



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

GUÍA PARA MUSEOS CON REALIDAD AUMENTADA

Alumno: José Reyes Romero

Tutor: D. Raúl Mata Campos

Depto.: Ingeniería de Telecomunicación

Septiembre, 2019



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares
Grado en Ingeniería de Tecnologías de Telecomunicación

Trabajo Fin de Grado

Guía para museos con realidad aumenta.

Septiembre, 2019.

Alumno: **José Reyes Romero**

Tutor: **Raúl Mata Campos**

TABLA DE CONTENIDO

1	Resumen.....	7
1.1	Resumen	7
1.2	Abstract.....	7
2	Introducción	8
2.1	Antecedentes	11
3	Objetivos y motivación.	16
4	Estado del arte	18
4.1	Tipos de realidad aumentada.	18
4.1.1	Realidad aumentada con marcador	18
4.1.2	Realidad aumentada sin marcador.....	18
4.1.3	Realidad aumentada a través de objetos tangibles	18
4.1.4	Realidad aumentada basada en sensores	19
4.1.5	Realidad aumentada basada en geolocalización	19
4.2	Técnicas de visualización.....	19
4.2.1	Gafas de realidad aumentada.	19
4.2.2	Pantalla de mano	19
4.2.3	Proyección espacial.....	20
4.3	Aplicaciones de la realidad aumentada	20
4.3.1	Arqueología	20
4.3.2	Arquitectura	21
4.3.3	Educación STEM	22
4.3.4	Comercio	23
4.3.5	Literatura.....	24
4.3.6	Arte visual	24
4.3.7	Videojuegos	24
4.3.8	Práctica Sanitaria.....	25
4.3.9	Navegación.....	26
4.3.10	Entornos industriales.....	27

4.3.11	Transmisión y eventos en vivo	27
4.3.12	Turismo.....	28
4.3.13	Traducción	28
4.4	Empresas que invierten en AR.....	29
4.4.1	Microsoft	29
4.4.2	Google	29
4.4.3	Apple	30
5	Desarrollo.....	31
5.1	Aplicaciones similares en el mercado.....	32
5.1.1	Skin & Bones	32
5.1.2	England Originals.....	33
5.1.3	Museo Carlos V	33
5.1.4	Sorolla Museum AR	34
5.1.5	Museo Nacional de Escultura.....	34
5.2	Requisitos del sistema	35
5.3	Implementación de herramientas	36
5.4	React Native – ViroReact.....	38
5.4.1	React Native	38
5.4.2	ViroReact.....	39
5.4.3	Funcionamiento e integración	39
5.5	Confrontación requisitos del sistema.....	40
5.5.1	Detección de superficies	40
5.5.2	Lectura de imágenes.....	41
5.5.3	Reconocimiento y comparación de imágenes	41
5.5.4	Representación de contenido.....	41
5.5.5	Anclaje de información.....	42
5.5.6	Interacción del usuario	42
5.6	Elección de componentes	43
5.7	Lógica del sistema	46

5.8	Programación del sistema.....	48
5.8.1	Editor de código fuente empleado.....	48
5.8.2	Proyecto.....	49
5.8.3	Interfaz de usuario	61
5.9	Requisitos y capacidades.....	64
6	Pruebas y resultados.....	66
6.1	Colección	67
6.2	Posicionamiento y escalado	71
6.3	Interacción	73
6.4	Comentarios y observaciones	73
7	Escalabilidad y migración a otros requerimientos	76
7.1	Introducir nuevos contenidos	76
7.2	Migración a otros requerimientos	77
8	Conclusión	78
9	Trabajo futuro.....	79
9.1	Mejoras en la aplicación.....	79
9.2	Enseñanza y mini juegos	79
9.3	Nuevas plataformas	79
10	Bibliografía	81
	Anexo 1. Creación de archivo ejecutable.....	85
	Anexo 2. Manual de usuario de la aplicación.....	89
	Experiencia Básica	89
	Experiencia Específica.....	90
	Requisitos y buenas maneras de uso.	91

TABLA DE ILUSTRACIONES

Figura 2.1 – Curva de Garther 2018	10
Figura 2.2 – Ejemplo de aplicaciones; a) comercial , b) turístico, c) médico, d) industrial	11

Figura 2.3 - Sensorama.....	12
Figura 2.4 - Ivan Sutherland head mounted display.....	13
Figura 2.5 – Sistema de AR Virtual Fixtures	13
Figura 2.6 - Kinect	14
Figura 2.7 – Google Glass.....	14
Figura 2.8 – Videojuego con AR Pokémon Go	15
Figura 4.1 – Sistema VITA.....	21
Figura 4.2 – Casco DAQRI	21
Figura 4.3 – Realidad Aumentada en la educación.....	22
Figura 4.4 – Escáner biométrico Bodymetrics.....	23
Figura 4.5 – Exposición de arte con RA con obras de Scarlett Raven	24
Figura 4.6 – Render del juego Star Wars: Jedi Challenges.....	25
Figura 4.7- Dispositivo Near-infrared vein finder.....	26
Figura 4.8 – Sistema de visión sintética híbrida.....	26
Figura 4.9 – Realidad Aumentada en entorno industrial	27
Figura 4.10 – Porcentaje de acierto de tiro con AR.....	28
Figura 4.11 – Aplicación Wikitude AR Travel Guide.....	28
Figura 4.12 – Ejemplo del sistema Word Lens.....	29
Figura 5.1 – Metodología para el desarrollo de la aplicación	32
Figura 5.2- Ejemplo aplicación Skin and Bones.....	32
Figura 5.3 – Ejemplo aplicación England Originals	33
Figura 5.4 – Ejemplo aplicación museo Carlos V.....	34
Figura 5.5 – Ejemplo aplicación Sorolla Museum AR	34
Figura 5.6 – Ejemplo aplicación Museo Nacional de Escultura.....	35
Figura 5.7 – Interconexión de los componentes	43
Figura 5.8 – Listado componentes API ViroReact	45
Figura 5.9- Sistema de referencia de coordenadas cartesianas	46
Figura 5.10 – Lógica del sistema – experiencia básica.....	46
Figura 5.11 – Lógica del sistema – experiencia específica	47
Figura 5.12 – Editor de código fuente ATOM.....	48
Figura 5.13 – Simulación de la interfaz para experiencia básica.....	62
Figura 5.14 - Simulación de la interfaz para experiencia específica.....	63
Figura 5.15 - Simulación de la interfaz para experiencia específica.....	64
Figura 6.1 – Metodología empleada en las pruebas.	67
Figura 6.2 – Imágenes que comparten estilo.	68
Figura 6.3 – Imágenes del mismo autor.....	69
Figura 6.4 – Imágenes con la misma técnica.....	69

Figura 6.5 – Imágenes que representan imágenes similares.....	70
Figura 6.6 – Imágenes con mucho y poco contraste.....	71
Figura 6.7 – Experiencia AR con tamaños y posiciones sin ajustar	72
Figura 6.8 – Experiencia AR con tamaños y posiciones ajustadas	72
Figura 6.9 – Experiencia específica con interacción	73
Figura 9.1 – Headset Lenovo Mirage.....	80
Figura 0.1 – Ejemplo experiencia básica	89
Figura 0.2 – Ejemplo experiencia específica.....	90

1 RESUMEN

1.1 Resumen

El principal objetivo de este trabajo de fin de grado es el crear una aplicación de realidad aumentada que sirva como guía para un museo o galería de arte. Con el objetivo de proporcionar al usuario una nueva experiencia de acercamiento al arte pictórico, mediante el empleo de la realidad aumentada.

Para ello, se va adquirir nuevos conocimientos de programación tanto de propósito general para el desarrollo de una aplicación móvil como de propósito específico para proporcionar a la aplicación la capacidad de realidad aumentada. En concreto, en este trabajo se pretende la creación de una aplicación móvil de realidad aumentada que sea capaz de reconocer las fotografías correspondientes a 30 cuadros y mostrar una serie de elementos de aumentados que proporcionaran información relacionada con la obra correspondiente.

La aplicación está desarrollada empleando un framework de JavaScript llamado React Native el cual permite la integración con la plataforma ViroReact. ViroReact nos permite la creación de aplicaciones de realidad aumentada y virtual. Al emplear este framework nos permite que nuestra aplicación se pueda convertir en una aplicación con funcionamiento nativo para dispositivos Android e iOS (no se realiza la conversión, pero si se indica dónde encontrar la información para ello).

Se tendrá entonces, una aplicación para dispositivos móviles Android que nos permitirá reconocer las obras, mostrar el contenido aumentado e interactuar con él.

1.2 Abstract

In this work, an augmented reality application will be developed to serve as a guide for a museum or art gallery. With the aim of providing the user with a new experience of approaching pictorial art, through the use of augmented reality.

For this, new programming knowledge will be acquired for the development of a mobile application and to provide you with the capacity of augmented reality.

The application will be able to recognize the photographs corresponding to 30 pictures and show a series of augmented elements that will provide information related to the corresponding work.

For its development a JavaScript framework called React Native has been used which allows integration with the ViroReact platform. ViroReact allows us to create augmented reality applications. Using this framework allows us to make our application an application with native operation for Android and iOS devices.

2 INTRODUCCIÓN

En las últimas décadas se ha producido un crecimiento exponencial en desarrollo de dispositivos móviles y facilidad al acceso continuo a Internet. Lo que ha provocado que los dispositivos móviles formen parte de nuestra vida cotidiana. Haciendo que cada vez se desarrollen más aplicaciones y que sean más complejas debido al aumento de capacidad de procesamiento de los microprocesadores. Desde las primeras aplicaciones que simplemente mostraban texto y permitían poca interacción han ido evolucionando hasta llegar a aplicaciones que permiten editar fotografías, videojuegos, GPS, etc.

Como ya se ha mencionado el campo de la tecnología está cada vez más presente en todos los ámbitos de la vida y se llevan décadas investigando y desarrollando sistemas que faciliten las tareas a las personas; y dentro de estas tecnologías se encuentran la Realidad Virtual (VR) y la Realidad Aumentada (AR) que presentan una manera diferente a la habitual en la interacción entre humano-maquina.

La **Realidad Aumentada** se podría definir como una experiencia interactiva de un entorno del mundo real donde los objetos que residen en el mismo se ven mejorados por la información perceptiva generada por ordenador, a través de múltiples modalidades sensoriales, incluyendo visual, auditiva y olfativa.

Un sistema de AR está compuesto por tres simples fases. Una primera fase de reconocimiento, una segunda de seguimiento y una última fase de integrar la información del mundo o del objeto virtual con la imagen del mundo real. Considerando esto podemos indicar cuales son los elementos mínimos necesarios para componer un sistema de AR.

- Un elemento que capture las imágenes de la realidad que está siendo vista por los usuarios. Para lo que sería suficiente con una cámara presente el dispositivo que se vaya a emplear como pudiera ser un teléfono móvil, Tablet, o gafas de AR.
- En segundo lugar, un elemento sobre que el mostrar la combinación de las imágenes reales con las imágenes creadas. Este elemento podría ser una pantalla de ordenador, de teléfono móvil o el display de unas gafas de AR.
- En tercer lugar, un elemento o varios de procesamiento cuya función es la de interpretar la información del mundo real que recibe el usuario y generar la información virtual necesaria e integrarla de forma adecuada. Este punto es importante para determinar el grado de madurez del dispositivo, mientras que en la actualidad el procesamiento de los smartphones y Tablet es suficiente aún está limitado en el caso de las gafas de realidad aumentada.

- Finalmente se necesita un el elemento activador, en el mejor de los casos el activador sería la imagen que están viendo los usuarios y a partir de ella el sistema debería reaccionar. Pero, dada la complejidad técnica de este proceso (cada vez menor), también se utilizan otros elementos de activador como el GPS, marcadores tipo RFID o códigos bidimensionales. Los activadores se pueden emplear de manera aislada o combinando varios de ellos.

Una parte también muy importante en el desarrollo y expansión de esta tecnología es la del software, del cual podemos encontrar soluciones gratuitas lo que permite que cualquier persona con conocimientos técnicos pueda desarrollar sus propias experiencias de Realidad Aumentada.

En este campo del software podemos destacar algunos de los SDKs (software development kit) más importantes y que están siendo más usados por los desarrolladores son:

- Vuforia.
- Arkit.
- ARCore.

A pesar de que la realidad aumentada es una tecnología que ha provocado muchas fantasías a día de hoy ha demostrado muy poco. Ya que muchas publicaciones lanzaron videos que mostraban increíbles experiencias con realidad aumentada, pero esas superaron las posibilidades reales. Provocando un rechazo de la AR por parte del gran público, que la experiencia más cercana que han tenido a la AR es el videojuego Pokémon Go.

Para ver el estado actual de la realidad aumentada, podemos observar la curva de Gartner para 2018 (Figura 2.1). La cual sitúa a la AR el abismo de la desilusión. Pero un análisis más detallado nos revela que en un periodo de 5 a 10 se encontrará en la “Meseta de Productividad” aunque estos plazos pueden variar dependiendo del interés por parte de las empresas y usuarios. La AR se está consolidado en sectores económicamente importantes como el comercio, el turismo y el entretenimiento, y en menor medida, pero no menos importantes en los sectores industriales y médico, lo que augura que los plazos se están cumpliendo.

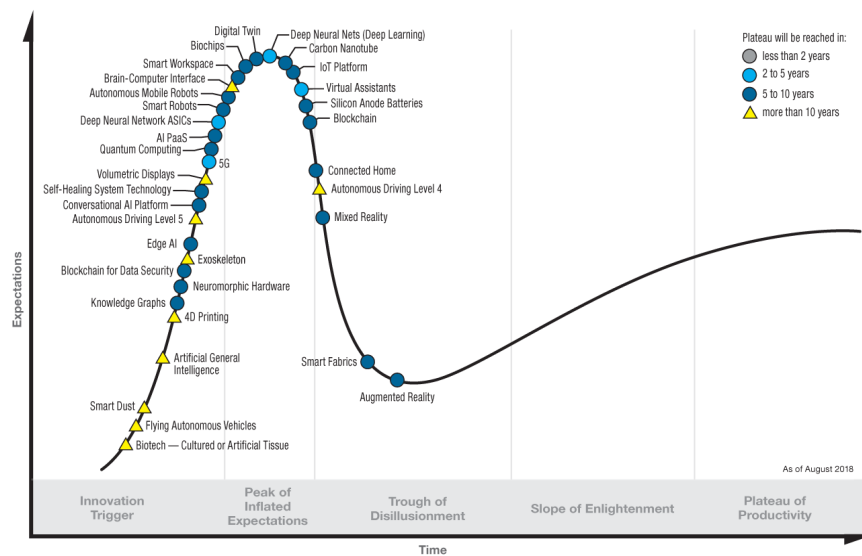


Figura 2.1 – Curva de Garther 2018

Fuente: Garther.com

Algunos ejemplos los podemos encontrar en las aplicaciones desarrolladas por IKEA o Wayfair, (Figura 2.2 a) que permiten mostrar un mueble en relación al entorno real. En el sector del turismo las aplicaciones suelen utilizar como activador el GPS (Sistema de Posicionamiento Global), para mostrarle al usuario recomendaciones sobre establecimientos cercanos o monumentos (Figura 2.2 b). Mientras que en el sistema del entretenimiento podemos encontrar, las mas que famosas aplicaciones de Pokémon Go y Snapchat también podemos destacar la aplicación Figment AR (realizada con la API ViroReact).

Y a nivel industrial que será este el que dote a la tecnología de AR de una madurez que le permita desarrollarse en todos los estratos de la sociedad, nos encontramos con algunos ejemplos como MiRA, desarrollada por Airbus Group Innovation, la cual permite al ingeniero aeronáutico verificar la integridad de la colocación correcta de miles de piezas instaladas en sus modelos de aviones (Figura 2.2 d).

De igual en el sector médico, donde la planificación para realizar cirugías es fundamental para que éstas tenga éxito, consiguen que se puedan superponer sobre el cuerpo del paciente datos correspondientes a exámenes médicos anteriores como podrían ser resonancias magnéticas, rayos X o construir con esos mismos datos un modelo en 3D de su anatomía, mediante la superposición de imágenes (Figura 2.2 c). Y en el transcurso de la cirugía, mediante el empleo de unas gafas de AR se podría conseguir que el paciente se volviese transparente a la vista del cirujano, y de esta manera podría visualizar estructuras dentro de los órganos (vasos, tumores ...) que de otra manera solamente las podría percibir a través de métodos invasivos como el tacto. Consiguiendo de esta manera

que la visión real del cirujano se vea enriquecida, lo que conduce a menos tiempo de intervención, lo que repercutiría en un periodo de hospitalización menor. Aunque los problemas asociados con las calibraciones (puntos comunes entre los datos), imponen restricciones al desarrollo de este tipo de técnicas.

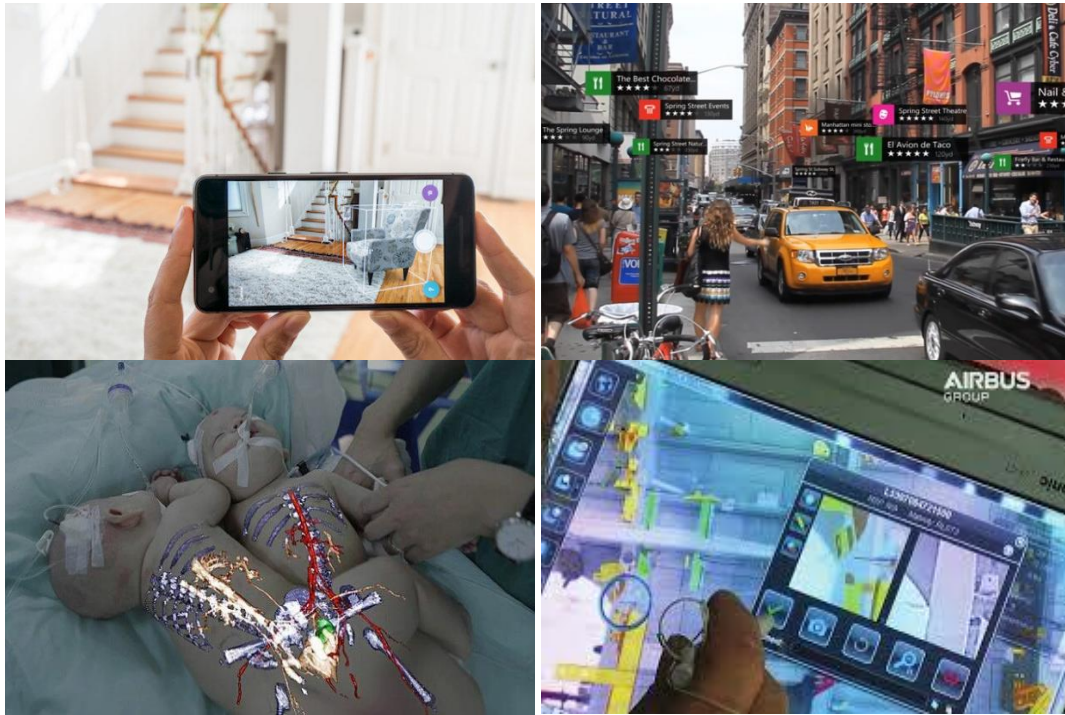


Figura 2.2 – Ejemplo de aplicaciones; a) comercial , b) turístico, c) médico, d) industrial

Fuente: Wikimedia Commons y Airbus Group

2.1 Antecedentes

La realidad aumentada, aunque parece una tecnología muy reciente, no es así, ya que a principios del siglo pasado se empezaron a desarrollar sistemas que, sin usar el término de realidad aumentada, si pretendían crear experiencias en las que los usuarios recibieran más estímulos sensoriales.

Pero antes de hacer un recorrido histórico desde los inicios de la realidad aumentada, vamos a definir lo que se considera la Realidad Aumentada y en qué se diferencia de la Realidad Virtual.

Realidad Aumentada (AR): es una tecnología que superpone o añade capas de información al mundo real, mediante elementos creados de manera digital, sobre el entorno que rodea al usuario.

Realidad Virtual (VR): es una tecnología que crea un entorno de escenas u objetos normalmente de apariencia real, la cual le crea al usuario la sensación de estar inmerso en dicho entorno.

La historia de la AR se puede resumir en los siguientes puntos, que se consideran hitos en esta tecnología:

- 1901: L. Frank Baun, menciona por primera vez la idea de una pantalla/gafas electrónicas que superponen información en la vida real, en este caso sobre personas. Se llama “carácter marker”.
- 1957-62: Morton Heiling, un director de fotografía, crea y patenta un simulador de moto llamada Sensorama con imágenes, sonido, vibración y olor. Podemos ver cómo era este simulador en la Figura 2.3.



Figura 2.3 - Sensorama

Fuente: Wikimedia Commons

- 1968: Ivan Sutherland inventa el casco de realidad virtual lo que sugiere una ventana a un mundo virtual. Como se puede observar en la Figura 2.4, se necesitaba una gran infraestructura.



Figura 2.4 - Ivan Sutherland head mounted display

Fuente: D'Source.in

- 1981: Dan Reitan asigna geoespacialmente múltiples imágenes de radar meteorológico, cámaras espaciales y de estudio a mapas terrestres y símbolos abstractos para transmisiones meteorológicas de televisión, llevando un concepto precursor a la realidad aumentada (imágenes reales / graficas mixtas) a la televisión.
- 1990: Nace el termino de “realidad aumentada” se atribuye a Thomas P. Caudell un ex investigador de Boeing.
- 1992: Louis Rosenberg desarrolló uno de los primeros sistemas de AR en funcionamiento, llamado Virtual Fixtures (Figura 2.5) el Laboratorio de Investigación de las Fuerza Aérea de los Estados Unidos, Armstrong que demostró beneficios para la percepción humana.



Figura 2.5 – Sistema de AR Virtual Fixtures

Fuente: Wikimedia Commons

- 1994: Julie Martin crea la primera, producción de teatro de realidad aumentada, Dancing In Cyberspace que presenta bailarines y acróbatas que manipulas objetos virtuales del tamaño del cuerpo en tiempo real, proyectados en el mismo espacio físico y plano de actuación.

- 1998: NASA X-38 voló utilizando superposición de mapas de video del software LandForm en el Centro de Investigación de Vuelo Dryden.
- 1999: El Laboratorio de Investigación Naval de los EE.UU. Participa en un programa de investigación de una década llamado Sistemas de Realidad Aumentada de Battlefield (BARS) para crear prototipos de algunos de los primeros sistemas portátiles para soldados a pie que operan en entornos urbanos para la concienciación y el entrenamiento de la situación.
- 2008: Wikitude AR Travel Guide se lanza al mercado el 20 de octubre de 2008 con el teléfono Android G1.
- 2010: Salida al mercado de Kinect (Figura 2.6) un controlador de juego desarrollado por Microsoft para la videoconsola Xbox 360, que permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico con un controlador.



Figura 2.6 - Kinect

Fuente: Wikimedia Commons

- 2012: Glass (Figura 2.7), es un dispositivo de visualización de tipo gafas de realidad aumentada desarrollado por Google y presentado en el congreso I/O del junio del 2012.



Figura 2.7 – Google Glass

Fuente: Wikimedia Commons

- 2016: presentación del HoloLens unas gafas de realidad aumentada, de Microsoft.
- 2016: Niantic lanza Pokémon Go.



Figura 2.8 – Videojuego con AR Pokémon Go

Fuente: Flickr.com

- 2017: Apple anuncia ARkit y Google anuncia ARCore.

3 OBJETIVOS Y MOTIVACIÓN.

El principal objetivo, es crear una aplicación móvil que suministre información ampliada mediante Realidad Aumentada para realizar visitas de museos.

Se va a desarrollar una aplicación que sea capaz de reconocer una serie de marcadores (códigos QR o las propias imágenes) que identifiquen las obras pictóricas presentes en el museo, y una vez reconocidas muestre información ampliada específica para cada una de las obras, con la cual podrá interactuar el usuario, consiguiendo de esta manera una guía para museos con realidad aumentada.

Además, la aplicación que se va implementar queda abierta para numerosas aplicaciones futuras como podría ser: enseñanza de bellas artes, educación STEM para completar los libros de texto y para ayudar a personas con deficiencias visuales o auditivas, ya que se podría añadir explicaciones mediante lenguaje de signos o sonora. Es posible plantear por tanto una serie de objetivos fundamentales para alcanzar nuestra meta, y que quedarán definidos como los objetivos del TFG.

- Revisión bibliográfica y estudio de las técnicas de AR en general y específicas para la visita de exposiciones y museos además de la documentación específica de la plataforma Viro que será la empleada para desarrollar la aplicación.
- Estudio de aplicaciones existentes en el mercado. Para conocer sus diferentes alcances, limitaciones y requisitos que servirán de ayuda para diseñar las funcionalidades que se van a pretender cubrir con la aplicación desarrollada en este TFG.
- Diseño e implementación de algoritmos/herramientas/librerías que integrarán la aplicación que desarrolle el análisis y procesamiento objetivos. Es importante conocer si hay alguna herramienta que pueda ser útil para la realización de este trabajo, y si está permitido su uso. Permitiendo así saber que herramientas tendrían que ser programadas y diseñadas, es posible desarrollar la aplicación siguiendo una metodología específica.
- Diseños de una aplicación específica para guía en museo/exposición en un entorno gráfico a partir del software creado y recopilado. Una vez decidido que herramientas serán desarrolladas y cuales, recopiladas, el siguiente paso será la implementación de estas en la aplicación. Para ello necesitaremos estudiar el modo de interconexión entre dichas herramientas cuales son las limitaciones que se nos plantean. Como punto fundamental, será necesario tener la posibilidad de capturar información del dispositivo móvil correspondiente a la cámara y demás sensores y el algoritmo o librería dedicada a reconocer las obras.

- Implementación y comprobación del correcto funcionamiento de la aplicación. Este apartado es fundamental para corroborar si se ha alcanzado el principal objetivo de este TFG. Una vez desarrollada la primera versión de la aplicación, se comprobará si es funcional y si los resultados son adecuados para el uso que se pretende dar. Conseguido esto, habrá que llevar a cabo pruebas de testeo más exigentes hasta que se determine que la aplicación desarrollada es adecuada para cumplir con el objetivo de este TFG.
- Desarrollo de la memoria, manuales y documentación habitual a la presentación de trabajo de fin de grado. Para ello se tendrá que cumplir con las normas básicas de estilo y estructura especificadas por la escuela para este tipo de trabajo. En ellas se determinan aspectos como el tipo de letra, la numeración de figuras, tablas y formulas o la norma a seguir para las referencias en la bibliografía entre otros aspectos. Además, para este TFG, será necesaria la inclusión de un anexo que contenga un manual para el usuario en el que se explicará cómo se utiliza la aplicación.

Siguiendo esta metodología, se habrá conseguido alcanzar el principal objetivo. Una vez conseguido, se pondrá la meta más allá para intentar lograr nuevos objetivos siempre con la intención de mejorar este trabajo.

4 ESTADO DEL ARTE

En este capítulo, se va a desarrollar el estado del arte de la realidad aumentada y de las aplicaciones que la emplean. Para ellos, se hablará de las diferentes maneras de interacción y de las distintas técnicas de visualización de la realidad aumentada. También distintas áreas en las que se puede aplicar la AR y cuales empresas están desarrollando e invirtiendo en ella.

4.1 Tipos de realidad aumentada.

Podemos diferenciar 5 tipos de realidad aumentada, dependiendo de la manera en la que el contenido es integrado en la experiencia, es decir, en función de los elementos activadores que sean empleados.

4.1.1 *Realidad aumentada con marcador*

Este tipo de realidad es la que emplea como activador un marcador. Los marcadores son símbolos 2D impresos en un papel o imágenes que presentan gran contraste ya que normalmente son negros sobre fondo blanco y sobre el cuál se superponen los elementos virtuales. El contenido correspondiente a la experiencia aparece cuando la app asociada reconoce el marcador y activa la experiencia. Para que funcione correctamente, es necesario que el marcador se encuentre en una superficie plana y el dispositivo mantenga una distancia adecuada.

4.1.2 *Realidad aumentada sin marcador*

Este tipo de realidad es la que se superpone a la realidad sin necesidad de marcadores, es decir, es capaz de analizar la imagen que capta que será mucho variable que en el caso de los marcadores. Por lo que para funcionar necesita más capacidad de procesamiento lo que repercute en más recursos computacionales, que no ocurría en el caso de emplear marcadores.

4.1.3 *Realidad aumentada a través de objetos tangibles*

Este tipo de realidad es el menos típico de todos, en esta se invierte la interacción dispositivo-realidad alterada, ya que, en vez de ser una realidad aumentada sensible al entorno, utiliza objetos concretos para desplegar la información.

4.1.4 Realidad aumentada basada en sensores

Este tipo de realidad aumentada se basa en la información proporcionada por sensores, como pueden ser giroscopios, acelerómetros y magnetoscopios. Esta tecnología está siendo utilizada con frecuencia aprovechando los recursos que incorporan los smartphones, que la mayoría suelen disponer de estos sensores. Presenta como desventaja que no se pueden realizar tareas como zoom, solamente permite realizar movimientos de rotación, lo que la convierte en un tipo de realidad aumentada estática. Pero suele emplearse en paralelo con los otros tipos de realidades.

4.1.5 Realidad aumentada basada en geolocalización

Este tipo de realidad aumentada se basa en identificar la realidad de objetos o indicadores mediante el posicionamiento en el espacio empleando coordenadas, ya sean proporcionadas con GPS o Beacon. La limitación que presenta este tipo de realidad es que, al no identificar elementos físicos, su uso se reduce a emplearlo como “mapa aumentado” o a ser usado como complemento de alguna de los otros tipos de realidad.

4.2 Técnicas de visualización

Las técnicas de visualización las podemos sintetizar en tres siendo estas las principales para mostrar la realidad aumentada.

4.2.1 Gafas de realidad aumentada.

Las gafas de realidad aumentada, visor o HMD (head-mounted- display) se utiliza tanto para mostrar las imágenes procedentes del mundo físico como los objetos virtuales sobre vista actual. Las gafas disponen de un sensor que detecta el movimiento de las gafas por lo que no es necesario que esté conectado a un ordenador. Este sensor permite al sistema añadir la información virtual dentro de mundo físico. Como principal ventaja destaca la integración de la información virtual dentro del mundo físico para el usuario. La información gráfica está condicionada a la vista de los usuarios.

4.2.2 Pantalla de mano

Se necesita un dispositivo informático que incorpore una pantalla pequeña que cabe en la mano de un usuario. Todas las soluciones utilizadas en dispositivos de mano, han empleado técnicas de superposición sobre la información gráfica. En un principio, los dispositivos de mano empleaban sensores tales como magnetoscopios y GPS que añadían marcadores al video. Y en la actualidad sistemas como ARToolkit y sistemas de visión como SLAM o PTAM.

La pantalla de mano promete ser la que lleve a todo el público la realidad aumentada. Dado el carácter portátil de los dispositivos y posibilidad de ser integrada en los smartphones.

4.2.3 Proyección espacial

Para esta técnica de proyección espacial (SAR), se hace uso de proyectores digitales para mostrar la información gráfica sobre los objetos físicos. Lo que conlleva que la pantalla está separada de los usuarios, permitiendo a grupos de usuarios utilizarlo a la vez y coordinar el trabajo entre ellos.

SAR plantea varias ventajas frente a los otros tipos de visualización, y es que el usuario no está obligado a llevar ningún tipo de equipo, ni gafas ni pantallas. Permitiendo que los usuarios puedan verse las caras y tengan las manos libres, convirtiendo a SAR en un buen candidato para realizar trabajos colaborativos.

Otro inconveniente que es solventado mediante el uso de SAR es el tamaño de visualización que se reduce al tamaño de los visores empleados en las otras técnicas. Mientras que el sistema de proyección permite incorporar más proyectores para ampliar el área de visualización, permitiendo mostrar un mayor número de superficies virtuales a la vez en un entorno interior.

4.3 Aplicaciones de la realidad aumentada

La realidad aumentada se ha explotado en muchas aplicaciones, desde juegos y entretenimientos hasta medicina, educación y negocios. En este apartado describiremos algunos ejemplos de uso de la realidad aumentada.

4.3.1 Arqueología

La AR se ha utilizado para ayudar a la investigación arqueológica. Al aumentar las características arqueológicas del paisaje moderno, permitiendo la AR a los arqueólogos formular posibles configuraciones de sitios a partir de estructuras existentes. Por ejemplos sistemas como VITA (Visual Interaction Tool for Archaeology) permitirá a los usuarios imaginar e investigar resultados de excavaciones instantáneas sin salir de su hogar. Cada usuario puede colaborar mutuamente “navegando, buscando y viendo datos”.



Figura 4.1 – Sistema VITA

Fuente: YouTube.com

Los sistemas de AR colaborativos proporcionan interacciones multimodales que combinan el mundo real con imágenes virtuales de ambos entornos.

AR también se ha adoptado recientemente en el campo de la arqueología subacuática para apoyar y facilitar de manera eficiente la manipulación de artefactos arqueológicos.

4.3.2 Arquitectura

AR puede ayudar a visualizar proyectos de construcción. Las imágenes generadas se pueden superponer a una vista de la vida real antes de que el edificio físico se construya allí. También como demostró Trimble Navigation, también se pueden emplear dentro del espacio de trabajo de un arquitecto, visualizando animaciones en 3D de sus planos en 2D.

Con las mejoras en la precisión del GPS, las empresas pueden usar la AR para visualizar modelos georreferenciados de sitios en construcción, estructuras subterráneas, cables y tuberías utilizando dispositivos móviles. Se aplica para presentar nuevos proyectos, resolver problemas de construcción en el sitio. Un ejemplo es el que presenta DAQRI, mediante el uso de un casco con Android que se usa para crear realidad aumentada para el trabajador en la industria, que incluye instrucciones visuales, alertas en tiempo real y mapeo 3D.



Figura 4.2 – Casco DAQRI

Fuente: falproyectos.com

Otro ejemplo de esta aplicación es la realizada por la Universidad de Canterbury que lanzó CityViewAR, después del terremoto de Christchurch, que permitió a los planificadores e ingenieros de la ciudad visualizar los edificios que habían sido destruidos. Esto no solamente proporciono a los planificadores herramientas para hacer referencia al paisaje urbano anterior, sino que también sirvió como recordatorio de la magnitud de la devastación resultante, ya que edificios completos habían sido demolidos.

4.3.3 Educación STEM

En entornos educativos, AR se ha utilizado para complementar un plan de estudios estándar. Los libros de texto, tarjetas de vocabulario y otros materiales educativos pueden contener activadores que cuando son detectados por un dispositivo AR, producen información suplementaria para el alumno en formato multimedia.

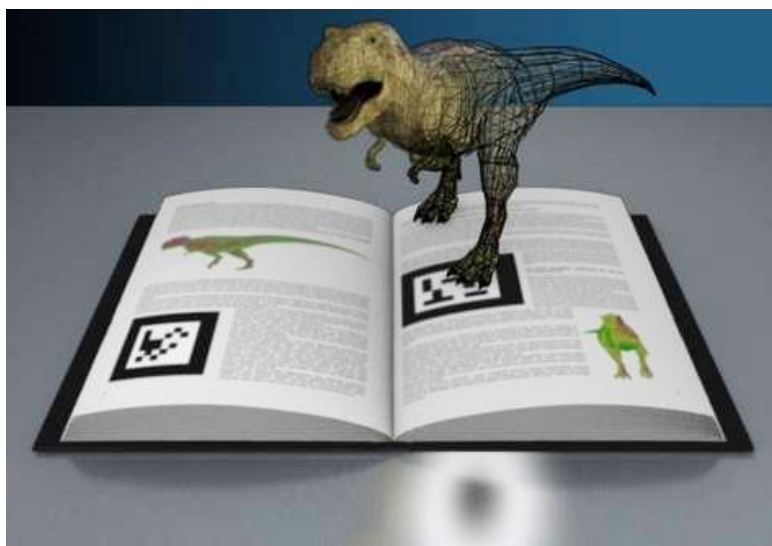


Figura 4.3 – Realidad Aumentada en la educación

Fuente: informaikta.blogspot.com

A medida que AR evoluciona, los estudiantes pueden participar interactivamente e interactuar con el conocimiento de manera más auténtica. En lugar de permanecer como receptores pasivos, los estudiantes pueden convertirse en aprendices activos, capaces de interactuar con su entorno de aprendizaje. Las simulaciones creadas por ordenador de eventos históricos permiten a los estudiantes explorar y aprender detalles de cada área significativa del evento.

En la educación superior, existen sistemas como Construct3D que permite a los estudiantes aprender conceptos de ingeniería mecánica, matemática o geometría. Los estudiantes de química pueden visualizar e interactuar con la estructura espacial de una molécula usando un objeto marcador sostenido en la mano. Otros han utilizado HP Reveal, para crear tarjetas de notas AR para estudiar los mecanismos de la química orgánica para

crear demostraciones virtuales de cómo utilizar la instrumentación de laboratorio. Y los estudiantes de anatomía pueden visualizar diferentes sistemas del cuerpo humano en tres dimensiones.

4.3.4 Comercio

AR se utiliza para integrar marketing de impresión y video. El material de marketing impreso puede diseñarse con ciertas imágenes como activadores que, cuando son escaneadas por un dispositivo con la app de AR correspondiente activan una versión de video del material promocional. Una diferencia entre la realidad aumentada mediante marcador y el reconocimiento directo de imágenes es que uno puede superponer múltiples medios al mismo tiempo en la pantalla de visualización, como los botones para compartir en las redes sociales, el video en la página, incluso el audio y los objetos 3D. Las publicaciones impresas tradicionales utilizan la realidad aumentada para conectar diferentes tipos de medios. También puede mejorar las vistas previas de los productos, como permitir que un cliente vea lo que se encuentra dentro del paquete de un producto sin abrirlo.

La realidad aumentada es cada vez más utilizada para publicaciones Web, y los comercios ofrecen la posibilidad de cargar una imagen y “probarse” varias prendas superpuestas en la imagen. Aún más compañías como Bodymetrics instalan cabinas de vestir en grandes almacenes que ofrecen un escaneo de cuerpo completo. Estas cabinas representan un modelo tridimensional del usuario, lo que le permite probarse diferentes atuendos sin la necesidad de cambiarse de ropa físicamente. Un ejemplo de estas cabinas las podemos ver en la Figura 4.4.



Figura 4.4 – Escáner biométrico Bodymetrics

Fuente: Bodymetrics

También la AR es utilizada por los comercios de muebles como IKEA, Wayfair, que permiten a los consumidores ver sus productos en su hogar antes de comprar cualquier cosa.

4.3.5 Literatura

La primera descripción de AR como se la conoce hoy fue en *Virtual Light*, la novela de 1994 de William Gibson. En 2011, AR fue mezclado con poesía por Ni Ka de Sekai Camera en Tokio. La prosa de estos poemas AR proviene de Paul Celan, *Die Niemandsrose*, que expresa las secuelas de terremoto y tsunami de Tōhoku.

4.3.6 Arte visual

La AR aplicado en las artes visuales permite que los objetos o lugares desencadenen experiencias artísticas multidimensionales e interpretaciones de la realidad.

La realidad aumentada puede ayudar en la progresión del arte visual en los museos al permitir a los visitantes del museo ver obras de arte en galerías de forma multidimensional a través de las pantallas de sus smartphones. Esto permite a las personas ver aspectos ocultos e información sobre las pinturas, y también poder tener una experiencia tecnológica interactiva con obras de arte.

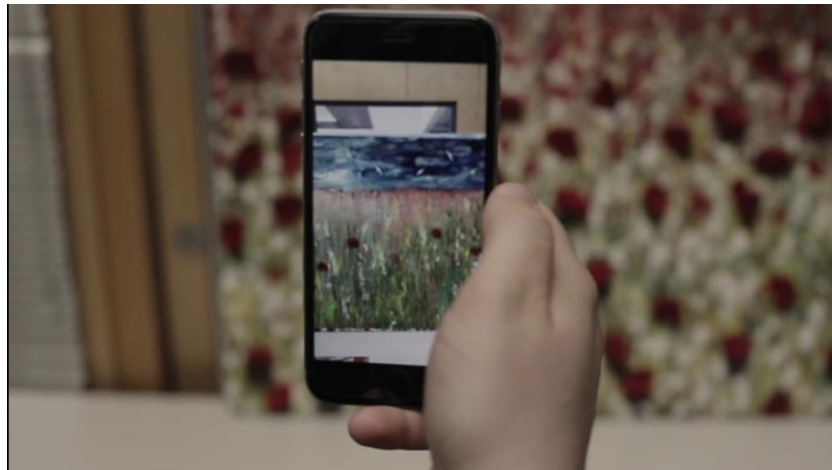


Figura 4.5 – Exposición de arte con RA con obras de Scarlett Raven

Fuente: stampaprint.net

La tecnología AR también se usó en dos de las piezas de arte público en la exposición *Desert X* de 2019.

La tecnología AR ayudó al desarrollo de la tecnología de seguimiento ocular para traducir los movimientos oculares de una persona discapacitada en dibujos en una pantalla.

4.3.7 Videojuegos

Aunque la industria de los videojuegos ha sido el adalid de la realidad virtual también adoptó la tecnología AR para desarrollar algunos juegos para entornos interiores,

como el air hockey AR, Titans of Space y los juegos de mesa como el billar mejorados por AR.

La realidad aumentada permitió a los jugadores experimentar el juego digital en un entorno del mundo real aumentada. Niantic lanzó el juego móvil de realidad aumentada Pokémon Go. Disney se ha asociado con Lenovo para crear el juego de realidad aumentada Star Wars: Jedi Challenges que funciona con un casco Lenovo Mirage AR, un sensor de seguimiento y un controlador Lightsaber.



Figura 4.6 – Render del juego Star Wars: Jedi Challenges

Fuente: StarWars.com

4.3.8 Práctica Sanitaria

Desde 2005, un dispositivo llamado near-infrared vein finder (buscador de venas infrarrojo cercano), graba las venas subcutáneas, procesa y proyecta la imagen de las venas sobre la piel se ha utilizado para localizar las venas. AR proporciona a los cirujanos datos de monitoreo del paciente al estilo de la pantalla de visualización de un piloto de caza y permite superponer registros de imágenes del paciente. Los ejemplos incluyen una vista de rayos X virtual basada en tomografía previa o en imágenes en tiempo real de sondas de ultrasonido y microscopía confocal. Visualizando la posición de un tumor en el video de un endoscopio, o riesgos de exposición a la radiación de los dispositivos de imágenes de rayos X. También puede mejorar la visualización de un feto dentro del útero materno. La realidad aumentada y otras utilidades informáticas similares se están utilizando para capacitar a profesionales médicos.



Figura 4.7- Dispositivo Near-infrared vein finder

Fuente: YouTube.com

4.3.9 Navegación

El X-38 de la NASA voló utilizando un sistema de visión sintética híbrida que superpuso los datos del mapa en video para proporcionar una navegación mejorada para la nave espacial durante las pruebas de vuelo del 1998 a 2002. Utilizó el software LandForm que fue útil en momentos de visibilidad limitada, incluido un ejemplo, cuando la ventana de la cámara de video se congeló dejando a los astronautas confiar en las superposiciones de mapas. También se probó en el Campo de Pruebas Yuma del ejército en 1999. En la Figura 4.1 se pueden ver los marcadores del mapa que indican pista, torre de control de tráfico aéreo, calles de rodaje y hangares superpuestos.

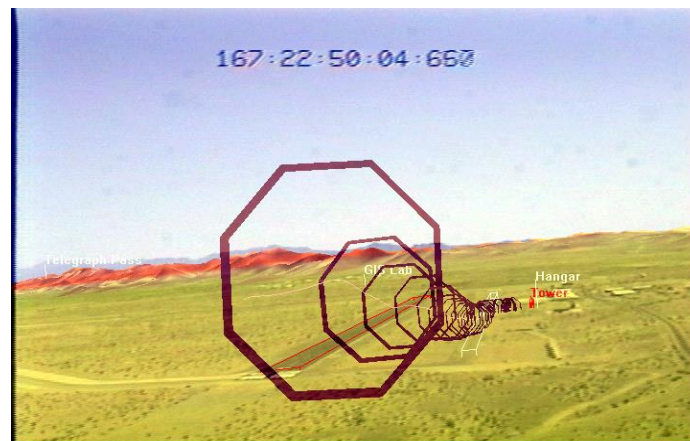


Figura 4.8 – Sistema de visión sintética híbrida

AR también puede aumentar la efectividad de los dispositivos de navegación. La información se puede mostrar en el parabrisas de un automóvil que indica las direcciones de destino, velocímetro, el clima, el terreno, las condiciones de la carretera y la información del tráfico, así como alertas de posibles peligros en su camino. A bordo de las

embarcaciones marítimas, AR puede permitir a los observadores de puentes monitorear continuamente información importante como el rumbo y velocidad de un barco mientras se mueve por el puente o realiza otras tareas.

4.3.10 Entornos industriales

En los entornos industriales, la realidad aumentada está demostrando tener un impacto sustancial con más presencia cada vez, en todos los aspectos del ciclo de vida del producto, desde el diseño del producto y la introducción de nuevos productos hasta la fabricación, el servicio y el mantenimiento.

Las líneas de montaje se beneficiaron del uso de AR. Además de Boeing, BMW y Volkswagen son conocidos por incorporar esta tecnología en las líneas de ensamblaje para monitorear las mejoras del proceso. Las máquinas grandes son difíciles de mantener debido a sus múltiples capas o estructuras. AR permite a las personas mirar a través de la máquina como si fuera una radiografía, señalando el problema de inmediato.



Figura 4.9 – Realidad Aumentada en entorno industrial

Fuente: neosentec.com

En la Harvard Business Review, sostienen que estas tecnologías aumentan la productividad al hacer que los trabajadores sean más cualificados y eficientes, y por lo tanto tienen el potencial de generar más crecimiento económico y mejores empleos.

4.3.11 Transmisión y eventos en vivo

En las previsiones meteorológicas en televisión fueron las primeras en emplear la realidad aumentada. Ahora es común que en la transmisión se muestren videos en movimiento de imágenes capturadas en tiempo real desde múltiples cámaras y otros dispositivos de imágenes. Junto con los símbolos gráficos en 3D y mapeados a un modelo geoespacial virtual común, estas visualizaciones animadas constituyen la primera aplicación verdadera de AR en televisión.

AR también se ha vuelto común en transmisiones deportivas, como puede ser la línea amarilla “primer down” en las transmisiones de fútbol americano, o la línea de fuera de juego en las del fútbol o la línea para indicar la posición del poseedor del récord actual en las transmisiones de natación o el porcentaje de acierto de los jugadores de baloncesto. También se utilizar para mostrar anuncios superpuestos en la vista del área de juego.



Figura 4.10 – Porcentaje de acierto de tiro con AR

Fuente: marketingregistrado.com

4.3.12 Turismo

Los viajeros pueden usar AR para acceder a información en tiempo real sobre una ubicación, sus características y comentarios o contenido proporcionado por visitantes anteriores. Entre las primeras aplicaciones que emplearon la realidad aumentada para el turismo, destaca Wikitude AR Travel Guide. Las aplicaciones AR avanzadas incluyen simulaciones de eventos históricos, lugares y objetos renderizados en el paisaje.



Figura 4.11 – Aplicación Wikitude AR Travel Guide

Fuente: wikitude.com

4.3.13 Traducción

Los sistemas de AR, como Word Lens, pueden interpretar el texto extranjero en signos y menús y en la vista aumentada de un usuario, volver a mostrar el texto en el idioma del usuario. Las palabras habladas de un idioma extranjero se pueden traducir y mostrar en la vista del usuario como subtítulos.



Figura 4.12 – Ejemplo del sistema Word Lens

Fuente: gizmodo.com

4.4 Empresas que invierten en AR

En este apartado vamos a describir algunas de las empresas más importantes que actualmente se encuentran invirtiendo y desarrollando dispositivos o software para aplicaciones de AR.

4.4.1 Microsoft

Microsoft dispone de Microsoft HoloLens Development Edition dispone de dos versiones de su equipo holográfico, la segunda versión presentada en febrero del 2019, se trata de un equipo totalmente autónomo y que permite interactuar con hologramas de alta definición en su entorno. La tecnología de rastreo que emplea es una evolución de la empleada en Kinect.

Y entre otras aplicaciones ya desarrolladas destacan, CAE VimedixAR que permite la capacitación inmersiva basada en simulación mediante ultrasonidos y educación anatómica a través de la realidad aumentada. Y Fragments, un juego de aventuras y suspense en el cual se mezcla la realidad aumentada y la realidad virtual. Además, se cuenta con una versión de la aplicación Skype, en la cual el usuario de la aplicación de PC o móvil vera la vista que está viendo el usuario de HoloLens y el usuario de HoloLens vera la vista capturada por la cámara del usuario del PC dispositivo móvil. Y en noviembre del 2018 se anunció que Microsoft está preparado HoloLens para incorporar la tecnología de AR al repertorio de armas de los soldados estadounidenses.

4.4.2 Google

Google ha tenido varios proyectos relacionados como el extinto Tango, pero sus desarrollos están actualmente concentrados en la plataforma ARCore.

ARCore es la plataforma de Google para construir experiencias de AR, mediante diferentes API, ARCore permite que su teléfono detecte el entorno, comprenda el mundo e interactúe con la información.

ARCore emplea tres capacidades fundamentales para integrar contenido virtual en el mundo real y como se ve a través de la cámara del teléfono.

- Seguimiento de movimiento, permite que el smartphone entienda y rastree su posición en relación con el mundo.
- Compresión ambiental, que permite que detecte el tamaño y la ubicación de las superficies.
- Estimación de luz, permitiendo que estime las condiciones de iluminación del entorno.

Como pequeño inconveniente se puede indicar que ARCore está diseñado para ser utilizado en smartphones que cuenten con una versión Android 7.0 o posterior, que contrasta con su antecesor Tango que estaba diseñado para ser empleado en todas las versiones.

Google también contaba con un dispositivo de visualización tipo gafas de realidad aumentada, se trataba de las Glass presentada en 2012, pero en 2015 la compañía decidió no continuar con el proyecto cancelando la venta masiva al público comercial y sin dar más información, si en un futuro presentara algún otro tipo de lentes.

4.4.3 Apple

Apple invierte y desarrolla en la realidad aumentada mediante su propia plataforma ARKit, que combina el seguimiento de movimiento del dispositivo, la captura de escenas de la cámara, el procesamiento avanzado de escenas y las comodidades de visualización para simplificar la tarea de crear una experiencia AR. Esta plataforma ha contado con varias versiones y a la fecha de la redacción se encuentra en su versión 3.

De igual manera que ocurre que otros casos, para poder usar las aplicaciones generadas con esta plataforma, los dispositivos necesitan unos requisitos mínimos, que en este caso son: iOS 11.0 o superior y un procesador A9 o posterior.

5 DESARROLLO

En el siguiente capítulo se centra el desarrollo de la aplicación. Para llegar a tener una aplicación adecuada al objetivo de este TFG, se propone la siguiente metodología (Figura 5.1) a seguir en su desarrollo la cual está formada por una serie de etapas que serán claves para tal fin:

- 1) Se investigará el mercado en busca de aplicaciones que realicen una tarea similar a la que desempeñará nuestra aplicación.
- 2) Se indicarán los requisitos o módulos que se pretenden desarrollar para formar parte del sistema. Es importante definir las funcionalidades que se quieren tener, antes de comenzar a programar. Lo que requiere de estrategia de estructuración y división del sistema en distintas partes.
- 3) Serán investigadas y analizadas las herramientas de desarrollo disponibles, indicando cual se empleará para cada parte. Justificando el porqué de dicha elección.
- 4) Se explicará el funcionamiento de la plataforma elegida y como se va a adaptar a la implementación de esta aplicación.
- 5) Dada la sobriedad que tendrá la aplicación se seleccionarán los componentes de la API que sean acordes a esto.
- 6) Se determinará la lógica del sistema. Una vez se hayan decidido las funcionalidades con las que contará la aplicación, para ello es necesario hacer un estudio de la lógica antes de comenzar a programarlo. Con esto se conseguirá una buena integración e interconexión entre las distintas partes que se van a construir. Se pretende conseguir robustez, evitando posibles bucles o caminos sin salida debidos a la falta de interconexión entre las distintas partes.
- 7) Por último, se procederá a programar dicha aplicación. Se indicará el editor de código fuente utilizado y algunas de sus características. Se explicará el menú que presentará la aplicación final ofrecida al usuario.

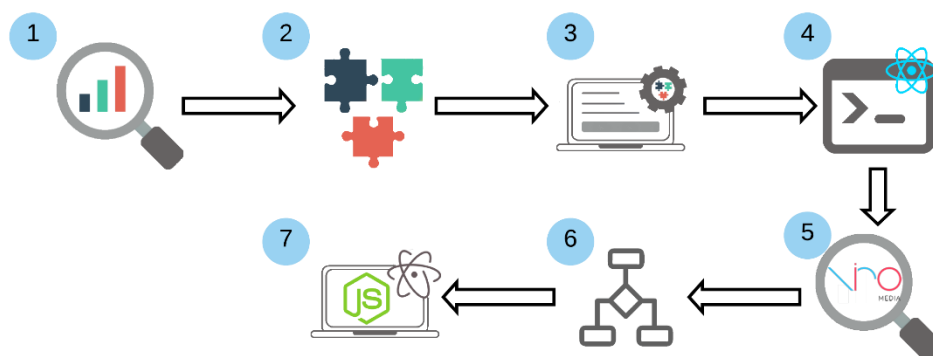


Figura 5.1 – Metodología para el desarrollo de la aplicación

5.1 Aplicaciones similares en el mercado

La realidad aumentada como ya hemos visto tiene muchas aplicaciones distintas, y entre ellas podemos encontrar la del turismo y la de la enseñanza. Y estas dos cualidades las podemos encontrar unificadas en los museos. Así que en este punto vamos a investigar el mercado para ver que aplicaciones emplean la realidad aumentada en los sectores del turismo, educación y en guías para museos en concreto.

Entre ellas podemos destacar las siguientes:

5.1.1 Skin & Bones

Se trata de una aplicación de realidad aumentada que corresponde al Museo Nacional de Historia Natural de Smithsonian. En ella se emplea la realidad aumentada para ver una representación sobre los esqueletos de los animales que se encuentran en el museo y así poder ver como serían en vida, además también aporta alguna información relativa al animal correspondiente. Un ejemplo de ellos lo podemos ver en la figura 5.1.

Se trata de una aplicación gratuita que solamente se encuentra disponible para dispositivos iPhone.



Figura 5.2- Ejemplo aplicación Skin and Bones.

Fuente: naturalhistory.si

5.1.2 England Originals

La aplicación England Originals, cuenta las historias de los sitios patrimoniales más simbólicos de Inglaterra. Emplea la realidad aumentada para mostrar nuevos conocimientos sobre 12 ubicaciones históricas. Emplea tanto el reconocimiento de imagen como la localización por GPS.

El contenido aumentado está formado por imágenes y texto, como podemos ver en la figura 5.2.

Se trata de una aplicación gratuita que está disponible tanto para Android como para iPhone.



Figura 5.3 – Ejemplo aplicación England Originals

Fuente: Englandoriginals.co.uk

5.1.3 Museo Carlos V

Se trata de la aplicación del Museo Carlos V, que emplea la realidad aumentada para mostrar información sobre algunas de las obras. Emplea reconocimiento de imágenes, pero empleando como activadores marcadores que se encuentran junto a las obras que disponen de este contenido. La información que muestra son tarjetas de texto que indican algunas curiosidades sobre el personaje o paisaje representado en la obra. También cuenta con un juego de preguntas y respuestas para demostrar lo que se ha aprendido durante la visita al museo,

La aplicación es gratuita y está disponible tanto para iPhone como para Android.



Figura 5.4 – Ejemplo aplicación museo Carlos V

Fuente: Museocarlosv.es

5.1.4 Sorolla Museum AR

Esta aplicación que está disponible para iPhone y Android de manera gratuita para ambas plataformas. Utiliza la realidad aumentada para conseguir que el pintor Joaquín Sorolla cobre vida y explique en primera persona algunas de sus obras, solamente algunas disponen de esa posibilidad (en el momento de la redacción solamente 10). Emplea el reconocimiento de imagen para activar el video que se reproduce sobre el lienzo del cuadro en el que Sorolla nos explica la obra, pero se necesita que el dispositivo se quede inmóvil, una vez que se ha reconocido la imagen sino desaparece el contenido y sería necesario que la aplicación volviera a reconocer la imagen.

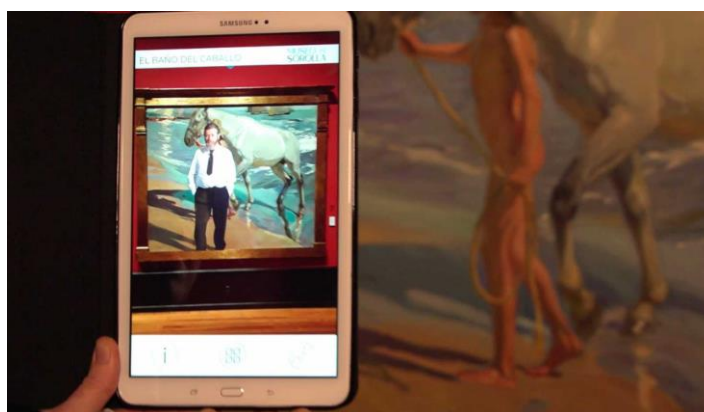


Figura 5.5 – Ejemplo aplicación Sorolla Museum AR

Fuente: Rtve.es

5.1.5 Museo Nacional de Escultura

Se trata de una aplicación gratuita realizada para el Museo Nacional de Escultura, dispone de versión para Android y para iPhone. En este caso no hace uso de la realidad aumentada, se trata de una guía del museo, pero en formato multimedia en la cual va explicando las distintas zonas del recorrido elegido. El contenido tiene que ser descargado

a parte de la aplicación, también se puede elegir si se quiere incluir contenido de accesibilidad como audio descripción, subtítulos y lengua de signos.



Figura 5.6 – Ejemplo aplicación Museo Nacional de Escultura

Fuente: Captura de pantalla de la aplicación MNE Valladolid

Como conclusiones de este apartado podemos decir que, todavía el número de museos que disponen de una aplicación que emplee la realidad aumentada es reducido, y las aplicaciones que si disponen de esta función suele venir acompañada de otros elementos de *engagement* para el usuario, como juegos, así como herramientas que permiten tomar fotografías con los personajes.

El contenido que muestran suele ser muy estático, no permitiendo interacción alguna por parte del usuario, tampoco permite adaptar el contenido que se muestra, haciendo que la experiencia se adapte al nivel del usuario. La mayoría requieren estar in situ en el lugar, evitando que estas aplicaciones pueden ser empleadas en otro lugar como podría ser la enseñanza empleando fotografías sobre las cuales generar el contenido.

También el número de obras, las cuales disponen de experiencia de realidad aumentada es muy limitado.

5.2 Requisitos del sistema

Al dividir el sistema en varios módulos o requisitos, nos aseguramos de conseguir una mejor integridad del sistema. Ya que permitirá detectar y corregir un fallo si únicamente afecta a un módulo en concreto. Además, su implementación será más fácil ya que se realizará de manera individual y después de comprobados se agruparán.

Por tanto, será necesario que el sistema cumpla con los siguientes requisitos:

- **Detección de superficies:** para conocer los planos sobre el cual se encuentra la obra.

- **Lectura de imágenes:** permitiendo que se lean las imágenes almacenadas en el servidor, que posteriormente serán comparadas con las captadas por la cámara del dispositivo.
- **Reconocimiento de imágenes:** clave fundamental si se quiere hacer una guía para un museo/galería de arte que sea capaz de reconocer las imágenes, a las que se le añadirá contenido aumentado.
- **Comparación de imágenes:** se comparará la imagen capturada por la cámara del dispositivo móvil, con todas las imágenes presentes en el servidor, para comprobar si se encuentra en él y en caso afirmativo activar la experiencia de la realidad aumentada.
- **Representación de contenido:** se representará el contenido aumentado, para de esta manera realizar la experiencia de realidad aumentada. Y que este contenido pueda ser de diferentes tipos.
- **Anclaje de información:** a la obra, permitiendo de esta manera que, aunque el usuario se desplace puede seguir observando el contenido aumentado, evitando que el usuario tenga que mantenerse inmóvil si quiere observar el contenido.
- **Interacción del usuario:** otorgar de funciones a los elementos mostrados, para que el usuario pueda interactuar con ellos, permitiéndole mostrar más o menos contenido, dependiendo de su interés.

Estos serán en un principio los requisitos mínimos que se han planteado para que el sistema tenga una buena estructura. Estos requisitos se irán añadiendo o modificando en función de las facilidades o contratiempos que vayan surgiendo durante el desarrollo.

5.3 Implementación de herramientas

En este apartado se van a analizar algunas de las herramientas, SDKs, programas, librerías, etc que se han encontrado y que podrían cubrir la mayoría de las necesidades de esta aplicación.

Para analizar las herramientas vamos a tener en cuenta que se pueda desarrollar la interfaz de usuario, que sería la aplicación tradicional, a la cual se le añadirá soporte para realidad aumentada.

Entre las posibles soluciones encontramos las siguientes duplas que cumplen los requisitos indicados en el punto anterior.

Las duplas son: iOS empleando ARKit, Android empleando ARCore, Unity empleando Vuforia y React Native empleando ViroReact.

Dado que todas las duplas **cumplen los requisitos anteriores**, vamos a emplear otros requisitos para decidir cuál de ella emplearemos. Los criterios que se van a tener en cuenta son:

- Tipo de licencia.
- Plataformas que soporta, Android, iOS o ambas.
- Sistema operativo en el que se puede desarrollar.
- Almacenamiento de marcadores en la nube.
- Aplicaciones nativas para dispositivos móviles.
- Nivel de conocimientos previos de programación sí son válidos conocimientos previos o se tiene que aprender de cero, se valorará de 0 a 5, siendo 0 no se necesitan conocimientos nuevos y 5 todo se tiene que aprender de nuevo.

Estos requisitos se verán representados en la siguiente tabla y ella nos servirá para decidir qué herramientas será la que usemos para desarrollar la aplicación.

	iOS - ARKit	Android - ARCore	Unity - Vuforia	React Native- ViroReact
Tipo de Licencia	Libre	Libre	Libre - Comercial	Libre
Plataformas que soporta	iOS	Android	Android - iOS	Android - iOS
Dispositivos para desarrollar	Mac	Windows - Mac	Windows - Mac	Windows -Mac - Linux
Almacenamiento de marcadores	Sí	Sí	Sí	Sí
Aplicaciones Nativas	Sí	Sí	No	Sí
Nivel de conocimientos previos [0 - 5]	5	4	3	3

Tabla 5.1

A la vista de los datos representados en la Tabla 5.1, se ha decidido que para realizar la aplicación se va a emplear React Native para la creación de la aplicación móvil y el soporte para la realidad aumentada será proporcionado por la plataforma ViroReact.

Los que más ha influido en la elección ha sido que no se requiere un nivel alto de conocimientos previos ya que React Native se trata de un framework de JavaScript lenguaje del cual ya se dispone de conocimiento básico, aparte de crear aplicaciones que

emplearían elementos nativos de cada plataforma, repercutiendo positivamente en el futuro rendimiento de la aplicación.

En los siguientes puntos explicaremos el funcionamiento de la misma y con cuales componentes que nos proporciona emplearemos finalmente para nuestra aplicación.

5.4 React Native – ViroReact

En este apartado se explicará lo que es React Native y ViroReact. También se indicará cual es el cometido de React Native y ViroReact dentro de la aplicación y el porqué de su elección. Además, se indicarán cuáles son los pasos necesarios para poder usarlos y desarrollar proyectos.

5.4.1 React Native

React Native es un framework de JavaScript para escribir aplicaciones móviles nativas para iOS y Android. Se basa en React, la librería JavaScript de Facebook para construir interfaces de usuario, pero orientada en este caso a plataformas móviles. Además, la mayoría del código que se escribe puede compartirse entre plataformas facilitando el desarrollo simultaneo tanto para Android como para iOS.

Las aplicaciones React Native se escriben usando una mezcla de JavaScript y etiquetas XML, conocido como sintaxis JSX. A continuación, invoca a la API de representación nativas Objective-C (para iOS) o Java (para Android), por lo tanto, la aplicación se representará utilizando componentes de interfaz de usuario reales, no vistas web, y se verá y sentirá como cualquier otra aplicación móvil.

Algunas de las ventajas que presenta emplear React Native son:

- Representa elementos nativos de las plataformas móviles, lo que le permite diferenciarse de otros métodos de desarrollo multiplataforma como Cordova o Ionic, reflejándose esto en el rendimiento.
- Dado que React Native es JavaScript “just”, no es necesario compilar por completo otra vez la aplicación para ver los cambios reflejados, en su lugar simplemente basta con guardar los cambios y actualizar la aplicación en el simulador.
- No es necesario utilizar Xcode para desarrollar para iOS o Android Studio para Android, se puede trabajar con cualquier editor de texto para la edición de JavaScript.

5.4.2 ViroReact

ViroReact es una plataforma para desarrollar aplicaciones realidad aumentada y realidad virtual usando React Native.

La plataforma admite ARKit y ARCore para el desarrollo de AR, de esta manera englobamos las dos grandes plataformas de desarrollo de realidad aumentada, bajo una misma plataforma.

ViroReact está compuesta por dos grandes componentes principales, un motor de renderizado de 3D nativo y una extensión personalizada de React para el desarrollo de AR y VR.

Algunas de las ventajas que presenta ViroReact son:

- Permite crear una aplicación desde cero o agregar características AR/VR a una aplicación ya existente.
- Posibilidad de multiplataforma móvil empleando un mismo código base.
- Al igual que React Native, permite comprobar los cambios realizados solamente con actualizar la app.
- Fácil de aprender ya que emplea un lenguaje de marcas, lo que lo hace bastante intuitivo.
- Se trata de una plataforma de código abierto, lo que proporciona que se puedan encontrar códigos que pueden ser reutilizados en nuestra aplicación.
- No es necesaria la utilización de Xcode ni Android Studio.

5.4.3 Funcionamiento e integración

Para poder hacer uso tanto de React Native como de ViroReact es necesario instalar:

- Node.js, que es un entorno en tiempo de ejecución multiplataforma, de código abierto construido con el motor de JavaScript, para la capa del servidor.
- Java SE Development Kit (SDK).
- React Native CLI se trata de la interfaz de la línea de comandos de React Native.

Los puntos anteriores son necesarios tanto para utilizar React Native, como ViroReact.

- Registro en la página Web de Viro para obtener la API KEY, su obtención es gratuita.
- Dependencias de ViroReact

Una vez instalados y realizados los pasos anteriores, ya podemos usar ambas plataformas para desarrollar nuestra aplicación. Solamente necesitaríamos un editor de texto o un IDE de nuestro gusto, en un punto posterior se comentará cual se empleará.

La integración entre React Native y ViroReact es total, ya que ambos parten de un mismo núcleo común como es React y además ViroReact fue desarrollado para integrarse con React Native, por tanto, no presentaremos dificultades para la interacción entre módulos creados mediante React Native y los creados por ViroReact.

5.5 Confrontación requisitos del sistema

En este apartado se van a confrontar los requisitos definidos para el sistema en el punto 5.2. Se indicará que funciones/componentes/librerías nos son proporcionadas por la plataforma y cuales tendrán que ser desarrolladas, para su posterior inclusión con el resto.

Recordamos cuales eran estos requisitos:

- Detección de superficies.
- Lectura de las imágenes.
- Reconocimiento de las imágenes.
- Comparación de las imágenes.
- Representación de contenido.
- Anclaje de la información.
- Interacción del usuario.

5.5.1 Detección de superficies

La detección de superficies es importante ya que ella nos permitirá detectar el plano sobre el cual se encuentra la imagen y otros planos los cuales pueden ser usados para superponer sobre ellos información aumentada.

Para realizar esta detección, la plataforma ViroReact nos proporciona un elemento llamado ViroARPlane, el cual nos permite detectar planos libremente, o planos con unas características indicadas por nosotros, tales como sus dimensiones y su alineación, vertical u horizontal.

También dispone del componente ViroARPlaneSelector, el cual permite que se pueda seleccionar el plano que se quiere usar.

Pero este componente no lo usaremos de manera directa, sino que será empleado por otras funciones, para indicar que se ha encontrado un plano en el que se podrán renderizar y anclar el contenido.

La propiedad fundamental mediante la cual se va a indicar que se ha encontrado será `onAnchorFound`, la cual también está disponible en otros muchos componentes.

5.5.2 *Lectura de imágenes*

Para realizar la lectura de las imágenes las cuales son las que sirven de referencia para después realizar la comparación, la plataforma nos proporciona el componente `ViroARTrackingTargets`. Este componente nos permite emplear las propiedades:

- `Source` en la que podemos especificar la dirección URL donde se encuentre la imagen o también podemos indicar la dirección de directorios si el archivo se encuentra en el almacenamiento local. Esta propiedad es obligatoria para esta propiedad.
- `Orientation` en la que se determina la orientación de la imagen.
- `PhysicalWidth` la cual permite especificar el tamaño de la imagen real, lo cual ayuda determinar más fácilmente el tamaño de la imagen durante el reconocimiento.

5.5.3 *Reconocimiento y comparación de imágenes*

El reconocimiento y comparación de las imágenes probablemente sean el elemento fundamental, ya que de ello dependerá que el funcionamiento sea el correcto o no. Para realizar esta función la plataforma nos proporciona el componente `ViroARImageMarker`, el cual nos proporciona la capacidad de reconocer las imágenes que captura la cámara realizando la comparación con las imágenes leídas mediante `ViroARTrackingTargets`.

Esta interconexión entre `ViroARTrackingTargets` y `ViroARImageMarker` se realiza mediante la propiedad `Target` la cual es requerida en ambos elementos, es de tipo `string` el cual identificará la imagen que se ha leído y la que se está buscando.

También dispone de otras propiedades como:

- `onClick`: es una función que permite que, al pulsar, se realice una acción programada por el desarrollador.
- `onAnchorFound`: es una función que permite anclar el contenido al plano, una vez que el plano haya sido encontrado.

5.5.4 *Representación de contenido*

La representación del contenido es la que da significado a la realidad aumentada, que se basa en incluir contenido sobre lo que se está viendo.

Este contenido puede ser de diferente tipo, 2d o 3d y dentro de la dimensión del mismo también podemos diferenciarlos dependiendo de si es texto, imagen o un objeto.

Para esto la plataforma nos proporciona una serie de componentes como pueden ser ViroImage, ViroVideo, Viro3DObject. Los componentes empleados para este cometido se especifican en el siguiente punto de esta memoria.

Estos componentes disponen entre otras propiedades de:

- Source en la que podemos especificar la dirección URL donde se encuentre la imagen o también podemos indicar la dirección de directorios si el archivo se encuentra en el almacenamiento local. Esta propiedad es obligatoria para esta propiedad.
- Animation, esta propiedad los permite indicar cómo será la forma en la que el elemento aparecerá, si será de manera inmediata o retardada o realizará algún recorrido antes de anclarse en su posición.
- onClick es una función que permite que, al pulsar sobre el elemento, se realice una acción programada por el desarrollador.
- Scale, esta propiedad permite determinar el tamaño con el que se mostrará el elemento.
- Visible, esta propiedad es de tipo booleano y nos permite indicar si el contenido será visible o no, por defecto siempre se encuentra en true indicando que sea visible.

5.5.5 Anclaje de información

El poder anclar la información, posibilita que, aunque el usuario se desplace la información se mantenga en la posición que fue definida para estar y no se desplace con el movimiento del usuario. También posibilita que la información continúe anclada, aunque se haya detectado otra imagen, permitiendo que si se vuelve a la imagen anterior no tendremos que realizar todos los pasos anteriores (detección, lectura...).

Para esto la plataforma nos proporciona la función onAnchorFound a la cual podemos acceder desde las propiedades de bastantes componentes. Entre ellos todos los componentes empleados para renderizar información, tanto imágenes, textos, videos y objetos 3d.

Esta función es la encargada de proporcionar la información de los planos en los que posteriormente se anclara la información.

5.5.6 Interacción del usuario

La interacción del usuario es un requisito fundamental y diferencial respecto a otras aplicaciones que se encuentran en el mercado. Esto permite que el usuario pueda sentirse más integrado en la experiencia de realidad aumentada.

Para realizar las interacciones la plataforma nos proporciona la propiedad onClick, que está disponible en multitud de componentes, como ViroText, ViroImage, ViroSound, ViroVideo Esta propiedad nos permite conectar con las funciones que serán creadas por nosotros. Tales como las que posibilitan el cambio del estado de una imagen, de visible a invisible, de un video de pause a play, abrir un enlace o cambiar a otro archivo donde se encuentra la experiencia específica de una obra.

Las funciones desarrolladas se programarán empleando el lenguaje JavaScript, y emplearemos la sintaxis JSX, propios de React Native.

Siguiendo el orden empleado en la ejecución de la aplicación, el cual se indica con detalle en la lógica del sistema punto 5.7. Podemos ver en el siguiente gráfico (Figura 5.7) cómo se implementan los requisitos del sistema y como se encuentran conectados entre ellos.

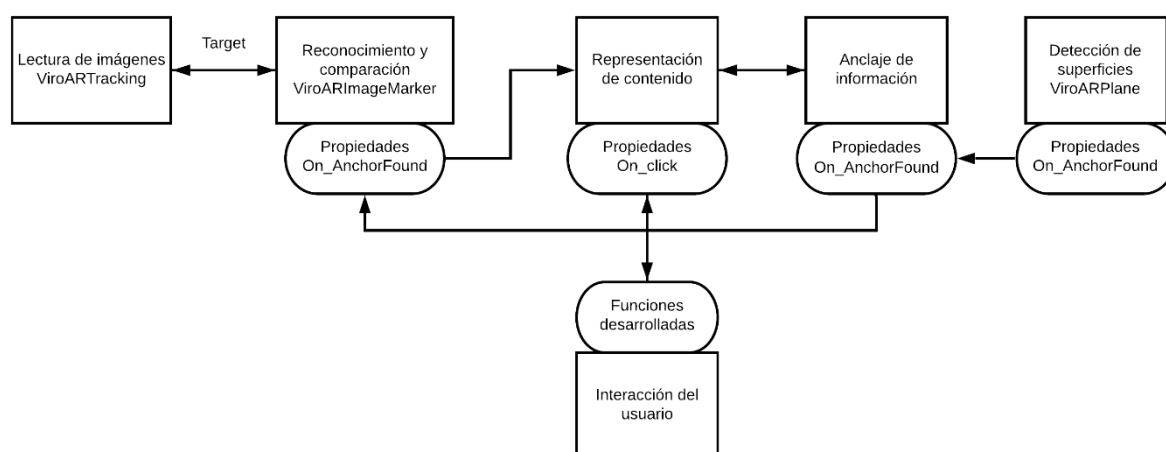


Figura 5.7 – Interconexión de los componentes

5.6 Elección de componentes

En este punto se indicarán cuáles serán los componentes fundamentales que se emplearán en nuestra aplicación, de todos los que nos proporciona ViroReact.

Tras la lectura de la documentación facilitada por ViroReact, los componentes que se van a emplear para la creación de la experiencia AR serán los siguientes:

Componentes básicos con lo que contarán todas las obras:

- **ViroARScene:** permite agrupar la lógica correspondiente a una experiencia AR y sus componentes y cambiar entre ellos utilizando ViroARSceneNavigator.
- **ViroARSceneNavigator:** es el punto de entrada para aplicaciones AR con Viro.
- **ViroNode:** genera un nodo 3D vacío en la escena. Las transformaciones establecidas sobre él (posición, rotación, escala) se aplican a todos los elementos hijos.
- **ViroFlexView:** es un componente que permite combinar varios componentes 2D para crear diseños que se pueden posicionar en el espacio 3D.
- **ViroARTrackingTargets:** contiene la información necesaria para que los componentes de seguimiento de AR, como ViroARImageMarker funcionen correctamente.
- **ViroARImageMarker:** es un componente que intenta encontrar el objeto dado en la vista del usuario.
- **ViroText:** es un componente que proporciona funcionalidades de texto 2D.
- **ViroImage:** un componente utilizado para mostrar imágenes en 2D.
- **ViroAnimations:** se usa para crear animaciones reutilizables que se aplican a un componente ViroReact.

Componentes que solamente se emplearán para algunas obras en función de si se dispone del material adecuado para incluirlos en la experiencia.

- **ViroSound:** un componente que permite al usuario reproducir y controlar efectos de sonido mono y estéreo.
- **ViroVideo:** componente que proporciona una superficie 2D para reproducir video. También permite el representar videos parcialmente transparentes sobre cualquier escena VR, AR o 3D, y para escena AR esto permite experiencias tipo “holograma”. Lo consigue aplicando un filtro de croma eliminando todos los colores del video que se encuentre dentro de un rango epsilon.
- **Viro360Video:** un componente que muestra el video en una esfera de 360° rodea al usuario.
- **Viro360Image:** un componente que muestra la fotografía en una esfera de 360° que rodea al usuario.

- **ViroPortal**: un componente que representa la forma de entrada en un **ViroPortalScene**, se debe proporcionar un objeto Viro3DObject.
- **ViroPortalScene**: un componente que permite configurar una subescena solo renderizada y accesible mediante el uso de un componente ViroPortal.
- **Viro3DObject**: un componente que permite mostrar un objeto 3D que se coloca en el espacio virtual.

Se puede encontrar información más detallada y ejemplos de uso de cada componente que van a ser empleados, y de los demás componentes que conforman la API de ViroReact. Esta información se puede consultar en el enlace que está disponible en la bibliografía [26]. También podemos observar en la Figura 5.8 la extensa lista de componentes que conforman esta API.

API REFERENCE	
Viro360Image	ViroMaterials
Viro360Video	ViroMaterialVideo
Viro3DObject	ViroNode
ViroAnimatedImage	ViroOmniLight
ViroAmbientLight	ViroOrbitCamera
ViroAnimatedComponent	ViroParticleEmitter
ViroAnimations	ViroPolygon
ViroARImageMarker	ViroPolyline
ViroARObjectMarker	ViroPortal
ViroARTrackingTargets	ViroPortalScene
ViroARPlane	ViroQuad
ViroARPlaneSelector	ViroScene
ViroARScene	ViroSceneNavigator
ViroARSceneNavigator	ViroSkyBox
ViroBox	ViroSound
ViroButton	ViroSoundField
ViroCamera	ViroSpatialSound
ViroConstants	ViroSphere
ViroController	ViroSpinner
ViroDirectionalLight	ViroSpotLight
ViroFlexView	ViroText
ViroGeometry	ViroVideo
ViroLightingEnvironment	ViroVRSceneNavigator
ViroImage	Viro3DSceneNavigator
	Styles
	PolarToCartesian
	IsARSupportedOnDevice

Figura 5.8 – Listado componentes API ViroReact

Nota: Para posicionar los componentes en el espacio 3D se emplea el sistema de coordenadas cartesianas para 3D, teniendo los ejes las posiciones que se muestran en la Figura 5.9, estas coordenadas se aplican pueden aplicar tanto a los elementos padres como hijos. También los componentes se pueden rotar en cada uno de los ejes, ocasionando que para ese componente los ejes de coordenadas se encuentren en otra posición debida a esa rotación, las rotaciones siguen el criterio “hand-right” siendo los ángulos positivos en el sentido hacia donde apuntan los dedos al cerrar la mano.

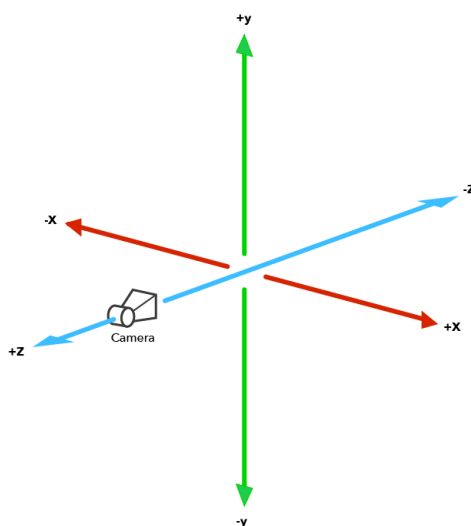


Figura 5.9- Sistema de referencia de coordenadas cartesianas

Fuente: Documentación ViroReact

5.7 Lógica del sistema

Conocidos ya los componentes y los requisitos, ya podríamos empezar a programar. Pero antes, es necesario determinar la lógica del sistema que debe seguir para conectar los distintos módulos y que formen en su conjunto el sistema que se ha desarrollado. Y para su correcto funcionamiento y que el sistema no conduzca a bucles

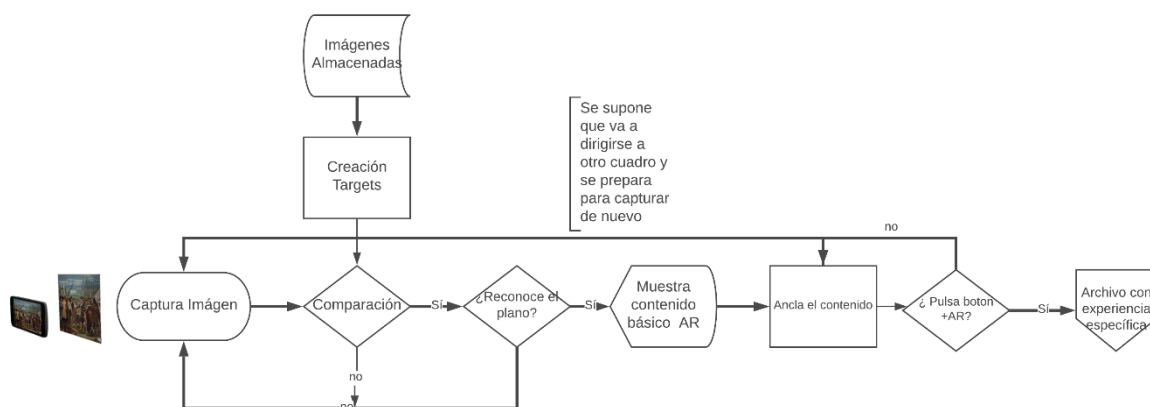


Figura 5.10 – Lógica del sistema – experiencia básica.

infinitos, caminos sin salida, o a módulos inaccesibles. Para ello la lógica propuesta es la que se muestra en las Figura 5.10 y 5.11

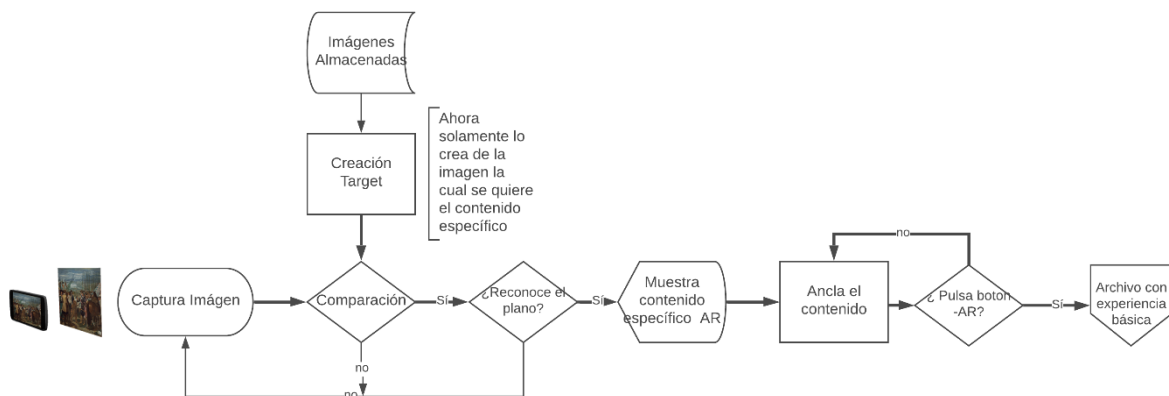


Figura 5.11 – Lógica del sistema – experiencia específica

- I. El dispositivo móvil, capta la información visual mediante la cámara, si la información captada coincide con alguna de las imágenes de las cuales disponemos de su “target”, sino coincide sigue a la espera y captando por si en el futuro se produce la coincidencia.
- II. Si la comparación ha sido exitosa, se procede a detectar el plano en el que se encuentra, sino es capaz de detectarlo volvería a estar a la espera de capturar imágenes. Pero si el plano se detecta correctamente, se pasa a renderizar la información del contenido básico de AR, y esta se anclaría al plano anteriormente identificado.
- III. Entre el contenido renderizado se encuentra un botón (+AR) el cuál, sino es pulsado no ocurre nada y el sistema mantiene la información anclada y se vuelve a mantener a la espera de su captura nuevo contenido que sea coincidente con los targets correspondientes a otras obras. Pero si el botón es pulsado entonces se pasaría a otro archivo en el cual se encuentra la experiencia específica para esa obra.
- IV. La lógica empleada para la experiencia específica es muy similar a la básica y su funcionamiento es homologo al explicado anteriormente con los cambios de que en este caso solamente se dispone del target correspondiente a una obra y en este caso de pulsar el botón ahora denominado -AR el cambio de archivo se produciría hacia el correspondiente a la experiencia básica.

Con esto, queda determinada la lógica que tendrá el sistema y la interconexión entre los módulos, por tanto, se puede comenzar con la programación.

5.8 Programación del sistema

Una vez diseñado el sistema completo, explicado y detallado cada módulo, podemos comenzar con la programación. Se debe programar una aplicación con la que el usuario pueda interactuar y que cuente con todos los requisitos planteados. En los siguientes apartados explicaran los elementos que conforman el proyecto. Se explicará el funcionamiento de las funciones. Además, se describirá la interfaz de la aplicación final ofrecida al usuario.

5.8.1 Editor de código fuente empleado

Para programar la aplicación, se va a utilizar JavaScript mediante el framework React Native, y también se empleará la plataforma ViroReact la cual también emplea JavaScript, el cual presenta una serie de ventajas que fueron indicadas en puntos anteriores.

Una de esas ventajas era que no era necesario el uso de un entorno de desarrollo específico, así que se ha decidido emplear el editor de código fuente Atom, el cual es de licencia libre y gracias a la comunidad dispone de muchos paquetes que aumentan sus capacidades, como el poder autocompletar texto para determinados framework como es el caso de React Native.

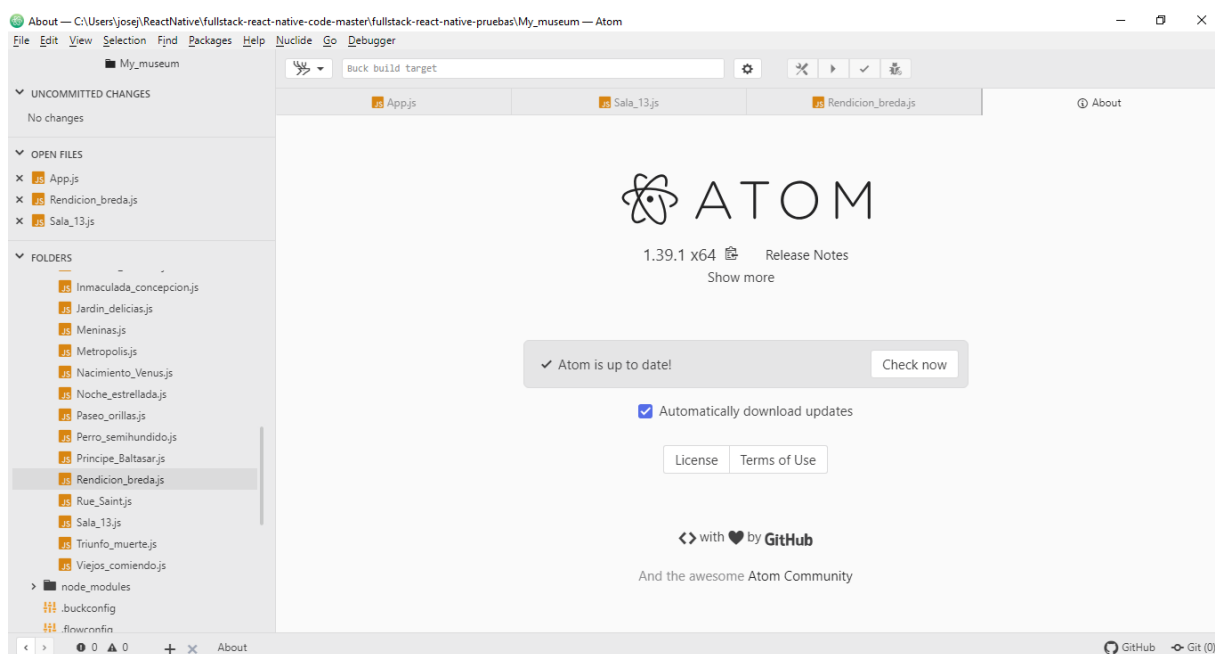


Figura 5.12 – Editor de código fuente ATOM

5.8.2 Proyecto

Para la creación del proyecto vamos a seguir las indicaciones de la documentación de ViroReact, las cuales nos crean un proyecto de ejemplo, en el cual sustituiremos los archivos de la experiencia de prueba por los que crearemos.

La creación la realizaremos desde la consola de comandos escribiendo los siguientes comandos en el mismo orden que se muestran:

```
react-native init My_museum --version=0.59.3  
cd My_museum  
npm install -S -E react-viro
```

Al realizar el proyecto de esta manera nos ahorramos de crear todo el árbol de directorios necesarios para una aplicación. El proyecto creado para el desarrollo y programación de la aplicación lo podemos dividir en los siguientes componentes:

- Archivos externos: corresponden a los archivos librerías y directorios que han sido creadas por defecto con la creación del archivo.
 - React: un directorio que contiene los archivos que nos permite dentro de nuestro código el importar los componentes de React que nos sean necesarios utilizar. En nuestro proyecto solamente importaremos “Component” que es el componente principal para crear aplicaciones con React, y de él heredan los demás componentes.
 - React-native: un directorio que contiene los archivos necesarios para poder importar los componentes de React Native que no sean necesarios para la creación de la aplicación y caracterización de la interfaz de usuario.
 - React-viro: un directorio que contiene los archivos con las funciones de la plataforma ViroReact para que puedan ser usadas para crear nuestras experiencias de realidad aumentada.
- Archivos propios: son los archivos de código fuente que han sido programados para desarrollar el sistema. Se encargan de lanzar la aplicación móvil y de crear las experiencias de realidad aumentada.

- App.js : es el primer archivo que se abre cuando se inicia la aplicación, en él se indica la API Key específica para nosotros que nos fue proporcionada tras el registro en la página de Viro media, y también se indica cual será la primera escena que se ejecutara, es decir, cuál será el primer archivo/experiencia AR en ejecutarse.
- Sala_13.js: este archivo contiene la primera experiencia que se ejecuta tras App.js y en él se encuentran todos los componentes que forman la experiencia básica de AR. Y la mayoría del tiempo de uso de la aplicación se encontrará ejecutando este archivo, así que lo podríamos considerar como el más importante de los que se han programado, para esta aplicación.
- Nombre_cuadro.js: existen una serie de archivos los cuales están nombrados con el nombre de la obra, de la cual contienen la programación de todos los componentes correspondientes a la experiencia de realidad aumentada específica para cada obra. Estos archivos se encuentran conectados con Sala_13.js y viceversa mediante los botones -AR y +AR respectivamente.

5.8.2.1 App.js

Como ya se indicó anteriormente este es el archivo principal y el primero que se ejecuta. En él se indica la API Key que identifica al usuario que está desarrollando la aplicación permitiéndole poder usar los componentes de la plataforma.

También se configura cual será la primera escena que se ejecutará, para esto se emplea el componente ViroARSceneNavigator, normalmente estas escenas se encuentran en distintos archivos, por lo que este archivo usualmente suele tener pocas líneas.

A continuación, podemos ver un fragmento del código empleado en nuestra aplicación.

```

...
var apiKey = "4B097EB2-7835-48E7-992C-EA3AF0DAD660";

var arScenes = {
  'My_museum_card' : require('./js/Sala_13.js'),
}

var My_museum_class = createReactClass({
  render: function() {
    return (
      <ViroARSceneNavigator
        initialScene={{
          scene: arScenes['My_museum_card'],
        }}
        apiKey={apiKey} />
    );
  }
});
...

```

5.8.2.2 Sala_13.js

En este archivo es donde se encuentra la programación relativa a la experiencia básica.

Como todos los archivos tiene una primera parte en la que se importan las librerías para poder usar los componentes incluidos en estas librerías en este archivo.

También definimos una serie de estados, estos estados son unas variables de tipo booleano del cual disponen todos los componentes, que normalmente se emplean su valor por defecto, pero cuando queremos que este estado se vea modificado tenemos que definirlo, como podemos ver en el siguiente fragmento de código.

```

state = {
  isTracking: false,
  initialized: false,
  runAnimation: false
}

```

Estos estados que están inicializados como falso, no cambian su estado hasta que no se inicia la aplicación y detecta alguna superficie cambiara el estado isTracking a verdadero y al mismo tiempo también initialized. Y en último lugar cambiara el estado runAnimation, cuando se detecte una de las

imágenes que se encuentran en la colección, este esta activa que los componentes como ViroText, ViroImage... realicen su animación para mostrarse.

Vamos a indicar los códigos correspondientes a los componentes que forman la experiencia básica de una obra en concreto, ya que es el código es el mismo para todas las obras solamente adaptando las variables correspondientes a cada obra.

En primer lugar, tendríamos que leer la imagen que se encuentra almacenada en el servidor, empleamos el componente ViroARTrackingTargets.

Como ya se ha indicado anteriormente, se define un target, que para este ejemplo lo llamamos Guernica_picture el cual se empleará para conectarlo con el componente encargado del reconocimiento y la comparación. También definimos la ubicación de la imagen en el servidor local, al igual su orientación y el tamaño físico del cuadro en el mundo real.

```
ViroARTrackingTargets.createTargets({  
  "Guernica_picture" : {  
    source : {uri:"http://192.168.0.103/resources/Guernica/Guernica.jpg"},  
    orientation : "Up",  
    physicalWidth : 0.33 // real world width in meters  
  },  
});
```

La parte del reconocimiento y comparación se realizará mediante el componente ViroARImageMarker. Como podemos ver en el siguiente código también encontramos el cambio de estado dentro de la propiedad onAnchorFound, el cual nos indica que se ha encontrado un plano y podemos anclar en él la información cambiando el estado correspondiente a runAnimation lo que provoca que los elementos puedan iniciar su animación para mostrar y posicionarse.

```
    <ViroARImageMarker target={"Guernica_picture"}
    onAnchorFound={
      () => this.setState({
        runAnimation: true
      })
    }
    >
    ...
  </ViroARImageMarker>
```

Entre las etiquetas de apertura y cierre del componente `ViroARImageMarker`, es donde se definen los componentes relativos a componente que renderizaran la información. Recordamos que para el posicionamiento se empleará el componente `ViroNode` o `ViroFlexView` empleándose uno u otro una combinación de ambos dependiendo de las características específicas del posicionamiento del elemento. También emplearemos el sistema de coordenadas para la posición de los elementos sobre la imagen.

Vamos a indicar cuales son los componentes que conforman la experiencia básica, y cuáles son sus códigos.

Tenemos un `ViroText` correspondiente al título de la obra, el cual lo creamos dentro de un `ViroNode` mediante el cual definimos la posición del componente y se lee el estado actual de la animación, que por el cambio realizado en la función `onAnchorFound` es verdadero, para que así el componente aparezca. Como podemos ver en el código también se definen una serie de propiedades como son el estilo, el cual lo modificamos mediante una hoja de estilo en la que definimos el color, tamaño, tipo de fuente y alineación del texto entre otras. Para el componente `ViroText` también tiene otras propiedades como la posición la escala, rotación y el tamaño tanto alto como ancho.

```

<ViroNode opacity={0} position={[0, 0, -0.15]}
  animation={{
    name:'animateImage',
    run: this.state.runAnimation
  }}
  >
  <ViroText text="Guernica"
    position={[0, 0, 0]}
    rotation={[0, 0, 0]}
    scale={[.05, .05, .05]}
    width={5} height={0.5}
    style={styles.textStyle_titulo}
  />
</ViroNode>

```

```

textStyle_titulo: {
  flex: .5,
  fontFamily: 'sans-serif',
  fontSize: 30,
  color: '#000000',
  textAlignVertical: 'top',
  textAlign: 'center',
  fontWeight: 'bold',
}

```

También tenemos otro componente ViroText para mostrar el texto correspondiente a una breve descripción sobre la obra, su definición es la misma que la empleada para el componente para el texto, con la diferencia del texto indicado en la propiedad text, la cual indica el texto que se va a mostrar, también el estilo definido es específico para los componentes que representaran el texto correspondiente a la descripción. Como lo podemos ver en los siguientes códigos.

```

... "
  <ViroNode opacity={0} position={[0, 0, 0]}
    animation={{
      name:'animateImage',
      run: this.state.runAnimation
    }} >
    <ViroText text="Guernica es un famoso cuadro
    textLineBreakMode="Justify"
    position={[0, 0, 0]}
    rotation={[0, 0, 0]}
    scale={[.05, .05, .05]}
    width={4} height={5}
    style={styles.textStyle_descripcion}
  />
</ViroNode>

```

```

textStyle_descripcion: {
  flex: .5,
  fontFamily: 'Roboto',
  fontSize: 20,
  color: '#000000',
  textAlignVertical: 'top',
  textAlign: 'left',
  fontWeight: 'bold',
},

```

En esta experiencia también se muestran una serie de características técnicas sobre la obra. Son un conjunto de componentes ViroText, los cuales los agruparemos dentro de un ViroFlexview, ya que estos elementos queremos que vayan en conjunto, y de esta manera al indicar posicionamiento en el ViroNode dentro del cual se encuentra, solamente tenemos que definir el posicionamiento una vez, y no tenemos que repetir el proceso de asignárselo a cada componente de texto.

ViroFlexView, dispone de una serie de propiedades como son rotation, position, alto y ancho que definen el espacio dentro del cual colocaremos los ViroText. Para organizar los elementos dentro de él, se hace mediante su propiedad style en la cual podemos indicar como queremos que alienen los componentes si en fila o en columna. Los ViroText empleados en este caso hacen uso de dos propiedades distintas a los explicados anteriormente, estas propiedades son textClipMode y textLineBreakMode las cuales debidas al valor con el que están definidas, permiten que el texto supere los limites si el texto es mayor y que no se produzcan salto de línea, respectivamente.

A continuación, encontramos los códigos, pertenecientes a los estilos de los ViroText del ViroFlexView y los códigos de los demás componentes que conforman el elemento de las características técnicas.

```
textStyle_caracteristicas:
{
    flex: .2,
    fontFamily: 'Roboto',
    fontSize: 25,
    color: '#000000',
    textAlignVertical: 'top',
    textAlign: 'left',
    fontWeight: 'bold',
}
```

```
flex_caracteristicas:{
    flexDirection: 'column'
}
```

```

<ViroNode opacity={0} position={[0, 0, 0]}
animation={{
  name:'animateImage',
  run: this.state.runAnimation
}}
>
<ViroFlexView
rotation={[[-90, 0, 0]}
position={[0.15,0,0]}
height={0.07}
width={0.07}
style={styles.flex_caracteristicas}
>
<ViroText text="Autor: Pablo Picasso"
textClipMode="None"
textLineBreakMode="None"
scale=[.05, .05, .05]}
width={0.01} height={0.01}
style={styles.textStyle_caracteristicas}
/>
<ViroText text="Técnica: Óleo sobre lienzo"
textClipMode="None"
textLineBreakMode="None"
scale=[.05, .05, .05]}
width={0.01} height={0.01}
style={styles.textStyle_caracteristicas}
/>
<ViroText text="Estilo: Cubismo"
textClipMode="None"
textLineBreakMode="None"
scale=[.05, .05, .05]}
width={0.01} height={0.01}
style={styles.textStyle_caracteristicas}
/>
...

```

```

...
<ViroText text="Tamaño: 776,6 cm x 349,3 cm"
    textClipMode="None"
    textLineBreakMode="None"
    scale={ [.05, .05, .05]}
    width={0.01} height={0.01}
    style={styles.textStyle_caracteristicas}
/>
<ViroText text="Localización: Museo Reina Sofía"
    textClipMode="None"
    textLineBreakMode="None"
    scale={ [.05, .05, .05]}
    width={0.01} height={0.01}
    style={styles.textStyle_caracteristicas}
/>
</ViroFlexView>
</ViroNode>

```

El último elemento que conforma la experiencia básica es la imagen que funciona como botón +AR. Para mostrar esta imagen empleamos el componente `ViroImage`, el cual lo ponemos entre el componente `ViroNode` para modificar su posicionamiento y activar su animación. Este componente como la mayoría de componentes para mostrar contenido dispone de las propiedades `position`, `rotation`, `height`, `width`.

Como diferencia en este componente tenemos la propiedad de `Source` que es fundamental ya que mediante ella indicamos donde se encuentra la imagen que queremos que se muestre.

También dispone de la propiedad `onClick` que es la que nos permite darle a la imagen el funcionamiento de botón, ya implica que cuando se pulsa sobre ella se ejecute la función creada por nosotros, llamada en este caso `_launchGuernicaScene`.

Las escenas se van almacenando en una pila y podemos acceder mediante `sceneNavigator` a ella y seleccionar la escena que queramos. Si la escena a la que queremos pasar ya se encuentra en la pila, simplemente la movemos a la parte superior mientras que si no se encuentra la cargamos y se coloca en la parte superior. Empleamos `jump`, por estar optimizado para hacer más rápido que lo que podríamos hacer con `push` y `pop`. Por tanto, `jump` es útil para aplicaciones que cambian con frecuencia entre varias escenas, como es nuestro caso. Identificamos la escena a la que queremos saltar con un

Key (Guernica_salto), lo que permitirá a Viro reconocer cuando dicha escena ya está en la pila para que podamos saltar rápidamente y seguidamente identificamos el nombre del archivo donde se encuentra la escena a la que queremos saltar.

```
_launchGuernicaScene={()=>{  
  this.props.sceneNavigator.jump("Guernica_salto",  
  {scene:require('./Guernica_visiones'))};  
}
```

```
<ViroNode opacity={0} position={[0, 0, -0.07]}  
  animation={{  
    name:'animateImage',  
    run: this.state.runAnimation  
  }}  
>  
<ViroImage  
  position={[0.20, 0, 0]}  
  rotation=[[-90, 0, 0]}  
  height={0.05}  
  width={0.05}  
  style={styles.image}  
  onClick={this._launchGuernicaScene}  
  source={require('./res/mas_AR.png')}  
</ViroImage>  
</ViroNode>
```

5.8.2.3 Nombre_cuadro.js

Como ya se ha comentado cada obra dispone de un archivo en el cual se encuentra su experiencia específica. En él se emplean mayoritariamente los mismos componentes que en el archivo Sala13.js pero con ligeras modificaciones adaptándolas a las necesidades requeridas, por eso aquí solamente vamos a indicar el funcionamiento y programación de los componentes que solamente se encuentran en los archivos de experiencias específicas.

Uno de estos componentes es ViroSound el cual, nos permite incluir un archivo de audio e interactuar sobre él. Dispone de una propiedad obligatoria como Source para indicar la ubicación del archivo.

Luego la propiedad volume, con un valor de uno indica que se reproduzca al máximo, y que sea limitado por el volumen del dispositivo, mientras que la propiedad onFinish invoca a una función propia del componente que lo que indica es que cuando el audio finalice se pare y no se vuelva a reproducir, a menos que se vuelva a interactuar sobre el elemento que en su propiedad onClick llame la función que modifique el estado de audio_paused el cual afecta al estado de la propiedad paused mediante la cual se indica si el audio se está reproduciendo o no.

```
<ViroSound
  source={{uri:"http://192.168.0.103/resources/Meninas/audio.mp3"}}
  volume={1}
  onFinish={this.onFinishSound}
  paused={this.state.audio_paused}
/>
```

Otro componente que podemos encontrar en algunas de las experiencias específicas es ViroVideo, el cual nos permite mostrar un video, este componente a diferencia del anterior si lo vamos a programar entre un ViroNode ya que indicaremos las coordenadas para su posicionamiento. También dispone de la propiedad obligatoria Source, para indicar la ubicación del vídeo, y ya que se indica su posicionamiento también dispone de las propiedades para poder indicar la escala, la rotation del display en el que se mostrara el vídeo. Además, podemos indicar el si queremos que comience a reproducirse de manera automática o no, indicándolo mediante la propiedad paused, y el estado de esta propiedad lo podemos modificar mediante la interacción de un elemento que tenga asociada una función que lo cambie.

```

<ViroNode opacity={0} position={[0, 0, 0]}
  animation={{
    name:'animateImage',
    run: this.state.runAnimation
  }}
>
<ViroVideo
  source={{uri:"http://192.168.0.103/res/myvideo.mp4"}}
  position={[0,0,0]}
  rotation=[[-90, 0, 0]}
  scale={[0.2, 0.2, 0.2]}
  volume={1}
  paused={true}
  paused={this.state.videoPaused}
>

</ViroVideo>
</ViroNode>

```

Por último, encontramos el componente Viro3DObject, el cual nos permite mostrar un objeto 3d en una posición del espacio. Lo que nos indica que tendremos que emplear un ViroNode para asignar la posición.

Este componente, dispone de propiedades obligatorias, como Source para indicar donde se encuentra el modelo 3d, el cual pueden ser de dos tipos de formatos .obj o .vrp, y resource en el cual se definen donde se encuentran los archivos de textura correspondientes al objeto.

También dispone de las propiedades para poder definir la posición, la escala, la rotación y el tipo de objeto que se va emplear.

Como particularidad, se tiene que definir un componente ViroAmbientLight, para que el componente sea visible, mediante este componente lo que se define es un punto de luz, para crear sobre el objeto zonas de luz y sombra y se pueda ver definida la forma del objeto. En este caso se ha elegido que el color de la fuente de luz sea blanco.

```

<ViroNode opacity={0} position={[0, 0, 0.1]}
  animation={{
    name:'animateImage',
    run: this.state.runAnimation
  }}
>
<ViroAmbientLight color="#ffffff" />
<Viro3DObject
  source={{uri:"http://192.168.0.103/resources/Atleta_cosmico/12338_Statue_v1_L3.obj"}}
  resources={[
    {uri:"http://192.168.0.103/resources/Atleta_cosmico/12338_Statue_v1_L3.mtl"},
    {uri:"http://192.168.0.103/resources/Atleta_cosmico/Podium.jpg"},
    {uri:"http://192.168.0.103/resources/Atleta_cosmico/Statue.jpg"}
  ]}
  position=[[-0.3, 0, 0]}
  scale={[0.001, 0.001, 0.001]}
  rotation={[0, 180, 0]}
  type="OBJ"
  />
</ViroNode>

```

5.8.3 Interfaz de usuario

En este apartado se explicará la interfaz con la que se encontrará el usuario cuando abra la aplicación. Esta será muy minimalista en un principio, ya que tras aceptar los permisos para que la aplicación puede usar la cámara, se encontrará que en toda la pantalla se visualizara lo que está viendo la cámara en ese momento.

Por tanto, vamos a mostrar que elementos se encontraría el usuario cuando se reconoce una imagen, ya que lo que experimentará será la experiencia básica la cual dispone de unos elementos comunes para todas las obras que son:

- Título de la obra: se indicará el título de la obra y estará ubicado en la parte central superior.
- Reseña de la obra: será una descripción breve sobre la obra y se encontrará situada a la izquierda de la obra.

- Datos técnicos de la obra: estos datos técnicos incluyen, el autor, técnica, estilo, tamaño de obra original y localización de la misma y estará ubicada a la derecha de la obra.
- Botón +AR: será un botón circular ubicado en la parte derecha sobre los datos técnicos, y su presencia indicará que la obra cuenta con más contenido de realidad aumentada, sirviendo este como puerta de entrada para la experiencia específica de AR.

La distribución de estos elementos los podemos ver en la Figura 5.13 que encontramos a continuación siendo esta un ejemplo recreado, del posicionamiento, además los tamaños y dimensiones del contenido aumentado respecto a la obra pueden no ser los mismos.

Como ya se ha indicado anteriormente la experiencia específica será distinta para cada obra y el único elemento que tendrán en común el botón -AR. Pero en la Figura 5.14 podemos ver un ejemplo de la experiencia específica para la obra del Guernica.

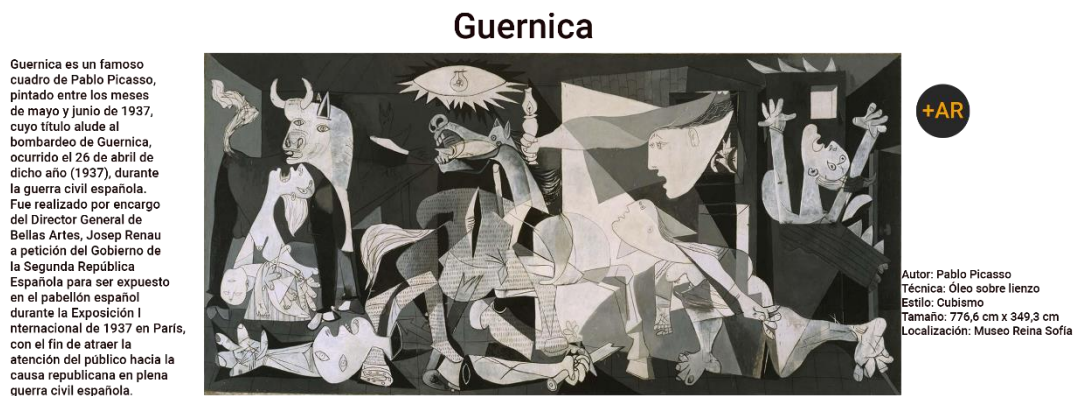


Figura 5.13 – Simulación de la interfaz para experiencia básica.

Guernica



Figura 5.14 - Simulación de la interfaz para experiencia específica.

En el ejemplo que se muestra en la Figura 5.14 lo que encontramos son:

- Título de la obra.
- Botón -AR: situado en la parte derecha superior junto a la obra, y servirá para salir de la experiencia específica y volver a la básica.
- 3 botones: situados en la parte derecha junto a la obra e indican distintas visiones sobre esta obra, las cuales nos permiten apreciar distintos detalles en cada una.
- Descripción de la visión: situada en la parte izquierda junto a la obra encontramos una explicación sobre el tipo de visión que se visualizando en cada momento.

Otro ejemplo es el que podemos ver en la Figura 5.15, en el que encontramos otra serie de elementos distintos, que son:

- Título de la obra.
- Botón -AR: situado en la parte derecha superior junto a la obra, y servirá para salir de la experiencia específica y volver a la básica.
- 2 botones : situados en la parte derecha junto a la obra que permiten escuchar la explicación del cuadro o ver esa explicación mediante un video.
- Un video: situado en la parte derecha junto a la obra y en el cual se explica la obra mediante lenguaje de signos, a modo de signoguía.
- Nombres de los personajes: sobre los personajes del cuadro se encuentra el texto con el nombre de los personajes.
- 3 botones: juntos a algunos de los nombres se encuentran unos botones que indican que se dispone de más información sobre el personaje en cuestion.

- Información sobre el personaje: en la parte derecha junto al cuadro encontramos un texto que nos amplía la información del personaje que hayamos seleccionado.

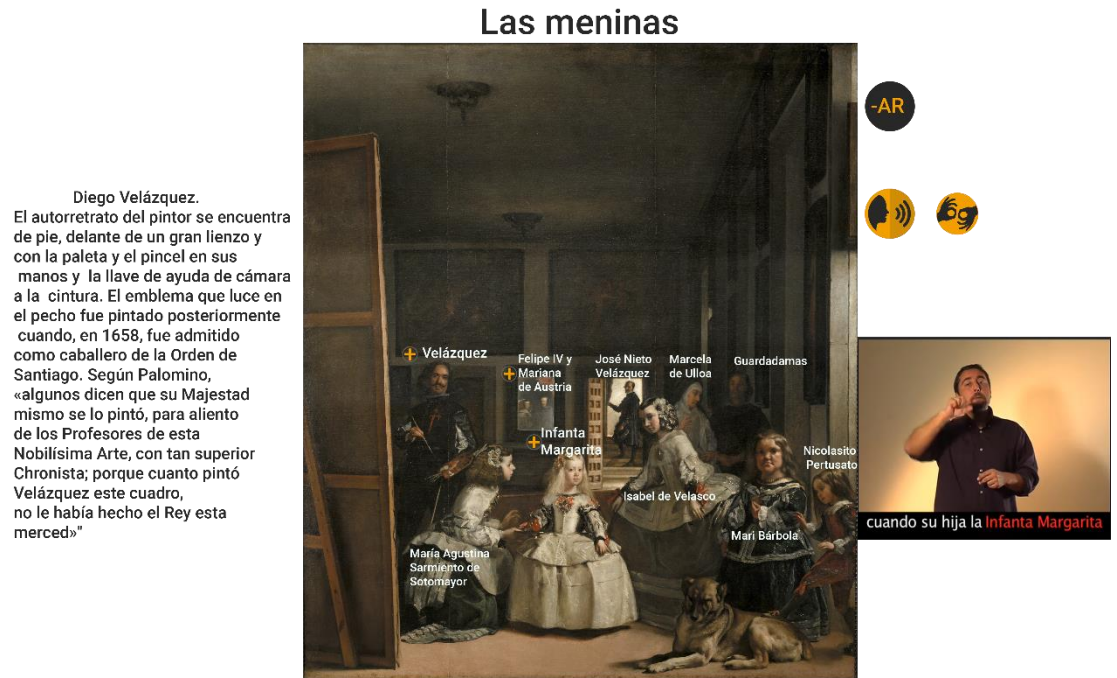


Figura 5.15 - Simulación de la interfaz para experiencia específica

5.9 Requisitos y capacidades

En este punto se van comentar los requisitos y buenas prácticas que se deberán respetar para el buen funcionamiento del sistema, estos requisitos vienen derivados de la plataforma y del lenguaje de programación empleados.

También se indicarán las capacidades con las que cuenta que son las suficientes para cumplir con los objetivos planteados.

- Dispositivos móviles: para poder usada, los dispositivos Android tendrán que disponer de la versión 7 o superior, y los dispositivos Apple requieren disponer de iOS 11.0 o superior y chip A9 o superior, aunque cumpliendo esos requisitos no todos los dispositivos son compatibles en la actualidad, se puede consultar en la lista de dispositivos en el enlace de la bibliografía [34].
- Para reconocer la imagen necesita que esté fija en un lugar o que se mueva lentamente.

- La imagen rellene al menos el 25% del marco de la cámara para la detección inicial.
- Que la superficie en la que se encuentra la imagen sea plana, y no se encuentre arrugado o alrededor algún objeto, por ejemplo, alrededor de una botella.
- Se tenga una visión lo más directa con la imagen, que no se encuentre parcialmente oculta, o en un ángulo oblicuo o que la cámara se esté moviendo rápidamente.
- Las imágenes deberán tener una resolución de 300 x 300 píxeles.
- Si se conoce el tamaño físico esperado de la imagen, especificarlo mejorara el rendimiento del seguimiento, especialmente para imágenes físicas grandes (más de 75 cm).

Entre las capacidades con las que cuenta podemos destacar las siguientes:

- La aplicación es capaz de rastrear hasta 20 imágenes simultáneamente en el entorno del usuario fuera de cámara y dentro de la cámara.
- La detección de las imágenes se basa en los puntos de alto contraste, así que es capaz de detectar imágenes en blanco y negro cuando la imagen de referencia usada es en color y viceversa.
- El uso de imágenes de alta resolución no hace que mejore el rendimiento, por tanto, evitamos el cargar con más peso la aplicación al no tener que usar imágenes de alta resolución.

6 PRUEBAS Y RESULTADOS

Con la aplicación ya diseñada y desarrollada es necesario probar su correcto funcionamiento para la finalidad de este TFG:

Para probar dicho sistema se va a tomar como objetivo el correcto reconocimiento de un conjunto de 30 imágenes correspondientes a cuadros. Se realizarán tantos ensayos y mejoras como sean necesarios hasta alcanzar el objetivo de este TFG y poder determinar que la aplicación cumple los cumple.

Se empleará la aplicación proporcionada por Viromedia la cual dispone de la función Testbed, con la cual nos permite ver el cómo sería el funcionamiento de la aplicación y ver rápidamente los cambios realizados hasta conseguir que la aplicación quede depurada.

Los pasos que se seguirán para realizar las pruebas serán:

1. Comprobar que se reconocen las imágenes que se encuentran en la colección, se considerará el reconocimiento correcto si se muestra sobre la imagen su correspondiente contenido de realidad aumentada.
2. Corrección del posicionamiento y tamaño de los elementos aumentados.
3. Probar el funcionamiento del botón +AR.
4. Comprobar que se representa el contenido aumentado correspondiente a la experiencia específica.
5. Ajuste de posición y tamaño de los elementos aumentados.
6. Comprobación de los elementos que permiten la interacción del usuario.
7. Comentarios y observaciones de los resultados y el proceso.
8. Decidir las mejoras que se harán en el siguiente ensayo en caso de que haya que mejorar, o si la aplicación ya es funcional.

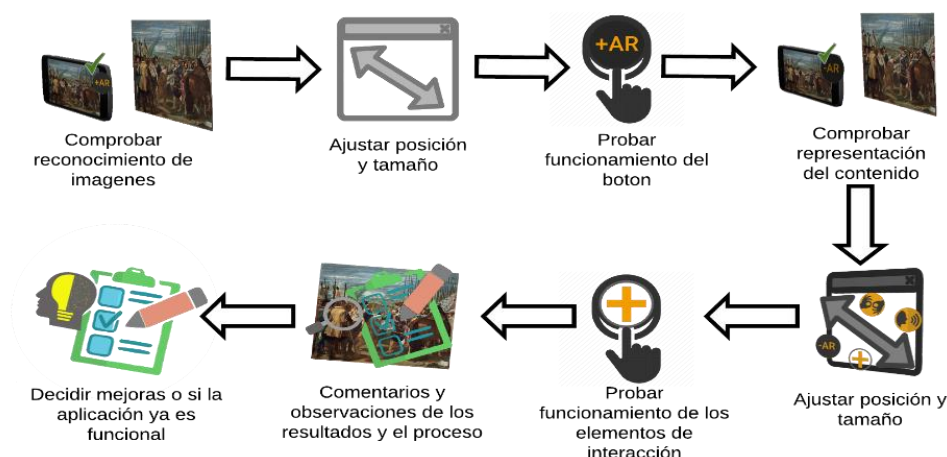


Figura 6.1 – Metodología empleada en las pruebas.

6.1 Colección

Un aspecto importante en la comprobación del correcto funcionamiento de la aplicación, es la elección de las obras que formaran la colección y que dispongan de características destacables entre las obras pictóricas.

Para intentar cubrir los aspectos y propiedades comunes que se pueden encontrar entre los cuadros disponibles dentro de la sala de un museo o una galería, se intentará simplificar solamente en 30 cuadros toda la historia pictórica, se emplearán principalmente obras que se pueden agrupar dentro de los siguientes puntos en común.

- Obras con el mismo estilo.
- Obras realizadas por el mismo autor.
- Obras que emplean la misma técnica,
- Obras que representan figuras similares.
- Obras con mucho y poco contraste.

Si se consigue que el sistema reconozca las 30 obras se dará por supuesto que el sistema será capaz de reconocer en mayor o menor medida cualquier otra obra que se introduzca.

Todas las obras que se han empleado actualmente se encuentran libres de derechos de derechos de autor, y las imágenes digitales empleadas han sido obtenidas de las páginas web de los museos, donde se encuentran expuestas, que nos permiten usarlas de modo personal o para uso académico, y las que no permiten la posibilidad de descargar

han sido de obtenida de Wikimedia Commons, repositorio de imágenes con licencia reutilizable.

Algunas de las imágenes empleadas y que las podemos agrupar dentro de los cinco puntos citados anteriormente son:

Atendiendo al estilo empleado contamos las siguientes imágenes que podemos ver en la Figura 6.2 y ambas comparte el estilo Manierismo.



Figura 6.2 – Imágenes que comparten estilo.

Teniendo en común el mismo autor podemos encontrar algunas de las siguientes imágenes que se muestran en la Figura 6.3, en este caso el autor es Diego Velázquez.



Figura 6.3 – Imágenes del mismo autor.

Atendiendo a obras que, realizadas con misma técnica, algunas de las que tenemos incluidas son las que se muestran en la Figura 6.4, las cuales comparten que han sido realizadas en Óleo sobre lienzo.



Figura 6.4 – Imágenes con la misma técnica.

Sí tenemos en cuenta que las imágenes representen una situación similar, tenemos como ejemplo las que se muestran en la Figura 6.5.



Figura 6.5 – Imágenes que representan imágenes similares.

Atendiendo al contraste de las figuras representadas en la obra, tenemos dos posibilidades que la imagen tenga mucho contraste o poco, y alguno de los ejemplos son lo que encontramos en la Figura 6.6.



Figura 6.6 – Imágenes con mucho y poco contraste.

Una vez comprobado que todas las imágenes que conforman la colección son reconocidas por la aplicación pasaremos a ajustar el contenido aumentado.

6.2 Posicionamiento y escalado

En este momento de las pruebas, lo que realizaremos es el posicionamiento idóneo de los elementos aumentados que como ya se comentó para la experiencia básica, son el título, descripción breve, botón +AR y datos técnicos.

Los elementos formados solamente por texto se realizan mediante el uso del componente ViroText, el cual cuenta con la propiedad “position” en la cual podemos indicar el posicionamiento del texto indicándolo mediante coordenadas. Tendremos que tener en cuenta que para el componente ViroText su punto de origen, punto en el que el valor de las tres componentes X, Y, Z son cero, se encuentra en la esquina superior izquierda.

También tenemos que mantener un compromiso entre el tamaño de la fuente empleada para el texto y el escalado del mismo, ya que el uso de fuentes más grandes y definidas utiliza más memoria, por otro lado, un tamaño de fuente demasiado pequeño hará que el texto se vea borroso. De igual manera utilizar la propiedad “scale” para aumentar directamente el tamaño del texto también puede provocar que esté se vea borroso.

Otro aspecto que también afectara a la experiencia es la distancia del texto desde el usuario. Estamos en un espacio 3D, por lo que cuanto más lejos este el texto de la cámara, más pequeño aparecerá.

Así que emplearemos el método de prueba y error para descubrir el tamaño de fuente y escalado correctos.

Un ejemplo de cómo se ve el contenido cuando los tamaños y posicionamiento no son los correctos lo podemos ver en la Figura 6.7 mientras que una vez elegidos los tamaños y posicionamiento correcto podemos observar como el contenido se ve más nítido y claro (Figura 6.8).



Figura 6.7 – Experiencia AR con tamaños y posiciones sin ajustar



Figura 6.8 – Experiencia AR con tamaños y posiciones ajustadas

6.3 Interacción

Una vez que estamos conforme con el posicionamiento de los elementos, pasaremos a comprobar el funcionamiento de los elementos sobre los que podemos interactuar como son el botón +AR y los botones que nos proporcionan más información sobre los personajes.

Estos elementos son imágenes en las cuales utilizamos la propiedad on_click para que al pulsar sobre ellas se produzca una acción indicada en la función que se ha creado específicamente para cada imagen.

El método que se usará para comprobar que el funcionamiento es correcto será el de prueba y error. Comprobando si al pulsar sobre las imágenes ocurre lo que se estaba buscando.

En la Figura 6.9 podemos observar un ejemplo de interacción, en el cual junto a algunos personajes aparece un símbolo con un “+” indicando que si lo pulsamos obtendremos más información. Una vez pulsado lo que ocurre es lo esperado, y junto al cuadro en la parte derecha se nos muestra más información. Otros ejemplos de interacción se pueden ver en el vídeo que se encuentra en el punto 6.4 de esta memoria.

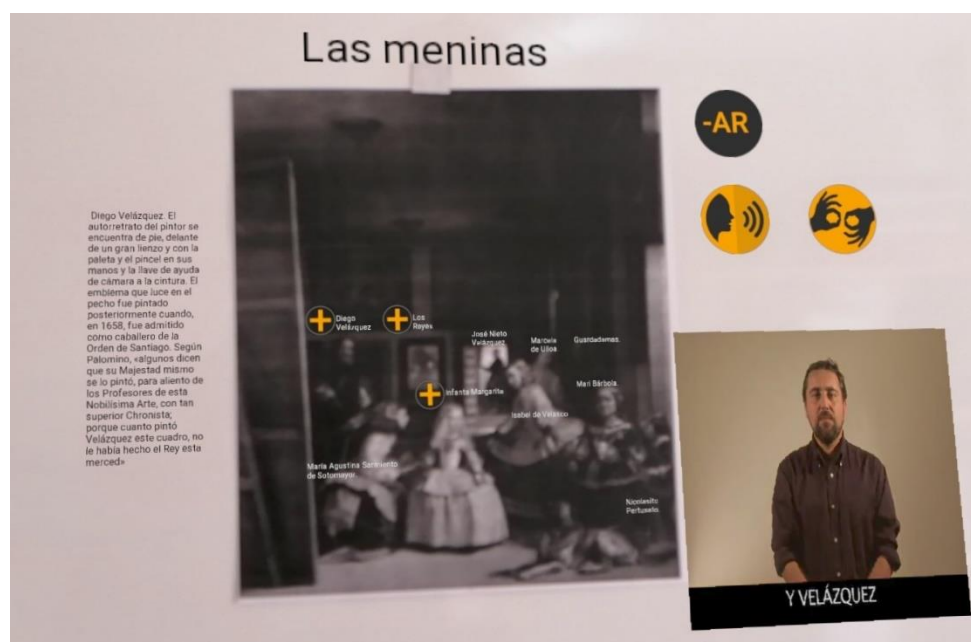


Figura 6.9 – Experiencia específica con interacción

6.4 Comentarios y observaciones

En este apartado explicaremos algunas de las observaciones que se han tenido en cuenta durante el desarrollo de las pruebas, y que se han ido implementado tras el transcurso de las pruebas.

Algunas de las que se han tenido en cuenta son:

Que la imagen la cual queremos reconocer no se encuentre junto a un punto de luz intensa, que provoca que la cámara del dispositivo se encuentre “a contraluz” impidiendo si la luz es demasiado intensa que se puedan distinguir los contrastes en la imagen. De igual manera si la imagen se encuentra en un entorno muy oscuro el resultado sería el mismo impidiendo distinguir los puntos de contraste. Inconveniente que se podría subsanar mediante el empleo de flash del dispositivo, pero dado que, no en todos los museos/salas de exposición se permite su uso, no se ha implementado en la aplicación. Además, consideramos que estos inconvenientes no se producirían, ya que las instalaciones para las cuales está diseñada esta aplicación dispondrían de un nivel de luminosidad adecuado para su uso.

Durante las pruebas de escalado y posicionamiento de los elementos, intentar que el dispositivo se encuentre siempre a la misma distancia para que los cambios realizados sean comparables, ya que al encontramos en un entorno 3D, la distancia a la que se encuentra la cámara y la imagen afecta al tamaño y nitidez de los elementos aumentados. Para solucionar esto se empleará un trípode u objeto similar para anclar el dispositivo en una posición fija durante las pruebas.

Otra observación que se detectó durante las pruebas es la necesidad de que los límites de la imagen se distingan claramente del fondo sobre el cual se encuentra, ya que si esta distinción no es suficiente no se podrá calcular el tamaño de la imagen, provocando que el posicionamiento de los elementos que se realizan mediante coordenadas no sea esperado.

Puede concluirse entonces que tras las modificaciones desarrolladas durante las pruebas se ha logrado el objetivo de que se reconozcan todas las imágenes y que el contenido se muestre de la manera adecuada. Por tanto, se puede indicar que la aplicación ya ha acabado su fase de pruebas y podemos desarrollar los archivos ejecutables que se pondrán a disposición de los usuarios.

La explicación para obtener los archivos ejecutables se encuentra en el Anexo 1.

A continuación, podemos encontrar una serie de pruebas en las que se corroboran las condiciones y capacidades en las que sí funciona la aplicación y en cuales presentan dificultades.

Algunas de las pruebas son:

- La imagen se encuentre fija o en movimiento lentamente.
- Rellene al menos el 25% del marco de la cámara.
- Que la superficie sea plana.

- Se tenga visión directa, no se encuentre parcialmente oculta.
- Capaz de rastrear imágenes fuera y dentro de cámara.
- Detección de imágenes, mediante puntos de alto y bajo contraste, permitiendo detectar imágenes en blanco y negro o color.
- Detección de la imagen dependiendo de la luminosidad.

La realización de estas y otras pruebas se pueden encontrar en el siguiente vídeo:

<https://youtu.be/oVjOCZ1Sg5k>

La aplicación en funcionamiento la podemos ver en el siguiente video:

<https://youtu.be/MPMksWuf3y4>

7 ESCALABILIDAD Y MIGRACIÓN A OTROS REQUIRIMIENTOS

En este punto se indicará como se pueden ir incrementando el número de imágenes que se puedan reconocer y cuáles serían las modificaciones que se tendrían que realizar en la aplicación para que se puede emplear en otro entorno distinto al que ha sido diseñada en primer momento.

Como se indicó anteriormente, la aplicación se podría emplear en otros ámbitos por ejemplo en la educación, el comercio.

7.1 Introducir nuevos contenidos

Para introducir nuevas imágenes que sean reconocidas necesitaríamos lo siguiente:

- Una imagen en formato digital con una resolución mínima de 300 x 300 pixeles de la imagen que se quiere que sea reconocida por la aplicación.

Luego emplearíamos el proceso ya explicado en otros apartados, para el cual tendríamos que crear una serie de componentes para cada uno de los pasos.

- Un componente ViroARTrackingTargets para leer la imagen del lugar donde esta almacenada.
- Un componente ViroARImageMarker el cual se encargará de buscar la imagen que se le indique mediante el target y comparar con lo que este capturando la cámara del dispositivo.

Con estos dos componentes ya sería capaz de detectar la nueva imagen que añadiríamos.

Ya estaríamos en disposición de comenzar a añadir el contenido aumentado, este contenido puede ser de distinto tipo, puede ser texto, imágenes, imágenes en movimiento (GIF), imágenes 360°, vídeos, videos en 360°, sonido, objetos 3d...

Para añadir contenido de cada uno de este tipo se dispone de componentes específicos, en los cuales de manera bastante intuitiva se indica donde se encuentra alojado ese recurso y ya se mostraría, posteriormente se puede adaptar mediante las propiedades propias de cada componente a las necesidades que se precisen.

El posicionamiento del contenido aumentado se realizaría mediante los componentes ViroNode, ViroFlexView o una combinación entre ambos, indicando las posiciones mediante el sistema de coordenadas cartesianas.

7.2 Migración a otros requerimientos

La aplicación puede ser fácilmente utilizada en otros ámbitos, ya que los requisitos para la detección son los mismos, solamente se necesita una imagen almacenada que al ser detectada servirá como activador para poder mostrar la realidad aumentada.

Esta aplicación se podría emplear en la educación, encontrándose la imagen empleada en los libros de texto. También la imagen de activador la puede tener el profesor y mediante *mirroring* de la pantalla del dispositivo móvil, les muestre la escultura o animal mitológico, aumentado que se encuentra en el centro del aula a tamaño real. Para lo cual simplemente se necesitaría las imágenes empleadas como activadores y los modelos 3d de lo que se quiere mostrar.

También, se puede emplear para ver en escala real un nuevo vehículo que se quiera comprar tanto de manera individual por parte del usuario, o en un confesionario para mostrarle al cliente un modelo del cual no se dispone uno de manera física. Para lo cual se podría emplear como activador el logo de la marca y al detectarlo poder elegir el modelo que se quiere ver, y se genera el modelo 3d aumentado del vehículo

Otro ejemplo, sería que las imágenes empleadas como activadores, sean los carteles de películas, lo detectaría y sobre ella se mostraría el contenido aumentado, el tráiler y dando la posibilidad de comprar la entrada para esa película.

8 CONCLUSIÓN

En este trabajo de fin de grado, la primera conclusión a la que podemos llegar una vez finalizadas las tareas descritas en esta memoria es la del cumplimiento de los objetivos propuestos en el apartado número 3.

Se ha desarrollado una aplicación móvil de realidad aumentada a lo largo de este trabajo, ofreciendo ciertas funcionalidades más haya de reconocer una imagen y añadir algún contenido multimedia. La sencillez de uso y la capacidad de que los contenidos estén almacenados de manera remota nos permiten el poder incluir un mayor número de obras que son clave frente a otras aplicaciones las cuales solamente disponen de un catálogo muy reducido.

Tras un estudio del estado del arte en el ámbito de la realidad aumentada, nos ha permitido conocer en mayor profundidad las técnicas que pueden ser empleadas por esta tecnología y sobre las cuales se apoya la plataforma empleada para la implementación de la aplicación.

También se ha realizado un estudio de las aplicaciones que emplean realidad aumentada en el ámbito del arte, para conocer su funcionamiento y características de las que disponen. Por tanto, se han podido observar los puntos fuertes y carencias de estas aplicaciones con el fin de mejorar en algunos aspectos nuestra aplicación.

Se ha realizado un estudio de la plataforma ViroReact que ha sido la con la que se ha desarrollado la aplicación permitiéndonos cuantificar que características y elementos podíamos emplear directamente y cuales se tendrían que desarrollar. Una vez, conocido esto se ha llevado a cabo la implementación dando lugar a la aplicación, la cual se ha sometido a una serie de testeos y pruebas de funcionamiento hasta determinar su correcto funcionamiento.

Como se podrá observar en el anexo 2 correspondiente al manual de usuario, se ha conseguido crear una aplicación que cumple con su propósito, el de ser una guía para museos con realidad, y que resulta fácil e intuitiva para el usuario.

9 TRABAJO FUTURO

Este proyecto de la posibilidad de expandirse a hacia otras líneas de trabajo. Quedando muchas puertas abiertas a pesar de desarrollar un extenso trabajo que cumple con los objetivos propuestos para este trabajo de fin de grado. Más allá de los objetivos que se tenían como finalidad, podemos encontrar las siguientes líneas de trabajo las cuales se intentarán abarcar en un futuro.

9.1 Mejoras en la aplicación

Uno de los principales objetivos futuros y de los cuales ya se encuentran en desarrollo, será el usar un servidor remoto en el que almacenar los recursos multimedia empleados para la creación de las experiencias de realidad aumentada.

Otra mejora para facilitar el uso por parte de usuarios extranjeros o que no dispongan de conexión a Internet durante la visita es la de posibilitar la descarga del contenido relativo a las obras que vaya a visitar y que este se almacene en el dispositivo móvil y pueda de esta manera utilizar la aplicación sin limitaciones. Para llevar a cabo esta mejora también implicaría la creación de un menú en el que el usuario pudiera elegir esta opción y otras.

9.2 Enseñanza y mini juegos

Ya que los museos son lugares frecuentemente visitados como parte de excursiones escolares, se podrían incluir una serie de preguntas correspondientes a cada obra para así comprobar si se ha prestado atención a la información relativa a la obra.

Para hacerlo más atractivo se podría emplear una estética similar a la de un juego en el que se van obteniendo puntos con las respuestas acertadas, y que posteriormente esos resultados se puedan compartir mediante las redes sociales

9.3 Nuevas plataformas

En esta línea, la idea principal es el desarrollo de la aplicación para su uso en unas gafas de realidad aumenta. Dependiendo del tipo de las mismas sería necesaria el desarrollo por completo de la aplicación simplemente realizar algunas modificaciones para adaptar la experiencia.

El caso más simple sería el de emplear unas gafas de realidad aumentada, por ejemplo, las Lenovo Mirage (Figura 9.1), las cuales de manera simplificada lo que proporcionan es la posibilidad de colocar el dispositivo móvil, en este caso tendría que ser

un teléfono móvil, en un *headset* permitiendo le al usuario tener las manos libres y sentir la experiencia de manera más realista.



Figura 9.1 – Headset Lenovo Mirage

Fuente: RoadtoVR.com

En el otro caso, sería el uso de las Holens, pero para este caso se tendría que desarrollar la aplicación desde cero ya que actualmente no se dispone de soporte por parte de ARCore ni ARKit. Aparte de su elevado precio lo cual limitaría el uso por parte de los usuarios y conllevaría una gran inversión por parte de los museos o galerías, que pudieran dejarlas durante el tiempo de la visita.

10 BIBLIOGRAFÍA

- [1] INFINITE RED, 2018.Chain React 2018: Building AR Apps with React Native by Vladimir Novick. En: *Youtube* [vídeo en línea]. Publicado el 27 de julio de 2018 [consulta: junio de 2019]. Disponible en: https://www.youtube.com/watch?v=WYCZqWEnHkg&list=LLIEPBWwROZ3La3yeT_7eflQ&index=11&t=729s.
- [2] GITNATION REACT CONFERENCES, 2018: React Native VR + AR Made Simple – Nader Dabit. En *Youtube* [video en línea]. Publicado el 6 de mayo de 2018 [consulta: junio de 2019]. Disponible en: https://www.youtube.com/watch?v=csgMNguUjDU&list=LLIEPBWwROZ3La3yeT_7eflQ&index=15 .
- [3] Riobó Iglesias, J., Aznar Relancio, S., Gracia Bandrés, M.A., Romero San Martín, D. Análisis de tendencias: Realidad Aumentada y Realidad Virtual. TecsMedia,2015.
- [4] Lacueva Pérez, F.J, et al. Análisis Realidad Aumentada para entornos Industriales”. TecsMedia,2015.
- [5] Arthur, Rachel, 2017. Augmented Reality is set to transform fashion and retail. *Forbes* [en línea]. 31 de octubre. Disponible en: <https://www.forbes.com/sites/rachelarthur/2017/10/31/augmented-reality-is-set-to-transform-fashion-and-retail/#5eda17813151> .
- [6] Piacente, Francisco Javier, 2010.Crean el primer dispositivo de realidad virtual tangible. *Tendencias21* [en línea]. 22 de julio. Disponible en: https://www.tendencias21.net/Crean-el-primer-dispositivo-de-realidad-virtual-tangible_a4705.html .
- [7] Churcher, Jason, 2013. Augmented Reality: Future or Fad?. *Augview* [en línea]. 13 de septiembre. Disponible en: <https://www.augview.net/News/Blog/archive-13September2013.html> .

- [8] Imascono Team, 2017. Tipos de Realidad Aumentada según sus formas de utilización. Imascono [en línea]. 19 de junio. Disponible en: <https://imascono.com/es/magazine/realidad-aumentada-segun-utilizacion> .
- [9] Stewart Smith, Hana, 2012. Education with Augmented Reality: AR textbooks released in Japan. ZDNet [en línea]. 4 de abril. Disponible en: <https://www.zdnet.com/article/education-with-augmented-reality-ar-textbooks-released-in-japan-video/> .
- [10] Shariar, Khaled, 2019. Best Augmented Reality apps & VR Games on Android. Getandroidstuff [en línea]. 6 de junio. Disponible en: <http://getandroidstuff.com/best-augmented-reality-apps-vr-games-android/> .
- [11] Borghino, Dario, 2012. Augmented reality glasses perform real-time language translation. Newatlas [en línea]. 29 de julio. Disponible en: <https://newatlas.com/language-translating-glasses/23494/> .
- [12] Mayer, David, 2012. Telefónica best on augmented reality with Aurasma tie-in. Gigaom [en línea]. 17 de septiembre. Disponible en: <https://gigaom.com/2012/09/17/telefonica-bets-on-augmented-reality-with-aurasma-tie-in/> .
- [13] Magid, Abraham, 2017. Augmented Reality is already improving worker performance. *Harvard Business Review* [en línea]. 13 de marzo. Disponible en: <https://hbr.org/2017/03/augmented-reality-is-already-improving-worker-performance> .
- [14] Luna, Úrsula, Ibáñez-Etxeberria, Alejandro, Rivero, Pilar, 2019, El patrimonio aumentado. 8 apps de Realidad Aumentada para la enseñanza-aprendizaje del patrimonio. *Revista Interuniversitaria de Formación del Profesorado*. Vol 94, no 43-62. ISSN 0213-8646.
- [15] Maier, Patrick, Klinker, Gudrun, Tönnis, 2009. Augmented Reality for teaching spatial relations. *Conference of the International Journal of Arts & Sciences*.

- [16] Griggs, Brandon, 2012. 'Augmented-reality' windshields and the future of driving. En: *CNN Business* [en línea]. Disponible en: <https://edition.cnn.com/2012/01/13/tech/innovation/ces-future-driving/index.html> .
- [17] Eisenman, Bonnie, 2015. *Learning React Native* [en línea]. Madrid: O'Reilly Media. ISBN 978191919049. Disponible en: <https://www.oreilly.com/library/view/learning-react-native/9781491929049/>
- [18] López Pombo, Héctor, 2010. *Análisis y desarrollo de sistemas de realidad aumentada* [en línea]. Trabajo de fin de máster. Madrid: Universidad Complutense de Madrid. Disponible en: https://eprints.ucm.es/11425/1/memoria_final_03_09_10.pdf
- [19] Sánchez Lavi, Eva M, 2019. *Carta de restaurante virtual mediante realidad aumentada AR* [en línea]. Trabajo de fin de grado. Jaén: Universidad de Jaén. Disponible en: <https://hdl.handle.net/10953.1/9125>
- [20] Garrote Núñez, Adrián, 2016. *Funciones básicas para reconocimiento automático de lenguaje de signos utilizando Leap Motion* [en línea]. Trabajo de fin de grado. Jaén: Universidad de Jaén. Disponible en: <https://hdl.handle.net/10953.1/3424>
- [21] Pavlik, John V y Shawn McIntosh, 2017. *Augmented Reality. Converging Media: A New Introduction to Mass Communication*. 5ª ed. Oxford University Press, pp.184-185. ISBN 9780199859931.
- [22] Página web con la documentación para desarrolladores de ARCore <https://developers.google.com/ar/discover/>
- [23] Página web con la documentación para desarrolladores de ARKit <https://developer.apple.com/documentation>
- [24] Página web con la documentación para desarrolladores de React Native <https://facebook.github.io/react-native/>
- [25] Repositorio de Github donde están disponibles los códigos de aplicaciones ejemplo de ViroReact. <https://github.com/viromedia/viro>

- [26] Página web de la plataforma ViroMedia de donde obtenemos la API Key y tenemos acceso a toda la documentación correspondiente de la API. <https://docs.viromedia.com/docs/viro-platform-overview>
- [27] Página web del Museo del Prado. <https://www.museodelprado.es/>
- [28] Página web del Art Institute of Chicago. <https://www.artic.edu/>
- [29] Enlace a la entrada Augmented reality correspondiente a Wikipedia. https://en.wikipedia.org/wiki/Augmented_reality
- [30] Página web del Museo Reina Sofía. <https://www.museoreinasofia.es/>
- [31] Página web del Museo Thyssen. <https://www.museothyssen.org/>
- [32] Página web del Museo Sorolla. <http://www.culturaydeporte.gob.es/msorolla/inicio.html>
- [33] Página web al repositorio de imágenes con licencia reutilizable Wikimedia Commons. https://commons.wikimedia.org/wiki/Main_Page.
- [34] Enlace web donde se pueden encontrar los dispositivos compatibles con ARCore. <https://developers.google.com/ar/discover/supported-devices>

ANEXO 1. CREACIÓN DE ARCHIVO EJECUTABLE

En este anexo se explicará cual es proceso a seguir para obtener el archivo ejecutable a partir del código programado empleando React Native y ViroReact.

Se puede obtener el archivo tanto para Android (.apk) como para iOS (.ipa). Pero solamente se explicará el proceso para Android ya que para iOS se necesitan credenciales de desarrollador.

Android requiere que todas las aplicaciones estén firmadas digitalmente con un certificado antes de poder instalarse, para poder distribuir su aplicación de Android a través de Google Play, debe estar firmada con una clave de lanzamiento que luego debe usarse para todas las actualizaciones futuras. Desde 2017, Google Play puede gestionar las versiones de firma automáticamente gracias a la funcionalidad de firma de aplicaciones de Google Play. Sin embargo, antes de que su aplicación se cargue en Google Play, debe firmarse con una clave de carga.

Se puede encontrar más información sobre el firmado de aplicaciones Android en la documentación para desarrolladores de Android, pero nosotros describiremos brevemente el proceso, así como los pasos necesarios para empaquetar el paquete de JavaScript.

1. Generación de la clave de carga: puede generar una clave de firma privada usando *keytool*. En Windows, *keytool* debe ejecutarse desde *C: \ Archivos de programa \ Java \ jdkx.x.x_x \ bin*. Tendríamos que usar el siguiente comando:

```
$ keytool -genkeypair -v -keystore my-release-key.keystore -alias  
my-key-alias -keyalg RSA -keysize 2048 -validity 10000
```

Este comando le solicita su contraseña para el almacén de claves y la clave y para los capos de Nombre para distinguir su clave. Luego genera el almacén de claves como un archivo llamado *my-upload-key.keystore*.

El almacén de claves contiene una única clave, valida por 10000 días. El alias es un nombre que usará más adelante cuando firme su aplicación, así que recuerde anotar el alias.

2. Configurar variables de Gradle:
 - Coloque el archivo *my-upload-key.keystore* en el directorio *android / app* dentro de la carpeta de su proyecto.

- Edite el archivo *android/gradle.properties* y agregue lo siguiente (sustituyendo por los datos empleados en el primer paso).

```
MYAPP_RELEASE_STORE_FILE=my-release-key.keystore  
MYAPP_RELEASE_KEY_ALIAS=my-key-alias  
MYAPP_RELEASE_STORE_PASSWORD=*****  
MYAPP_RELEASE_KEY_PASSWORD=*****
```

Estas serán variables globales de Gradle, que luego podemos usar en nuestra configuración de Gradle para firmar nuestra aplicación.

3. Agregar la configuración de firma a la configuración de Gradle de su aplicación: el último paso que se debe hacer es configurar las versiones de lanzamiento para que se firmen con la clave. Edite el archivo *android / app / build.gradle* de la carpeta de su proyecto y agregue la configuración de firma.

```

...
android {
    ...
    defaultConfig { ... }
    signingConfigs {
        release {
            if (project.hasProperty('MYAPP_RELEASE_STORE_FILE')) {
                storeFile file(MYAPP_RELEASE_STORE_FILE)
                storePassword MYAPP_RELEASE_STORE_PASSWORD
                keyAlias MYAPP_RELEASE_KEY_ALIAS
                keyPassword MYAPP_RELEASE_KEY_PASSWORD
            }
        }
    }
    buildTypes {
        release {
            ...
            signingConfig signingConfigs.release
        }
    }
}
...

```

4. Generando la versión APK: simplemente tenemos que ejecutar lo siguiente en un terminal.

```

$ cd android
$ ./gradlew assembleRelease

```

AssembleRelease de Gradle incluirá todo el JavaScript necesario para ejecutar su aplicación en el APK.

El APK generado se puede encontrar en *android/app/build/outputs/apk/release/app-release.apk* y está listo para ser distribuido.

ANEXO 2. MANUAL DE USUARIO DE LA APLICACIÓN

En este anexo se explicará cómo usar la aplicación y que indican cada símbolo/icono que aparecen durante las experiencias de realidad aumentada.

En primer lugar, tendremos que proceder a instalar la aplicación, una vez instalada procedemos a abrirla y nos solicitara permiso para utilizar la cámara.

Una vez aceptado veremos en la pantalla lo que este captando en ese momento la cámara. Y ya está preparada la aplicación para poder reconocer las imágenes.

La aplicación dispone de dos tipos de experiencias, la experiencia básica y la experiencia específica para cada obra.

Experiencia Básica

En la experiencia básica se es capaz de reconocer todas las obras de la colección. La información que se ofrece sobre las obras es título, descripción breve y datos técnicos sobre la obra. Sí la obra dispone de experiencia específica también aparecerá el botón +AR (Tabla 0.1), mediante el cual puede acceder a la experiencia específica.

En la Figura 0.1, se puede ver como se distribuyen estos componentes entorno a la imagen en esta experiencia.

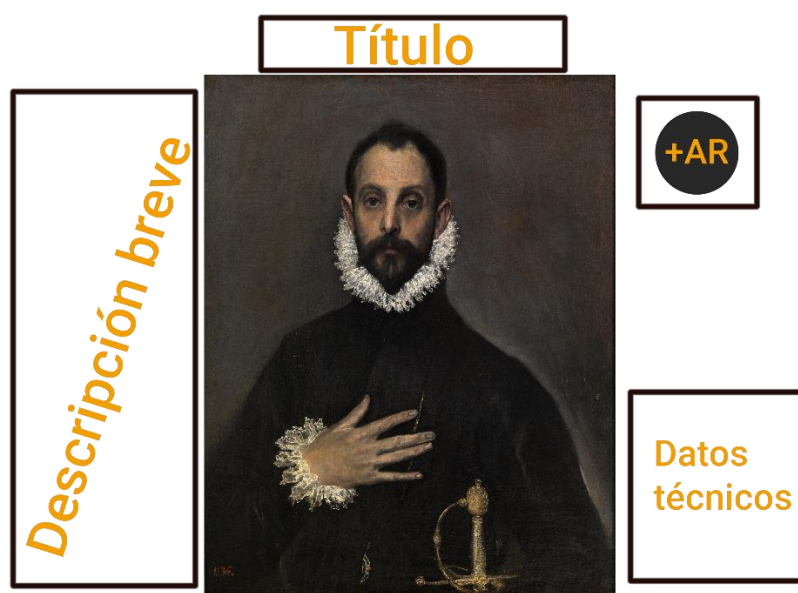


Figura 0.1 – Ejemplo experiencia básica

Experiencia Específica

En la experiencia específica se muestra más contenido aumentado sobre la obra a la cual se ha pulsado el botón +AR.

El tipo de contenido en esta experiencia es distinto dependiendo de la obra, el único elemento común entre todas ellas es que disponen del botón -AR para volver a la experiencia básica.

Alrededor y sobre la imagen se encontrará información como los nombres de los personajes, interpretaciones del cuadro... También se mostrarán una serie de iconos que indican distintas cosas, el significado de cada uno de estos iconos se muestra en la Tabla 0.1

Un ejemplo de una experiencia específica es el que se puede ver en la Figura 0.2 a continuación.

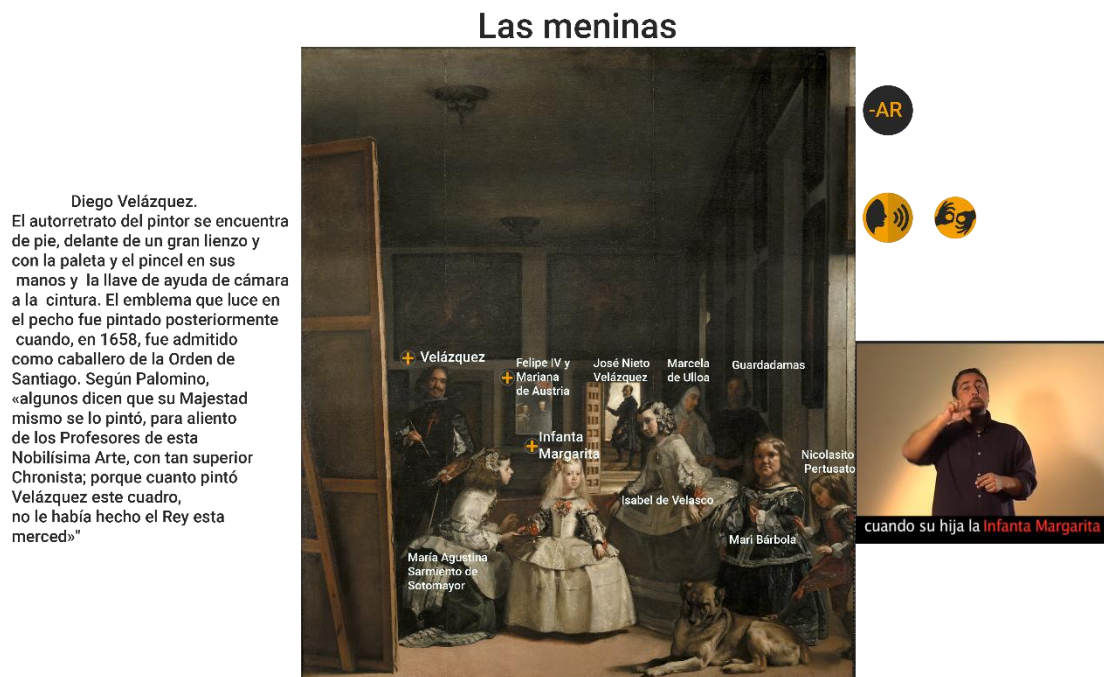


Figura 0.2 – Ejemplo experiencia específica

Icono	Descripción
	Aparece durante la experiencia básica e indica que la imagen dispone de más contenido AR, al pulsar sobre él se ejecuta la experiencia específica.
	Aparece durante la experiencia específica y al pulsar sobre el se sale de ella y se vuelve a la experiencia básica.
	Aparece durante la experiencia específica sobre la imagen reconocida e indica que se puede obtener más información sobre el elemento que se encuentre.
	Aparece durante la experiencia específica e indica que se dispone de una explicación de la obra mediante audio, al pulsar sobre ella comenzará a reproducirse.
	Aparece durante la experiencia específica e indica que se dispone de una explicación de la obra mediante un vídeo en lenguaje de signos y subtítulos.
	Puede aparecer en ambas experiencias e indica que se puede cerrar cierto contenido que se está mostrando.
	Aparece durante la experiencia específica e indica que si pulsamos sobre él se mostrará la imagen en el tamaño real de la obra.
	Aparece durante la experiencia específica e indica que si pulsamos sobre él la imagen con tamaño real desaparecerá y volveremos a la experiencia específica normal.

Tabla 0.1 – Significado de los iconos

Requisitos y buenas maneras de uso.

Para poder ejecutar la aplicación en su dispositivo es necesario que cumpla con los siguientes requisitos:

Si es Android tendrá que disponer de la versión 7 o superior, y los dispositivos Apple requieren disponer de iOS 11.0 o superior y chip A9 o superior, aun cumpliendo esos

requisitos puede ser que su dispositivo no sea compatible, lo puede comprobar en el siguiente enlace donde encontrara un listado con los dispositivos compatibles.

Enlace para consultar los dispositivos compatibles:
<https://developers.google.com/ar/discover/supported-devices>.

Para que la aplicación funcione de manera adecuada tenga en cuenta los siguientes consejos de uso:

- Para reconocer la imagen necesita que esté fija en un lugar.
- La imagen rellene al menos el 25% del marco de la cámara para la detección inicial.
- Que la superficie en la que se encuentra la imagen sea plana, y no se encuentre arrugado.
- Se tenga una visión lo más directa con la imagen, que no se encuentre parcialmente oculta, o en un ángulo oblicuo.

Puede encontrar un vídeo de la aplicación en funcionamiento en el siguiente enlace:
<https://youtu.be/MPMksWuf3y4>