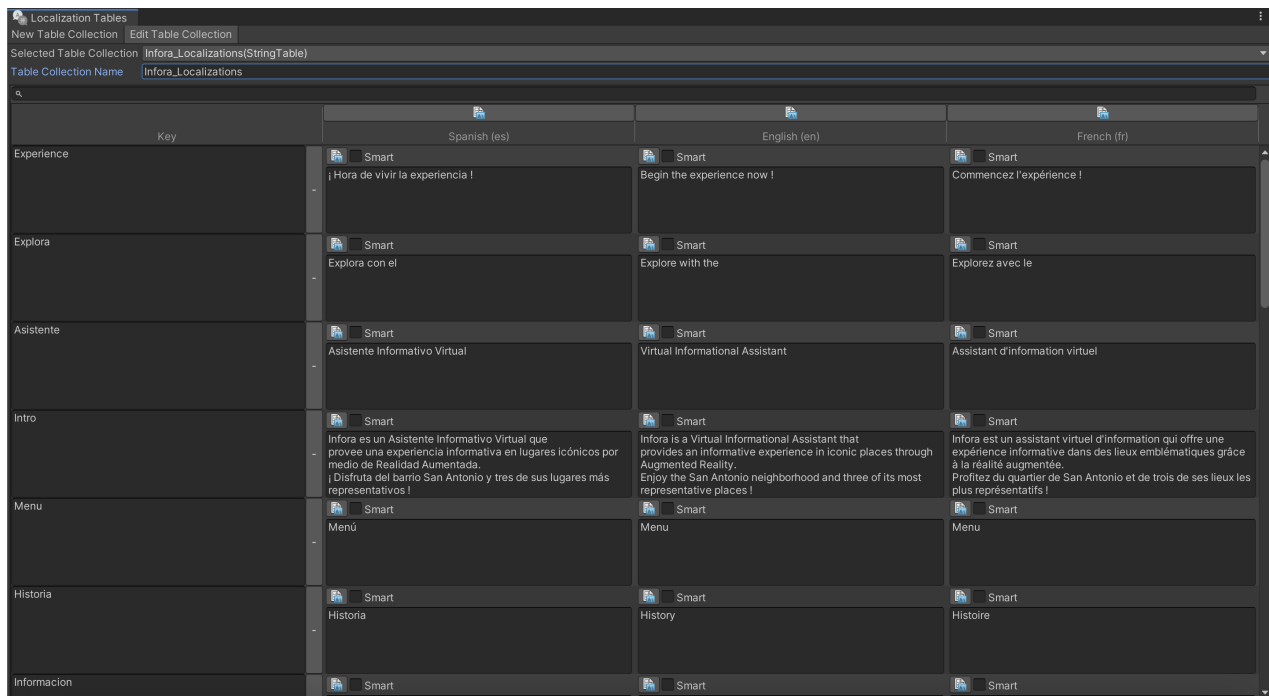
En definitiva, aquellas zonas elegidas con una metodología de tipo informativa tuvieron la capacidad de hacer uso del posicionamiento de los avatares de manera libre a través de las dos clases y los dos conceptos de experiencia de usuario recién mencionados. Con esto, los avatares humanoides se posicionarían libremente conforme a la decisión del usuario, logrando ser observados con mayor precisión, técnica de gran valor para la experiencia de los usuarios al hacer uso del prototipo.

4.2.5. Locales del sistema

El proceso metodológico elegido generó diferentes cambios en el alcance planteado para el proyecto, puesto que fases como compartir y reflexionar (Ver **Metodología 3.1**), permitieron llevar a cabo ciclos de mejora al sistema desarrollado.

Inicialmente, el proyecto tendría la capacidad de informar acerca de las tres localizaciones elegidas por medio del idioma local Español. Sin embargo, teniendo en cuenta el impacto de la metodología y el ideal del termino **Computación Cultural** documentado en la **Sección 2.1.2.4**, en el proyecto se implementó la utilización del paquete de localizaciones de la plataforma de desarrollo Unity llamado Localization.

Este paquete le proporcionó al proyecto herramientas para el soporte de múltiples idiomas y variantes regionales en la aplicación, por ejemplo, el soporte de interfaces gráficas en múltiples idiomas o características específicas de aspectos idiomáticos, como los clip de audio de los avatares.



Key	Spanish (es)	English (en)	French (fr)
Experience	Smart ¡ Hora de vivir la experiencia !	Smart Begin the experience now !	Smart Commencez l'expérience !
Explora	Smart Explora con el	Smart Explore with the	Smart Explorez avec le
Asistente	Smart Asistente Informativo Virtual	Smart Virtual Informational Assistant	Smart Assistant d'information virtuel
Intro	Smart Infora es un Asistente Informativo Virtual que provee una experiencia informativa en lugares icónicos por medio de Realidad Aumentada. ¡ Disfruta del barrio San Antonio y tres de sus lugares más representativos !	Smart Infora is a Virtual Informational Assistant that provides an informative experience in iconic places through Augmented Reality. Enjoy the San Antonio neighborhood and three of its most representative places !	Smart Infora est un assistant virtuel d'information qui offre une expérience informative dans des lieux emblématiques grâce à la réalité augmentée. Profitez du quartier de San Antonio et de trois de ses lieux les plus représentatifs !
Menu	Smart Menú	Smart Menu	Smart Menu
Historia	Smart Historia	Smart History	Smart Histoire
Informacion	Smart	Smart	Smart

Figura 4.27: Tabla de textos configuraciones regionales: Español, Inglés y Francés.

El proceso se enfocó en la creación de tres diferentes configuraciones regionales. Esta decisión fue realizada bajo el reconocimiento de la variedad idiomática de los usuarios finales del sistema y el objetivo general del proyecto documentado en la **Sección 1.2**. Estas configuraciones regionales se usaron para reconocerse por medio de una tabla de entradas de texto. Cada fila de la tabla contiene

una clave (el texto original) y una entrada de texto de traducción para cada configuración regional (Ver **Figura 4.27**).

Posterior a la configuración de la tabla, el sistema debía proveer la capacidad de cambiar el idioma en tiempo real, lo que implicó el desarrollo de un nuevo componente en la interfaz de usuario que permitiría completar esta función de gran valor para la investigación y los usuarios finales del sistema.



Figura 4.28: Componente interfaz de usuario de elección de idioma.

4.2.6. Interacción por voz

La metodología de innovación y exploración implementada le permitió al equipo tener un mejor conocimiento del medio al que iba dirigido el prototipo. Con esto, se lograron reconocer funcionalidades no propuestas inicialmente, pero que serían gran utilidad en el contexto de la investigación, por ejemplo la interacción por voz.

Por esto, dado el uso de avatares humanoides modelados en 3D, se plantearon nuevas capacidades del sistema a nivel interactivo. En este caso, se optó por la idea de entablar comunicación entre los usuarios finales y los avatares construidos para cada una de las locaciones de la investigación.

Este planteamiento conllevaría a la implementación de un método tecnológico con el cual habilitar una comunicación entre usuario y avatar. En este orden de ideas, el equipo decidió hacer uso del hardware del dispositivo, exactamente el micrófono. Para lograrlo, la implementación se basó en reconocer de manera particular las capacidades en hardware y software de los sistemas operativos móviles usados en el proyecto.

4.2.6.1. Técnica de habla a texto

Tras un proceso de investigación basado en el chequeo de las capacidades tecnológicas actuales, el equipo reconoció la existencia de clases nativas para convertir habla en texto en los dos sistemas operativos trabajados. Además, descubrió que pueden ser configuradas con cualquiera de los idiomas soportados por ambos sistemas operativos, lo que permitió la continuidad de una implementación basada en múltiples idiomas, incluso con la interacción vocal (Español, Inglés, Francés) de los usuarios.

Para lograr la implementación de clases nativas en un desarrollo de tipo cruzado, la plataforma de desarrollo Unity permite el uso de una capa en la cual se pueden definir componentes de tipo plugin para comunicaciones nativas con los sistemas operativos objetivo del proyecto. Con base en esto, se adaptó y tomó como base un proyecto de código abierto documentado en la plataforma GitHub [j1m18]. Su implementación se centra en la capacidades de los sistemas operativos Android y iOS en convertir habla en texto y texto en habla por medio de sus librerías nativas. Además, implementa un puente de comunicación directa con Unity.

Por ende, para la técnica de habla a texto se implementaron archivos de conexión nativa en lenguajes de programación como Java para Android y Objective-C para iOS, haciendo uso de las siguientes librerías:

- **SpeechRecognizer:** Clase que provee acceso al servicio de reconocimiento de la voz por medio de la conexión al micrófono en dispositivos con sistema operativo **Android**.

Código fuente usado en el proyecto para la conexión nativa en Java:

```
// MainActivity.java
import android.content.Intent;
import android.os.Bundle;
import android.speech.RecognitionListener;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
```

```

public class MainActivity extends UnityPlayerActivity
{
    // Speech To Text Method Android
    public void OnStartRecording()
    {
        intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_PREFERENCE,
            Bridge.languageSpeech);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            Bridge.languageSpeech);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Bridge.languageSpeech);
        intent.putExtra(RecognizerIntent.EXTRA_SPEECH_INPUT_COMPLETE_SILENCE_LENGTH_MILLIS,
            2000);
        intent.putExtra(RecognizerIntent.EXTRA_CALLING_PACKAGE,
            this.getPackageName());
        intent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS, 3);

        this.runOnUiThread(new Runnable() {

            @Override
            public void run() {
                speech.startListening(intent);
            }
        });
        UnityPlayer.UnitySendMessage("SpeechToText", "onMessage", "CallStart,
            Language:" + Bridge.languageSpeech);
    }

    // ...
}

```

- **SFSpeechRecognizer:** Clase que provee acceso al servicio de reconocimiento de la voz por medio de la conexión al micrófono en dispositivos con sistema operativo **iOS**.

Código fuente usado en el proyecto para la conexión nativa en Objective-C:

```

// SpeechRecorderViewController.m
#import "SpeechRecorderViewController.h"
#import <Speech/Speech.h>

// Speech To Text Method iOS
- (void)startRecording
{

```

```
if (!audioEngine.isRunning) {
    if (recognitionTask) {
        [recognitionTask cancel];
        recognitionTask = nil;
    }

    inputNode = audioEngine.inputNode;

    recognitionRequest = [[SFSpeechAudioBufferRecognitionRequest alloc] init];
    recognitionRequest.shouldReportPartialResults = NO;
    AVAudioFormat *format = [inputNode outputFormatForBus:0];

    [inputNode installTapOnBus:0 bufferSize:1024 format:format
     block:^(AVAudioPCMBuffer * _Nonnull buffer, AVAudioTime * _Nonnull when)
     {
        [recognitionRequest appendAudioPCMBuffer:buffer];
    }];
    [audioEngine prepare];
    NSError *error1;
    [audioEngine startAndReturnError:&error1];
    NSLog(@"errorAudioEngine.description: %@", error1.description);
}
}

// ...
```



Figura 4.29: Componente de interfaz de usuario del menú de realidad aumentada para el acceso al micrófono.

Con el fin de integrar en un solo código fuente (patrón propio de la plataforma de desarrollo utilizada) el uso de las dos clases usadas, se desarrolló un algoritmo de comunicación entre la interfaz gráfica de la aplicación y los servicios de cada sistema operativo.

Esto se llevó a cabo gracias a la existencia de un puente de comunicación directa entre la plataforma de desarrollo Unity y sistemas operativos móviles como Android y iOS. Además, el equipo se encargó del diseño de la interfaz de usuario con la cual hacer uso de la lógica construida (Ver **Figura 4.29**).

De esta manera, se implementó un script escrito en lenguaje de programación C# con el cual se accedió a los servicios nativos documentados anteriormente. Su desarrollo se basó en particularizar el llamado de los métodos para cada uno de los sistemas operativos móviles elegidos. Esto se llevó a cabo a partir del uso de directivas específicas para cada una de las plataformas objetivo del proyecto, en este caso Android y iOS. Estas directivas se usan como condiciones que permiten ejecutar porciones de código diferente cuando se compila el proyecto a cada uno de los sistemas

operativos:

- `#if UNITY_ANDROID` : `#define` directiva para la plataforma Android.
- `#if UNITY_IOS` : `#define` directiva para la plataforma iOS.

A continuación un fragmento del código fuente de la solución:

```
// SpeechToText.cs
/// <summary>
/// Speech To Text class implementation with Unity Bridge
/// </summary>
public class SpeechToText : MonoBehaviour
{
    // Speech To Text Method Unity
    public void StartRecording(string _message = "")
    {
#if UNITY_IOS
        _TAG_startRecording();
#elif UNITY_ANDROID
        AndroidJavaClass javaUnityClass = new
            AndroidJavaClass("com.starseed.speechootext.Bridge");
        javaUnityClass.CallStatic("OnStartRecording");
#endif
    }
    // ...
}
```

4.2.7. Procesamiento del lenguaje natural

Al continuar con la implementación de una comunicación entre los usuarios finales y los avatares, el equipo debió centrar la investigación en reconocer cómo llevar a cabo este tipo de enfoque de interacción o control por voz. Al tratarse de una conversación, las mejores opciones se dirigían en el uso de técnicas de inteligencia artificial con las cuales entablar de manera satisfactoria una conversación o comunicación.

Por ello, el equipo planteó reducir el problema de una comunicación por voces, en una comunicación por texto. Con esto, de manera interna (no evidente para el usuario final) el sistema controló toda la información como entradas de texto. Inicialmente haciendo uso de los métodos y clases mencionados en la **Sección 4.2.6**, se capturó la voz de los usuarios finales y se convirtió directamente en texto. Posteriormente, al reconocer lo dicho por los usuarios por medio del micrófono, el siguiente paso se enfocaba en estudiar el texto con el fin de generar una respuesta acorde y asertiva.

Bajo este ideal, con el fin de desarrollar un prototipo basado en una comunicación interactiva y

clara, el equipo decidió hacer uso de modelos para el reconocimiento del texto y el entendimiento de las peticiones o las posibles preguntas de los usuarios dentro del nicho de la investigación.

Estos modelos fueron creados por medio de Microsoft Azure, exactamente, a través de dos servicios cognitivos de computación en la nube, QnAMaker y Language Understanding (LUIS). Esta elección fue realizada gracias a las capacidades de clasificación de palabras clave y de reconocimiento de intenciones de cada una de las herramientas. Además, los modelos fueron entrenados desde el servidor y a través de una interfaz de usuario, lo que posibilitó su uso a través de servicios web que permitieron la generación de respuestas fundamentadas en cada una de las preguntas realizadas por los usuarios finales del aplicativo Infora.

4.2.7.1. Capa conversacional - QnAMaker

El servicio de procesamiento del lenguaje natural para la creación de robots conversacionales de Microsoft Azure, recibe como nombre QnAMaker. Este servicio le permitió al equipo implementar una base de datos del conocimiento para cada uno de los lenguajes en los que se desarrolló el proyecto. En concreto, se desarrollaron tres bases de datos del conocimiento: una en idioma español, otra en inglés y finalmente otra en francés.

Con esto, se definieron tres conjuntos de llaves y valores, que representarían las posibles preguntas y respuestas del asistente informativo virtual. Así mismo, es importante resaltar que su elección se basó en el uso de datos abiertos para entrenamiento de Microsoft Azure [Mic21] con los que se tuvo la capacidad de entrenar los modelos con preguntas contextuales (aquellas que incluían o nombraban alguno de los tres lugares elegidos como base de la investigación) y no contextuales (aquellas de cualquier índole). En otras palabras, se le brindó al sistema la capacidad de responder a preguntas en hasta 100 diferentes tipos de escenario, incluso por fuera del contexto propio de la investigación, en la **Figura 4.30** se pueden observar dos ejemplos.

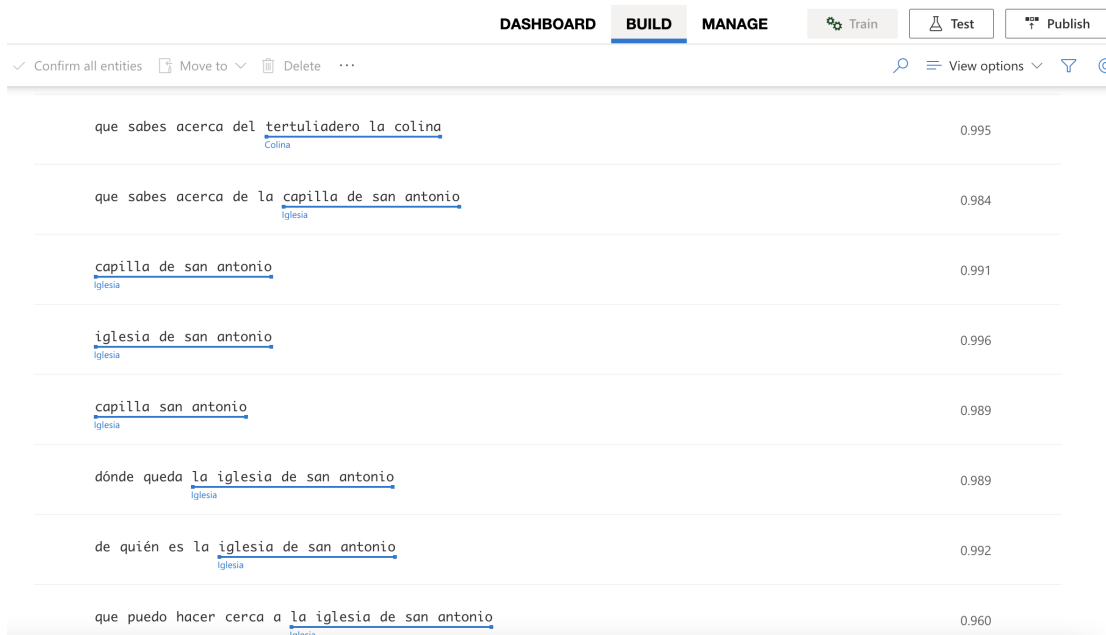


Figura 4.30: Ejemplo de llaves y valores para preguntas de tipo contextual (menciona a la Iglesia de San Antonio) y no contextual (se refiere a cualquier tema).

4.2.7.2. Entendimiento del lenguaje - LUIS

Microsoft Azure permite a través de su servicio LUIS reconocer las intenciones dentro de un determinado texto o petición. Para lograrlo, hace uso de un enfoque de etiquetado de palabras, las cuales denomina entidades de aprendizaje automático.

Estas entidades se definen como posibles fragmentos de texto que hacen parte de la petición general. Por esto, en el caso particular del proyecto se decidió hacer uso del servicio de entendimiento del lenguaje gracias a que proveía la capacidad de reconocer aquellas preguntas de tipo contextual, es decir, aquellas con contenido referente a los tres lugares elegidos para la investigación con el previo entrenamiento de las entidades de aprendizaje automático (Ver **Figura 4.31**).



Text	Entity	Score
que sabes acerca del <u>tertuliadero la colina</u>	Colina	0.995
que sabes acerca de la <u>capilla de san antonio</u>	Iglesia	0.984
<u>capilla de san antonio</u>	Iglesia	0.991
<u>iglesia de san antonio</u>	Iglesia	0.996
<u>capilla san antonio</u>	Iglesia	0.989
dónde queda <u>la iglesia de san antonio</u>	Iglesia	0.989
de quién es la <u>iglesia de san antonio</u>	Iglesia	0.992
que puedo hacer cerca a <u>la iglesia de san antonio</u>	Iglesia	0.960

Figura 4.31: Ejemplo de entidades de aprendizaje automático (marcas azules) definidas por el equipo.

Por medio de la técnica de etiquetado anteriormente mencionada se implementó un algoritmo de selección aleatoria de animaciones corporales de explicación (mover los brazos y la cabeza) únicamente cuando la petición del usuario se trataba de una pregunta de tipo contextual.

4.2.7.3. Técnica de texto a habla

Debido a que el prototipo se construyó con un enfoque multicultural, en temas como el lenguaje, el genero y la etnia de los avatares, también fue necesaria la incursión de diversidad en el habla de cada uno de ellos.

Para lograrlo, se utilizó el lenguaje de marcado de síntesis de voz (SSML), el cual es un lenguaje de marcado basado en XML que permitió especificar cómo se convierte el texto de entrada en voz sintetizada a partir del servicio cognitivo de texto a voz de Microsoft Azure. En comparación con el texto plano, el SSML permite ajustar el tono, la pronunciación, la velocidad del habla, el volumen y otros aspectos de la salida de texto a voz. Los signos de puntuación normales, como la pausa después de un punto o la entonación cuando una frase termina con un signo de interrogación, se manejaron de manera automáticamente.

En este orden de ideas, se definió un conjunto de nueve diferentes tipos de voces que estuvieron asociadas a cada avatar y los tres idiomas de su dominio. Además, se desarrolló un método con

el cual construir el archivo de marcado de síntesis de voz en tiempo de ejecución según el avatar y el lenguaje seleccionado en el sistema. El siguiente fragmento de código muestra las voces seleccionadas y la creación del XML del texto en SSML.

```
// VoiceController.cs
/// <summary>
/// Avatars voices (each language) and SSML runtime method creator
/// </summary>

// Christopher Voices - La Iglesia
ttsLanguages.Add(new List<string>
{ "es-ES-AlvaroNeural", "en-US-ChristopherNeural", "fr-FR-HenriNeural" });

// Alonso Voices - La Linterna
ttsLanguages.Add(new List<string>
{ "es-US-AlonsoNeural", "en-US-GuyNeural", "fr-CA-AntoineNeural" });

// Aria Voices - La Colina
ttsLanguages.Add(new List<string>
{ "es-MX-DaliaNeural", "en-US-AriaNeural", "fr-FR-DeniseNeural" });

// ...

public XDocument CreateXML(string value)
{
    var doc = new XDocument(new XElement("speak",
        new XAttribute("version", "1.0"),
        new XAttribute("xmlns", "http://www.w3.org/2001/10/synthesis"),
        new XAttribute(XNamespace.Xml + "lang", ttsLangSelected[0]),
        new XElement("voice", value,
            new XAttribute("name", ttsLangSelected[1]))));

    return doc;
}

/// Sample result
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis" xml:lang="es-MX">
    <voice name="es-MX-DaliaNeural">
        Texto proveniente de los servicios QnAMaker y LUIS.
    </voice>
</speak>
```

4.2.8. Implementación base de datos no relacional Firebase

La elección de una base de datos no relacional se centró en la capacidad de tener esquemas de fácil creación y acceso.

Además, el equipo se inclinó por el servicio de base de datos en tiempo de real de Firebase puesto que proveía el ambiente completo para su desarrollo, manejo y despliegue, lo que permitió mejorar los tiempos de desarrollo de las funcionales que se comunicaban con la base de datos no relacional, en este caso, los comentarios y las valoraciones de los usuarios finales.

Así mismo, esta tecnología fue usada por sus características de uso con baja latencia y porque le permitía al prototipo escalar fácilmente de manera horizontal, agregando nuevas funcionalidades y comunicándolas con nuevos esquemas con bajo esfuerzo.

4.2.9. Máquina de estados finita controlada por eventos

Haciendo énfasis en la arquitectura desarrollada para el Asistente Informativo Virtual, el equipo se encargó de la implementación de un controlador de eventos.

Este controlador de eventos funcionó prototipalmente como un sistema de transiciones entre peticiones. Por ello, el enfoque utilizado se basó en el uso de una máquina de estados finita controlada por eventos.

Esta máquina de estados le permitió al equipo controlar las diferentes acciones de los usuarios como eventos de comunicación entre el cliente y el servidor (Ver **Arquitectura 3.14**). En definitiva, por medio del lenguaje de programación C# se desarrolló un conjunto de directivas para el cambio de estados según los eventos que ocurriesen en la aplicación.

A continuación, una porción de código que evidencia la implementación y declaración del algoritmo basado en casos para cuando se selecciona un lugar o un lenguaje:

```
// StateMachine.cs
/// <summary>
/// Event-Driven Finite State Machine
/// </summary>

void CheckScene(string scene)
{
    switch (scene)
    {
        case "ARIglesiaScene":
            messageSelected = saludosIniciales[0];
            sceneVoices = ttsLanguages[0];
            break;
    }
}
```

```
        case "ARLinternaScene":
            messageSelected = saludosIniciales[1];
            sceneVoices = ttsLanguages[1];
            break;
        case "ARColinaScene":
            messageSelected = saludosIniciales[2];
            sceneVoices = ttsLanguages[2];
            break;
    }
}

void CheckLanguage(int language)
{
    switch (language)
    {
        case 0:
            QnA_Uri = QnA_UriSP;
            QnA_EndKey = QnA_EndKeySP;
            ttsLangSelected.Add(codesLanguage[0]);
            ttsLangSelected.Add(sceneVoices[0]);
            break;
        case 1:
            QnA_Uri = QnA_UriEN;
            QnA_EndKey = QnA_EndKeyEN;
            ttsLangSelected.Add(codesLanguage[1]);
            ttsLangSelected.Add(sceneVoices[1]);
            break;
        case 2:
            QnA_Uri = QnA_UriFR;
            QnA_EndKey = QnA_EndKeyFR;
            ttsLangSelected.Add(codesLanguage[2]);
            ttsLangSelected.Add(sceneVoices[2]);
            break;
    }
}
```

4.2.10. Comunicación y servicios REST

Los tres servicios usados para el procesamiento del lenguaje natural (Ver **Sección 4.2.7**) fueron virtualizados y accedidos por medio de la web haciendo uso de protocolo HTTP a través de peticiones de transferencia de estado representacional o REST, por sus siglas en inglés.

Para lograrlo se implementó la virtualización de cada uno de los servicios por medio de App Service, un servicio de alojamiento propio de Microsoft Azure y de alta compatibilidad con las prestaciones técnicas usados en el proyecto. La elección de esta tecnología se centró en su capacidad para el

alojamiento y ejecución de manera dedicada y segura de cada uno de los servicios con la implementación de tokens de acceso para su utilización.

La comunicación se basó en el envío y recibo de objetos tipo JSON que fueron serializados y deserializados haciendo uso de la librería Newtonsoft JSON [New21]. A continuación se evidencia uno de las respuestas de los servicios, en este caso LUIS con la definición de la intención y la entidad encontrada en el cuerpo de la petición:

```
{
  "query": "Cuentame acerca de la iglesia de san antonio",
  "prediction": {
    "topIntent": "Explanation",
    "intents": {
      "Explanation": {
        "score": 0.9967927
      }
    },
    "entities": {
      "Iglesia": [
        "iglesia de san antonio"
      ]
    }
  }
}
```

4.2.11. Manejo de versiones

En todo proyecto es de vital importancia tener un control sobre el manejo de versiones. De esta manera se da paso a trabajo en equipo de manera organizada, haciendo cambios concurrentemente y revisando versiones anteriores de ser necesario.

En el caso de la plataforma para desarrollo de vídeo juegos Unity, existe un recurso llamado **Collaborate** que permite guardar, compartir y sincronizar proyectos [Teca], de manera muy sencilla dentro de la aplicación. A través de un tablero web se listan las fechas, el autor y las versiones de los cambios subidos, además permite el manejo de usuarios y los permisos que le correspondan dado un proyecto en específico. Esta herramienta es de gran utilidad en contraposición a las demás existentes, puesto que se integra directamente con Unity lo que permite que el tiempo utilizado en actualizar y respaldar el proyecto sea mucho menor.

Aseguramiento y Control de Calidad

En este capítulo se lleva a cabo el registro de la metodología usada para el aseguramiento y control de calidad del sistema, donde teniendo en cuenta la metodología escogida para el proyecto (Ver **Metodología 3.1**) se definió la etapa de Jugar de la espiral creativa (Ver **Figura 3.1** creativo) como aquella donde en cada uno de los ciclos de iteración se ejecutan y validan los requerimientos establecidos y de calidad para el ciclo correspondiente, el conjunto de tareas establecidas para este proceso en particular contribuyeron por lo tanto a un desarrollo ágil y eficaz del proyecto identificando errores y correcciones a tiempo.

De esta manera, para el proceso de aseguramiento se revisó que Infora contara con la complejidad de las características definidas dentro del proyecto y para el proceso de control de calidad se escogieron diferentes tipos pruebas de software, con el fin de comprobar el correcto funcionamiento del sistema y aceptación por parte de los usuarios finales. Para esto se hizo uso del modelo Verificación y Validación proveniente del modelo CMMI (Capability Maturity Model Integration) y con el cual se logra evidenciar la existencia de lo expuesto en el Capítulo **Metodología, Análisis y Diseño**, y su funcionamiento en cuanto pruebas de calidad.

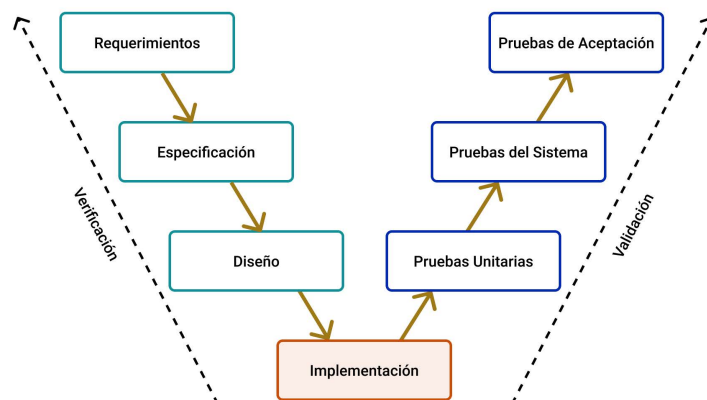


Figura 5.1: Modelo de verificación y validación adaptado.

El modelo de verificación y validación son un conjunto de actividades para la comprobación y análisis, con el objetivo de asegurar que el sistema se desarrolle acorde a su especificación. Este consta de dos sentidos, el de verificación que se centra en las fases iniciales de planteamiento, especificación y diseño de la aplicación; en donde se asegura que la implementación cuente con lo que se estableció en dichas fases [Ste06], para esto se hace una revisión estática de los puntos dichos que debían de tener el sistema, tales como requerimientos funcionales, no funcionales, implementación de la arquitectura y flujo acorde a sus diagramas de especificación. Su otro sentido es el de validación en donde se analiza el sistema con base a requerimientos de prueba, atributos de calidad y pruebas a usuarios finales, para así identificar oportunidades de mejora y por lo tanto llevar a su correcto funcionamiento en cuanto al sistema en general [Eri03].

5.1. Verificación

Ahora bien, para desarrollar el proceso de verificación se hizo uso de una lista de chequeo, con el fin de mantener el control sobre los requerimientos y especificaciones conforme se avanzaba en cada iteración. Previamente para llevar el orden y listas de actividades se seleccionó la herramienta wrike 5.2, esta provee la funcionalidad de "Tabla Scrum" en la cual se consignaron cada una de las tareas correspondientes a los ciclos de desarrollo y a partir de estas se realizó la verificación del cumplimiento de las mismas.

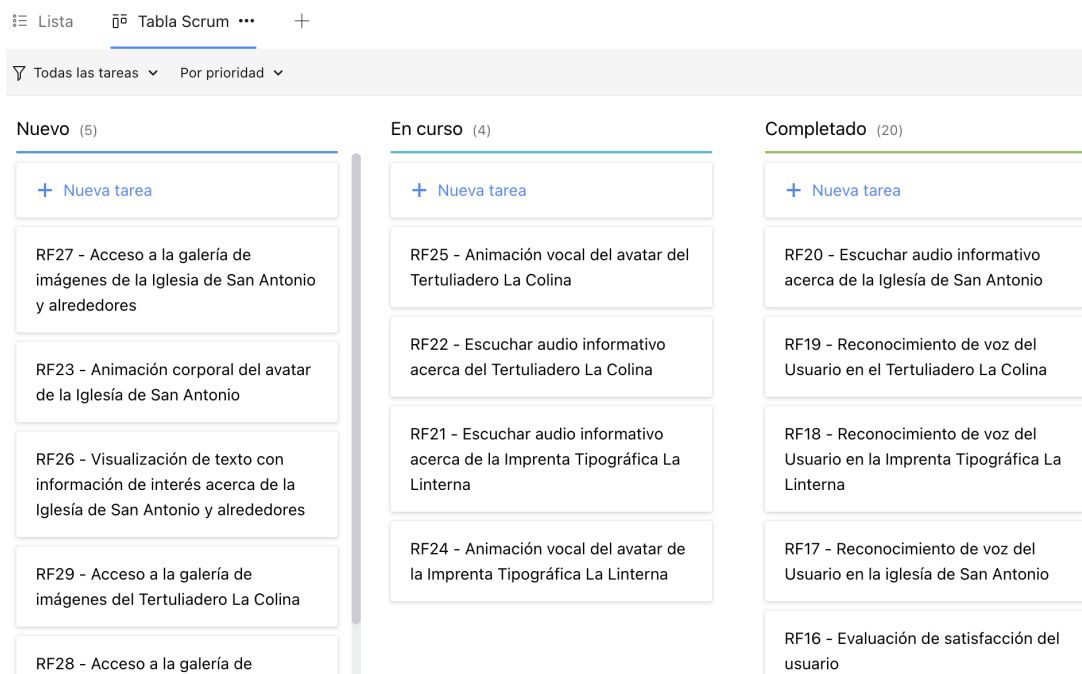


Figura 5.2: Tablero de seguimiento en Wrike.

5.2. Validación

En el proceso de validación se llevan a cabo todas las actividades relacionadas con las pruebas de calidad como tal, por lo tanto se plasman las estrategias utilizadas para la planeación, ejecución y resultados de las mismas junto con su respectivo análisis y corrección de errores o mejoras dentro de las funcionalidades que se consideraron pertinente, esto reconociendo que el ambiente en el que se debió desplegar el sistema requirió de la clasificación oportuna de diferentes tipos de pruebas, exactamente tres diferentes categorías:

- **Pruebas unitarias (UT):** Las pruebas unitarias permiten identificar si una unidad de código o fragmento de código se comportan correctamente [SA12], disminuyendo la probabilidad de errores antes de realizar la prueba de la funcionalidad en sí y promoviendo la identificación de fallas rápidamente.

- **Pruebas de producto (PQT):** Las pruebas de producto son aquellas que validan la existencia de los requerimientos funcionales y no funcionales del sistema, buscan cerciorarse de la calidad del producto o sistema.

- **Pruebas de aceptación de Usuario (UAT):** Las pruebas de aceptación del usuario evidencian el correcto funcionamiento del sistema desde la perspectiva del usuario final y la experiencia de los mismos a la hora de utilizarla, brindando herramientas de toma de decisión claves para el desarrollo y finalización del proyecto.

Según los tipos de prueba expuestos anteriormente se utilizaron herramientas y tiempos distintos para su ejecución, con el fin de ubicar cada una en espacios estratégicos en el diseño, durante y en la finalización de la implementación (ver **Figura 5.3**), por lo cual se abordará cada una desde los métodos utilizados.

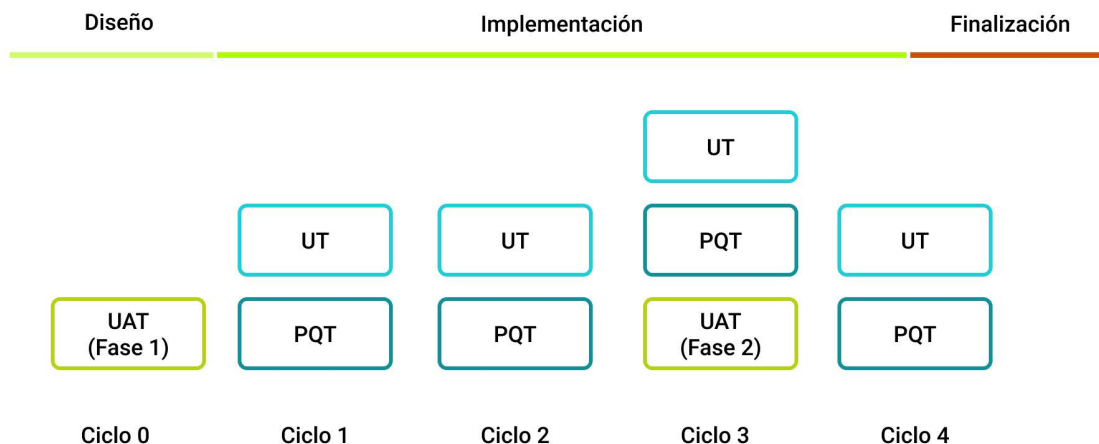


Figura 5.3: Distribución de tipos de pruebas en los ciclos de desarrollo del Asistente Informativo Virtual.

5.2.1. Pruebas unitarias (UT):

Las pruebas unitarias se desarrollan antes de realizar cualquier otro tipo de pruebas, ya que es mucho más eficiente identificar errores antes de entrar a una etapa de PQT, por lo tanto fueron ejecutadas en cada uno de los ciclos en la fase de pruebas como primer paso.

Para llevar a cabo el desarrollo de estas pruebas se hizo uso del **Unity Test Framework (UTF)** provisto por la herramienta utilizada para desarrollar la aplicación Unity, por lo que la integración y uso se da de forma natural y rápida [Tecc].

En este orden de ideas, el principal módulo para el cual se probó el sistema fue el de realidad aumentada. Por ende, se hizo uso de scripts de pruebas como el que se encuentra a continuación.

```
// Tests/VuforiaCamera.cs
/// <summary>
/// Vuforia Camera Unit Test
/// </summary>

public class VuforiaCamera
{
    private GameObject mARCamera;
```

```
[SetUp]
public void SetUp()
{
    mARCamera = new GameObject("ARCamera");

    var camera = mARCamera.AddComponent<Camera>();
    camera.nearClipPlane = 0.05f;
    camera.farClipPlane = 2000f;
    camera.clearFlags = CameraClearFlags.Color;
    camera.backgroundColor = Color.black;
    camera.depth = 1f;
    camera.allowHDR = false;

    mARCamera.AddComponent<AudioListener>();
    mARCamera.AddComponent<VuforiaBehaviour>();
}

[Test]
public void VerifyARCamera()
{
    Assert.IsNotNull(VuforiaBehaviour.Instance);
}
}
```

Por medio de este algoritmo, se probó el despliegue de la cámara de realidad aumentada de Vuforia. Incluso, el marco de trabajo de pruebas crea un ambiente aislado en el cual llevar a cabo pruebas tanto en tiempo de ejecución como edición y genera un resultado de aceptación o no. Esto se evidencia en la **Figura 5.4**.

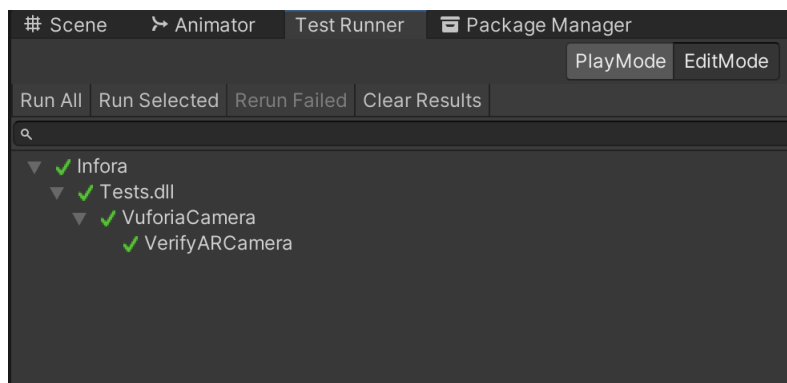


Figura 5.4: Prueba satisfactoria del despliegue de la cámara de realidad aumentada.

5.2.2. Pruebas de producto (PQT):

Las pruebas de producto están enfocadas en cerciorarse del correcto funcionamiento de las funcionalidades correspondientes a la aplicación, en estas se lleva a cabo la siguiente diferenciación:

5.2.2.1. Pruebas funcionales:

Las pruebas funcionales permiten validar la existencia de los requerimientos funcionales en el sistema y que efectivamente estos cumplan con lo propuesto. En este caso las pruebas fueron realizadas de manera manual, siguiendo lo planteado en los requerimientos de prueba (Ver **Tabla 5.1**). Posteriormente se anotaron los resultados obtenidos, y aquellos errores o comportamientos encontrados en la aplicación se procedían a solucionar en el ciclo siguiente.

5.2.2.2. Pruebas no funcionales:

Las pruebas no funcionales verifican los requerimientos a partir de atributos de calidad de software como rendimiento, portabilidad, usabilidad (Ver [Tablero de requerimientos no funcionales](#)).).

Las pruebas de rendimiento son necesarias para identificar cuellos de botella que se pudieran presentar con respecto al uso de la CPU, memoria u otros factores, generando tiempos de respuesta altos y uso excesivo de recursos de los dispositivos. La plataforma de desarrollo Unity provee dentro de sus herramientas el **Unity Profiler**. Profiler recopila y muestra datos acerca del rendimiento de la aplicación en cuanto a CPU, memoria y capacidad de renderización; identificando la relación de su alto o bajo rendimiento con el código fuente, escenas y configuraciones de la aplicación [[Tecn](#)]. Además permitía el acceso a Android y iOS, por este motivo se hizo uso de esta herramienta para llevar a cabo el perfilamiento del aplicativo en los sistemas operativos usados.

En este orden de ideas, Vuforia define un conjunto de cuadros por segundo objetivos para el correcto funcionamiento de las aplicaciones según los dispositivos y los sistemas operativos [[PTC21a](#)]. Con base en esto, si las experiencias cuentan con un tiempo de duración mayor a diez minutos, el número de cuadros objetivo sería 30 (FPS) tanto para Android como para iOS.

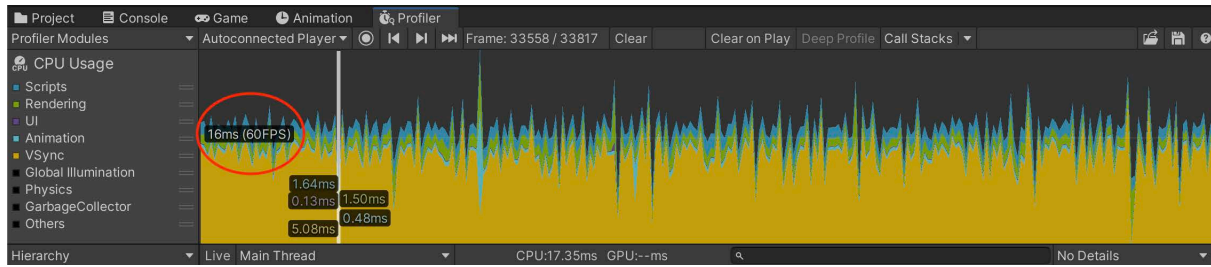


Figura 5.5: Toma de pantalla del Unity Profiler mostrando un promedio de cota inferior de 17.35 ms de tiempo de respuesta de CPU.

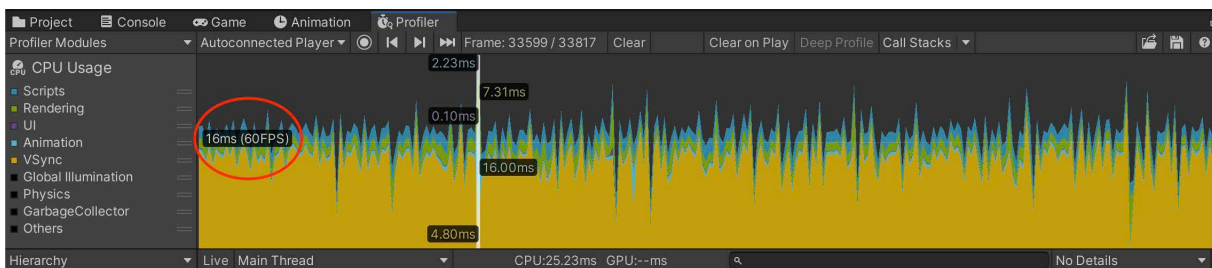


Figura 5.6: Toma de pantalla del Unity Profiler mostrando un promedio de cota superior de 25.23 ms de tiempo de respuesta de CPU.

En ambos casos, hacía el costado izquierdo de la imagen (circulo rojo) se encuentra una etiqueta que contiene "16 ms (60FPS)", esta define que en esa altura se está llevando a cabo un procesamiento con estas características.

Según las **Figuras 5.5 y 5.6** el número de cuadros por segundo se encuentra entre los 30 y 60 FPS, lo que demuestra un funcionamiento esperado basado en lo planteado por Vuforia. Además, al momento de ejecutar tareas con mayor procesamiento como la animación facial y corporal de los avatares, el sistema demostró a través del Unity Profiler responder (comunicación con la CPU, no con el usuario final) en un tiempo promedio no superior a los 30 ms. Mientras que en tareas de bajo procesamiento como entrar al menú de selección, se lograron tiempos de repuesta promedio de 17 ms. Se concluye que, con respecto a lo planteado por Vuforia [PTC21a] las características encontradas en este perfilamiento siguen los estándares documentados y esperados para este tipo de aplicaciones.

Igualmente, como se muestra en la Figura 5.3, las pruebas de PQT se desarrollaron en cada ciclo durante la etapa de JUGAR, donde cada vez que se avanzaba se realizaban las pruebas de forma acumulativa, por lo que se ejecutaban de nuevo las pruebas de ciclos anteriores junto con las correspondientes al actual, de esta manera se cercioraba que ningún componente o requerimiento pasado se hubiera afectado y se generaran retrocesos en ciclos futuros.

Se realizó una lista de requerimientos de prueba (Ver **Anexo 8.5**) donde se relaciona cada elemento con su respectivo requerimiento en caso de aplicar, una descripción corta de la prueba, el resultado esperado después de llevarse a cabo y el ciclo al cual corresponde, en la **Tabla 5.1** se resaltan dos ejemplos de los requerimientos de pruebas utilizados y la estructura descrita:

Requerimiento	Descripción	Resultado esperado	Ciclo
RF-01, RF-04, RF-05	Seleccionar Iglesia, reconocer área objetivo	Avatar Christopher desplegado en pantalla	1
RNF-01	Instalar aplicación en dispositivo con SO Android, ingresar a todas las interfaces	funcionamiento correcto del sistema	1

Cuadro 5.1: Requerimientos de prueba

5.2.3. Pruebas de aceptación del usuario (UAT):

En la realización de las pruebas de aceptación del usuario se utilizaron métodos para evaluar la usabilidad del sistema de acuerdo con Kate Moran [Kat], esto con el fin de identificar aquellas características y propiedades en la aplicación que pudieran ser mejoradas o para validar la satisfacción y la viabilidad de las mismas, desde la opinión y comportamiento del usuario objetivo.

Los métodos de evaluación de usabilidad utilizan distintos medios y técnicas encargados de medir un grupo de aspectos en específico [SGP12]. Para la aplicación de infora se utilizaron métodos de indagación y métodos de test.

- **Métodos de Indagación:** Permiten obtener información acerca de la usabilidad de un sistema para evidenciar la aceptación positiva del usuario [SGP12]. Se seleccionó la siguiente actividad de los métodos de indagación:
 - **Cuestionarios:** Los cuestionarios permitieron a través de una serie de preguntas, recolectar la información necesaria acerca de la satisfacción de los usuarios frente a la aplicación final de Infora.
- **Métodos de test:** En los métodos de test se realizan actividades para identificar cómo los usuarios interactúan con los prototipos y con esto dar soporte del cumplimiento de los objetivos planteados. Para esta categoría se utilizaron las siguientes actividades:
 - **Pensamiento en voz alta:** Se le solicita al usuario que exprese sus inquietudes frente a la aplicación verbalizando sus pensamientos en voz alta, mientras atraviesan la interfaz del sistema; esto sin darle una explicación a profundidad de cómo debe ser usada, así, el usuario expresa su punto de vista desde sus primeras impresiones [Jakb].

Es así como la realización de los métodos en las pruebas de UAT llevaron a consignar los análisis y resultados en dos fases, la primera fase de **Evaluación de interfaz de usuario y usabilidad** y la segunda fase de **Evaluación de satisfacción**. Para cada una de las evaluaciones se seleccionaron 5 personas [Jakc] que quisieran participar y cumplieran con las características del arquetipo de usuario de la aplicación (Ver **Sección 3.2.3**).

5.2.4. Evaluación de interfaz de usuario y usabilidad

La evaluación de interfaz de usuario y usabilidad se encontró pertinente realizarla en el ciclo 0 del proyecto (Ver **Figura 5.3**), es decir en la fase de diseño del prototipo (Ver **Sección 3.3.3**) y de esta manera obtener oportunidades de mejora en cuanto a aspectos estéticos de diseño y de las funcionalidades presentadas por parte de los usuarios antes de desarrollar el prototipo funcional.

Como primera medida se utilizó el método de test, pensamiento en voz alta (Ver 5.2.3), para generar una participación natural en la primera impresión del prototipo. Teniendo en cuenta que fue llevado de manera remota se desarrolló de la siguiente forma por medio de la aplicación de mensajería WhatsApp:

1. Se da al participante una breve contextualización del proyecto y el objetivo de la evaluación.
2. Se comparte el link a la simulación del prototipo diseñado con Figma.
3. Después del usuario haber utilizado la aplicación envía sus apreciaciones en cuanto a la interfaz y usabilidad del prototipo de forma abierta, por medio de mensajes de texto o notas de voz.
4. Se reúnen todas las respuestas y se analizan para finalmente resaltar los elementos específicos que necesitan modificarse.

Resultados

Después de realizar la evaluación a las 5 personas seleccionadas, se recogieron sus apreciaciones y se analizaron para concretar cuales eran aquellos comentarios en común y por lo tanto permitieran realizar una mejora, por lo que se pudo dar cuenta de lo siguiente:

- 3 de las 5 personas proponen que la interacción con los avatares podría ser mayor, puesto que es el elemento que más les llamó la atención.
- 4 de las 5 personas consideran que la interfaz al mostrar los lugares contiene muchos colores y elementos que la hacen ver saturada (Ver **Figura 3.22**).
- 3 de las 5 personas mencionan que les gustaría poder conocer acerca de las experiencias de los demás usuarios al utilizar la aplicación.

Si bien las opiniones en este método tienden a ser subjetivas y cualitativas, permiten al equipo identificar dónde se presentan mayores detalles a tener en cuenta, por lo tanto se realizaron cambios pertinentes en base a la evaluación realizada.

Como primera medida, se modificó la manera de comunicación entre los usuarios y el Asistente Informativo, permitiendo a los mismos entablar una conversación con el avatar y no solo la reproducción automática de audio, como se planteó inicialmente. La implementación de esta funcionalidad y su proceso fue descrita en la Sección (Ver **Sección 4.2.6**).

Teniendo en cuenta la saturación a nivel de interfaz identificada en la evaluación, se realizaron ajustes basados en las heurísticas de Nielsen (Ver **Sección 3.3.3**), donde las características estéticas de color y proporción fueran adecuadas a las demás pantallas de la aplicación y por lo tanto no generaran molestia y/o incomodidad al usuario final. El resultado se puede ver en la Figura 3.22.

Siguiendo con las apreciaciones encontradas se propuso agregar el uso de comentarios a la aplicación, si bien ya se había planteado la opción de brindar una reseña, queja o reclamo por medio del módulo de ayuda(3.18b), los comentarios permitirían satisfacer la necesidad expuesta de reconocer las experiencias de los demás usuarios, vale aclarar que el sistema no contiene creación de usuario ni inicio de sesión, puesto que la idea es acceder de manera fácil y rápida al lugar que se quiere conocer, sin embargo los comentarios se hacen de manera anónima, donde se tiene la posibilidad de agregar una reacción y el texto propio del comentario, y así, poder conocer las experiencias de los demás y aportarle valor a la aplicación.

5.2.5. Evaluación de satisfacción

Teniendo en cuenta que el objetivo de la aplicación es brindar a los turistas y foráneos una experiencia de aprendizaje de lugares, específicamente los escogidos dentro Barrio San Antonio (Ver 3.2). Se realizó esta evaluación para dar cuenta de la efectividad del Asistente Informativo en cuanto a satisfacción y usabilidad para los usuarios finales. De esta manera, la evaluación fue realizada en el ciclo 3 (Ver **Figura 5.3**) de la implementación, es decir, se realizó con un prototipo completamente funcional.

Para llevar a cabo la evaluación se realizó un test de usabilidad [Kat], el cual consta de tres elementos principales: Tareas, Facilitador, Participante. En la **Figura 5.7** se describen cada uno de estos elementos.

Core Elements of Usability Testing



Figura 5.7: Elementos en un test de usabilidad Tomada de: [Kat]

Este tipo de test permite observar y analizar el comportamiento del usuario frente al proceso de aprendizaje con el sistema y cómo este se ve afectado positiva o negativamente por medio de sus expresiones faciales y verbales. Con esto se pretende evidenciar el nivel de compromiso que obtiene el usuario en el tiempo de iniciación, transición y progreso [CD10]. Los comportamientos de iniciación están dados por la primera impresión de usuario, ya sea observando a otros utilizándola o cuando se da una mínima interacción. Los comportamientos de transición se dan cuando el usuario termina de realizar alguna tarea y muestra su reacción positiva frente a su experiencia y comodidad con la respuesta del sistema. Los comportamientos de progreso son aquellos que reconocen un entendimiento profundo del sistema y posible uso futuro. Es así, como todos hacen parte de un conjunto progresivo de reacciones que contribuyen a demostrar la aceptación de la aplicación.

5.2.5.1. Proceso de evaluación

La recolección de datos se realizó en una prueba en conjunto con visitantes del Barrio San Antonio, en los lugares de la Iglesia de San Antonio (Ver 3.2) y la Imprenta La Linterna (Ver 1).

La evaluación tuvo lugar en dos momentos, durante el uso del software y después del uso del software, de este modo, se podría evaluar el comportamiento durante el uso de la aplicación y posteriormente su apreciación general del sistema. Se realizaron procesos de observación y cuestionarios (Ver 5.2.3), registrados en las tablas 5.4 y 5.5, con el fin de medir cuantitativamente los resultados del test.

Durante el uso del software, el facilitador se encargó de guiar a los usuarios por medio de las tareas previamente establecidas (Ver **Tabla 5.2**), dichas tareas se leyeron en voz alta a los participantes conforme se avanzaba una por una, mientras se realizaba cada tarea, el facilitador llevó a cabo un proceso de observación siguiendo el plan establecido en la tabla 5.4, con el fin de convertir

las expresiones faciales y verbales evidenciadas del comportamiento del usuario, en indicadores (Ver **Tabla 5.3**). En algunas ocasiones, el facilitador acudió al método de pensamiento en voz alta (Ver 5.2.3) para confirmar las interpretaciones que tuviera acerca del comportamiento del usuario.

Después del uso del software se realizó un cuestionario (Ver **Tabla 5.5**) que no tomó más de un par de minutos a los usuarios, este fue resuelto a través de Microsoft Forms [Link a cuestionario](#). A partir de las respuestas obtenidas en el cuestionario se realizó posteriormente un análisis que permitiera profundizar en los resultados obtenidos, estos resultados fueron registrados en (Ver **Sección 5.2.5.2**).

5.2.5.2. Método de análisis e Instrumentos

Como se mencionó anteriormente, para llevar a cabo la evaluación se utilizó un test de usabilidad, en el cual las tareas permiten dar un flujo natural y ordenado al uso de la aplicación, además de abarcar a nivel general todas las funcionalidades y posibles caminos que esta pueda tener para obtener un análisis completo del comportamiento y aceptación por parte de los usuarios finales. Teniendo en cuenta esto, se plantearon 6 tareas para ser realizadas en los lugares disponibles en la aplicación, consignadas en la tabla (Ver **Tabla 5.2**). Estas tareas permitieron articular el test de manera objetiva y precisa para ser realizado bajo condiciones de uso realistas.

Para realizar el análisis de los resultados obtenido en el test, se hizo uso de una evaluación sumativa informal, expuesta en [RP12] y [ADG14]. Con esto se logró enfocar la medición de los objetivos en base a indicadores que evidencian la aceptación y el disfrute por parte de los usuarios al interactuar con la aplicación, para seleccionar dichos indicadores el equipo se basó en escenarios de uso real tales como [ADG14] y [MBTO14], además de recomendaciones evidenciadas en [RP12].

Se estableció un plan de observación [ADG14] con el fin de relacionar los indicadores con las acciones ejecutadas por parte de los usuarios mientras realizaban las tareas, estas acciones a su vez se clasificaron a partir de los distintos tiempos de comportamiento que evidencian los participantes del test.

Junto con la estrategia del plan de observación, se realizó un cuestionario adoptando las recomendaciones de otros modelos de cuestionario, en este caso el USE (usabilidad, satisfacción y facilidad de uso) [RP12]. Para medir los resultados de manera cuantitativa por medio de estadística descriptiva, el cuestionario fue evaluado en una escala semántica diferencial de 1(poco de acuerdo) a 5(completamente de acuerdo).

Tarea	Objetivo	Lugar
1. Ingresa a (La Iglesia / La Linterna).	Ingresar al lugar.	Ambos
2. ¿Cuándo fue construida la Iglesia de San Antonio?.	Obtener información acerca de (Iglesia/Linterna).	Iglesia
3. ¿Qué se realiza en la Imprenta La Linterna?.	Obtener información acerca de (Iglesia/Linterna).	Linterna
4. Observe Bailar al Asistente Informativo	Animar el Asistente.	Ambos
5. ¿Dónde se encuentra ubicada (La Iglesia / La Colina)	Encontrar la dirección.	Ambos
6. Envié un comentario acerca de (La Iglesia / La Colina)	Enviar un comentario.	Ambos
7. Cambie el idioma de la aplicación a Inglés	Cambiar Idioma de la aplicación.	Ambos

Cuadro 5.2: Tareas realizadas durante el test

Indicador	Descripción
A. Satisfacción	Recomendaría la aplicación a alguien más. La aplicación funciona como el usuario quiere que funcione. Es divertida de usar.
B. Mejor estado de ánimo después de usar la aplicación	Se siente satisfecho y feliz. Brinda opiniones positivas.
C. Fácil de utilizar	Es simple de usar. El usuario utiliza la aplicación intuitivamente.
D. Intención de uso futuro	Utilizaría la aplicación de nuevo.
E. Frustración	El usuario se sintió frustrado al no poder utilizar correctamente la aplicación.
F. Inmersión	El usuario se concentra en la aplicación, aprende a utilizarlo rápidamente

Cuadro 5.3: Indicadores para medir satisfacción y usabilidad para los usuarios durante el uso de la aplicación

Acción	Comportamiento	Indicador
Realiza la tarea	Iniciación	C
Repite la tarea	Transición	D
Muestra emociones positivas	Transición	A
Recomienda la aplicación a cercanos	Transición	A
Muestra emociones negativas	Transición	E
Identifica rápidamente los pasos a seguir	Progreso	C
Se involucra con la aplicación	Progreso	F
Mejor estado de animo después de usar la aplicación	Progreso	B

Cuadro 5.4: Plan de Observación durante la realización del test

Pregunta	Indicador
¿ Disfrutó de la actividad ?	A
¿ Disfrutó de la conversación con el Asistente Informativo ?	F
¿ Volvería a utilizar la aplicación ?	D
¿ Recomendaría la aplicación a un amigo ?	A
¿ Cree que podría utilizar la aplicación sin un apartado de instrucciones ?	C
¿ Considera que la aplicación es fácil de utilizar ?	C
¿ Considera que la Interfaz gráfica de la aplicación es clara y adecuada ?	Usabilidad

Cuadro 5.5: Cuestionario después la realización del test.

Resultados

Para realizar las evaluación en los distintos lugares, el equipo se dirigió al barrio al Barrio San Antonio para tener un escenario bajo condiciones reales, en la **Figura 5.8** se adjuntan algunas fotos registradas en conjunto con usuarios, mientras estos realizaban las tareas (Ver **Tabla 5.2**).

Dicho esto para resumir los resultados obtenidos, se presenta la **Tabla 5.6** en donde se consignan los puntajes de la escala semántica recolectados con los 5 participantes del test, tanto en el plan de observación (Ver **Tabla 5.4**) realizado durante la ejecución de tareas, como en el cuestionario (Ver **Tabla 5.5**) resuelto al finalizar el test. Por lo tanto se sumaron los valores obtenidos y se dividieron sobre el puntaje máximo que podían lograrse, teniendo en cuenta los indicadores que se relacionaban con cada tarea y pregunta.



Figura 5.8: Imágenes recolectadas en el proceso de evaluación con usuarios finales, en el domo de la Iglesia(a) e Imprenta La Linterna(b).

Indicador	Plan de observación	Cuestionario	Total	Puntaje máximo	Resultado (%)
A	47	24	71	75	95
B	22	n/a	22	25	88
C	43	43	86	100	86
D	23	21	44	50	88
E	0	n/a	0	25	0
F	24	22	46	50	92
Usabilidad	n/a	24	24	25	96

Cuadro 5.6: Resultados de evaluación de satisfacción en indicadores.

De esta manera, pudimos evidenciar que la satisfacción e inmersión por parte de los usuarios fue bastante aceptada, siendo ambos indicadores de los más altos con 95 % y 92 % respectivamente; los usuarios además, brindaron comentarios positivos frente a la implementación del aplicativo en el barrio, resaltando lo divertido y útil que puede llegar a ser en la zona, algunos mencionando que no conocían acerca de La Colina y La Linterna. Unido a esto se encuentra que los usuarios presentan la intención de utilizar la aplicación en el futuro (88 %), mostrando la viabilidad del Asistente Informativo. Sin embargo, valdría la pena seguir trabajando en el diseño de interfaz para mejorar la facilidad de aprendizaje, puesto que aunque no se presentaron inconvenientes, según lo observado

podría mejorar el resultado del indicador (86%), esto podría realizarse por medio de más ciclos de prueba.

5.2.6. Resultados generales:

El orden ejecutado para dar cuenta del aseguramiento y control de calidad, sugerido por el modelo de verificación y validación, permitió al equipo llevar a cabo un plan mucho más ordenado y rápido, sumado a esto las herramientas proveídas por las herramienta Unity, brindaron gran apoyo para el desarrollo de esta fase.

Los resultados indican que la capacidad proveída a nivel no funcional, en cuanto a rendimiento cumple con estándares de calidad brindados por la herramienta Vuforia [PTC21a]. Los requerimientos de prueba lograron cumplirse en su totalidad al finalizar el ciclo 4, por lo tanto alcanzando la totalidad de funcionalidades planteadas para el proyecto. Además, si bien en las pruebas de UAT, el aplicativo no fue evaluado en iOS, con los demás tipos de prueba se logró probar que cumplía con la portabilidad para los dos sistemas operativos móviles iOS y Android. de esta evaluación se puede concluir que los usuarios disfrutaban de la aplicación y esta les brinda satisfacción al utilizarla, además, a algunos de los participantes les permitió conocer acerca de lugares que no conocían antes, por lo tanto, permite dar visibilidad a los mismos. Es así como logra fomentar actividades de turismo que resalten aquellos lugares tradicionales dentro del barrio San Antonio.

Conclusiones

Una vez finalizada la etapa de pruebas, la presentación, animación y voz de los 3 avatares propuestos para la aplicación, evidencia que el prototipo cumple con los objetivos propuestos en la **Sección 1.2**.

Para lograrlo, el presente proyecto de grado desarrolla de forma exitosa un prototipo de aplicación móvil que integra tecnologías como la realidad aumentada y el procesamiento del lenguaje natural. Por medio de estas tecnologías se logró comunicar de manera consciente y natural acerca de los tres lugares escogidos como base la investigación en el barrio San Antonio en Cali, Colombia, lo cual fomenta el turismo inteligente basado en nuevas tecnologías de interacción.

Así mismo, al abordar la investigación desde una mirada de los usuarios finales, la implementación de lenguajes le permitió al prototipo ser usado tanto por locales como por extranjeros. Lo que validó el uso de términos como turismo inteligente y computación cultural. Además, el equipo encargado hizo uso de tecnologías de reconocimiento del ambiente mezclando hardware y software, con las cuales se recreó de manera virtual las fachadas de lugares como la Iglesia de San Antonio y el Domo de cuenteros de San Antonio, esto con el fin de mejorar la experiencia de usuario al momento de desplegar los avatares en realidad aumentada, haciéndolo con un reconocimiento de la infraestructura real existente.

Por otro lado, la interfaz gráfica definida para el Asistente Informativo Virtual se implementó bajo la sugerencia de las heurísticas de usabilidad de Nielsen mencionadas y detalladas en la **Sección 3.3.3**, generando una interfaz estética y fácil de usar. Finalmente, gracias a la conjunción de diferentes técnicas y tecnologías caracterizadas en el **Capítulo 4** se logró alcanzar un prototipo funcional aceptado por un arquetipo de usuarios finales.

Trabajo Futuro

Teniendo en cuenta las actividades realizadas dentro de este trabajo de grado y todo el proceso de investigación de desarrollo que surgió, se considera importante como trabajo futuro continuar con el crecimiento del Asistente Informativo Virtual, mejorando y añadiendo funcionalidades que beneficien positivamente el desarrollo turístico y comercial de la ciudad de Santiago de Cali.

Sin duda, la ciudad cuenta con lugares que vale la pena añadir dentro del enfoque planteado, tanto dentro del Barrio San Antonio como fuera de él (ej: Cristo Rey, Gato Tejada, museo La Tertulia), por lo tanto, se considera interesante continuar con el proceso de investigación de la mano de entidades locales que busquen resaltar la historia y patrimonio de la ciudad, aportando también información necesaria para enriquecer las bases de datos del conocimiento.

De igual manera, es importante seguir un proceso de evaluación exhaustivo con usuarios finales, para identificar acciones que puedan añadirse a los Asistentes Informativos, con el fin de cautivar la atención del público en general y mejorar la experiencia de usuario. Sumado a esto se integrarían funcionalidades, directamente relacionadas con sugerencias de establecimientos comerciales, con el objetivo de brindar visibilidad a los pequeños y grandes empresarios de la región que además, permitan crear un modelo de negocio para la escalabilidad y mantenimiento de la misma aplicación.

Bibliografía

- [ADG14] Navarro-Newball. Andres, Linares. Diego, and Álvarez Gloria. *Talking to Teo: Video game supported speech therapy*. Elsevier, 2014.
- [Ale] Lenk. Alex. *Advanced People Pack*. Unity Asset Store.
- [Anj18] Ken. Anjyo. *Blendshape Facial Animation*. Springer, 2018.
- [ARC] ARCore. Uso de scene viewer para mostrar modelos 3d interactivos en ar desde una aplicación o navegador de android. [urlhttps://developers.google.com/ar/develop/java/scene-viewer](https://developers.google.com/ar/develop/java/scene-viewer).
- [ARK] Arcoachingoverlayview. [urlhttps://developer.apple.com/documentation/arkit/arcoachingoverlayview](https://developer.apple.com/documentation/arkit/arcoachingoverlayview)
- [ASV⁺18] Darwin. Alulema, Betsabe. Simbaña, Christian. Vega, Derlin. Morocho, Alexander. Ibarra, and Verónica. Alulema. *Design of an Augmented Reality-based Application for Quito's Historic Center*. IEEE, 2018.
- [BGDT17] Răzvan. Boboc, Florin. Gîrbacia, Mihai. Duguleană, and Aleš. Tavčar. *A handheld Augmented Reality to revive a demolished Reformed Church from Braşov*. ACM, 2017.
- [CD10] Barriault. Chantal and Pearson. David. *Assessing Exhibits for Learning in Science Centers: A Practical Tool*. Visitor Studies, 2010.
- [CIEA15] Perea-Tanaka. Carlos, Moreno. Isidro, Prakash. Edmond, and Navarro-Newball. Andres. *Towards tantalluc: Interactive mobile augmented reality application for the Museo de América in Madrid*. IEEE, 2015.
- [CMS21] Crazy Minnow Studio. CMS. *SALSA LipSync v2 suite*. SALSA LipSync Suite - Online Documentation, 2021.
- [CMU] Carnegie Mellon. CMU. *Carnegie Mellon University Motion Capture Database*. CMU Graphics Lab Motion.
- [Dav18] Eck David. *Introduction to Computer Graphics*. Hobart and William Smith Colleges, 2018.
- [DPD⁺16] Che Samihah Che. Dalim, Thammathip. Piumsomboon, Arindam. Dey, Mark. Billinghurst, and Shahrizal. Suna. *TeachAR: An Interactive Augmented Reality Tool for Teaching Basic English to Non-Native Children*. IEEE, 2016.
- [DPW07] Hearn. Donald, Baker. Pauline, and Carithers Warren. *Computer Graphics with Open GL*. Pearson New International Edition, 2007.

- [Eri03] Braude Eric. *Ingeniería de Software. Una Perspectiva Orientada a Objetos*. Alfaomega, 2003.
- [Fir21] Yohannes. Firzal. *Architectural photogrammetry: A low cost image acquisition method in documenting built environment*. International Journal of GeoMate, 2021.
- [III15] Tomayess. Issa and Pedro. Isaias. *Usability and Human Computer Interaction (HCI)*. Springer, 2015.
- [Inc] Adobe Systems Incorporated. *Mixamo Part of the Adobe family*. Adobe.
- [j1m18] j1mmyto9. *Speech-And-Text-Unity-iOS-Android*. GitHub, 2018.
- [Jaka] Nielsen Jakob. 10 heurísticas de usabilidad para el diseño de interfaces de usuario. [urlhttps://www.nngroup.com/articles/ten-usability-heuristics/](https://www.nngroup.com/articles/ten-usability-heuristics/).
- [Jakb] Nielsen Jakob. Thinking aloud: The 1 usability tool. [urlhttps://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/](https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/).
- [Jakc] Nielsen Jakob. Why you only need to test with 5 users. [urlhttps://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/](https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/).
- [Kat] Moran Kate. Crea proyectos conjuntos de una manera sencilla. [urlhttps://www.nngroup.com/articles/usability-testing-101/](https://www.nngroup.com/articles/usability-testing-101/).
- [KEA⁺18] Masayoshi. Keynes, Lamounier. Edgard, Cardoso. Alexandre, Lopes. Ederaldo, and Mendes de Lima. Gerson. *Augmented Reality System for Aiding Mild Alzheimer Patients and Caregivers*. IEEE, 2018.
- [Kha08] Siciliano. Khatib. *Springer HandBook of Robotics*. Springer, 2008.
- [Kib13] R. Kibble. *Introduction to natural language processing*. University of London, 2013.
- [LLH18] Sauter. Loris, Rosetto. Luca, and Schuldt. Heiko. *Exploring Cultural Heritage in Augmented Reality with GoFind!* IEEE, 2018.
- [Mar93] Weiser. Mark. *Hot topics-ubiquitous computing*. IEEE, 1993.
- [MBTO14] Elisa D. Mekler, Julia Ayumi Bopp, Alexandre N. Tuch, and Klaus Opwis. *A Systematic Review of Quantitative Studies on the Enjoyment of Digital Entertainment Games*. Association for Computing Machinery, 2014.
- [MCM07] Mark. Maimone, Yang. Cheng, and Larry. Matthies. *Two years of visual odometry on the Mars exploration rovers, J. Field Robot.* California Institute of Technology, 2007.
- [MGA16] Cavallo. Marco, Rhodes. Geoffrey, and Forbes Angus. *Riverwalk: Incorporating Historical Photographs in Public Outdoor Augmented Reality Experiences*. IEEE, 2016.

- [Mic21] Microsoft. *Personality Chat Datasets*. GitHub, 2021.
- [MK94] Paul. Milgram and Fumio. Kishino. *A Taxonomy of Mixed Reality Visual Displays*. IEICE Transactions on Information Systems. vol. E77-D, no. 12, 1994.
- [MTUK94] Paul. Milgram, Haruo. Takemura, Akira. Utsumi, and Fumio. Kishino. *Augmented Reality: A class of displays on the reality-virtuality continuum*. SPIE, vol. 2351, pp. 282–292, 1994.
- [New21] Newtonsoft. *Newtonsoft Json.NET*. GitHub, 2021.
- [PS09] Shirley. Peter and Marschner Steve. *Fundamentals of Computer Graphics*. CRC Press, 2009.
- [PTC21a] PTC. *Framerate and Performance Optimization*. Vuforia Library, 2021.
- [PTC21b] PTC. *Vuforia Fusion*. Vuforia Library, 2021.
- [Res07] Mitchel. Resnick. *All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten*. Cambridge, MA 02139 USA, 2007.
- [RM15] Hewitt. Robert and Joshua. Marshall. *Towards Intensity-Augmented SLAM with LiDAR and ToF Sensors*. ResearchGate, 2015.
- [RP12] Hartson. Rex and Pyla Pardha. *The UX Book Process and Guidelines for Ensuring a Quality User Experience*. Elsevier, 2012.
- [SA12] Kumar. Singh and Singh. Amarjeet. *Software testing*. Vandana Publications, 2012.
- [SELdFG09] Alexandre. Santos, Bruno. Evangelista, José Antonio. Leal de Farias, and Riemer. Grootjans. *Beginning XNA 3.0 Game Programming From Novice to Professional*. Apress, 2009.
- [SGP12] Maris. Stella, Armando De Giusti., and Pesado. Patricia. *Métodos de evaluación de usabilidad: una propuesta de aplicación en Objetos de Aprendizaje*. WICC, 2012.
- [Ste06] Schach Stephen. *Ingeniería de Software Clásica y Orientada a Objetos*. McGraw Hill, 2006.
- [tDJ18] M.Claudia. tom Dieck and Timothy. Jung. *Augmented Reality and Virtual Reality The Power of AR and VR for Business*. Springer, 2018.
- [Teca] Unity Technologies. Crea proyectos conjuntos de una manera sencilla. [urlhttps://unity.com/es/unity/features/collaborate](https://unity.com/es/unity/features/collaborate).
- [Tech] Unity Technologies. Profiler overview. [urlhttps://docs.unity3d.com/Manual/Profiler.html](https://docs.unity3d.com/Manual/Profiler.html).

-
- [Tecc] Unity Technologies. Unit testing. url<https://docs.unity3d.com/Manual/testing-editor-test-runner.html>.
- [Val20] Universidad Valladolid. *Nuevas estrategias educativas para el patrimonio industrial, arquitectónico y cultural 2da Edición*. MIRIADAX, 2020.
- [WCCL18] Chun-Chun. Wei, Fanfan. Chen, Chien-Hsu. Chen, and Yang-Cheng. Lin. *Virtual and Augmented Reality to Historical Site Reconstruction: A Pilot Study of East Taiwan Old Railway Station*. Association for Computing Machinery, 2018.
- [XDYY16] Wei. Xiaodong, Wengb. Dongdong, Liuc. Yue, and Wangd. Yongtian. *A tour guiding system of historical relics based on augmented reality*. IEEE, 2016.

El siguiente capítulo reúne los diferentes anexos referenciados en el documento.

8.1. Análisis de lugares escogidos

Teniendo en cuenta las circunstancias en las que inició el proyecto, se hizo uso de herramientas que permitirán disminuir el número de visitas presenciales al lugar. Primero se estableció una comunicación entre conocidos y familiares del equipo que residieran en el barrio San Antonio en la ciudad de Cali. Estos mencionaron La Linterna y La Colina como un "voz a voz tradicional de la zona", puesto que se revisó su información y no aparecen en plataformas como TripAdvisor, pero sí en páginas gubernamentales, blogs y recomendaciones de Google. En las **Figuras 8.1 y 8.2**, se puede resaltar la aceptación por parte de los visitantes a estos lugares. En la **Figura 8.3** por medio de la herramienta Google Maps, se encuentra la distancia y tiempo que tomaría un recorrido de la zona a pie, siendo un tiempo de 10 minutos en promedio, por lo tanto podría realizarse el recorrido en un tiempo no muy largo.



Figura 8.1: Comentarios y calificación de visitantes del Tertuliadero la Colina.

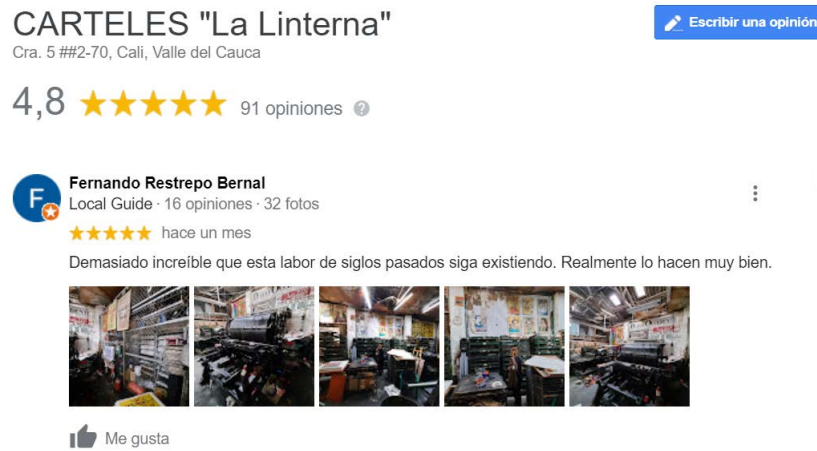


Figura 8.2: Comentarios y calificación de visitantes de la Imprenta la linterna.



Figura 8.3: Distancia encontrada entre los tres lugares escogidos.

8.2. Encuesta acerca de los 3 lugares escogidos

Para esta encuesta se realizaron 12 preguntas, las cuales se encontraron registradas en Microsoft Forms ([Link a encuesta](#)). Información obtenida con 32 participantes de la encuesta (Ver **Figuras 8.4, 8.5 y 8.6**).

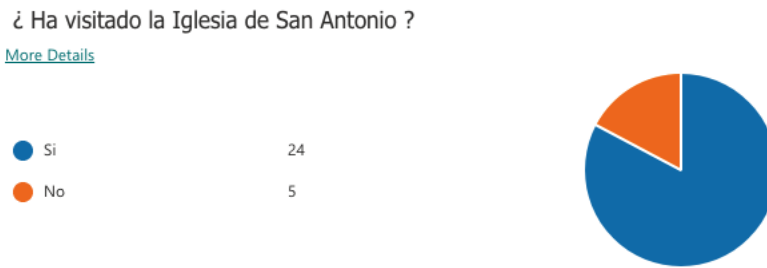


Figura 8.4: Porcentaje de visitas Iglesia de San Antonio Cali.

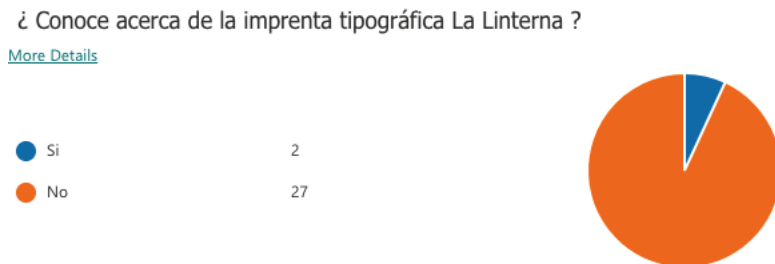


Figura 8.5: Porcentaje de conocimiento acerca de la Imprenta La Linterna.

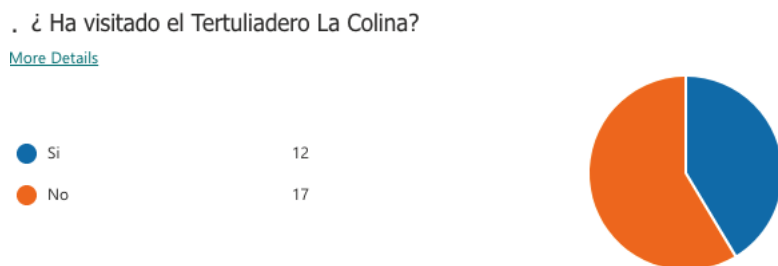


Figura 8.6: Porcentaje de visitas Tertuliadero La Colina.

8.3. Requerimientos funcionales

A continuación se agregan capturas de pantalla de la totalidad de los requerimientos funcionales del tablero de la aplicación **Miro**:

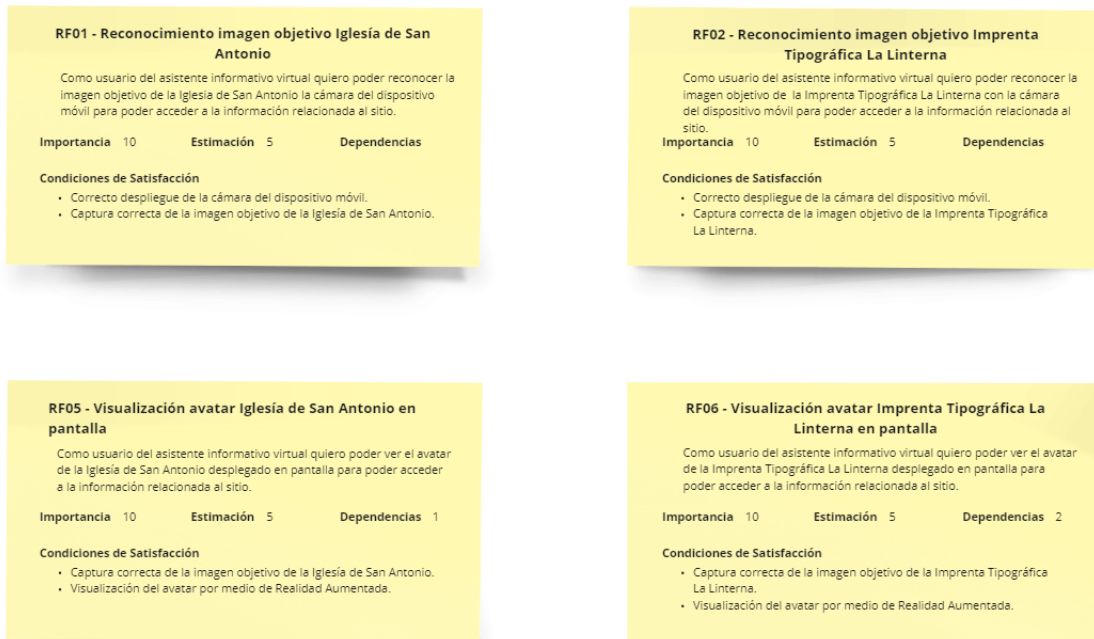


Figura 8.7: Ciclo 1 Parte 1.



Figura 8.8: Ciclo 1 Parte 2.

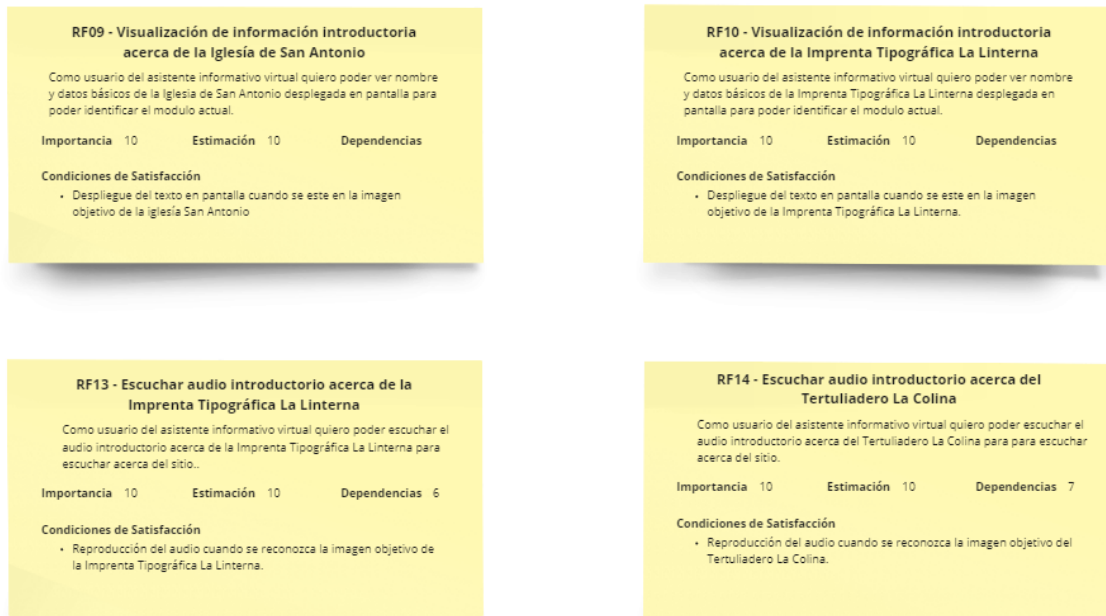


Figura 8.9: Ciclo 2 Parte 1.

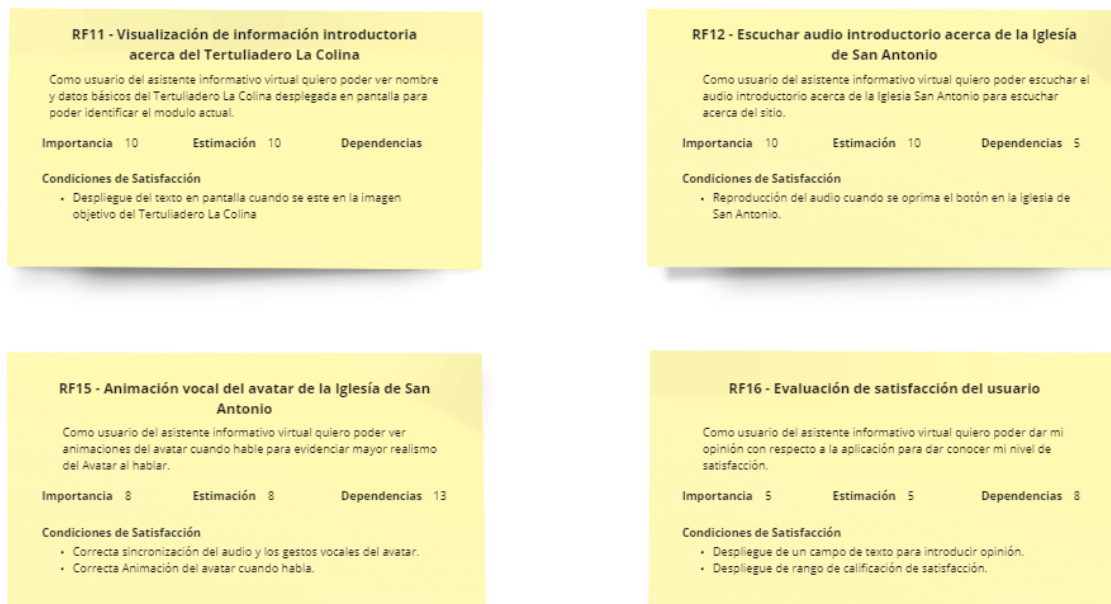


Figura 8.10: Ciclo 2 Parte 2.

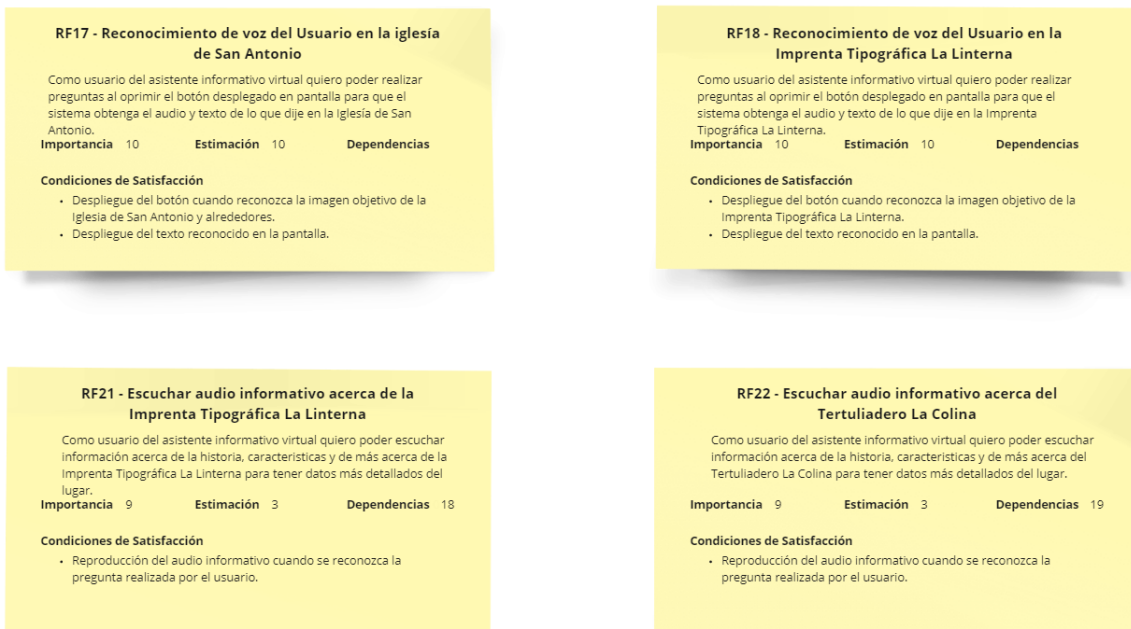


Figura 8.11: Ciclo 3 Parte 1.

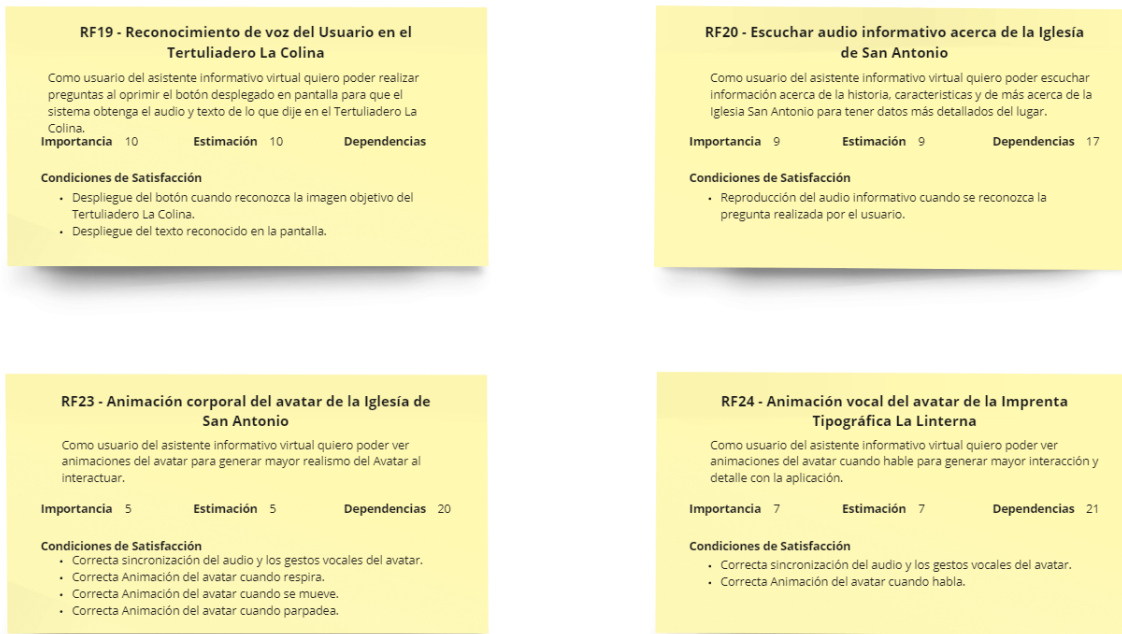


Figura 8.12: Ciclo 3 Parte 2.

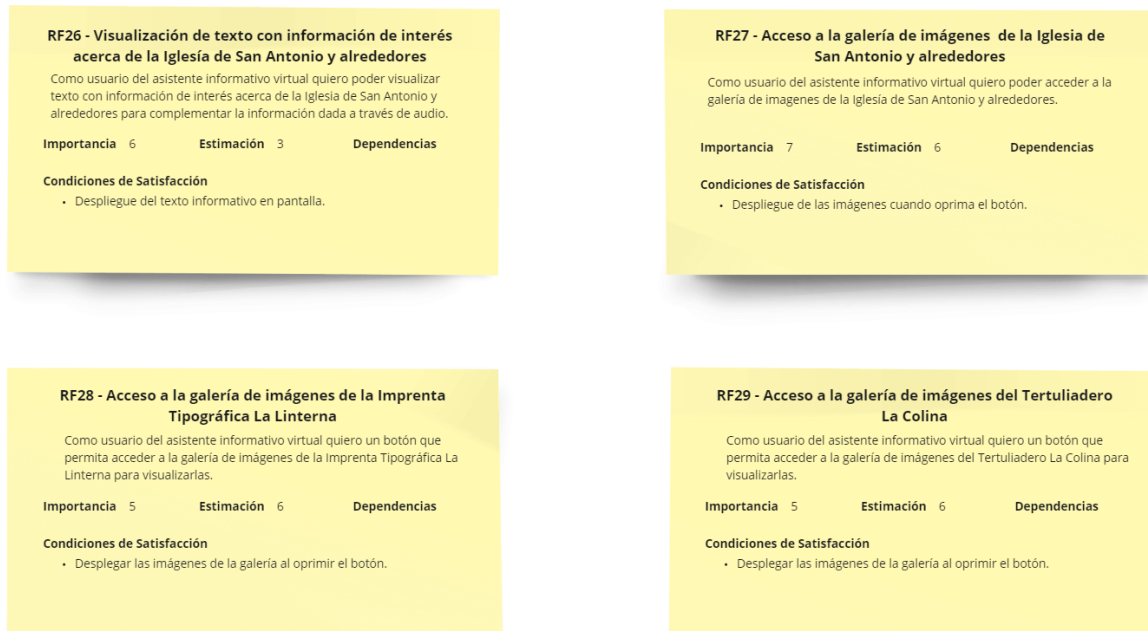


Figura 8.13: Ciclo 4.

8.4. Código fuente del sistema

Código fuente de la definición de cada una de las partes del avatar (faciales y corporales).

```
// CharacterCustomization.cs
/// <summary>
/// Recalculate all blendshapes of character parts using current data
/// </summary>
public void RecalculateBodyShapes()
{
    SetBodyShape(BodyShapeType.Fat, bodyShapeWeight["Fat"]);
    SetBodyShape(BodyShapeType.Muscles, bodyShapeWeight["Muscles"]);
    SetBodyShape(BodyShapeType.Slimness, bodyShapeWeight["Slimness"]);
    SetBodyShape(BodyShapeType.Thin, bodyShapeWeight["Thin"]);
    SetBodyShape(BodyShapeType.BreastSize, bodyShapeWeight["BreastSize"]);

    SetFaceShapeByArray(new Dictionary<FaceShapeType, float>()
    {
        { FaceShapeType.Ear_Size, bodyShapeWeight["Ear_Size"] },
        { FaceShapeType.Ear_Angle, bodyShapeWeight["Ear_Angle"] },
        { FaceShapeType.Jaw_Width, bodyShapeWeight["Jaw_Width"] },
        { FaceShapeType.Jaw_Offset, bodyShapeWeight["Jaw_Offset"] },
    });
}
```

```

    { FaceShapeType.Cheek_Size,    bodyShapeWeight["Cheek_Size" ]},
    { FaceShapeType.Chin_Offset,   bodyShapeWeight["Chin_Offset" ]},
    { FaceShapeType.Eye_Width,     bodyShapeWeight["Eye_Width" ]},
    { FaceShapeType.Eye_Form,      bodyShapeWeight["Eye_Form" ]},
    { FaceShapeType.Eye_InnerCorner, bodyShapeWeight["Eye_InnerCorner" ]},
    { FaceShapeType.Eye_Corner,    bodyShapeWeight["Eye_Corner" ]},
    { FaceShapeType.Eye_Rotation,  bodyShapeWeight["Eye_Rotation" ]},
    { FaceShapeType.Eye_Offset,    bodyShapeWeight["Eye_Offset" ]},
    { FaceShapeType.Eye_ScaleX,    bodyShapeWeight["Eye_ScaleX" ]},
    { FaceShapeType.Eye_ScaleY,    bodyShapeWeight["Eye_ScaleY" ]},
    { FaceShapeType.Eye_Size,      bodyShapeWeight["Eye_Size" ]},
    { FaceShapeType.Eye_Close,     bodyShapeWeight["Eye_Close" ]},
    { FaceShapeType.Eye_Height,    bodyShapeWeight["Eye_Height" ]},
    { FaceShapeType.Brow_Height,   bodyShapeWeight["Brow_Height" ]},
    { FaceShapeType.Brow_Shape,    bodyShapeWeight["Brow_Shape" ]},
    { FaceShapeType.Brow_Thickness, bodyShapeWeight["Brow_Thickness" ]},
    { FaceShapeType.Brow_Length,   bodyShapeWeight["Brow_Length" ]},
    { FaceShapeType.Nose_Length,   bodyShapeWeight["Nose_Length" ]},
    { FaceShapeType.Nose_Size,     bodyShapeWeight["Nose_Size" ]},
    { FaceShapeType.Nose_Angle,    bodyShapeWeight["Nose_Angle" ]},
    { FaceShapeType.Nose_Offset,   bodyShapeWeight["Nose_Offset" ]},
    { FaceShapeType.Nose_Bridge,   bodyShapeWeight["Nose_Bridge" ]},
    { FaceShapeType.Nose_Hump,     bodyShapeWeight["Nose_Hump" ]},
    { FaceShapeType.Mouth_Offset,  bodyShapeWeight["Mouth_Offset" ]},
    { FaceShapeType.Mouth_Width,   bodyShapeWeight["Mouth_Width" ]},
    { FaceShapeType.Mouth_Size,    bodyShapeWeight["Mouth_Size" ]},
    { FaceShapeType.Face_Form,     bodyShapeWeight["Face_Form" ]},
    { FaceShapeType.Chin_Width,    bodyShapeWeight["Chin_Width" ]},
    { FaceShapeType.Chin_Form,     bodyShapeWeight["Chin_Form" ]},
    { FaceShapeType.Neck_Width,    bodyShapeWeight["Neck_Width" ]},
    { FaceShapeType.Smile,         bodyShapeWeight["Smile" ]},
    { FaceShapeType.Sadness,       bodyShapeWeight["Sadness" ]},
    { FaceShapeType.Surprise,      bodyShapeWeight["Surprise" ]},
    { FaceShapeType.Thoughtful,    bodyShapeWeight["Thoughtful" ]},
    { FaceShapeType.Angry,         bodyShapeWeight["Angry" ]},
});

SetHeadOffset(bodyShapeWeight["Head_Offset"]);
SetHeight(heightValue);
SetHeadSize(headSizeValue);
}

```

Código fuente de la corrutina desarrollada para la obtención del audio y su uso en el avatar.

```

// VoiceController.cs
/// <summary>
/// Download audio Corroutine

```

```

/// </summary>
IEnumerator GetAudio(XDocument doc)
{
    var fileUrl = "https://eastus.tts.speech.microsoft.com/cognitiveservices/v1";
    var fileType = AudioType.MPEG;

    RequestHelper currentRequest = new RequestHelper {
        Uri = fileUrl ,
        Method = "POST",
        Headers = new Dictionary<string, string> {
            { "Content-Type", "application/ssml+xml"},
            { "X-Microsoft-OutputFormat", "audio-16khz-32kbitrate-mono-mp3"},
            { "Authorization", "Bearer " + PlayerPrefs.GetString("BearerToken")}
        },
        BodyString = doc.ToString(),
        DownloadHandler = new DownloadHandlerAudioClip(fileUrl, fileType),
        ParseResponseBody = false,
        DefaultContentType = false
    };

    RestClient.Post(currentRequest).Then(response =>
    {
        if (response.StatusCode.ToString() == "200")
        {
            // Download handler - Avatar play audio
            answerAudio.clip =
                ((DownloadHandlerAudioClip)response.Request.downloadHandler).audioClip;
            answerAudio.Play();
        }
    }).Catch(err =>
    {
        var error = err as RequestException;
    });
    yield return null;
}

```

8.5. Requerimientos de prueba

Los requerimientos de prueba fueron creados a partir de los requerimientos funcionales y no funcionales del sistema. Ver **Tabla 8.1**

Requerimiento	Descripción	Resultado esperado	Ciclo
RF-01, RF-04, RF-05	Seleccionar Iglesia, reconocer área objetivo	Avatar Christopher desplegado en pantalla	1
RF-02, RF-06	Seleccionar La Linterna, reconocer imagen objetivo	Avatar Alonso desplegado en pantalla	1
RF-03, RF-07	Seleccionar La Colina, reconocer imagen objetivo	Avatar Aria desplegado en pantalla	1
RF-08	Seleccionar Ayuda	Lista de preguntas y respuestas	1
RNF-01	Ingresar a la aplicación desde dispositivo con SO Android, ingresar a todas las interfaces	funcionamiento correcto del sistema	1
RNF-02	Ingresar a la aplicación desde dispositivo con SO iOS, ingresar a todas las interfaces	funcionamiento correcto del sistema	1
RNF-03	Ingresar a la aplicación, Interactuar con avatares	Profiler muestra FPS entre(30-60)	1
RNF-05	Ingresar a la aplicación, a travasar las interfaces(dos dispositivos)	Interfaces responsive	1
RF-09, RF-10, RF-11	Ingresar al menú de selección	despliegue de información acerca de los 3 lugares	2
RF-12, RF-15	Ingresar a la Iglesia	Avatar saluda y realiza gestos vocales	2
RF-13	Ingresar a la Linterna	Avatar saluda	2
RF-14	Ingresar a la Colina	Avatar saluda	2
RF-16	Ingresar a ayuda, enviar valoración	valoración enviada	2
RF-17, RF-20, RF-23	Ingresar Iglesia, oprimir micrófono, realizar pregunta "¿que sabes acerca de la Iglesia?"	Avatar Christopher responde a la pregunta con animaciones de cuerpo y labios	3
RF-18, RF-21, RF-24	Ingresar Linterna ,oprimir micrófono, realizar pregunta "¿cuenta-me sobre la Imprenta La Linterna?"	Avatar Alonso responde a la pregunta con animaciones de cuerpo y labios	3
RF-19, RF-22, RF-25	Ingresar Colina, oprimir micrófono, realizar pregunta "¿Que hay sobre la Colina?"	Avatar Aria responde a la pregunta con animaciones de cuerpo y labios	3
RF-27, RF-28, RF-29	Ingresar al menú de selección, Ingresar a galería	Despliegue de fotos sobre los 3 lugares	4
RF-30	Ingresar, seleccionar idioma Español	etiquetas y avatar muestran contenido en español	4
RF-30	Ingresar, seleccionar idioma Inglés	etiquetas y avatar muestran contenido en Inglés	4
RF-30	Ingresar, seleccionar idioma Francés	etiquetas y avatar muestran contenido en Francés	4

Cuadro 8.1: Requerimientos de prueba

