



Universidad
de La Laguna

Universidad de La Laguna
Escuela Superior de Ingeniería y Tecnología
Sección de Ingeniería Informática

PROYECTO FINAL DE CARRERA

Proyecto Tahut: Los Graffiti del Templo de Debod

Aplicación móvil con tecnología aumentada

(Curso 2015- 2016)

Tutor:

Jesús Miguel Torres Jorge

Alumno:

Sergio Aarón Afonso Suárez

Agradecimientos

A mis padres y a mi hermano, por el apoyo
y la paciencia que han mostrado.

A Cintia, por el dulce apoyo, ayuda e
interés mostrado a lo largo de este
proyecto.

A Pablo por su innegable ayuda.

A Miguel A. Molinero y a Jorge Martín por
su ayuda y consejo.

A Jesús Torres por sus ánimos.

Índice general

CAPÍTULO 1	12
INTRODUCCIÓN	12
CAPÍTULO 2	13
OBJETIVO Y JUSTIFICACIÓN	13
2.1 <i>Objetivo</i>	13
2.2 <i>Justificación</i>	13
CAPÍTULO 3	15
MOTIVACIÓN	15
CAPÍTULO 4	17
ANTECEDENTES	17
4.1 <i>Herramientas similares en Debod</i>	17
4.2 <i>Aplicaciones turísticas existentes</i>	18
4.2.1 <i>MTrip</i>	18
4.2.2 <i>Go Augmented</i>	19
4.2.3 <i>Street Museum</i>	20
4.2.4 <i>Exposición en el MOMA</i>	20
4.2.5 <i>Museo de Allard Pierson</i>	21
4.2.6 <i>Costa del Sol-Málaga</i>	21
4.2.7 <i>Guía turística de Segovia</i>	21
CAPÍTULO 5	22
MARCO TEÓRICO	22
5.1 <i>Aplicación Móvil (app)</i>	22
5.1.1 <i>Preámbulo</i>	22
5.1.1.1 <i>Características</i>	23
5.1.1.2 <i>Distribución</i>	23
5.1.2 <i>Historia de las Aplicaciones Móviles (App)</i>	24
5.1.3 <i>Tipos de aplicaciones</i>	25
5.1.3.1 <i>Nativas</i>	26
5.1.3.2 <i>Híbridas</i>	27
5.1.3.3 <i>Web</i>	28
5.2 <i>Realidad Aumentada</i>	29
5.2.1 <i>Preámbulo</i>	29
5.2.1.1 <i>TUI</i>	30
5.2.1.2 <i>Realidad Aumentada</i>	30
5.2.1.3 <i>Virtualidad Aumentada</i>	31
5.2.1.4 <i>Realidad Virtual</i>	31
5.2.2 <i>Realidad aumentada</i>	33

5.2.3 Un poco más de historia...	35
5.2.4 Tecnología	39
5.2.4.1 Hardware	39
5.2.4.2 Software	39
5.2.4.3 Técnicas de visualización	42
5.2.5 Aplicaciones	44
5.2.5.1 Educación	44
5.2.5.2 Turismo.....	44
5.2.5.3 Televisión	44
5.2.5.4 Entretenimiento.....	44
5.2.5.5 Arquitectura	45
5.2.5.6 Apoyo en tareas complejas.....	45
5.2.5.7 Publicidad	45
5.2.5.8 Comercio	45
5.3 Plataformas.....	46
5.3.1 APP	46
5.3.1.1 Android	47
5.3.1.2 iOS	49
5.3.1.3 Windows phone	50
5.3.2 Realidad aumentada	51
5.3.2.1 Qualcomm Vuforia	52
5.3.2.2 Metaio.....	52
5.3.2.3 Layar.....	53
5.3.2.4 ARToolKit.....	53
5.3.2.5 Total Immersion	54
5.3.2.6 Tango	54
5.3.3 Plataforma de desarrollo	55
5.3.3.1 Unity	55
CAPÍTULO 6	57
TRABAJO DESARROLLADO	57
6.1 Requisitos.....	57
6.2 Contenidos	59
6.3 Análisis del problema.....	60
6.3.1 Primeros pasos	60
6.3.2 Prototipado.....	61
6.3.2.1 Explicación del prototipo	64
6.3.3 Alternativa	72
6.3.4 Visita de Campo (viaje a Madrid).....	76
6.4 Realización	78
6.4.1 Unity.....	78
6.4.2 Interfaz	80
6.4.3 El Folleto	82
6.4.4 Realidad aumentada	86
6.4.5 Opciones	92
6.4.6 Idioma	92
6.5 Diseño Final.....	93
6.5.1 El Folleto	94

6.5.2 App.....	95
6.5.3 Opciones	100
CAPÍTULO 7	103
CONCLUSIONES	103
7.1 Conclusión.....	103
7.2 Líneas futuras.....	105
7.3 Aprendizaje.....	105
ANEXOS	108
ANEXO I. BUENAS PRÁCTICAS PARA LA SELECCIÓN DE TARGETS.	109
ANEXO II. CONTENIDOS OFRECIDOS POR ÉL CLIENTE.	111
TRÍPTICO.....	111
TEXTOS EN ESPAÑOL.....	112
TEXTOS EN INGLÉS	120
IMÁGENES	128
OTROS CONTENIDOS	131
<i>Videos</i>	131
<i>Audios</i>	131
<i>Modelo 3D</i>	131
ANEXO III. PROTOTIPO.	132
ANEXO IV. SCRIPTS.	138
USADOS EN EL FOLLETO	138
<i>Estructura ScripTableObject</i>	138
<i>Carga de texto, imágenes y audio</i>	138
USADOS EN LA REALIDAD AUMENTADA.....	141
<i>Objeto global, usado para mantener valores globales a todas las escenas.</i>	141
<i>Controlador del menu aumentado</i>	142
<i>Script que cambia la imagen según el targuet detectado.</i>	143
<i>Script que cambia el texto y el audio según el targuet detectado.</i>	144
<i>Script usado para manejar la rotación del modelo 3D, así como la activación de las posiciones.</i>	146
<i>Parser para controlar el lenguaje.</i>	149
ANEXO V. MARCADORES FINALES.....	150
PRIMER MARCADOR.....	150
SEGUNDO MARCADOR.....	150
TERCER MARCADOR	151
CUARTO MARCADOR.....	151
QUINTO MARCADOR.....	152
SEXTO MARCADOR	152
SEPTIMO MARCADOR	153
OCTAVO MARCADOR	153
BIBLIOGRAFÍA.....	154
ÍNDICE BIBLIOGRÁFICO	155

Índice de figuras

FIGURA 1 TRÍPTICO.....	17
FIGURA 2 MTRIP	19
FIGURA 3 GO AUGMENTED.....	19
FIGURA 4 STREET MUSEUM	20
FIGURA 5 MOMA AR EXPO	20
FIGURA 6 ALLARD PIERSON AR MUSEUM.....	21
FIGURA 7 FEATUREPHONES	24
FIGURA 8 LOGOTIPO DE GOOGLE PLAY Y APP STORE.....	25
FIGURA 9 VENTAJAS E INCONVENIENTES DE LAS APLICACIONES NATIVAS.	26
FIGURA 10 VENTAJAS E INCONVENIENTES DE APLICACIONES HÍBRIDAS.....	27
FIGURA 11 VENTAJAS E INCONVENIENTES DE APLICACIONES WEB.	28
FIGURA 12 GRÁFICO DEL CONTINUO DE VIRTUALIDAD.	29
FIGURA 13 REACTABLE.....	30
FIGURA 14 APLICACIÓN DE REALIDAD AUMENTADA.....	30
FIGURA 15 EYETOY	31
FIGURA 16 VIDEOJUEGO	31
FIGURA 17 VR SEMI-INMERSIVA.....	32
FIGURA 18 HTC VIVE	32
FIGURA 19 ESQUEMA DE UN SISTEMA AR.....	33
FIGURA 20 EJEMPLO DE AR.....	34
FIGURA 21 IMAGEN DEL SENSORAMA.....	35
FIGURA 22 IMÁGENES DEL PRIMER HDM.....	36
FIGURA 23 VIDEOPLACE.....	37
FIGURA 24 KARMA.....	37
FIGURA 25 VIDEO MAPPING.....	43
FIGURA 26 GRÁFICO DE IMPLANTACIÓN EN EL MERCADO DE SMARTPHONE.....	46
FIGURA 27 LOGOTIPO DE ANDROID.....	47
FIGURA 28 LOGOTIPO DE IOS.....	49
FIGURA 29 LOGOTIPO DE WINDOWS.....	50
FIGURA 30 LOGOTIPO DE VUFORIA.....	52
FIGURA 31 LOGOTIPO DE METAIO.....	52
FIGURA 32 LOGOTIPO DE LAYAR.....	53
FIGURA 33 LOGOTIPO DE UNITY.....	55
FIGURA 34 TARGET E INSTRUCCIONES EN EL TRÍPTICO.....	58
FIGURA 35 JUSTINMIND	62
FIGURA 36 EJEMPLOS DE ANUNCIOS Y REALIDAD.....	63
FIGURA 37 PORTADA DEL PROTOTIPO.....	64

FIGURA 38 PANTALLA PRINCIPAL DEL PROTOTIPO	65
FIGURA 39 PANTALLA DE REALIDAD AUMENTADA DEL PROTOTIPO.....	66
FIGURA 40 DATOS DE LA REALIDAD AUMENTADA DEL PROTOTIPO.	67
FIGURA 41 PANTALLA DE UNA DE LAS PUERTAS DEL PROTOTIPO.....	68
FIGURA 42 PANTALLA DE LA FACHADA DEL TEMPLO EN EL PROTOTIPO.....	69
FIGURA 43 PANTALLA DEL PLANO DEL PRIMER PISO DEL TEMPLO EN EL PROTOTIPO.....	70
FIGURA 44 PANTALLA DE DATOS DEL INTERIOR DEL TEMPLO EN EL PROTOTIPO.....	71
FIGURA 45 APARTADO DE LICENCIAS.....	72
FIGURA 46 CREACIÓN DE TARGUET.....	73
FIGURA 47 TARGETS EVALUADOS.	73
FIGURA 48 PRUEBA DE MARCADORES EN LA LAGUNA (OBJETO).....	74
FIGURA 49 PRUEBA DE MARCADORES EN LA LAGUNA (INTERIORES)	74
FIGURA 50 PRUEBA DE MARCADORES EN LA LAGUNA (EDIFICIOS)	75
FIGURA 51 LOGOTIPO DEL PROYECTO TAHUT.....	77
FIGURA 52 LOGOTIPO UTILIZADO COMO PRUEBA DE MARCADORES.	77
FIGURA 53 PRUEBA DE MARCADOR EN LA IMAGEN DE LA DIOSA.....	77
FIGURA 54 ESCENAS.....	78
FIGURA 55 ASSET.....	79
FIGURA 56 ESQUEMA BÁSICO DE UNITY.	79
FIGURA 57 EJEMPLO GUI.....	80
FIGURA 58 PRUEBAS DE UI.	81
FIGURA 59 UIBUILDER.....	82
FIGURA 60 COLORFUL BUTTONS.....	83
FIGURA 61 PRIMERA APROXIMACIÓN. FOLLETO	84
FIGURA 62 PRIMERA PRUEBA MARCADOR.....	86
FIGURA 63 SEGUNDA PRUEBA MARCADOR.	87
FIGURA 64 PRUEBAS AR. MENU.....	88
FIGURA 65 PRUEBAS AR. IMAGEN.	88
FIGURA 66 PRUEBAS AR. VIDEO.	89
FIGURA 67 PRUEBAS AR. MODELO 3D.	89
FIGURA 68 MARCADORES FINÁLES.	91
FIGURA 69 EJEMPLOS DEL IDIOMA DEFAULT.	93
FIGURA 70 ESCENA INICIAL	93
FIGURA 71 ESCENA DEL FOLLETO.....	94
FIGURA 72 ESCENA DE LA AR.	95
FIGURA 73 ICONO IMÁGENES.	96
FIGURA 74 ESCENA DE IMÁGENES.	96
FIGURA 75 ICONO TEXTO.....	97
FIGURA 76 ESCENA DEL TEXTO/AUDIO.	97
FIGURA 77 ICONO VIDEO.	98
FIGURA 78 ESCENA DEL VIDEO.....	98
FIGURA 79 ICONO MAPA.	99
FIGURA 80 ESCENA MODELO 3D.....	99
FIGURA 81 PANEL DE OPCIONES.....	100
FIGURA 82 PANEL DE AYUDA DE LA PANTALLA INICIAL Y DEL FOLLETO.....	101
FIGURA 83 PANEL DE AYUDA DE LA PARTE DE AR.	102
FIGURA 84 EJEMPLO BUEN TARGET.....	109
FIGURA 85 EJEMPLO BUEN TARGET MOSTRANDO BLOBS.	110
FIGURA 86 CARA EXTERIOR DEL TRÍPTICO.....	111

FIGURA 87 CARA INTERIOR DEL TRÍPTICO.....	111
FIGURA 88 MAPAS.....	128
FIGURA 89 DIBUJOS DE GRAFFITIS.....	128
FIGURA 90 IMÁGENES DE GRAFFITIS 1.....	129
FIGURA 91 IMÁGENES DE GRAFFITIS 2.....	130
FIGURA 92 IMÁGENES DE PORTADA.....	130
FIGURA 93 PANTALLAS DEL PROTOTIPO A.....	132
FIGURA 94 PANTALLAS DEL PROTOTIPO B.....	133
FIGURA 95 PANTALLAS DEL PROTOTIPO C.....	134
FIGURA 96 PANTALLAS DEL PROTOTIPO D.....	135
FIGURA 97 PANTALLAS DEL PROTOTIPO E.....	136
FIGURA 98 PANTALLA DEL PROTOTIPO F.....	137
FIGURA 99 FIRSTTARGET.....	150
FIGURA 100 SECONDTARGET.....	150
FIGURA 101 THIRDTARGET.....	151
FIGURA 102 FOURTHTARGET.....	151
FIGURA 103 FIFTHTARGET.....	152
FIGURA 104 SIXTHTARGET.....	152
FIGURA 105 SEVENTHTARGET.....	153
FIGURA 106 EIGHTHTARGET.....	153

Índice de tablas

TABLA 1 DISTRIBUCIÓN DE APP EN EL 2014.	46
TABLA 2 COMPARATIVA SDK'S SE AR.	51

Capítulo 1

Introducción

La realidad aumentada es una tecnología emergente, que combina elementos virtuales con la visión del mundo real, creando así experiencias visuales, que pueden ser usadas con gran utilidad en distintas aplicaciones, tanto en el ámbito social, como en los diversos sectores del mercado productivo, comercial y de servicios.

Hasta hace relativamente poco tiempo, la utilización de esta tecnología resultaba muy costosa y aparatosa, como para poder hacer uso de ella habitualmente y mucho menos por el usuario medio. No obstante a día de hoy, cualquiera puede hacer uso de la realidad aumentada, gracias a los dispositivos móviles de últimas generaciones, lo cuales nos ofrecen la capacidad computacional requerida para ello, que no podríamos haber imaginado hace tan solo unos pocos años.

En estos últimos años, hemos ido oyendo hablar cada vez más, del concepto de “Realidad aumentada” y “Realidad virtual” (AR/VR). Conceptos que hasta la fecha se tenían como “curiosos” y “futuristas”, pero que a día de hoy, han tomado repentinamente un carácter de “necesario” y/o “deseado” y “posible”.

La carrera tecnológica llevada a cabo por las principales compañías de entretenimiento visual, como *PlayStation*, *Microsoft*, *Valve* o *Nintendo*; así como por parte de los dos gigantes del software, *Facebook* y *Google*. Han dado lugar a un salto generacional muy importante dentro de la tecnología virtual y aumentada, la cual se ha ido aprovechado por la comunidad de desarrolladores, para crear nuevas experiencias y herramientas que puedan ser usadas en otros campos.

Un ejemplo del uso de la realidad aumentada, se puede encontrar hoy en día, en museos y centros culturales; donde se ofrece al visitante una herramienta de uso voluntario y autónomo, que completa la oferta cultural proporcionada por el centro; renovando además la imagen del mismo, al incorporar nuevas tecnologías que enlazan pasado con presente y futuro, pero que a la vez mejoran mucho y complementan la oferta convencional de estas instituciones, captando con ello nuevos usuarios y/o clientes, con un perfil adaptado a los tiempos.

Capítulo 2

Objetivo y justificación

2.1 Objetivo

La realización de este proyecto tiene como objetivo principal, la obtención del título de Ingeniero Informático, el cual, según el plan de estudios de la Universidad de La Laguna, se obtendrá tras la realización y defensa de un proyecto de final de carrera (PFC).

No obstante, también se busca aprender tecnologías de actualidad, que puedan favorecer una futura contratación en las nuevas empresas emergentes con las que nos podemos encontrar en los tiempos que corren.

2.2 Justificación

El proyecto, queda justificado, por la necesidad de la entidad tutelar gestora, encargada de los servicios de conservación del Espacio Cultural del Templo de Debod, dependiente del servicio de museos del Ayuntamiento de Madrid, pero con autonomía suficiente, para tomar las decisiones que atañen a ese espacio arquitectónico del Antiguo Egipto (2200 años de antigüedad), proveniente de una donación de dicho estado a la ciudad de Madrid, en agradecimiento por la colaboración de España, ante el llamamiento internacional que realizó la UNESCO, para salvar los templos de Nubia, ante el peligro de desaparición por la inminente construcción de la gran presa de Asuán.

La donación, se hizo en 1968, aunque se trasladaron los bloques hasta Alejandría, dos años más tarde, para ser embarcados hacia España, concretamente hasta Valencia, para posteriormente llevarlos a su destino en Madrid, que sería el emplazamiento del antiguo solar del cuartel de la montaña.

Tras resolver múltiples problemas burocráticos y de toma de decisiones técnicas de rehabilitación (por la falta de muchas partes); fue inaugurado el julio de 1972. Aunque hasta abril de 2008, no sería declarado Bien de Interés Cultural (BIC), por la Comunidad de Madrid.

Estas demoras en tratar adecuadamente como se merece este “espacio”, unido al uso inadecuado que se le dio en múltiples ocasiones durante este tiempo, hizo saltar la voz de alarma en Congresos de Egiptología Ibérica, celebrados en Madrid, Barcelona y La Laguna; que con el apoyo de la propia UNESCO, han hecho tomar conciencia de la

necesidad de tratar adecuadamente elementos patrimoniales de la importancia del Templo de Debob, sin dejar por ello de permitir visitas controladas para darlo a conocer y darle el uso cultural que se merece, priorizando ahora sí, su adecuada conservación.

Es en este contexto, en el que nos llega la oportunidad de participar en la elaboración de una propuesta moderna', adecuada tecnológicamente, económica y no agresiva. Propuesta que me hacen como objeto del proyecto de fin de carrera, utilizando la Realidad Aumentada en la aplicación que se me encarga, para que complementara a la recursos de los que ya disponían; que fuera posible de seguir ampliando y mejorando; que evitara los folletos convencionales en papel, por costosos económicamente y con rápida obsolescencia.

Me pareció un reto muy interesante, que cumplía con el objetivo, que tenía una justificación por sí mismo y que me motivaba a investigar y a afrontar el proyecto pendiente para poder titular.

Capítulo 3

Motivación

Antes de aceptar este proyecto, pasé algún tiempo de despacho en despacho tanteando en distintos departamentos a varios tutores; preguntando por posibles proyectos que tuviesen en mente, buscando uno que me llamase la atención. No quería realizar un proyecto que se basara en la optimización de un algoritmo, o que estuviera repleto de tablas, ecuaciones y textos aburridos. Deseaba encontrar algo más visual, actual y a ser preferible, que me ayudara a dar mis primeros pasos en el mundo de las aplicaciones móviles o los videojuegos.

Al comenzar la carrera, mi objetivo era poder dedicarme al mundo de los videojuegos, llegando algún día a formar parte de alguna empresa que se dedicase al sector y con suerte, llegar a desarrollar de alguno. No obstante, la manera más habitual de entrar en este mundo es mediante la rama artística, pero como no dispongo de dotes artísticas, la mejor alternativa que me quedaba era formarme en informática, por lo que acabe estudiándola.

No obstante a lo largo de la carrera, se presentaron pocas oportunidades que mostrasen algo de ese mundo de los videojuegos, lo cual visto con perspectiva es lógico. Para poder entrar en ese mundo hace falta una base sólida de conceptos esenciales y eso es lo que otorga la universidad en la carrera de informática.

Es por eso que, al poder elegir el campo de desarrollo de mi proyecto final, busqué algo que tuviera que ver con la industria del videojuego. Es por esto, que no me conformé y acepté propuestas de proyectos, que me podrían haber dado algo de interés o prestigio académico y seguí buscando.

Cuando me encontré con la posibilidad de desarrollar este proyecto, me ilusioné bastante. Si bien es cierto que al principio no parecía tener nada que ver con el campo que yo estaba buscando, ofrecía algunas características que me motivaban en ciertos aspectos.

Las aplicaciones móviles, eran algo que recientemente habían captado mi atención y la idea de incluir una tecnología, que por ese entonces se veía muy poco en las aplicaciones comerciales, como es la realidad aumentada (AR), lo hacía más atractivo; por lo que rápidamente comencé a interesarme muchísimo, más allá quizás, de las exigencias de realizar un buen proyecto.

Ciertas partes del proyecto tenían una buena expectativa de futuro, pues las compañías buscan continuamente desarrolladores de aplicaciones móviles; por lo que la idea de tener conocimientos en ese campo me llamaba bastante la atención.

Por si fuera poco, la realidad aumentada comenzaba a mencionarse, como una tecnología no tan futurista como se pensaba en un principio y los rumores de las grandes compañías dando sus primeros pasos comenzaba a motivar a la comunidad de desarrolladores. La idea de formar parte de ese mundo desde los comienzos me pareció una oportunidad que pocas veces te ofrece la vida formativa y profesional.

Por lo tanto, me resultó fácil tomar una decisión. Al final había tenido suerte y había encontrado un proyecto que me motivaba, pero mis sorpresas no terminaban ahí.

Al poco de empezar con el proyecto, se me dieron a conocer los requisitos que llevaba implícito y tras un estudio preliminar, me di cuenta de que la mejor manera de realizar el producto final, era mediante la plataforma Unity.

De esta forma, trabajaría con herramientas propias del ámbito de los videojuegos sin realizar un videojuego, lo cual ampliaba mi abanico formativo, hacia donde siempre había deseado, sin haber tenido la oportunidad, a la vez que me enfrentaba a desarrollar de una vez, el aplazado proyecto, que requería para poder titular, después de todos estos años.

Por lo tanto, a pesar de las sorpresas, dificultades y requerimientos, el proyecto ha resultado ser una motivación desde el principio hasta el final, por varias razones, pero la principal y más importante, por permitirme trabajar con algunas de las tecnologías que más me habían atraído a lo largo de mis estudios en la ingeniería informática y con las que siempre deseé encontrarme.

Capítulo 4

Antecedentes

4.1 Herramientas similares en Debod

El espacio cultural Templo de Debod, se encontraba actualmente desarrollando el denominado “*Proyecto Ta-hut*”, el cual constaba de un tríptico con texto breve sobre la explicación de la historia del templo, además de imágenes del mismo.

Dicho proyecto tenía el añadido de disponer de una aplicación de realidad aumentada, que se apoyaba en varios marcadores incluidos en el tríptico, los cuales, tras ser reconocidos por la aplicación de realidad aumentada, mostraban imágenes adicionales, enriqueciendo un poco más de esa forma, el contenido contemplado en el tríptico.

Esta aplicación estaba basada en una herramienta que hacía uso de elementos alojados en la nube, por lo que era necesaria una conexión a internet para su correcto funcionamiento. Este requisito esencial entraba en conflicto con la naturaleza del recinto, puesto que dentro del templo no llega señal de ningún tipo.



Figura 1 Tríptico

En el Anexo II, se podrá apreciar una imagen completa y más nítida del Tríptico.

4.2 Aplicaciones turísticas existentes

Como es lógico, no somos los primeros en apostar por la inclusión de tecnología aumentada, dentro de un espacio cultural y turístico; existen diversas aplicaciones enfocadas a mejorar la experiencia del turista como usuario, incluso existen compañías enfocadas en proporcionar a museos y atracciones turísticas, aplicaciones con tecnología aumentada.

Es algo cada vez más habitual, puesto que es una forma de ampliar la información que se puede poner a disposición del usuario, sin la necesidad de realizar grandes inversiones por parte del centro que las utilice, puesto que se aprovecha la tecnología de la que ya dispone el propio usuario, es decir su dispositivo móvil.

De esta forma, el centro solo tiene que preocuparse por el desarrollo y mantenimiento de la aplicación, lo cual es mucho más barato que editar miles de folletos informativos, como se venía haciendo hasta la fecha y que por otro lado, el contenido de la información quedaba limitado por el reducido espacio del propio folleto, además de limitar o impedir, los cambios en la información que ya se había impreso.

Por ello, no es extraño comprobar que existe diversidad de aplicaciones en el ámbito turístico, que hacen uso de la realidad aumentada, tanto en el ámbito internacional como incluso ya, en el nacional.

A continuación, a modo de ejemplo, mostraré las aplicaciones más interesantes que he encontrado a lo largo del desarrollo del proyecto:

4.2.1 MTrip

MTrip nacía en una época donde cualquier turista que se encontrara fuera de su nación, se veía en la necesidad de pagar una tarifa extra y a menudo excesiva para poder hacer uso de sus datos móviles, el denominado “roaming”, lo cual, negaba la posibilidad de los turistas de acceder a contenidos útiles, como mapas o guías turísticas alojadas en la nube.

Por ello, MTrip te permitía organizar tus viajes antes de realizarlo, guardando las rutas y lugares de interés en la memoria interna del dispositivo, lo cual permitía usarlo sin una conexión a internet.

Entre sus funcionalidades se encontraba el uso de la realidad aumentada, basada en el posicionamiento, la cual te indicaba elementos cercanos a tu posición e incluso te los remarcaba en la pantalla.



Figura 2 MTrip

4.2.2 Go Augmented

Aplicación de realidad aumentada desarrollada por el Museo Británico, la cual permite a los visitantes disfrutar de contenido virtual al enfocar los usuarios, sus dispositivos móviles hacia los objetos expuestos.

Esta aplicación hará uso, presumiblemente, de marcadores dispuestos cerca de los objetos expuestos y contemplados en la aplicación.

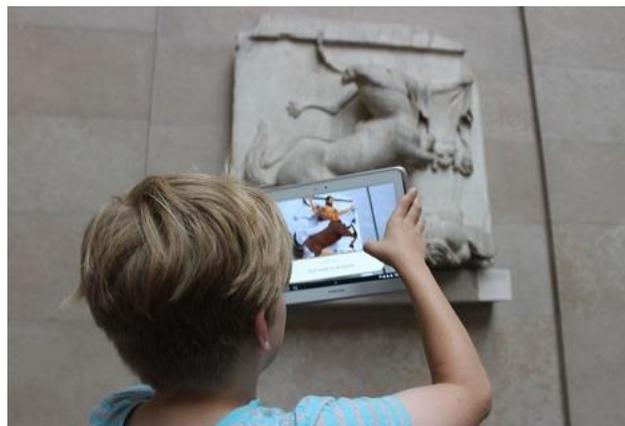


Figura 3 Go Augmented

4.2.3 Street Museum

Desarrollada por el Museo de Londres, la aplicación no solo se limita al museo y sus alrededores, sino que es utilizable a lo largo de toda la ciudad.

Dicha aplicación tiene como objetivo mostrar fotos tomadas a lo largo de la ciudad, dependiendo del punto en el que se encuentre el usuario, haciendo uso del GPS del dispositivo.



Figura 4 *Street Museum*

4.2.4 Exposición en el MOMA

El museo Moma de Nueva York, realizó una exposición que no era física, puesto que las habitaciones se encontraban completamente vacías. De esta forma, para poder ver las obras de arte de la exposición, era necesario hacer uso de una aplicación de realidad aumentada, la cual mostraba una habitación llena de obras virtuales.

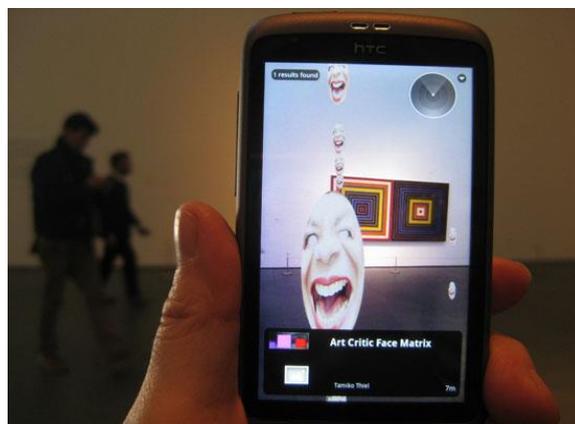


Figura 5 *MOMA AR Expo*

4.2.5 Museo de Allard Pierson

Este museo, optó por la opción de colocar pantallas giratorias en cada uno de sus objetos expuestos, de tal manera que el usuario podía hacer uso de dichas pantallas para ver el contenido aumentado asociado.



Figura 6 Allard Pierson AR museum

Pero no solo se utilizan estas aplicaciones turísticas, en países tecnológicamente muy avanzados; pues ya en diversos espacios de muchas de las CCAA de España, se van viendo lugares y espacios donde se aplica esta tecnología, pues la velocidad a la que se extiende su uso, es cada vez mayor; especialmente, en el ámbito turístico, como valor añadido o elemento diferenciador, para mejorar la oferta de experiencias a sus visitantes:

4.2.6 Costa del Sol-Málaga

Se trata de una aplicación desarrollada por la Diputación General de Málaga, en conjunto con la empresa Orange, la cual hace uso de la realidad aumentada, para mostrar los puntos de interés turísticos que se encuentren cerca de la posición del usuario.

4.2.7 Guía turística de Segovia

Realizada por el Ayuntamiento de Segovia, tiene la misma funcionalidad y objetivos que la aplicación realizada en Málaga. Esto es, mostrar según la posición del usuario, los puntos de interés turístico cercanos al lugar donde se encuentra.

Capítulo 5

Marco Teórico

En este capítulo intentaremos dar a conocer una serie de conocimientos básicos necesarios para el correcto entendimiento del proyecto realizado. Por lo tanto, al tratarse de una app con tecnología aumentada, empezaremos explicando qué es una **aplicación móvil** y cuando surgieron, seguidamente intentaremos aclarar el concepto de **realidad aumentada**, cuando surgió y a que hace referencia. Para terminar haciendo referencia, a las **plataformas** existentes que pueden servir, para realizar una aplicación como la que se ha llevado a cabo en este trabajo.

5.1 Aplicación Móvil (app)

5.1.1 Preámbulo

Una App es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes (*Smartphone*), tabletas y otros dispositivos móviles, las cuales, permiten al usuario realizar alguna tarea específica, ya sea de ámbito profesional o de ocio.

Dichas aplicaciones, se suelen encontrar disponibles a través de diversas plataformas de distribución, ya sean oficiales o no. Me refiero, a distribución oficial a las operadas, por las compañías propietarias de los sistemas operativos móviles vigentes u otras compañías asociadas.

Podemos encontrarnos con aplicaciones gratuitas, de pago, o las denominadas “*freemium*”, las cuales inicialmente son gratuitas, pero dan la posibilidad de realizar compras integradas, para mejorar la experiencia del usuario (normalmente este tipo de aplicación está más orientada al entretenimiento).

El término App se volvió rápidamente popular con el auge de los teléfonos inteligentes, tanto es así, que en 2010 fue declarada como palabra del año por la *American Dialect Society*.

5.1.1.1 Características

Al ser aplicaciones residentes en los dispositivos, están por necesidad, escritas en algún lenguaje de programación compilado (normalmente distinto según el sistema operativo en el que operen) y su funcionamiento, diseño y recursos se encaminan a aportar una serie de funcionalidades tales como:

- Acceso rápido y sencillo a los datos sin necesidad de autenticación por parte del usuario, para ello, se suelen pedir los permisos necesarios en la instalación de la app.
- Almacenamiento de datos personales.
- Gran versatilidad respecto a su utilización.
- Funcionalidades específicas.
- Mejorar la conectividad y disponibilidad de servicios.

5.1.1.2 Distribución

En cuanto a su distribución, existen distintas tiendas donde descargar las aplicaciones, normalmente enfocadas según el SO (*Sistema Operativo*) del dispositivo, las cuales suelen ser gestionadas por la misma empresa que desarrolla el sistema, u otras independientes. Estos son algunos de los más conocidos:

- Google Play: Perteneciente a Google Inc. es la plataforma por excelencia, donde se alojan las aplicaciones disponibles para teléfonos con sistema operativo Android.
- App Store: Fue el primer servicio de distribución de aplicaciones en salir, concretamente, el 10 de julio de 2008. En esta plataforma se encuentran todas las aplicaciones disponibles para el sistema iOS, ya sea smartphone ó PC convencional.

Además de esta versión, en 2012 se creó la *App Store Volume* la cual solo está disponible en EEUU y para empresas, dicha versión de la tienda de Apple, permite la compra en grandes volúmenes de aplicaciones, las cuales pueden ser usadas para mejorar el negocio o para regalarlas a modo de promoción.

- Windows Store: Es la plataforma de distribución de Microsoft, para los dispositivos que cuentan con el sistema operativo Windows, unificado

recientemente con Windows 10 en todas sus plataformas (Smartphone, PC, Consolas).

- Amazon Appstore: Se trata de una aplicación móvil, que a su vez realiza la función de distribución de otras aplicaciones móviles, actualmente solo para los dispositivos con SO Android.
- F-Droid Es un repositorio de aplicaciones para Android, donde únicamente se alojan aplicaciones de software libre y código abierto.

5.1.2 Historia de las Aplicaciones Móviles (App)

Si bien es cierto que no se sabe a ciencia cierta cuál fue el inicio de las App, si se sabe, que llevan con nosotros mucho más tiempo, de lo que comúnmente se pueda pensar. Hoy en día cuando hablamos de App, nos vienen a la mente las actuales y modernas aplicaciones de nuestros teléfonos con sus respectivos iconos que nos dan acceso directo a sus funcionalidades, pero sus orígenes, van mucho más allá.

En los años 90 se empezó a popularizar el uso de los teléfonos móviles; al principio, no eran más que teléfonos con contestador integrado, pero poco a poco, se le fueron añadiendo más funcionalidades: calendario, tonos de llamada (*Ringtone*), mensajería, fotografía, juegos... Todo ello como, aplicaciones móviles integradas directamente en el dispositivo, los cuales, se denominaban “featurephones”, distinguibles por sus pantallas pequeñas, las cuales aún no eran táctiles.



Figura 7 Featurephones

Uno de los ejemplos de este tipo de aplicaciones fue el famoso “Snake”. El juego lanzado por Nokia en sus dispositivos móviles, se convirtió en el videojuego para móviles con mayor aceptación en la historia, tanto es así, que muchas personas preferían comprarse el famoso nokia 6110, sólo por tener dicho videojuego.

En el año 2000 la irrupción de la tecnología WAP (*Wireless Application Protocol*) permitió una mayor capacidad a la hora de descargar juegos y aplicaciones distribuidas por los operadores de telefonía. No obstante su coste era elevado y posiblemente esto produjo, que no tuviese mucha aceptación por parte del mercado.

Fue en 2008 cuando Apple lanza la Apple Store y publica su primer SDK, orientado a sus primeros dispositivos, poco después Android se subía al carro del desarrollo de aplicaciones móviles, liberando su SDK y creando su propia plataforma de distribución, la “Android Market”, la cual pasaría a llamarse Google Play en 2012.



Figura 8 Logotipo de Google Play y App Store

Desde entonces, las App forman parte de nuestra vida cotidiana, convirtiéndose en una herramienta conocida por casi todo el mundo, dando lugar a una innumerable cantidad de aplicaciones, enfocadas a todo tipo de tareas, lo cual le ha generado un mercado muy considerable.

5.1.3 Tipos de aplicaciones

A la hora de realizar una App, llegará el momento en el que se tenga que pensar cómo va a estar construida técnicamente. De manera general, se pueden dividir las aplicaciones móviles en tres tipos distintos:

- Nativas.
- Híbridas.
- Web.

5.1.3.1 Nativas

Una App nativa, en principio es una aplicación desarrollada directamente en el lenguaje propio de cada terminal. Estos son:

- iOS Swift, aunque hasta hace poco se usaba Objective C.
- Android Java.
- Windows C# Y .NET.
- BlackBerry C++

Por lo tanto, si vamos a desarrollar nuestra aplicación en nativo, tendremos que crear una distribución, en cada uno de los lenguajes que queramos, si queremos abarcar un mercado mayor, ya que si solo lo hiciéramos en un lenguaje, nos estaríamos limitando a los dispositivos que soporten ese SO.

Una de las características fundamentales de este tipo, es que suelen proporcionar un resultado más robusto y fluido, puesto que al estar programado en el lenguaje nativo, aprovecha mejor las cualidades de cada dispositivo. No obstante tiene algunas ventajas, al igual que por otro lado cuenta con ciertos inconvenientes, que podemos resumir en la siguiente tabla:

Ventajas	Inconvenientes
<ul style="list-style-type: none">• Acceso completo al dispositivo• Mejor experiencia del usuario• Visibilidad en APP Store• Envío de notificaciones o "avisos" a los usuarios• La actualización de la app es constante	<ul style="list-style-type: none">• Diferentes habilidades / idiomas / herramientas para cada plataforma de destino• Tienden a ser más caras de desarrollar• El código del cliente no es reutilizable entre las diferentes plataformas

Figura 9 Ventajas e inconvenientes de las aplicaciones nativas.

5.1.3.2 Híbridas

Consisten en aplicaciones que contienen en su interior el navegador web del dispositivo. Son desarrolladas mediante frameworks externos, basados en lenguajes de programación web (HTML, CSS, Js), los cuales realizarán las conversiones y llamadas al sistema, dependiendo de a qué dispositivo se haga la compilación (Build).

Uno de los mayores exponentes en este campo es *PhoneGap*, en las tres capas de programación web, en el que el mismo se encarga, de hacer las traducciones necesarias a la hora de compilar.

Si bien es cierto que hoy en día es más fácil acceder al hardware de los dispositivos, mediante este tipo de desarrollo, de lo que era hace unos años, sigue siendo una opción menos robusta y fluida que una aplicación en nativo, básicamente porque la conversión a lenguaje nativo desde el lenguaje base, no siempre será tan buena como la opción de programarla íntegramente en la opción nativa.

Normalmente se escoge esta opción, si sabemos que no haremos mucho uso de elementos internos de nuestros dispositivos (giroscopio, acelerómetro...), ya que su acceso puede ser complicado, engorroso o incluso puede dar lugar a fallos.

En la siguiente tabla podemos comprobar algunas de sus ventajas y desventajas.

Ventajas	Inconvenientes
<ul style="list-style-type: none">• Es posible distribuirla en las tiendas de iOS y Android.• Instalación nativa pero construida con JavaScript, HTML y CSS• El mismo código base para múltiples plataformas• Acceso a parte del hardware del dispositivo	<ul style="list-style-type: none">• Experiencia del usuario más propia de la aplicación web que de la app nativa• Diseño visual no siempre relacionado con el sistema operativo en el que se muestre

Figura 10 Ventajas e inconvenientes de aplicaciones híbridas.

5.1.3.3 Web

No es una aplicación móvil propiamente dicha, sino una web a la que se accede desde el navegador del dispositivo y la cual adapta el formato de su contenido, según las dimensiones de la pantalla del smartphone en cuestión.

Dicha adaptación suele reubicar el contenido (imágenes, texto, botones), de tal manera que el resultado tenga aspecto de app, dándole al usuario la impresión de que está manejando una app como otra cualquiera.

Sobra decir que el lenguaje usado en este tipo de aplicaciones es puramente web (HTML, CSS, Js), y que es portable a todo tipo de plataforma (Android, iOS, Windows.), además de ser la más barata de todas, ya que no necesitas un especialista en las plataformas correspondientes, sino que es suficiente con especialistas de desarrollo web.

Sin embargo, al tratarse de una web modificada, su mayor inconveniente es la necesidad de una conexión a la red, ya que sin internet, la app no funcionaría.

En la siguiente tabla se reflejan el conjunto de ventajas e inconvenientes de este tipo de aplicaciones web.

Ventajas	Inconvenientes
<ul style="list-style-type: none">• El mismo código base reutilizable en múltiples plataformas• Proceso de desarrollo más sencillo y económico• No necesitan ninguna aprobación externa para publicarse (a diferencia de las nativas para estar visibles en app store)• El usuario siempre dispone de la última versión• Pueden reutilizarse sitios "responsive" ya diseñados	<ul style="list-style-type: none">• Requiere de conexión a internet• Acceso muy limitado a los elementos y características del hardware del dispositivo• La experiencia del usuario (navegación, interacción..) y el tiempo de respuesta es menor que en una app nativa• Requiere de mayor esfuerzo en promoción y visibilidad

Figura 11 Ventajas e inconvenientes de aplicaciones Web.

5.2 Realidad Aumentada

5.2.1 Preámbulo

Para poder comprender mejor las diferencias y conceptos que vendrán a continuación, deberíamos explicar qué tipos de “realidades” existen, desde el punto de vista computacional.

Según Paul Milgram y Fumio Kishino (1994), la realidad se podría definir como una escala continua que abarca desde el entorno real, hasta un entorno virtual puro. Es lo que denominaron como *Virtuality Continuum* (Continuo de la Virtualidad).



Figura 12 Gráfico del continuo de virtualidad.

En dicho gráfico, podemos observar la realidad tal y como la conocemos, pero a medida que nos vamos desplazando hacia la izquierda, se van incluyendo más elementos virtuales, hasta que llegado el punto, se reemplaza el mundo real, por uno distinto e irreal, creado totalmente con contenidos virtuales.

El extenso campo que se encuentra entre ambos opuestos es lo que denominamos realidad mixta. Dentro de esta clasificación entran varios tipos de realidad mixta. Si partimos desde nuestra “realidad”, diríamos, que nos encontramos con los distintos tipos:

5.2.1.1 TUI

Las *Tangible User Interfaces (TUI)* o Interfaces Tangibles de Usuario permiten, mediante el uso de objetos reales, generar contenido digital. El ejemplo más conocido de este tipo de tecnología es quizás *Reactable*, una mesa con pantalla digital, la cual genera diferentes sonidos según el objeto que pongas. Llegando a poder generar pequeñas piezas musicales con la conjunción de los distintos objetos que se coloquen.



Figura 13 *Reactable*

5.2.1.2 Realidad Aumentada

La *Augmented Reality (AR)* o Realidad Aumentada, se sitúa en el extremo del continuo más cercana a la realidad propiamente dicha. Esto se debe a que su funcionalidad radica en incluir elementos virtuales en el mundo real. Dentro de esta categoría hay varias opciones, las cuales explicaremos detalladamente más adelante.



Figura 14 *Aplicación de Realidad Aumentada.*

5.2.1.3 Virtualidad Aumentada

Este concepto no es tan famoso como los anteriores, dado que suele confundirse con la realidad virtual. Fundamentalmente, se basa en la inclusión de un elemento real, dentro de un entorno virtual.

Un claro ejemplo de esta tecnología se vio con la salida al mercado del EyeToy, el complemento de PlayStation de Sony. En él, el usuario era captado por una cámara e incluido en el videojuego, de tal manera que sus movimientos se trasladaban al propio videojuego.



Figura 15 EyeToy

5.2.1.4 Realidad Virtual

La Virtual Reality (VR) o Realidad Virtual, es el extremo opuesto en el continuo a nuestra realidad sensorial. Se trata de un entorno generado completamente por un ordenador, en el que existen tres grados básicos:

- Realidad Virtual No inmersiva son los más conocidos, al tratarse de la proyección en una pantalla del mundo virtual. Los videojuegos típicos se pueden categorizar dentro de este tipo.



Figura 16 Videojuego

- Realidad Virtual Semi-inmersiva se trata de entornos donde el usuario se ve rodeado de pantallas, en las cuales se proyecta el mundo virtual. A veces se pueden acompañar las imágenes, de otros dispositivos que enriquezcan la experiencia, tales como olores, aire, sonidos... pero el usuario siempre notará la diferencia entre la realidad y el mundo virtual.



Figura 17 VR Semi-inmersiva.

- Realidad Virtual Inmersiva es la variante más completa, donde el usuario tiene la sensación de estar dentro del videojuego. Esta idea se consigue involucrando todos los sentidos que tengan que ver con el entorno simulado. Actualmente está muy de moda, gracias a los últimos dispositivos sacados al mercado, tales como las HTC Vive, Oculus rift...



Figura 18 HTC Vive

5.2.2 Realidad aumentada

En este apartado, definiremos en profundidad el concepto de Realidad Aumentada, ya que es el que nos atañe para este proyecto.

La Realidad Aumentada (AR) se define como la idea de visualizar a través de un dispositivo tecnológico, el entorno físico, junto con elementos virtuales, creando así una realidad mixta en tiempo real. De ahí la elección del nombre de “realidad aumentada” (añadir/aumentar la realidad con contenido virtual). Esta es la principal diferencia con la Realidad virtual, puesto que no se sustituye la realidad física.

Aunque quizás la definición más aceptada por el colectivo es la que dio Ronald Azuma en 1997. Dicha definición se fundamenta en tres conceptos básicos que ha de tener todo aquello que denominemos realidad aumenta:

- Combina elementos virtuales y reales. Totalmente necesario.
- Es interactiva en tiempo real. Esto quiere decir, que los contenidos son generados en el mismo instante en el que el usuario los visualiza en el mundo real.
- Esta registrada en 3D. Hace referencia a que solo puede mostrar modelos en 3D, aunque por razones obvias no se toma mucho en cuenta esta última característica, ya que si así fuera, quedarían excluidas las muestras de video, imágenes, audios, etc.

Un esquema básico de un sistema de realidad aumentada suele contener, por lo tanto, una cámara que captura las imágenes del mundo real, conectada a un ordenador que realiza los cálculos necesarios para lograr introducir los elementos virtuales dentro de la escena real, proyectando el resultado final hacia los ojos del usuario.

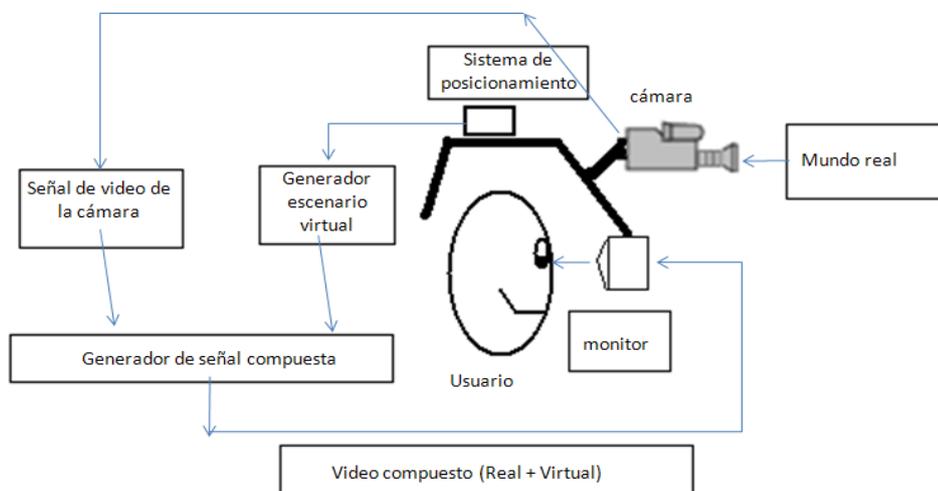


Figura 19 Esquema de un sistema AR.

Un ejemplo de realidad aumentada podría ser el siguiente, el usuario está dando un paseo por la zona comercial y visualiza en un escaparate cercano, un par de zapatos de muestra, se acerca y saca de su bolsillo su smartphone, al enfocarlo hacia el escaparate, van apareciendo los precios de los distintos productos allí mostrados, así como su nombre. Al hacer click en alguno de ellos, aparece una breve descripción y varias reseñas de otros usuarios.



Figura 20 Ejemplo de AR.

En nuestro ejemplo, podemos diferenciar dos tipos de objetos, perteneciente cada uno de ellos a dos realidades distintas:

- Realidad física: los zapatos, las estanterías, los complementos, etc.
- Realidad virtual: los bocadillos en los que se muestra el nombre y precio del producto, la interfaz gráfica, etc.

De donde se aprecia, que en nuestro ejemplo, se cumplen dos de las características antes mostradas:

- **¿Contienen elementos reales y virtuales?** Si, pues zapatos y precio son ejemplos de ello.
- **¿Es interactiva en tiempo real?** En nuestro ejemplo suponemos que sí, que los elementos aparecen nada más enfocar los productos con el móvil y que al hacer click en ellos aparecen nuevos datos.
- **¿Está registrada en 3D?** En este caso podemos ver como el contenido aumentado no es un modelo 3D sino una textura 2D. Es por ello que decimos que no es una característica muy necesaria, ya que la mayoría de las aplicaciones se basan en contenido que no necesariamente tiene que ser 3D.

5.2.3 Un poco más de historia...

En 1957 Morton Heilig, filósofo, visionario y realizador de cine, comienza a trabajar en el prototipo de una máquina con aspecto similar a las futuras, máquinas de arcade de las recreativas (inexistentes en aquel momento). Dicha máquina se llamó Sensorama.

El Sensorama, tenía una cabina en la que se proyectaban imágenes en 3D, junto con sonido envolvente, vibraciones en el asiento y viento simulado, etc. De esta forma, el usuario tenía la sensación de estar dentro del metraje.

A primera vista y siguiendo la clasificación que hicimos anteriormente, el sensorama tiene más similitudes con la realidad virtual (a partir de ahora nos referiremos a este tipo de realidad por sus siglas en inglés "VR"). No en vano, se considera a Heilig el padre de la VR. Sin embargo, se podría discutir que se trataba de realidad aumentada (AR, a partir de ahora me referiré a esta realidad mediante sus siglas), puesto que el video que se mostraba en la pantalla, no estaba editado. El contenido adicional (vibraciones, aire, sonido, etc.) se añadían a lo largo del metraje, dependiendo de en qué momento se encontrase, por lo tanto, era información adicional al video mostrado.

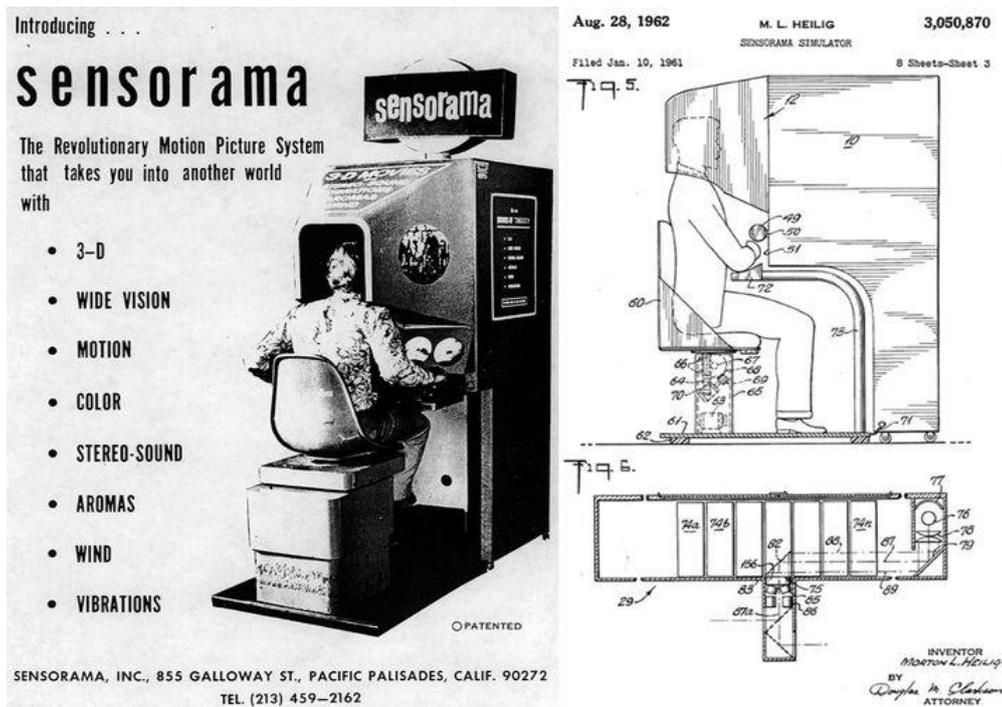


Figura 21 Imagen del Sensorama.

Por lo tanto, podemos comprobar como en sus comienzos, la VR/AR han ido de la mano, compartiendo sus orígenes, confundándose muchas veces a lo largo de su historia y no será hasta bastante más adelante, cuando la tecnología se perfeccione lo suficiente, como para dividir las en dos ramas claramente diferenciadas.

Más adelante, en 1966, el profesor de Ingeniería Eléctrica de Harvard, Ivan Sutherland creó un dispositivo, que podría definirse como el sucesor de las actuales Oculus Rift y compañía. Lo llamó *“Human Mounted Display”* (HDM). No obstante, lejos de parecerse a sus sucesoras, dicha máquina no era en absoluto algo que pudieses llevar en el día a día; de hecho, era tan pesada que se tenía que colgar del techo. Sin embargo, supuso un gran avance, a pesar de que solo mostraba modelos 3D básicos (de alambre).

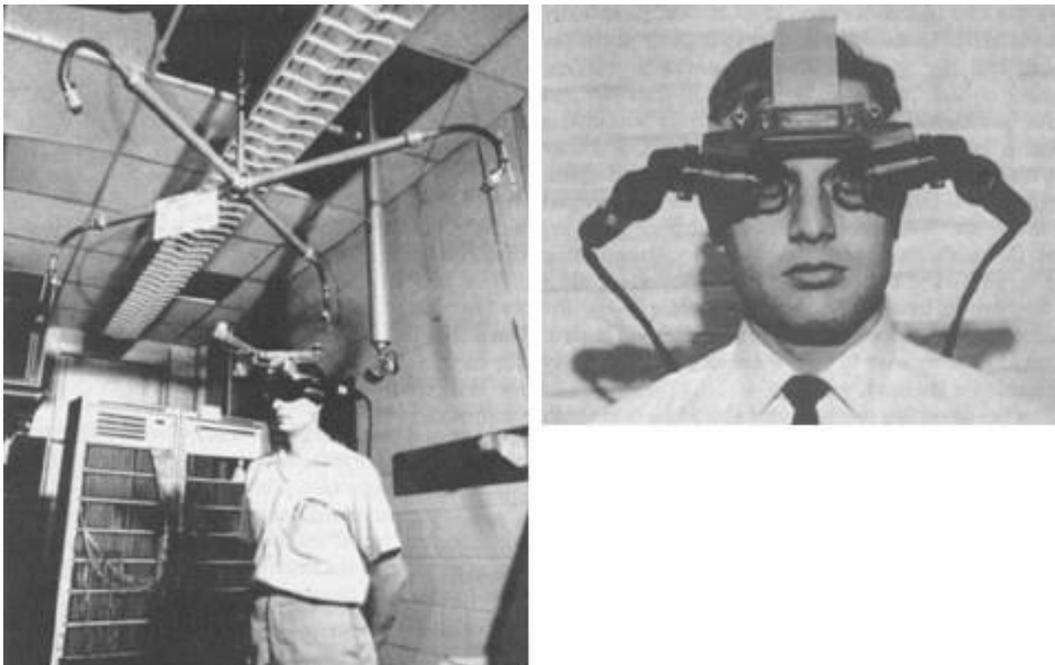


Figura 22 *Imágenes del primer HDM.*

En 1985, Myron Krueger creó VideoPlace. Esta tecnología es la predecesora del famoso Kinect de Microsoft, ya que era el ser humano el que interactuaba con el contenido digital proyectado en una pantalla. Fue la primera vez que los dos mundos se unían sobre el mismo plano y aunque es cierto que si hoy día vemos imágenes y videos de esta tecnología, como algo bastante básico; también cabe destacar que para la época, fue un gran hito, ya que daba la oportunidad de interactuar los dos mundos en tiempo real, con total libertad por parte del usuario.

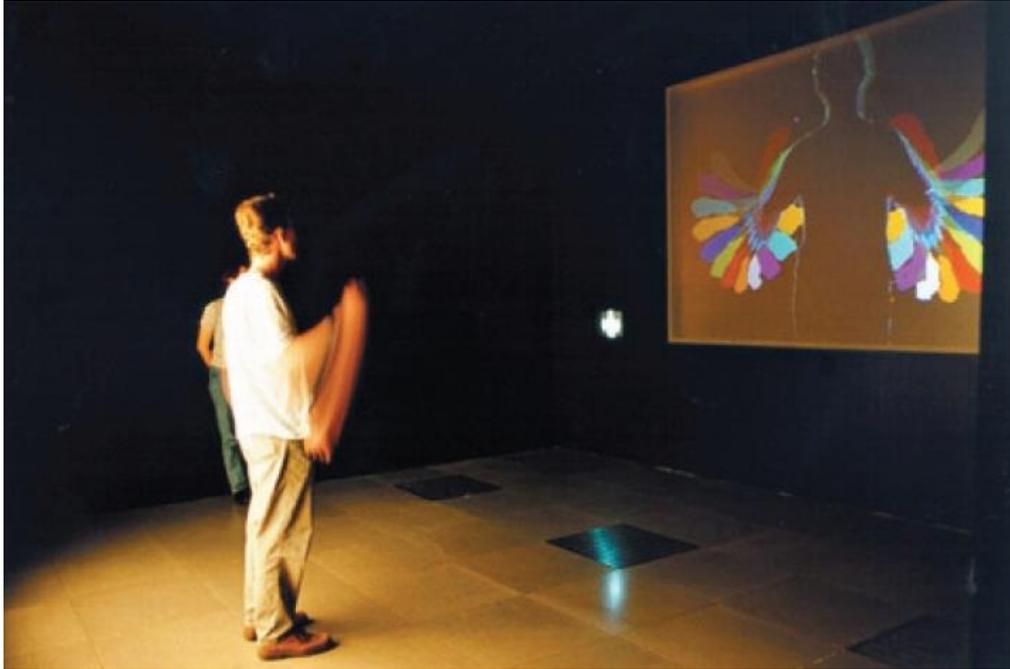


Figura 23 *VideoPlace.*

En 1992, Tom Caudell, ingeniero en Boeing, crea el término Realidad Aumentada al explicar su software, el cual imprimía los planos del cableado sobre las piezas producidas. Al mismo tiempo L-B.Rosenberg desarrollaba para las fuerzas aéreas de los Estados Unidos, la que se considera como la primera aplicación de realidad aumentada, en la que la misma aplicación, ayudaba al usuario a resolver ciertas tareas a medida que se presentaban, siendo así una especie de guía.

Otro gran hito dentro de la realidad aumentada sucedía durante el mismo año en la Universidad de Columbia, donde Steven Feiner, Blair MacIntyre y Doree Seligmann, creaban KARMA (Knowledge-based Augmented Reality for Maintenance Assistance), un HMD que proyectaba imágenes 3D sobre una impresora, para dar instrucciones sobre cómo cambiar los cartuchos de tinta.



Figura 24 *KARMA*

A partir de entonces, los avances en el campo de la realidad VR/AR fueron dando pasos agigantados, con la creación por parte de Nintendo en 1995 de Virtual Boy, un HDM que compatibilizaba su consola de sobremesa, con la creación de ARToolkit por parte de Hirokazu Kato en 1999, "AR wiktude"; saliendo a la venta con los primeros dispositivos Android en el 2008, mientras que Google presentaba sus gafas de AR (las Google Glass) en el 2012 y Niantic (en colaboración con Google), sacaba a la venta Ingress en 2013, el juego más exitoso para dispositivos móviles con tecnología aumentada hasta la fecha.

Y en la actualidad, nos encontramos con que varias grandes compañías están apostando fuertemente por la AR/VR:

- Valve: HTC Vive.
- Facebook: Oculus Rift.
- Microsoft: Hololens.
- Sony: PlayStation VR.
- Google: project Tango.
- Apple: la compra de Metaio.
- Nintendo: Pokemon GO.

Las AR/VR están viviendo sus inicios dorados, todo el mundo quiere ser el pionero en este campo, puesto que como veremos más adelante, sus aplicaciones son innumerables, abarcando casi todos los sectores del mercado, por lo que quien se posicione como líder, obtendrá una considerable ventaja en el futuro próximo.

5.2.4 Tecnología

5.2.4.1 Hardware

Normalmente, los dispositivos de Realidad aumentada constan de dos partes básicas. El “headset”, que lleva incorporado el sistema GPS, necesario para localizar la posición del usuario y un “display”, el cual mostrará al usuario el contenido virtual añadido a la imagen real. Cómo “display” se suelen emplear:

- Pantallas ópticas transparentes (Optical See-through Display), las cuales proyectan el contenido virtual sobre una lente a través de la cual, el usuario puede ver sin necesidad de un ordenador. Un ejemplo podrían ser las Google Glass.
- Pantallas de mezcla de imágenes (Video-mixed Display), el contenido virtual es proyectado sobre una pantalla digital; el ejemplo más claro de esto es un smartphone, donde podemos ver la realidad, a través de la cámara del dispositivo.

Los smartphone de hecho, se están convirtiendo en el medio más usado por la realidad aumentada, básicamente por la comodidad que ello implica. La tecnología aumentada aún es muy costosa computacionalmente, ya que se necesita de un ordenador lo bastante potente como para moverla sin problema. Los smartphone de hoy en día, son ordenadores de bolsillo más potentes, que muchos ordenadores de sobremesa de hace 10 años y disponen de todo el hardware necesario, para que funcione una aplicación de realidad aumentada en ellos (display, videocámara, gps, etc.), y más aún (acelerómetro, giroscopio, sensores...), por no contar las increíbles capacidades de la CPU's modernas, con la capacidad de memoria RAM actuales en dichos dispositivos.

Así las cosas, con el auge del mercado en pleno crecimiento en este ámbito, muchas compañías están optando por gafas con la tecnología suficiente, como para poder representar el contenido aumentado en tiempo real; algunas de ellas con el soporte del smartphone, otras con su propio sistema (los cuales suelen hacerlas más pesadas y voluminosas), pero aun soportables para el uso cotidiano.

5.2.4.2 Software

Obviamente el hardware es una parte fundamental de esta tecnología, pero es en el software donde radica la magia, al menos en la mayoría de los casos.

Para entender mejor este apartado, conviene explicar los distintos grados de complejidad, que se puede encontrar en cualquier aplicación de realidad aumentada.

Dichos niveles fueron propuestos por Lens-Fitzgerald, cofundador de Layar, uno de los navegadores de AR más extendidos actualmente:

- Nivel 0 (*Physical World Hyper Linking*). Son las que se apoyan en códigos de barra o códigos QR, no mostrando ningún contenido 3D; solo sirven como enlaces a páginas web o a contenido alojado en la nube u otro servidor.
- Nivel 1 (*Marker Based AR*). Las aplicaciones se apoyan en elementos físicos de pequeño tamaño, para mostrar el contenido virtual. Originalmente solo existían los llamados FrameMarkers; imágenes cuadradas con bordes específicos, que guardaban gran similitud en su funcionamiento con un código de barras típico. Poco después se permitieron el uso de imágenes, objetos rectangulares, cilíndricos y finalmente cualquier tipo de objeto pequeño.

Cabe destacar que hay un conjunto de buenas prácticas [Anexo I], a la hora de elegir un marcador, ya que la tecnología actual prefiere un determinado tipo de imágenes u objetos, para asegurar el correcto funcionamiento de la aplicación.

- Nivel 2 (*Markless AR*). Hacen uso del hardware del dispositivo, concretamente del GPS, giroscopio o acelerómetro. Detectan la posición en la que se encuentra el usuario y hacia donde está dirigiendo el dispositivo, mostrando el contenido aumentado y superponiéndolo en los objetos reales.
Este tipo de aplicaciones suele fallar mucho, debido a que las condiciones meteorológicas y el entorno en sí es muy variable. Dentro de este nivel se puede incluir las técnicas de SLAM o PTAM, las cuales al igual que la robótica, hacen uso de una nube de puntos, para reconocer el entorno y decidir si con esos datos recolectados, hay algún contenido virtual que añadir.
- Nivel 3 (*Augmented Vision*). Es el menos habitual de ver en el mercado real y el más habitual de ver en los anuncios comerciales como fácilmente factible. Estarían formados por una tecnología personalizada, que sin necesidad de marcadores, pudiera entender el entorno, proporcionando a la vez, un contenido aumentado específico para cada usuario. Es lo que pretendía ser Google glass.

Una vez mencionados los distintos tipos de niveles existentes, podemos comprobar que nos encontramos en una época, donde la mayoría de las aplicaciones de realidad aumentada se encuentran comprendidas entre el nivel 1 y 2, ya que o bien hacen uso de marcadores (imágenes, objetos, etc.), o bien utilizan la posición del dispositivo para mostrarnos el contenido (típicas guías turísticas que nos marcan el camino, o nos dicen la distancia hacia nuestro objetivo).

Lejos quedan los años, donde se nos vendían productos con realidad aumentada que no eran más que códigos QR, los cuales nos abrían el navegador con la página web del producto ya marcada. Pero al igual que hoy en día, nadie que se haga llamar desarrollador de AR utilizaría códigos QR, salvo circunstancia mayor.

En unos pocos años pasará lo mismo con los marcadores, ya que esta tecnología avanza a pasos agigantados, por lo que cualquier desarrollador que se precie, se verá obligado a profundizar en las tareas más complejas de la AR, si quiere tener un hueco en el futuro que nos espera.

Teniendo en cuenta todo esto, hoy por hoy, las aplicaciones de AR actuales, dividen sus tareas principales en 3 pasos:

- Reconocimiento de objetos.
- Tracking, o seguimiento de objetos.
- Iluminación y renderizado.

5.2.4.2.1 Reconocimiento de objetos

En este punto, como ya hemos comentado, depende del nivel de nuestra aplicación. Si se trata de una aplicación de *nivel 2*, se hará uso de procesos de registro de imágenes, tales como detección de esquinas/bordes, de umbral, detección de puntos importantes (feature detection), contraste de imágenes, etc.; con diferentes tipos de procesamiento de imágenes.

Si por el contrario se trata de una aplicación de *nivel 3*, se utilizarán los distintos componentes de posicionamiento, para comprobar si existen dichas coordenadas en la base de datos, mostrando el contenido en caso afirmativo.

Otra técnica muy habitual es el uso de SLAM (o PTAM para dimensiones pequeñas). El SLAM (*Simultaneous Localization And Mapping*), realiza una nube de puntos, para detectar los "feature" o puntos de interés del objeto que está visualizando y los compara, con una BBDD para determinar si lo que está viendo, contiene algún elemento virtual asociado para mostrar, esto aplicado con el GPS, puede dar lugar a aplicaciones más visuales que las de nivel 2. El problema aquí es el entorno variable en el que vivimos.

Por ejemplo, si queremos que en la fachada de cierto edificio emblemático de nuestra ciudad, tenga asociada la aparición de un monstruo gigantesco, tendremos que tener en cuenta de que la nube de puntos solo reconozca como objeto de referencia al edificio en sí (cosa que no suele ocurrir), puesto que si toma como referencia o parte de ella, o un edificio anexo, un árbol, una fuente o cualquier elemento de mobiliario urbano; la desaparición de cualquiera de ellos, afectarían a los datos previamente guardados; por no mencionar el tránsito de personas o el cambio de iluminación en el ambiente.

En el caso de PTAM (*Parallel Tracking and Mapping*) ocurre lo mismo, pero se ve menos afectado tratándose de una tecnología asociada a esquemas más pequeños (escritorios, habitaciones, etc.).

En ambos tipos de aplicaciones, las referencias pueden estar alojadas en la nube, haciendo menos pesada la aplicación.

5.2.4.2.2 Tracking

Cuando hablamos de Tracking, hacemos referencia al seguimiento del objeto reconocido y su correcto posicionamiento en el entorno físico de acuerdo a la posición del usuario, teniendo en cuenta la posibilidad de que alguno de los dos, el objeto reconocido o el usuario que esté haciendo uso de la app, se encuentre en movimiento. Para ello usaremos las coordenadas del objeto y las nuestras, comparándolas frame a frame de la ejecución y actualizando el contenido según las variaciones.

Al ser un punto tan importante y tan estudiado en otros ámbitos como la robótica, las aplicaciones de hoy en día han paliado casi todos los posibles problemas que pudiesen surgir, llegando al punto de ser capaces de reconocer vehículos en movimientos, o guardando la posición de un objeto “trackeado”, de tal manera que si mueves tu dispositivo, el objeto se quede estático, viéndose parcialmente si los bordes del dispositivo lo cortan.

5.2.4.2.3 Iluminación y renderizado

Una vez los objetos de la escena son identificados y analizados, es necesario introducir los elementos virtuales creados, mediante programas de representación gráfica. Sin embargo no es suficiente con generar los gráficos, sino que se deben intentar adaptar a las condiciones del entorno, a fin de que la experiencia de Realidad Aumentada sea lo más realista posible. Esto se consigue mediante dos vías:

- Coherencia geométrica o de posición: se refiere a que una vez se ha detectado el entorno real, el objeto virtual se ha de introducir en una posición determinada, para que se mimetice con la escena, de forma que parezca que realmente pertenece a ella.
- Coherencia en la iluminación: se refiere a la adaptación de la iluminación del objeto virtual, dependiendo de la iluminación en el entorno real. Sería muy raro ver bien iluminado un objeto, en medio de una noche oscura.

5.2.4.3 Técnicas de visualización

Una vez comprendido cómo funcionan las aplicaciones AR, distingamos las tres técnicas principales:

5.2.4.3.1 Display en la cabeza

Las ya mencionadas con anterioridad HDM (Head-Mounted Display), tratan de una pantalla transparente localizada entre el mundo real y los ojos del usuario; normalmente se suelen poner en gafas o cascos, aunque ya se están estudiando lentes de contacto, que se prevé que muy pronto salgan al mercado.

En este tipo de dispositivos, el usuario está viendo en todo momento el mundo físico, a través de la lente sobre la cual aparecen los contenidos virtuales; por ello, su mayor pega es que la información gráfica, se ve condicionada al campo visual del usuario.

5.2.4.3.2 Display de mano

En este campo el smartphone es el claro ganador. En este tipo de dispositivos, el contenido virtual se superpone sobre un video grabado en tiempo real, por lo que el coste computacional es muy grande, no obstante, eso no es un gran problema hoy en día. Sus principales ventajas son la portabilidad y la multifuncionalidad, al estar incluida en dispositivos capaces de realizar otras tareas.

5.2.4.3.3 Display espacial

La realidad aumentada espacial (SAR) es un concepto poco utilizado hoy en día, pero es uno de los más prometedores. En este tipo de visualización, el display no está asociado a los usuarios, de modo que varios pueden hacer uso de él al mismo tiempo. Se consigue mediante el uso de proyectores, que proyectan el contenido virtual sobre un objeto real.

De esta manera, el usuario no se ve obligado a llevar el dispositivo, pudiendo interactuar con otros usuarios. Este modelo soluciona fácilmente muchos de los problemas de los tipos anteriores, como por ejemplo el de la resolución, ya que para ampliar el tamaño, solo harían falta añadir proyectores.

Un ejemplo básico de este tipo sería el Video Mapping, el cual proyecta un video sobre fachadas, creando una ilusión de que se rompe, cambia de color, se cuartea u otros efectos.



Figura 25 Video Mapping.

5.2.5 Aplicaciones

La Realidad aumentada ofrece infinidad de nuevas aplicaciones en diversos campos de la actividad humana, no solo en el comercial y/o los negocios, como por ejemplo:

5.2.5.1 Educación

Es uno de los campos que mejor ha acogido a este tipo de tecnologías, incluyéndola en algunos libros de texto como apoyo a la hora de presentar ejemplos. Un alumno puede estar estudiando a los romanos en historia y con el dispositivo en cuestión, enfocar al dibujo del coliseo y ver una representación en 3D de cómo era en la época.

Existen ya varias aplicaciones, saliendo algunas de ellas con los primeros smartphones; unas traducían el texto enfocado, a otros idiomas; otras mostraban el nombre de las constelaciones que estabas viendo, etc.

5.2.5.2 Turismo

En este ámbito, se han creado aplicaciones diversas, unas que muestran ruinas reconstruidas, que narran explicaciones a medida que entras en distintas salas del museo, que muestran videos explicativos de cómo se confeccionaban los objetos que se vislumbran; o en otros ámbitos, que te indiquen el lugar más cercano donde saborear un plato típico de la zona, etc. Tecnologías que podemos encontrarnos con cierta frecuencia ya, en algunos lugares, en este sector de servicios.

5.2.5.3 Televisión

Uno de los ejemplos más vistos, se aplican en los eventos deportivos y en especial, en los partidos de fútbol, con la más que habitual línea que muestra cuando un jugador se encontraba en fuera de juego, o los escudos de ambos equipos, impresos en el lado del campo que les corresponde según el tiempo de juego, son algunos de los ejemplos más habituales; pero eso no tiene que acabar ahí, con la introducción de las televisiones inteligentes, el usuario podría seleccionar opciones, como elegir al jugador que desee y ver sus características, así como seguirlo durante el partido.

5.2.5.4 Entretenimiento

Uno de los campos en los que más se espera la inclusión de realidad aumentada, es en los videojuegos. Ya lo hizo Niantic con Ingress en el 2013 y fue el prelude de futuros juegos basados en esta tecnología. Su mayor postor ha sido Microsoft con sus Hololens, las cuales ha mostrado diversas veces, transformando un salón en un campo de batalla.

Actualmente Niantic, en colaboración con Nintendo, va a sacar (ya ha salido en EEUU y en el continente asiático) un videojuego basado en su famosa franquicia Pokemon,

donde el usuario podrá ver, capturar y luchar con pokemons en el mundo real, mediante su dispositivo móvil.

5.2.5.5 Arquitectura

Es otro de los campos que ha hecho uso de la AR, para mostrar cómo quedaría un edificio al finalizar la obra, usando el plano como “marcador”; así como en el diseño de interiores, para ver cómo quedaría un determinado diseño antes de actuar, o cambiarlos si se estima.

5.2.5.6 Apoyo en tareas complejas

Como por ejemplo en medicina y especialmente en cirugía, pues se podría mostrar un mapa sobre el cuerpo, para saber dónde pinchar o donde no hacerlo; o para distinguir determinadas zonas, o comparar posibles anomalías.

Otro ámbito sería apoyando distintas tareas en el ámbito industrial, no ya solo en cuanto a diseños y cambios en proyectos y procesos, sino en las propias aplicaciones de los procesos, comprobando niveles de presión o indicando qué hacer en caso de avería.

Este es el caso de DAQRI, una compañía que está diseñando un casco (un HDM) el cual proyecta sobre las gafas del mismo, ayuda e instrucciones sobre tareas dentro del ámbito de la ingeniería y las obras.

5.2.5.7 Publicidad

Es otro de los grandes campos donde la AR ha encontrado un gran nicho de mercado; las mismas etiquetas de los productos pueden servir como marcadores, pudiéndonos mostrar vídeos , música, juegos o personajes 3D, que se muevan a nuestro antojo.

5.2.5.8 Comercio

Nuestro ejemplo principal era de este ámbito, se nos mostraba el precio de unos productos gracias a su reconocimiento. Pero no solo eso, también nos podría mostrar en qué otros lugares comprarlos, que opiniones tienen otros usuarios...o incluso, en un futuro, gracias al internet de las cosas, decirnos si disponemos de ello en nuestras casas y cuánto nos queda.

Otro ejemplo de comercio serían tarjetas de visita con el CV del titular de la tarjeta; o en el ámbito laboral y en las técnicas de búsqueda de empleo.

5.3 Plataformas

5.3.1 APP

La aplicación se desarrolla para dispositivos móviles como smartphones y tablets, así pues es necesario conocer, qué plataformas existen actualmente y qué presencia tienen en el mercado. Las principales plataformas en cuanto a volumen de mercado son actualmente Android e iOS, seguidos muy de lejos de Windows Phone.

Según los datos de la consultora IDC (*International Data Corporation*) publicados en 2014, el mercado de los smartphone se encontraba distribuido de la siguiente forma:

Operating System	2014 Unit Volumes	2014 Market Share	2013 Unit Volumes	2013 Market Share	Year-Over-Year Change
Android	1,059.3	81.5%	802.2	78.7%	32.0%
iOS	192.7	14.8%	153.4	15.1%	25.6%
Windows Phone	34.9	2.7%	33.5	3.3%	4.2%
BlackBerry	5.8	0.4%	19.2	1.9%	-69.8%
Others	7.7	0.6%	2.3	0.2%	234.8%
Total	1,300.4	100.0%	1,018.7	100.0%	27.7%

Tabla 1 Distribución de app en el 2014.

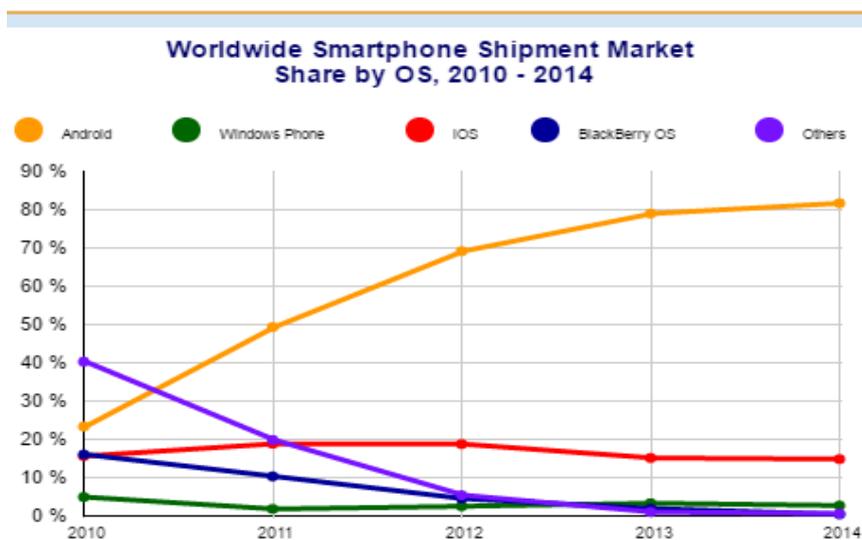


Figura 26 Gráfico de implantación en el mercado de Smartphone.

Como se puede comprobar en los gráficos, Android e Iphone abarcan el 96% de todo el mercado mundial de móviles, por lo tanto son las dos plataformas más propensas a recibir aplicaciones por parte de los desarrolladores. De este 96%, de esto, 81% prefiere un dispositivo Android, mientras que el 15% restante, opta por un Iphone; por lo que se llega a la conclusión, de que la gran mayoría de usuarios desarrolla en ambas plataformas.

5.3.1.1 Android



Figura 27 Logotipo de Android.

El SO creado por la Open Handset Alliance (OHA) en 2007, encabezados por Google, es el mayor Sistema Operativo para móviles existentes hasta la fecha, como indican los datos anteriores.

Basado en Linux y con la ayuda y el interés de los diversos integrantes de la OHA (fabricantes de chips, de terminales, operadoras, desarrolladores de software, etc.), ha conseguido colocarse a la cabeza en poco tiempo, esto es debido en parte al gran abanico de terminales que contiene dicho sistema operativo.

Al estar basado en Linux, hereda su filosofía de Open Source, dando todas las facilidades de desarrollo a cualquiera que se descargue su SDK gratuito. En este punto, cabe decir que existe una inmensa comunidad detrás del SO, dispuesta a ayudar e indicar en caso de dudas a aquel que se aventure dentro del mundo del desarrollo de aplicaciones móviles. La misma compañía ofrece tutoriales y buenas prácticas para facilitar el trabajo.

En cuanto a la publicación de las aplicaciones, se hacen después de pagar la cuota de desarrollador, la cual solo ha de hacerse una vez en vida y su precio no llega a los 30€, circunstancia que sirve de motivación para elegirla a la hora de decidir sobre qué plataforma se ha de desarrollar el trabajo.

Las aplicaciones subidas a la plataforma de distribución, son minuciosamente controladas para evitar el mal uso de ellas, o de las posibles aplicaciones fraudulentas. En este sentido, Google trata con mano dura, a los desarrolladores que se saltan sus reglas de comportamiento.

Hablando de su plataforma de distribución, Google Play es sin embargo, una de las plataformas con mayor contenido de app del mercado, siendo más de 2,2 millones de aplicaciones las alojadas en esa plataforma según los últimos datos.

5.3.1.1.1 Dato curioso

Como sabrá mucha gente, los nombres de las versiones del sistema operativo desarrollado por Google, suelen ser nombres de dulces famosos. Esta curiosidad tiene una historia interesante detrás, que aunque no es trascendente para este trabajo, la contaremos.

En las primeras versiones, se comenzó a desarrollar el sistema a base de hitos (*milestones*), por lo que se denominaban a las versiones con una M, seguido del número de versión. Al poco tiempo el desarrollo fue aumentando su velocidad, llegando las compilaciones a ser semanales (*week-build*), añadiendo otra nomenclatura WB junto con el número de versión.

Por último, se añadió a esa nomenclatura el ciclo de test correspondiente (*test-cycle*) con lo que al final, cada nueva versión del SO quedaba con un nombre ininteligible para el usuario medio, al estilo de: M8-WB19-TC5. Ante esto, los desarrolladores pensaron poner un nombre más visual y llamativo, llegando a la conclusión de que debían seguir un orden alfabético.

Al principio, pensaron en llamar a las distintas versiones del SO como androides o robots famosos, ya que el sistema se llamaba Android. De ahí que las versiones anteriores a la 1.0 se llamen Astroboy y Bender pero cuando fueron a sacar la versión 1.1, se optó por llamarlo Petit Four, ya que el manager amaba esos pequeños dulces. Desde entonces cada versión con cambios importantes de este SO, tiene un nombre que hace referencia a algún dulce, que comience por la letra que toca según el orden alfabético.

Así hoy en día tenemos la sucesión de Apple Pie, Petit Four, Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Beam, KitKat, Lollipop y (el recientemente anunciado), Marshmallow.

5.3.1.2 iOS



Figura 28 Logotipo de iOS.

Al principio salió bajo el nombre de iPhone OS (en el año 2007), basándose en OSX, el sistema operativo de los Mac.

Es el segundo SO operativo más importante del mercado, ya que si bien es cierto que está por debajo de Google, en número de aplicaciones en su plataforma de distribución, la App Store, con unos 2 millones de app; también es cierto que en cuanto a ganancias mediante las mismas app supera a Google en casi 100 millones.

Esto se debe al elitismo característico de los usuarios de Apple, ya que son usuarios que están más predispuestos a pagar algo más por las aplicaciones de esta compañía. Por no contar con el hecho de que Apple se asegura de obtener ganancias por partes, tanto de los desarrolladores, como de los usuarios; pues para desarrollar en iOS, has de pagar una cuota anual cercana a los 100€, por no mencionar la necesidad de disponer de un Mac, ya que Apple solo deja desarrollar y probar las aplicaciones en dispositivos de su propia casa.

No obstante, hay que reconocer que el sistema es muy potente y fluido, tanto es así, que Apple lo presenta como el SO más avanzado del mundo y si bien es cierto que en algunos aspectos flojea (la ausencia de Flash en todos sus dispositivos), en otros aspectos va a la cabeza en cuanto a innovaciones tecnológicas.

5.3.1.3 Windows phone

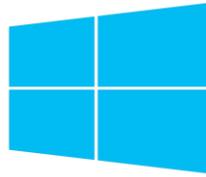


Figura 29 Logotipo de Windows.

El SO móvil de Microsoft, nunca ha tenido mucho éxito dentro de los smartphone, las razones son claras, ya que tardó mucho tiempo en aceptar el cambio de paradigma y cuando quiso entrar en el mercado, Apple y Google ya se habían afianzado como los grandes líderes.

Después de varios intentos fallidos, de varias versiones, de varios enfoques, optó por una línea distinta de lo habitual, aprovechó su amplio espectro comercial como punto fuerte, mostrando “*Windows 10 mobile*”, el cual promete aunar todas sus plataformas en un mismo SO, haciendo más fácil e intuitivo el uso de sus aplicaciones.

Si el diseño y funcionamiento de tu dispositivo móvil, es el mismo que el del ordenador de tu casa, mucho mejor, pues no tienes que estar aprendiendo a manejar dos entornos distintos, y si además eres un usuario de *Xbox* (la videoconsola de Microsoft), mejor aún, pues dicha consola también tiene la misma interfaz.

De esta manera, se aprovecha de que la mayoría de la población, está más familiarizada con sus SO para computadores y se les hace más fácil p asarse al uso de dispositivos móviles.

Pero como ya se dijo, esta estrategia es la más recientemente mostrada, falta ver cómo responden los usuarios, ya que con las versiones anteriores de su SO Windows Phone, solo disponía de 700 mil aplicaciones, en su plataforma de distribución Windows Marketplace; nada que ver si se compara, con los millones de aplicaciones de las que disponen Google y Apple.

5.3.2 Realidad aumentada

Antes de comenzar a explicar algunas de las tecnologías más importantes, que nos permiten crear contenido aumentado, hay que aclarar que este campo está creciendo a pasos agigantados; tanto es así, que en el periodo comprendido entre el inicio de este proyecto y la redacción de la memoria, han aparecido diversidad de plataformas que permiten un uso básico de la realidad aumentada, pero sin llegar al nivel de las que llevan más tiempo en el mercado. Es por ello que nos centraremos en las tecnologías más usadas y más robustas dentro de este campo.

Hace dos años, el campo de la realidad aumentada se encontraba dividido principalmente en el ámbito de tres compañías: Qualcomm, Metaio y LayAR. Como podemos ver, en el siguiente cuadro (el cual ha quedado algo desfasado, por la actual presencia de otras), se comparan las diferencias entre estas tres compañías y otras:

SDK	Purpose	Tracking	Platform	Graphics	Cloud	GP S	License
Qualcomm Vuforia	2D Images, Markers	NFT, Marker, Text	iOS, Android	Unity3D & Alt.	yes	no	Free
Metaio SDK	2D Images, GEO, 3D, Anywhere	NFT, GPS, 3D, SLAM	iOS, Android	Unity3D & Alt.	yes	yes	0-\$10,000
Totallmm. D'fusion	2D Images, faces	NFT, Face	iOS, Android Dev: MS only	Unity3D & Alt.	no	no	0-\$10,000
WikitudeSDK	2D Images, GEO	GPS, NFT	iOS, Android, Blackberry	HTML, Proprietary	yes	yes	0-\$2,300
Layar SDK	2D images	NFT, QR	iOS, Android	Proprietary	yes	no	\$3,250/app/yr + \$20/page
13 th Lab Point Cloud	2D Images, 3D, Anywhere	NFT, 3D, SLAM	iOS, Android	Unity3D & Alt.	no	no	0-\$5,000

Tabla 2 Comparativa SDK's se AR.

5.3.2.1 Qualcomm Vuforia



Figura 30 Logotipo de Vuforia.

Perteneciente a la multinacional Qualcomm, y disponiendo del soporte económico que ello conlleva, permitía licencias gratuitas a los usuarios que quisieran desarrollar aplicaciones bajo su SDK, el cual se podía descargar gratuitamente desde su web.

Intentaba seguir muy de cerca la estela de Metaio, si bien es cierto que no podía ponerse a su nivel, también es verdad que era su más claro competidor. Esto llevó a un creciente número de desarrolladores que optaban por esta versión, quizás resignados por no poder pagar una licencia de Metaio, por lo que consiguió que mucha gente trabajase con ellos, impulsándolos considerablemente en el mercado y mucho más aún, cuando Metaio fue comprado y retirado del mercado por Apple, lo que conllevó, a que en pocos años Vuforia se pusiera en cabeza en cuanto a plataformas de AR.

Si bien es cierto que aún no dispone de tecnología SLAM, si ha evolucionado, hasta llegar a disponer de una especie de PTAM, botones virtuales, extend tracking, etc. Una de las últimas noticias, hace referencia a la compatibilidad con las futuras gafas de Microsoft, las Hololens, lo cual es un gran incentivo para los desarrolladores.

Por no mencionar, que hasta la fecha son las únicas que permiten desarrollar aplicaciones que hagan uso de todas sus características, sin coste alguno.

5.3.2.2 Metaio



Figura 31 Logotipo de Metaio.

Sin duda alguna, la mejor de las tres, a pesar de que la licencia fuera de pago; el coste no era excesivo, a no ser que se fuesen añadiendo mejoras en cuanto almacenamiento en la nube. Además, disponía de una herramienta gratuita “Creator” la cual permitía hacer aplicaciones muy pequeñas de forma muy intuitiva. Y de otra herramienta, llamada “Junaio”, la cual al igual que Layar, se basaba en el GPS para mostrar contenidos

cercanos a la posición del usuario, además daba la opción de utilizar distintos “canales”. Dichos canales funcionaban como una capa, en la que solo se encontraban ciertos contenidos asociados a dicho canal, de tal forma que podías crear contenidos, basados en el posicionamiento mediante “Creator” y asociarle un canal específico, el cual, al seleccionarlo en “Junaio” mostraba dicho contenido, separando así tus propias creaciones de las mostradas por el resto de la comunidad las cuales por defecto se mostraban en el mismo canal.

A parte de esto su mayor punto fuerte, era su tecnología SLAM, la cual daba muy buenos resultados y estaba optimizada para seguir funcionando incluso con variaciones del entorno. No obstante, fue comprada por Apple y retirada del mercado, dejándola sólo para uso interno, seguramente con la idea de quitarse competidores futuros, de cara a la carrera por la realidad aumentada.

5.3.2.3 Layar



Figura 32 Logotipo de Layar.

Fue una de las primeras plataformas, que nos permitía desarrollar aplicaciones de realidad aumentada basadas en el posicionamiento. Esto quiere decir que usaba únicamente el GPS y el acelerómetro; poco después añadió el reconocimiento de códigos QR, pero nada tenía que ver con sus competidores.

En sus orígenes sólo estaba disponible para Android, pero no tardó en dar soporte para iOS, llegando finalmente a dar soporte incluso para las aun inexistentes Google Glass. Aprovechó el hecho de ser una de las primeras en aparecer, para abarcar un amplio número de dispositivos, pero su auge fue disminuyendo poco a poco al comprobarse que sus competidores daban más posibilidades de desarrollo.

Una de las últimas noticias es que fue comprada por la compañía BlippAR, la cual se dedica a ofrecer a distintos clientes soluciones comerciales basadas en aplicaciones de realidad aumentada.

5.3.2.4 ARToolKit

Se trata de una biblioteca Open Source, que permite la creación de aplicaciones de realidad aumentada. Al tratarse de una herramienta de código libre, nos encontramos con una gran cantidad de derivados con más o menos documentación, con licencias gratuitas o de pago, según sean más potentes unas que otras.

Entre sus características principales nos encontramos con:

- Un código de calibración de cámara bastante sencillo.
- Distribución al completo de su código fuente
- Enfoque casi exclusivo al uso de marcadores, lo cual les ha permitido mejorar la técnica de reconocimiento de los mismos.

Si bien es cierto que poco a poco se va utilizando cada vez más, también es cierto que aún le falta mucho camino por recorrer, para ponerse al nivel de sus competidores.

5.3.2.5 Total Immersion

Se trata de la plataforma de AR más usada en todo el mundo, ello se debe en gran medida a que ha optado por enfocarse en la optimización del reconocimiento de partes del cuerpo humano, dando lugar a crear aplicaciones de realidad aumentada que sus competidoras no pueden ofrecer.

Su plataforma patentada D`Fusion da la posibilidad al usuario de crear, aplicaciones de realidad aumentada con determinada facilidad. No obstante, Total Immersion solo puede ser desarrollada bajo su plataforma, por lo que no da soporte a herramientas de terceros, obligando al desarrollador a comprar y aprender a manejar D´Fusion.

5.3.2.6 Tango

Desarrollada por Google, se prevé que comiencen a salir aplicaciones que usen esta tecnología dentro de poco, aunque su coste computacional es muy elevado, por lo que muchos dispositivos no podrán manejarlo.

Por sus características pueden dar lugar a pensar que está más orientado a la realidad virtual, sin embargo también puede ser usado para la realidad aumentada, siendo una herramienta que trabaja básicamente mediante SLAM; para ello hace uso de varias técnicas de movimiento de percepción y de aprendizaje del entorno.

Como ya he comentado, el SDK acaba de ser liberado en estos últimos meses, por lo que aún no he podido probarlo para este proyecto por la premura de acabarlo; pero mucho menos, si no existe ningún tipo de documentación o reseña que explique detalladamente lo que tenemos entre manos; sin embargo, por lo que se ha visto, viene con grandes expectativas.

5.3.3 Plataforma de desarrollo

Como hemos visto, para desarrollar aplicaciones que se puedan utilizar en la mayoría de los dispositivos tenemos dos opciones:

- O bien hacer dos o más desarrollos en paralelo, cada uno por un SO que deseemos abarcar.
- O utilizamos aplicaciones híbridas que nos permitan abarcar más dispositivos distintos con un mismo desarrollo, pero que nos limitan en cuanto a uso interno de los dispositivos

A simple vista, la mejor opción es la segunda, pero teniendo en cuenta que nuestra aplicación hará uso de la realidad aumentada, la cual generalmente hace uso del hardware interno de los dispositivos; esto hace que sea más difícil de acceder correctamente en diseños híbridos, resultando así que la decisión a priori obvia, se complica bastante.

No obstante hay ciertas plataformas, que nos permiten desarrollar en un lenguaje intermedio (generalmente JavaScript), que a la hora de compilar, genera un código nativo para cada SO que deseemos. Esto es ideal para nuestro proyecto, pero a la vez necesitamos, que de soporte para trabajar con las herramientas de desarrollo, de la mayoría de tecnologías de realidad aumentada. Este es el caso de Unity.

5.3.3.1 Unity



Figura 33 Logotipo de Unity.

Unity, no es una plataforma de desarrollo de aplicaciones móviles comerciales, es un motor gráfico, orientado a la creación de videojuegos, el cual con el paso del tiempo ha ido aumentando el número de plataformas para las que permite desarrollar (Windows, Xbox, Playstation, Wii, etc.), que con el auge de los juegos en dispositivos móviles, también dio soporte para iOS y Android.

A medida que han ido pasando los años, ha ido mejorando y añadiendo características a su desarrollo, permitiendo por último crear aplicaciones íntegramente en la plataforma, haciendo uso de interfaces de usuario (*UI*), los cuales permiten añadir, botones, imágenes, videos, textos..., como si de una web se tratase, pero con el beneficio de que a la hora de compilar, puedes hacerlo nativo para cada SO.

A esta capacidad, cabe añadir que su licencia básica, la cual permite trabajar y hacer productos plenamente funcionales y capaces de competir en el mercado, es gratuita. Esa circunstancia ha hecho, que hoy en día sea de los motores físicos más utilizados para la realización de videojuegos, consiguiendo así una comunidad que lo respalda y una documentación envidiable.

Como dato a añadir, decir que al igual que Google, Apple o Windows; Unity dispone de su propia plataforma de distribución (*Asset Store*), donde se puede encontrar desde modelos 3D, hasta interfaces gráficas, audios e incluso código (*scripts*), que los usuarios dan gratuitamente o venden, según la calidad y el esfuerzo que conlleve el contenido creado.

Esta realidad, ha hecho posible que muchas tecnologías diversas, opten por dar soporte a esta plataforma. Y la realidad aumentada (AR), no es una excepción. Unity tiene compatibilidad con la mayoría de herramientas de realidad aumentada existentes (Metaio, Layar, Vuforia, ARToolKit., incluso con Tango), lo que la convierte en una excelente elección, para realizar aplicaciones de realidad aumentada, ya que nos permite trabajar con nuestra tecnología favorita y a la vez, exportarlo a todos los sistemas que deseemos de forma nativa.

Capítulo 6

Trabajo desarrollado

En este capítulo se explicará el desarrollo de la aplicación creada, desde sus inicios; Ideas originales, conceptualización, estudio y el desarrollo hasta el resultado final.

Para ello, se dividirá el capítulo en varios subapartados, que irán en orden de ejecución, de tal manera que, empezaremos explicando cuál era la idea original, los **requisitos** que se buscaban en el resultado final, seguido de los **contenidos** con los que contábamos inicialmente; para después meternos con el **análisis** inicial del problema a resolver, **la realización**, los problemas y decisiones que se tomaron a lo largo del proyecto; finalizando con el **diseño** final de la aplicación.

6.1 Requisitos

“El templo de Debod, fue el mayor templo donado por el Estado Egipcio, a un país de los que colaboraron en la Campaña internacional de salvamento de Nubia, organizada por la UNESCO con motivo de la construcción de la presa de Asuán. Desde 1973 se eleva en la Montaña del Príncipe Pío, en plena urbe de Madrid.

El trabajo del Proyecto Tahut, que comenzó en el año 2003, consiste en el estudio de los graffiti antiguos y de los relieves presentes en el templo. Es una colaboración entre el Museo de los Orígenes de Madrid, institución encargada de la conservación, estudio y difusión del templo de Debod, y la Universidad de La Laguna (Tenerife).”

El Templo de Debod, en su emplazamiento original, ha sido testigo del paso de innumerables generaciones y culturas que han pasado contribuido a su formación. Por ello, se pueden encontrar diversos grafitos, realizados por todas esas culturas que en algún momento han tenido contacto con el antiguo templo. Desde los antiguos devotos hasta los exploradores han ido dejando su huella, las cuales, son ahora objeto de estudio.

El proyecto Tahut, disponía antes de la realización de esta aplicación, como muchos otros centros turísticos culturales, de un tríptico informativo [Anexo II] el cual, contenía cierta información explicativa acerca de los graffiti del Templo de Debod. A que hacían referencia, quien los había hecho, donde se encontraban...

Recientemente, en colaboración con la Universidad de La Laguna, se le había añadido contenido aumentado, realizado mediante la herramienta comercial “Metaio Creator” y

apoyado sobre la plataforma “Junaio”; el resultado mostraba ciertas imágenes o videos cuando el usuario escaneaba algunos marcadores del tríptico. Cabe destacar, cómo para dichos marcadores se había optado por figuras geométricas con varios vértices, concretamente hexágonos, con colores claramente contrastados, con la intención de aumentar la calidad de dichas figuras a la hora de ser reconocidas por la aplicación.



Figura 34 Target e instrucciones en el tríptico.

Aun así, se pretendía mejorar este servicio de realidad aumentada en el templo, creando una aplicación que junto a la información proporcionada por el tríptico completara la experiencia turística del visitante, convirtiendo así el Templo de Debod, en un destino turístico renovado, de mano de las últimas tecnologías.

La idea original era una aplicación de AR, que el visitante pudiera descargar en cualquier dispositivo, la cual mostraría imágenes, audios, textos y videos; dependiendo de en qué lugar del templo se encontrara el usuario.

Por lo tanto, como idea inicial teníamos lo siguiente:

- Una app móvil disponible para el mayor número de dispositivos, es decir Android e iOS, tanto Smartphones como Tablets.
- Que hiciera uso de AR, sin importar cual fuera la tecnología.
- Multilinguaje, de momento solo Inglés y Español.
- Respetuoso con el templo, es decir, que en caso de necesitar marcadores, que fuesen lo más acorde con el templo, osea, sin códigos QR o imágenes que lo estropearan.
- Un único tipo de reconocimiento, ya que no se deseaba marear al usuario con varios tipos de reconocimiento (marcadores, gps, etc.).

Una vez conocidos estos requisitos previos iniciales, comenzó el estudio de las posibles herramientas a desarrollar y del problema a solventar.

6.2 Contenidos

Antes de continuar con el desarrollo de la aplicación, hagamos un alto para comentar y enumerar los contenidos de partida, los cuales deberían incorporarse al trabajo, según los objetivos buscados. Dichos contenidos habían sido aportados tras gestiones, por la dirección del templo de Debod, para facilitar el trabajo.

En este paquete inicial, que constaba de una copia del tríptico, más un alto número de contenidos realizados por diversas personas a lo largo de los últimos años, nos encontramos los siguientes archivos:

- Imágenes y fotografías de los diversos graffitti que debíamos mostrar.
- Textos explicativos tanto de los graffitti como de la historia del templo. En ambos idiomas, Español e Inglés.
- Locuciones que narraban los mismos textos. También ofrecidos en Ambos idiomas.
- Video que mostraba el Templo de Debod, desde el interior hasta el exterior; sin audio.
- Logo, imágenes de portada y mapas.
- Modelo 3D del templo. Sin texturas.

Dichos contenidos se detallan en el Anexo II. Cabe destacar que el resto de elementos que aparecen en la aplicación, son creación propia o perteneciente a librerías gratuitas distribuidas en la tienda oficial de Unity.

6.3 Análisis del problema

6.3.1 Primeros pasos

Antes de comenzar a desarrollar, había que documentarse sobre las opciones que existían, para afrontar el trabajo con la certeza de estar haciéndolo lo mejor posible. Lo primero que debía de aclararse era como desarrollar la aplicación; pues como uno de los requisitos exigidos era, que el producto final sirviese tanto para Android como para iOS, descarté desde el principio la opción de hacerlo puramente nativo, pues ello me obligaba a hacer un doble esfuerzo, necesitando crear el mismo producto en dos versiones distintas, una en Java y otra en el recién anunciado Swift, del cual no sabía nada.

Por lo que el enfoque se orientó a las plataformas que me permitían trabajar para ambos sistemas.

Titanium era una buena opción, al tratarse de una plataforma muy conocida para soluciones de este tipo. Se encontró mucha documentación y cursos al respecto, pero cuando busqué las posibilidades de añadir funcionalidad aumentada, se descubrió que (por entonces), aún era bastante inestable y las soluciones que permitía eran muy básicas.

En este punto fue cuando se decidió que resultaría más práctico decidir con qué tecnología aumentada se trabajaría, ya que una vez decidido dicho punto, resultaría más fácil escoger una plataforma de desarrollo, si se partía desde la base de que dicha plataforma, diera soporte a la tecnología aumentada que se fuera a utilizar.

La elección fue fácil, partiendo de que el paquete de contenidos previos que ya usaba era de Metaio y tras realizar un pequeño estudio de mercado, me di cuenta de que era de mucho, la mejor opción; la única pega para desarrollar aplicaciones con ellos era el coste de la licencia, pero el cliente ya disponía de una, la cual no le importaba compartir conmigo por lo que analizando todas las posibilidades que me ofrecía Metaio y comprendiendo que era afortunado por poder desarrollar bajo dicha tecnología sin coste por mi parte, me decidí por esa tecnología.

Una vez decidido que Metaio sería la tecnología que usaríamos para mostrar la realidad aumentada, la búsqueda de una plataforma que permitiese generar una distribución, para ambos sistemas operativos y que diera soporte a Metaio se redujo bastante, entre las opciones factibles, había una opción que resaltaba sobre las demás. Unity (Unity3D por entonces), tenía todas las características mencionadas y además acababa de incluir en sus posibilidades el sistema GUI, el cual permitía realizar interfaces de manera simple.

De nuevo, la fortuna ofrecía la posibilidad de trabajar con una tecnología, que hacía tiempo deseaba probar y sobre la que quería aprender todo lo que pudiera, no en vano, había participado en un curso que exploraba el scripting en Unity, por lo que lo tomé

como algo positivo, ya que se podría usar lo aprendido en el curso para el proyecto. Por lo tanto, las tecnologías a usar serían: Unity (Android/iOS) y Metaio.

Una vez decididas estas tecnologías base, se afrontó otro paso que se debería seguir siempre que se trabaje con un producto enfocado a usuarios con bajo nivel tecnológico. El prototipado.

6.3.2 Prototipado

Siempre que se realice una aplicación comercial, esto es, enfocada a usuarios que probablemente no tengan mucha habilidad manejando las nuevas tecnologías, es recomendable realizar un prototipo, por muy básico que sea. Ya que de esa forma, podremos hacernos una idea más fiable de que espera el cliente de nuestro trabajo. Al no realizar este paso, corremos el riesgo de entregar un producto que poco tenga que ver con la idea de nuestro cliente, con lo que podría dar lugar a descontento, o a trabajo extra; o peor aún, rechazo del trabajo, con los prejuicios y mala imagen que ello pueda acarrear.

Cuando hablamos de prototipado nos referimos a una simulación gráfica del producto que vamos a realizar. No tiene que ser exacto al resultado final, es más, muy pocas veces resulta así, por razones obvias.

Existen muchas herramientas, alguna de ellas gratis y otras de pago, que nos permiten la realización de prototipos. Si bien es cierto que un dibujo en una servilleta de cafetería podría considerarse un prototipo, estas herramientas nos ofrecen muchas ventajas: Videos demostrativos, plantillas de interfaces específicas para cada plataforma, inclusión de imágenes, botones con enlaces a otras pantalla, etc.

En este caso, se buscaba una herramienta que no solo permitiese realizar dibujos de una aplicación, ya que había comprobado anteriormente, que el usuario medio, podía no llegar a entender cómo se pasaba de una pantalla a otra, al tratarse de papeles impresos. A todos se nos hace raro simular que tocamos un botón, cuando lo que tenemos delante es un papel; además, deseaba poder exportar un video explicativo, que pudiera enviar al tutor sin necesidad de llevar mi ordenador a las reuniones.

Después de un par de días y de varias pruebas con distintos programas (Balsamiq, Axure, Visio..) se encontró el programa llamado JustInMind.

Las razones de la elección de este programa fueron varias:

- Gratuito (en principio, después de 3 meses me empezaron a cobrar).
- Permitía la simulación de funcionalidad, enlazando las distintas pantallas mediante botones.
- Disponía de una gran cantidad de interfaces simuladas, tanto de Android como de iOS.

- Daba almacenamiento gratuito en la nube de nuestros distintos proyectos.
- Permitía ejecutar la simulación en dispositivos móviles.
- Daba la opción de grabar un video de la simulación.

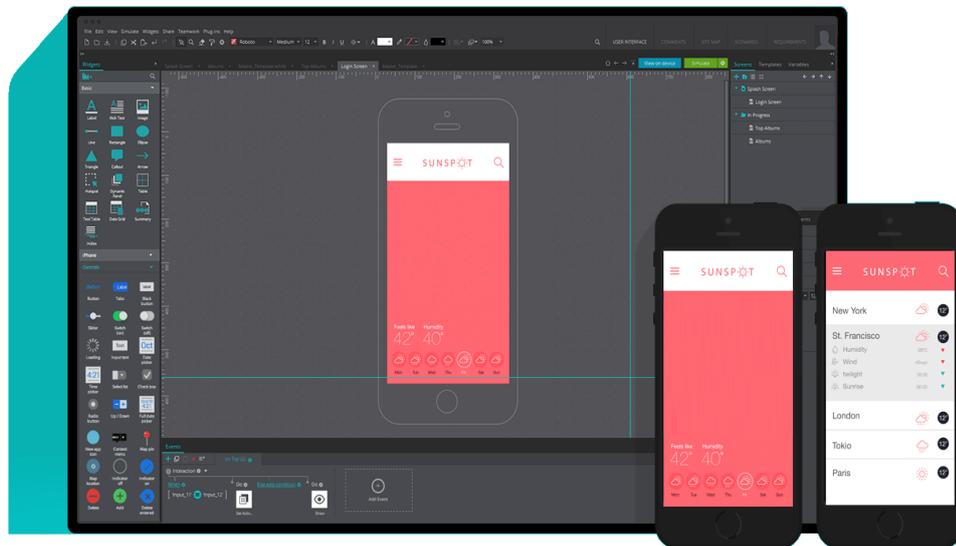


Figura 35 *JustInMind*

Antes de empezar a realizar el prototipado, se estudió superficialmente las capacidades que ofrecía el SDK de Metaio, opciones de detección de imágenes sobre todo y más concretamente qué era eso del SLAM. Una vez comprendida la potencia de la herramienta que se tenía entre las manos, comenzó la fase del prototipado.

A decir verdad, viéndolo con perspectiva, dicho prototipo resultó ser muy utópico, pero en mi defensa diré que a cualquiera le podría pasar. Si hay algo que se le pueda recriminar a la realidad aumentada, es la forma que tienen de mostrarla al mundo. Estamos acostumbrados a ver esos anuncios, donde el usuario ve modelos 3D perfectos, donde los manejan sin ningún tipo de problema, pero verdad es muy diferente y aún nos quedan muchos años, para llegar a lo que muestran esos anuncios comerciales.

Anuncio



Realidad



Figura 36 Ejemplos de anuncios y realidad.

Antes de explicar el prototipo elaborado, es recomendable aclarar que en este capítulo solo sólo se mostrará algunas capturas de pantalla. En el Anexo IV se podrá ver el resto de pantallas.

6.3.2.1 Explicación del prototipo



Figura 37 Portada del prototipo.

La primera pantalla era la portada de la aplicación como es natural, donde además de mostrar una imagen del templo con su nombre, nos encontrábamos con dos botones distintos, uno de ellos nos redirigía a la página web del templo y el otro entraba en nuestra aplicación.



Figura 38 Pantalla principal del prototipo

Una vez en la aplicación, nos encontramos con la pantalla principal. Empezando desde arriba a la izquierda nos encontramos con:

- Título. Nombre del templo.
- Botón de AR. Este botón activaba la realidad aumentada. Aquí diremos que la aplicación fue pensada desde el principio para dos modalidades, casera y en el propio templo.
 - No presencial o casera (no in situ): permitía al usuario acceder a toda la visita desde casa o desde cualquier parte, mostrando el contenido (imágenes y textos), sin necesidad de usar la realidad aumentada.

- Presencial (en el propio templo): estaba orientada a usar la realidad aumentada, y requería la presencia del usuario en el templo, para mostrar su contenido.
- Gráfico del templo. En él se mostraban las dos puertas y el templo en sí. Cuando el usuario hacía click sobre alguna puerta (los arcos), o sobre el templo, se cargaba la pantalla correspondiente a dicha sección.
- Texto explicativo. En la imagen se muestra el “Lore Ipsum”, puesto que fue el usado para el prototipo, pero la idea original era que ahí, se encontrase alguno de los textos que el cliente me había dado.
- Botones. Dichos botones actualizaban el texto en caso de que existiesen más textos que mostrar. Cabe decir que en el prototipo no cambiaba el texto, solo aparecía el botón de “Atrás” cuando hacías click en “Siguiete” y desaparecía cuando hacías click en él; dando la sensación de que la pantalla cambiaba.

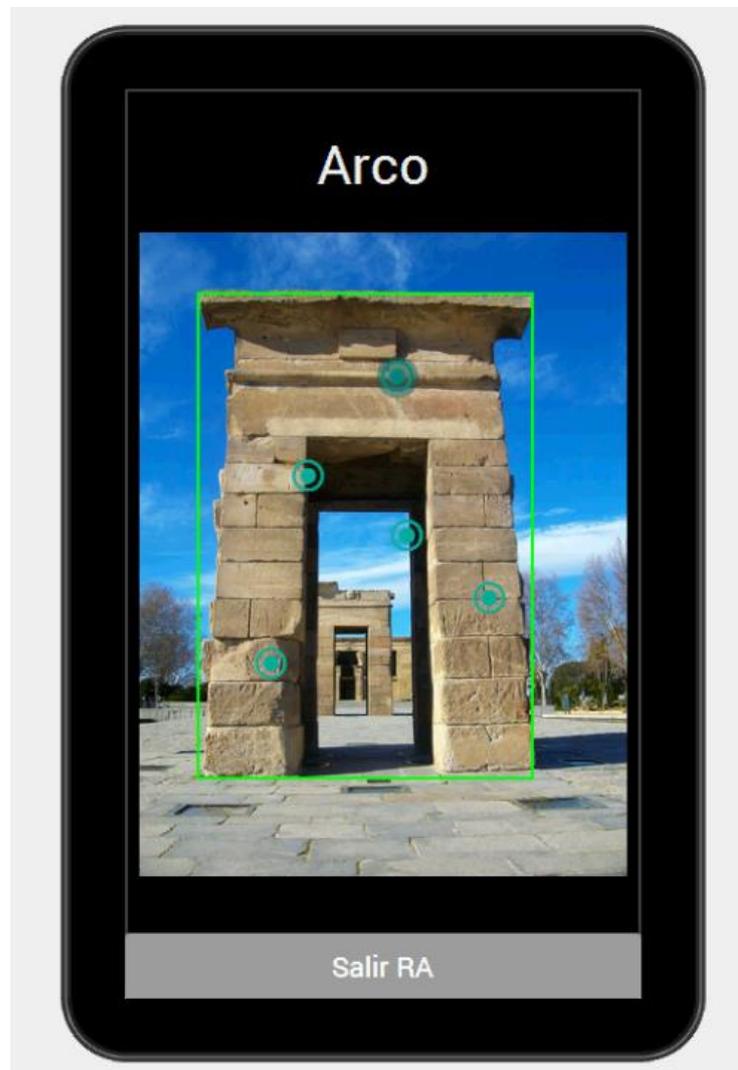


Figura 39 Pantalla de Realidad Aumentada del prototipo

Cuando seleccionábamos la opción de AR se encendía la máquina y comenzaba el reconocimiento. En el ejemplo se suponía que al estar fuera, el primer objeto a reconocer era una de las puertas (arco) y es por eso, que se muestra esta imagen cuando se pulsaba el botón de realidad aumentada.

La idea era que la aplicación usara el GPS para detectar que se encontraba fuera del templo; una vez hecho esto, haría uso de la tecnología SLAM para detectar el objeto que tenía delante (en la imagen se hace referencia a este punto con el recuadro verde alrededor de la puerta), para que una vez hecho, lo compararía con una base de datos en la nube (proporcionada por Metaio), con lo que detectaría que se encontraba frente a una de las puertas, mostrando el contenido aumentado, que en estos casos se representaban como unos botones virtuales, los cuales al pulsarlos cargarían la siguiente pantalla.

La idea es que cada uno de esos puntos, hiciera referencia a un graffito en concreto e indicara donde estaba. Es decir, la posición de dichos puntos no sería aleatoria.



Figura 40 Datos de la Realidad Aumentada del prototipo.

Cada uno de los puntos mostrados en la pantalla anterior, lleva asociado una vista como esta, en la que se mostraba una imagen, la cual hacía referencia al graffito en cuestión y un texto explicativo. De esa manera, el usuario podía acercarse a ver más de cerca el graffito y leer su significado. Al pulsar el botón de “Atrás” esta vista se cerraba y se mostraba la anterior.

Al pulsar el botón de “Salir” se volvía a la visión normal de la aplicación, esto es, sin realidad aumentada.



Figura 41 Pantalla de una de las puertas del prototipo

Como dijimos para la primera imagen, cuando el usuario hacía click en alguna de las partes del esquema del templo, se cargaba la pantalla correspondiente a la parte seleccionada. En el caso de las puertas principales se cargaba una vista como esta. Exactamente lo mismo que la pantalla anterior (el ejemplo de realidad aumentada), solo que sin la realidad aumentada; no obstante el usuario siempre podría pulsar el botón azul de realidad aumentada, para ver una vista como la ya explicada.

En esta pantalla cuando el usuario pulsaba sobre los grafitos marcados (con puntos claros en la imagen), se mostraba la imagen del grafito y el texto correspondiente. En la parte baja, a la derecha, se puede distinguir una flecha, que a su vez pulsándola, cargaba la pantalla principal.



Figura 42 Pantalla de la fachada del templo en el prototipo

En el caso de pulsar la imagen del templo, veríamos esta pantalla, la cual se diferencia de las puertas en que este tiene un botón que indica "Entrar", esto se debe a que dentro del templo existen más grafitos y esta pantalla solo servía de preludeo entre la parte exterior del templo y la interior.



Figura 43 Pantalla del plano del primer piso del templo en el prototipo.

Al entrar en el templo nos encontrábamos con un plano o mapa del templo, dividido por habitaciones. En esta pantalla, además de los elementos comunes (Título, texto, botón AR), de los cuales ya conocemos su función; tenemos un plano orientativo, en el cual, al seleccionar cualquiera de los espacios que lo componen, se abriría una pantalla específica, como la siguiente.



Figura 44 Pantalla de datos del interior del templo en el prototipo.

Cada espacio o habitáculo, disponía así de su propia pantalla, que mostraba al estilo corrousel, imágenes de los graffitis existentes dentro de esa habitación, así como un texto explicativo de cada uno de ellos.

Una vez el tutor le dio el visto bueno al prototipo, comenzó la fase de documentación en profundidad sobre la herramienta Metaio; comenzando por descargar Unity y el SDK de Metaio. Una vez todo estuvo listo, ejecute y probé los distintos ejemplos que ofrecía la herramienta y las cuales se detallaban paso a paso en su web.

Después de haber visto los distintos tipos de reconocimiento de los que disponía la herramienta y cómo funcionaba cada uno, siguió la fase de estudio de toda la documentación oficial que aportaba la página de desarrolladores de Metaio, (en Inglés por supuesto). Finalmente, cuando todo estaba por fin estudiado y listo para comenzar a realizar las pruebas en Unity, Apple compra Metaio. No tardaron ni dos semanas en avisar de que iban a dejar de dar soporte en breve y así fue. De nada sirvió que me pusiera en contacto con la empresa para ver si existía la posibilidad de seguir trabajando con ellos.

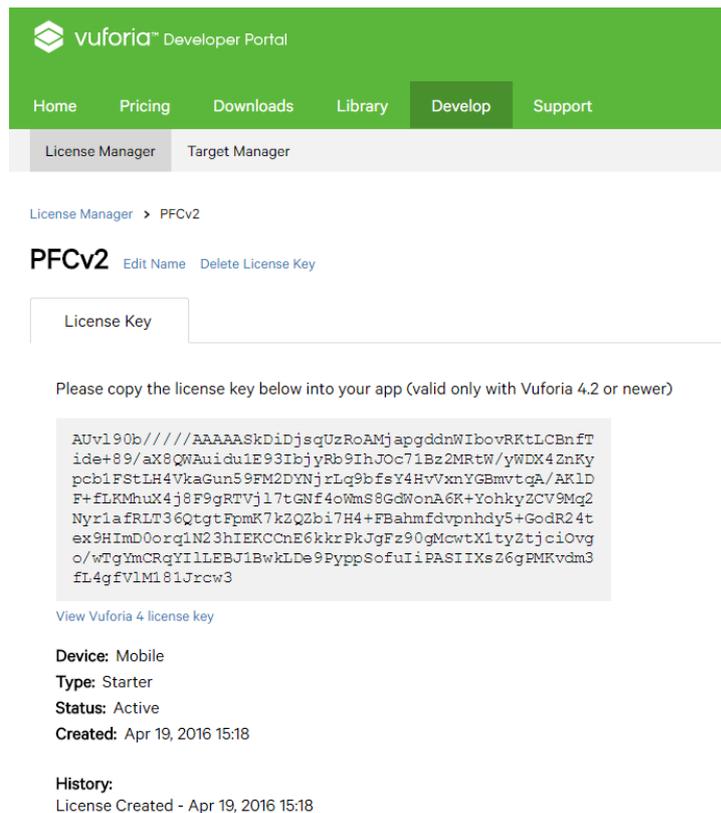
6.3.3 Alternativa

El estudio realizado tras meses de preparación no había dado sus frutos (por circunstancias ajenas). Por lo que se optó por buscar una alternativa rápida, basándonos en el estudio preliminar que habíamos realizado. Así que partiendo del uso de Unity como plataforma de desarrollo, se optó por la segunda mejor opción (algo que se vieron obligados a hacer muchos más desarrolladores, por lo visto) y así fue como se comenzó a trabajar con Vuforia.

Vuforia no era tan potente como Metaio, aunque si es verdad que desde entonces ha avanzado mucho más rápido de lo que lo había hecho hasta la fecha, por lo que mi planteamiento inicial se vio obligado a cambiar; básicamente porque Vuforia no disponía (ni lo hace aun) de reconocimiento SLAM.

Para trabajar con este SDK (Kit de Desarrollo de Software) es necesario crearse una cuenta de desarrollador en la plataforma de Vuforia, en la cual, se encuentra todo lo necesario para poder desarrollar aplicaciones junto con esta tecnología. Descarga de librerías, de ejemplos, tutoriales, foros y la parte más importante. La lista de licencias y targets.

En este apartado el desarrollador ha de crearse una licencia la cual deberá incluir en todos sus proyectos de realidad aumentada, si quiere que estos funcionen.



The screenshot shows the Vuforia Developer Portal interface. At the top, there is a navigation bar with links for Home, Pricing, Downloads, Library, Develop, and Support. Below this, there are tabs for License Manager and Target Manager. The current page is the License Manager page for a license key named PFCv2. The page includes a text input field for the license key, a copy instruction, and a code block containing the license key itself. Below the code block, there are details for the license key, including the device type (Mobile), type (Starter), status (Active), and creation date (Apr 19, 2016 15:18). A history section shows the license was created on the same date and time.

```
AUv190b/////AAAAAskDiDjsqUzRoAMjapgdnnWIbovRKtLCBnft
ide+89/aX8QWauidu1E93IbjyRb9IhJoc71Bz2MRtW/yWDX4ZnKy
pcb1FStLH4VkaGun59FM2DYNjrlq9bfsY4HvVxnYGBmvtgA/AK1D
F+fLKMhuX4j8F9gRTVj17tGNf4oWmS8GdWonA6K+YohkyZCV9Mq2
Nyr1afRLT36QtgtFpmK7k2QZbi7H4+FBahmfdvphndyS+GodR24t
ex9HImD0orq1N23hIEKCCnE6kkrPkJgFz90gMewtX1ty2tjciOvg
o/wTgYmCRqYI1LEBJ1BwkLDe9PyppSofuIiPASIIxs26gPMKvdm3
fL4gfV1M181Jrcw3
```

Device: Mobile
Type: Starter
Status: Active
Created: Apr 19, 2016 15:18

History:
License Created - Apr 19, 2016 15:18

Figura 45 Apartado de licencias

Una vez creada, el desarrollador podrá crearse una base de datos asociada a dicha licencia, donde podrá incluir sus propios marcadores creados en base con las imágenes que previamente suba en este mismo apartado. Vuforía evaluará dicho marcador indicando al desarrollador si es una imagen apta.

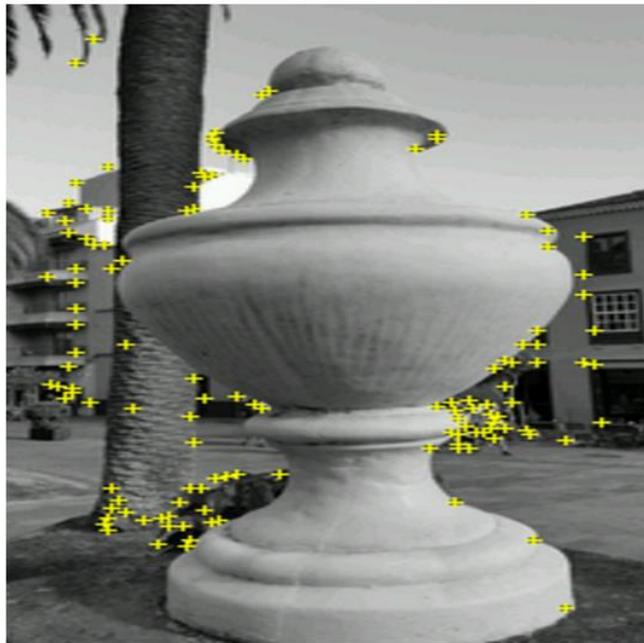
Figura 46 Creación de targuet.

<input type="checkbox"/>	R2D2	Single Image	★★★★★	Active	Jul 29, 2015 12:59
<input type="checkbox"/>	mustang	Single Image	★★★★☆	Active	Jul 29, 2015 12:57
<input type="checkbox"/>	Book	Single Image	★★★★☆	Active	Jul 29, 2015 12:56
<input type="checkbox"/>	Soda	Cylinder	n/a	Active	Jul 27, 2015 21:02
<input type="checkbox"/>	Ejemplo	Cuboid	n/a	Active	Jul 27, 2015 20:51

Figura 47 Targets evaluados.

Se realizaron varias pruebas con edificios y monumentos en la propia ciudad de La Laguna, para comprobar cuan factible podría ser usar las mismas fotos del objeto a reconocer (ya fuera la puerta o una fachada interior), como marcador en sí; por suerte los resultados hicieron que se rechazara esa pésima idea.

Edit Name Remove



Type: Single Image

Status: Active

Target ID: 6d70e34f050a4b13b8dbd210e374fcb1

Augmentable: ★★★★★

Added: Jul 27, 2015 20:00

Modified: Jul 27, 2015 20:00

Figura 48 Prueba de marcadores en La Laguna (objeto)

Edit Name Remove



Type: Single Image

Status: Active

Target ID: 6a8a781442a94b639c9dc89f6d14580c

Augmentable: ★★★★★

Added: Jul 27, 2015 19:46

Modified: Jul 27, 2015 19:46

Figura 49 Prueba de marcadores en La Laguna (interiores)

Edit Name Remove



Type: Single Image

Status: Active

Target ID: 1bde386cfca34176bcd31fff4b13bc61

Augmentable: ★★☆☆☆

Added: Jul 27, 2015 19:58

Modified: Jul 27, 2015 19:58

Figura 50 Prueba de marcadores en La Laguna (edificios)

Aun así, tenía la esperanza de poder usar la tecnología de reconocimiento mediante el posicionamiento para la parte exterior del templo, no obstante, todo trabajo en remoto conllevaba un alto índice de conjeturas, por lo que se organizó un viaje a Madrid, para ver in situ las características del templo.

6.3.4 Visita de Campo (viaje a Madrid)

Una vez en el Templo de Debod, y después de haber pasado la noche rehaciendo el prototipo, ya que, por lo visto los servidores de JustInMind habían borrado el anterior proyecto al pasar más de un mes inactivo, se hicieron las comprobaciones oportunas para estudiar la viabilidad de los distintos tipos de reconocimiento posibles.

- GPS, solo funcionaba en los exteriores del templo, en el interior no había señal de ningún tipo, ni siquiera había conexión a internet.
- Iluminación, dentro del templo era escasa, la opción de tomar como marcados las fachadas internas se esfumó nada más pisar la zona interior.
- Imagen de la diosa, una proyección luminosa sobre la piedra dentro del templo, tenía opciones de servir como marcador, al menos se podría intentar usar esa como tal.

De igual manera, se comprobó que no había tantos graffitis como se podía imaginar en un principio y se pudo comprobar además, que sus localizaciones también eran limitadas.

En el cambio de opiniones con el tutor y los responsables del centro, se comentó la posibilidad de usar el posicionamiento del usuario en la parte exterior y dentro del templo usar marcadores al uso para mostrar la realidad aumentada (a falta de hacer las comprobaciones de la imagen de la diosa como posible marcador).

No obstante, según el manual de buenas prácticas de Vuforia, la idea de tener dos tipos de realidad aumentada (basada en el GPS y en marcadores, por ejemplo), puede llegar a confundir al usuario, ya que podría no entender muy bien cuando usar uno y cuando usar otro, lo que podría llevar lugar a frustración y el desuso de la aplicación, con lo que el Templo vería afectada su imagen.

Con esto en mente y dejando claro que dentro del templo no cabía la posibilidad de usar el GPS, se optó por la posibilidad de usar imágenes como marcadores, pero después de intercambiar nuevamente opiniones, sobre las pocas posibilidades que existían de que las mismas paredes pudieran servir como tal, se decidió usar alguna imagen que pasara desapercibida dentro del templo.

Finalmente, después de muchas propuestas, se llegó a la conclusión de que el mismo logotipo del proyecto Ta-hut, podría servir como marca si se cambiase la X de lugar, pudiendo conseguirse hasta 12 marcadores distintos, colocando la X en las posiciones del reloj.

Esta idea a priori parecía factible y buena, pues daba juego para utilizar una docena de marcadores y a la vez era fácil de asimilar por los usuarios; por lo que se optó por llevarla a cabo.



Figura 51 Logotipo del proyecto Tahut.



Figura 52 Logotipo utilizado como prueba de marcadores.

Lo primero que se hizo al llegar al apartamento en Madrid, fue comprobar si la imagen que había tomado de la diosa dentro del templo, se podía usar como marcador, pero como era de esperar, no era el caso. Definitivamente solo quedaba la opción de los marcadores.

Prueba_Debod

[Edit Name](#) [Remove](#)



Type: Single Image
Status: Active
Target ID: 99407633a34844cdbcf21791ee59ac05
Augmentable: ★★☆☆☆
Added: Aug 23, 2015 14:38
Modified: Aug 23, 2015 14:38

Figura 53 Prueba de marcador en la imagen de la diosa.

6.4 Realización

Una vez aclarados todos los puntos, se comenzó con la preparación del proyecto, usando para ello los conocimientos que había adquirido en las pruebas, cursos y tutoriales.

6.4.1 Unity

Antes de comenzar a explicar el desarrollo de la aplicación, conviene comentar algunos de los conceptos básicos y característicos de Unity. Si bien; no se trata de redactar un tutorial sobre cómo trabajar con dicha plataforma, llegados a este punto, sí parece conveniente explicar los términos más habituales dentro de cualquier desarrollo realizado en esta plataforma, con el objetivo de que el lector no se sienta confundido en un futuro.

Unity divide sus proyectos en **Escenas** (*Scenes*). Estas pueden verse como los distintos niveles de un juego o aplicación; en cada escena estarán alojados un cierto número de **objetos** (*GameObjects*) los cuales suelen estar compuestos por varios **componentes** (*Components*) tales como, scripts, texturas...

Components \subseteq GameObjects \subseteq Scenes \subseteq Projects

Un ejemplo de esta jerarquía en un videojuego sería, la pantalla principal (*escena*) donde se muestra la animación de fondo y el botón de comienzo (*gameobject*) el cual esta compuesto por: un texto que pone “*Start*”, un script que dota de funcionalidad al botón, una textura que da color y forma al botón, y una animación que hace que el botón se agrande al ser pulsado (*componentes*). El conjunto de varias escenas similares a estas formarían nuestro proyecto.

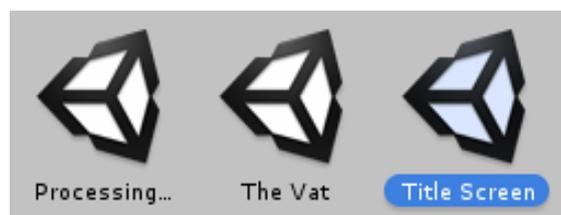


Figura 54 Escenas.

El material que se usará en la aplicación se denomina **Asset**, esto puede abarcar desde objetos individuales (imágenes, textos, audios, etc.); hasta objetos más complejos, formados por la unión de varios objetos simples; a éste tipo de conjuntos se les denomina **Prefabs**.

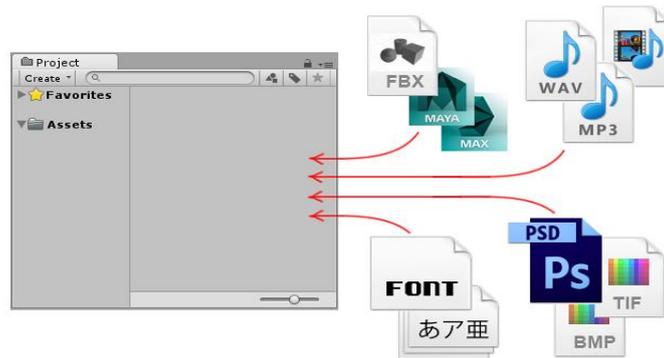


Figura 55 Asset

Unity mantiene un orden jerárquico estricto, donde todos los componentes, recursos, escenas y librerías han de estar bajo la librería **Assets**. Además, según las buenas practicas fomentadas en la comunidad de desarrolladores, es conveniente crear nuevos directorios donde alojar los distintos elementos principales de nuestro proyecto, es decir:

- Escenas
- Scripts
- Recursos
- Materiales (texturas)
- Prefabs

Y en definitiva cualquier otro directorio que el desarrollador crea conveniente.

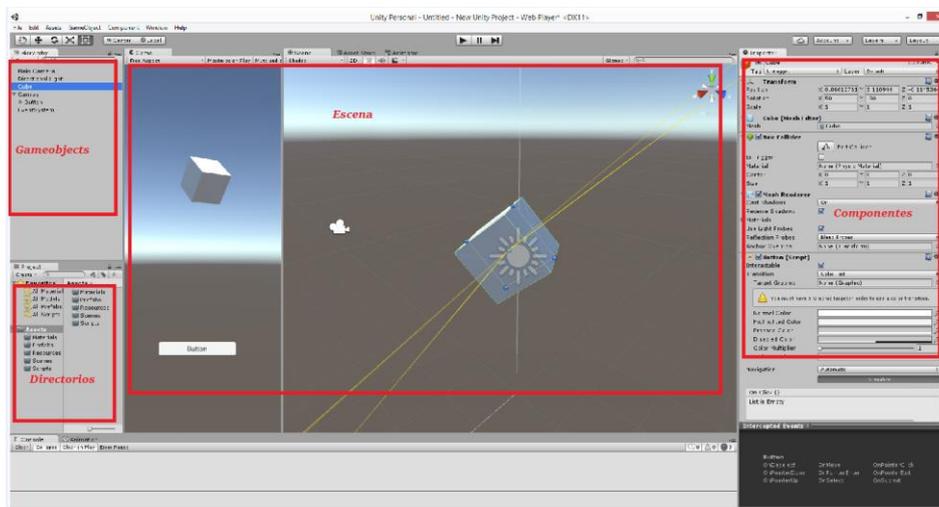


Figura 56 Esquema básico de Unity.

6.4.2 Interfaz

Cuando se comenzó el proyecto, Unity tenía capacidad de realizar interfaces gráficas gracias a sus sistema GUI, el cual dotaba a la plataforma de la capacidad de añadir textos, imágenes, botones, animaciones..., además de componentes típicos de una interfaz de usuario (UI). No obstante, al tratarse de una plataforma de desarrollo de videojuegos, dicha interfaz estaba orientada a menús (de pausa, inventario, opciones) y por lo tanto, tenían un aspecto más acorde con su finalidad; esto es: fondos transparentes, botones de cierto tamaño y grandes bordes redondeados...

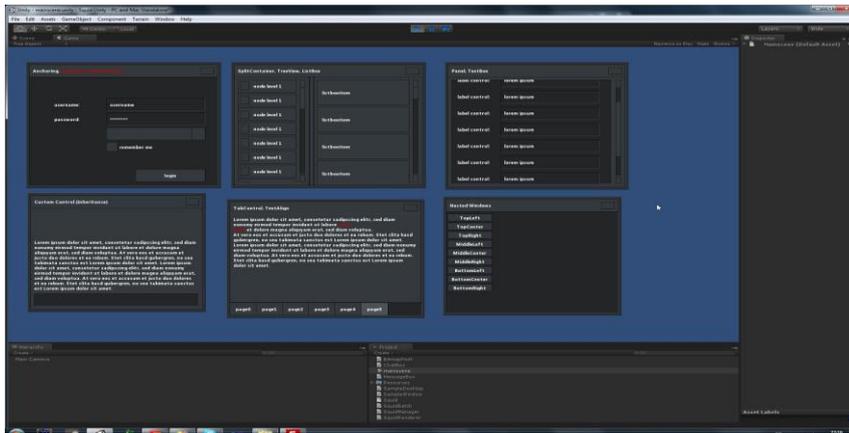


Figura 57 Ejemplo GUI

Pero al volver de Madrid, Unity había sacado una nueva versión y con ella, un nuevo soporte para la interfaz gráfica. Con esta nueva versión, se podía desarrollar una aplicación móvil al completo, usando solo el nuevo sistema de UI, el cual se basaba en el diseño web, de hecho, todo componente de la interfaz de usuario se realiza sobre un “canvas”.

A raíz de ello se comenzó a formar un sector dentro de los desarrolladores de Unity, el cual más tarde se propagó a todo el sector de los videojuegos, que se dedicaban única y exclusivamente a trabajar en la parte de la interfaz, dando lugar a grandes avances en cuanto a funcionalidad y diseño.

Al ver que el nuevo sistema se había complicado, me dedique a buscar tutoriales en internet y se encontró un curso de “Imagina Formación” el cual enseñaba los conceptos básicos de dicha interfaz, además de a integrar Vuforia con la plataforma.



Figura 58 Pruebas de UI.

Después de realizar este curso y de haber realizado unas cuantas pruebas, tomando el prototipo como modelo, estábamos preparados para llevar a cabo la primera parte del proyecto. **“El folleto”**.

6.4.3 El Folleto

Al quedar Metaio y sus variantes (Creator y Junaio), fuera del mercado, todas sus aplicaciones sin excepción, dejaron de funcionar. En el caso del proyecto Tahut, los miles de trípticos que tenían, se habían quedado desfasados antes incluso de salir, pues en él, anunciaban una aplicación móvil que ya no existía.

Este hecho llevó a pensar que no hacía falta tener una guía impresa en papel para el templo. La misma aplicación podía servir para informar al visitante acerca del templo sin necesidad de estar en él.

Para el diseño de los botones se hizo uso de herramientas o bien gratuitas o muy baratas que se pueden encontrar en la tienda de Unity. La más importante y la más usada fue UIBuilder, la cual tenía un coste de 50€. Dicha librería otorgaba una inmensa cantidad de diseños, tanto de botones como de marcos, específicamente diseñados para aplicaciones móviles, así como ejemplo de aplicaciones realizadas con la librería.

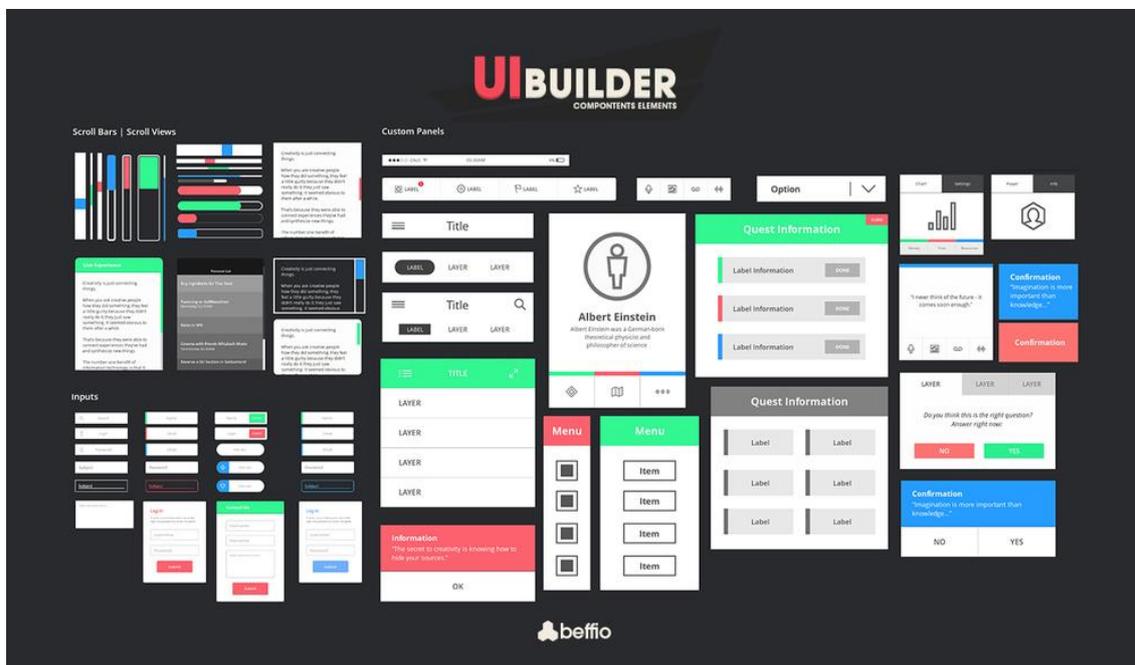


Figura 59 UIBuilder.

La otra librería usada y también de elementos gráficos, fue Colorful Buttons; librería gratuita, pero de mucha menos calidad gráfica.

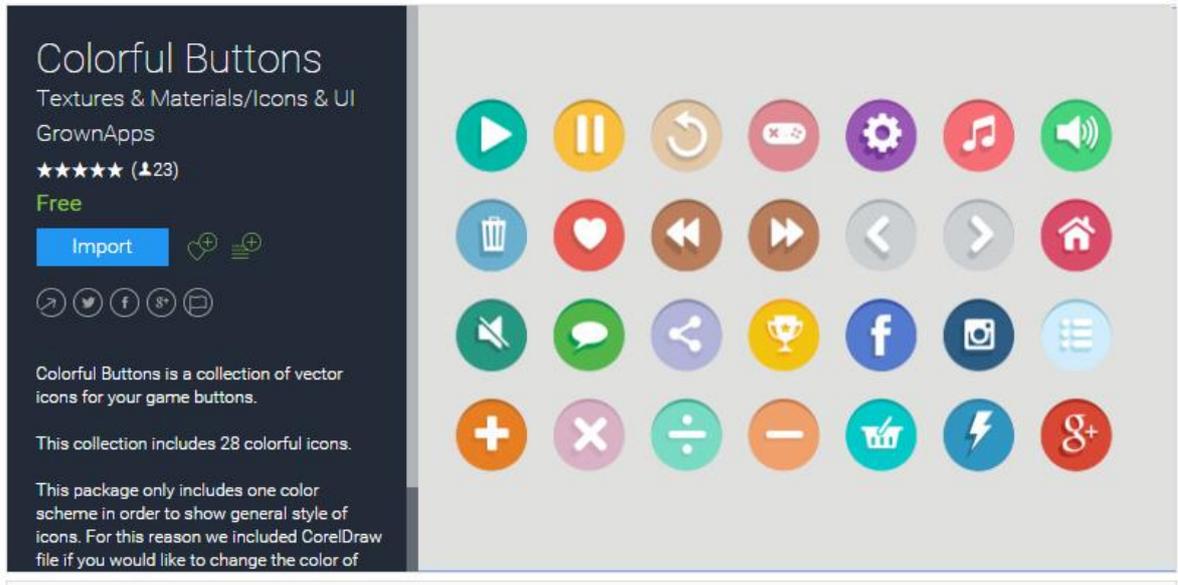


Figura 60 Colorful Buttons

Con el material gráfico necesario para realizar la primera fase de la aplicación, se comenzó a desarrollar las primeras escenas.

Algunos conceptos y prácticas, fueron difíciles de comprender, llevádo mucho tiempo y estudios de diversos tipos de tutoriales, tanto en Inglés como en Español hasta lograr comprenderlos; en esta parte del desarrollo quizás lo más complicado, fue lograr que la aplicación resultase adaptable a cualquier tipo de dispositivos (*responsive*).

El resultado fue bastante satisfactorio; la aplicación en este punto de desarrollo se parecía bastante al prototipo, solo que con una presentación mucho mejor. Mostraba una pantalla principal que daba dos opciones: cargar la página web o mostrar el folleto (También tenía el botón preparado para la segunda fase, pero estaba deshabilitado); la primera opción habla por sí sola, la segunda en cambio cargaba una escena nueva que emulaba al tríptico original, en la que se mostraban los siguientes elementos:

- Una imagen.
- Un texto en el cual el usuario podía hacer scroll.
- El logo del proyecto en el cual, al ser pulsado se iniciaba un audio, que narraba el texto que le precedía.
- Una serie de botones:
 - Home (La casita): volvía a la pantalla inicial.

- Flechas: cargaba las otras escenas (nueva imagen, texto y audio).
- Opciones: aun no tenía funcionalidad.



Figura 61 Primera aproximación. Folleto

Como se explicó en el apartado anterior de los botones, cuando el usuario hacía click en alguna de las flechas, se cargaba una nueva escena, la cual tenía su propio audio, imagen y texto. La aplicación estaba prevista para tener ocho textos lo que significaba que con este planteamiento, solo con la parte del folleto, la aplicación tendría que cargar nueve escenas distintas (las ocho del folleto más la inicial), lo que hacía que la aplicación no solo ocupase más espacio del necesario, sino que además fuera demasiado lenta. Esto llevó a investigar una solución para solucionar el problema llegando así a encontrar un tutorial que hablaba sobre los **ScriptableObject**.

El objetivo principal del **ScriptableObject** es ahorrar espacio en memoria, de tal manera que me permitía tener una única escena, con el único objeto de que contuviese todas las imágenes, textos, así como audios y al cual podía acceder y cargar el que quisiera en cada momento dado; de esta forma, las flechas que antes cargaban una escena nueva, ahora solo llevaban la cuenta de en qué página del folleto nos encontrábamos, mostrando de tal forma el contenido adecuado.

Gracias a esta metodología se logró disminuir los tiempos de carga, pero al ser una única escena con todos los contenidos precargados, la transición entre la escena inicial y la escena del folleto era muy lenta. Ello se debía en gran parte por la carga de las imágenes y los audios. En ese momento la aplicación solo estaba disponible en español,

si además del contenido actual, tenía que precargar también los textos y audios en Inglés, la aplicación resultaría inviable. Por lo tanto se decidió intentar un último cambio.

En lugar de usar los ScripTable para cargar el contenido, lo cargaría a la antigua usanza, usando un Switch Case, el cual dependiendo de la página en la que nos encontrásemos cargaría por ruta, los recursos previamente alojados en la carpeta correspondiente.

En este apartado Unity es muy estricto y cualquier elemento que desees cargar mediante la función Load (), ha de encontrarse bajo el directorio "Resources", ya que si no es así, no funcionará, independientemente de que indiques la ruta exacta. De esta manera los tiempos de carga se redujeron bastante, dando lugar a una aplicación usable.

En el Anexo V se podrán encontrar una muestra de los códigos usados.

6.4.4 Realidad aumentada

Una vez terminada la primera fase del proyecto, tocaba ponerse a desarrollar la parte más complicada. La realidad aumentada.

El primero de los pasos era crear los marcadores según se había acordado en el viaje a Madrid, siendo el logo el que hiciera a su vez de marcador. Para ello había que realizar algunos pequeños cambios en la posición de uno de sus elementos, consiguiendo así crear varios marcadores parecidos, partiendo de un mismo componente.

Pero no iba a resultar tan fácil, pues la imagen escogida como marcador no servía, ni siquiera cambiando el color de la imagen para obligarla a tener un mayor contraste.



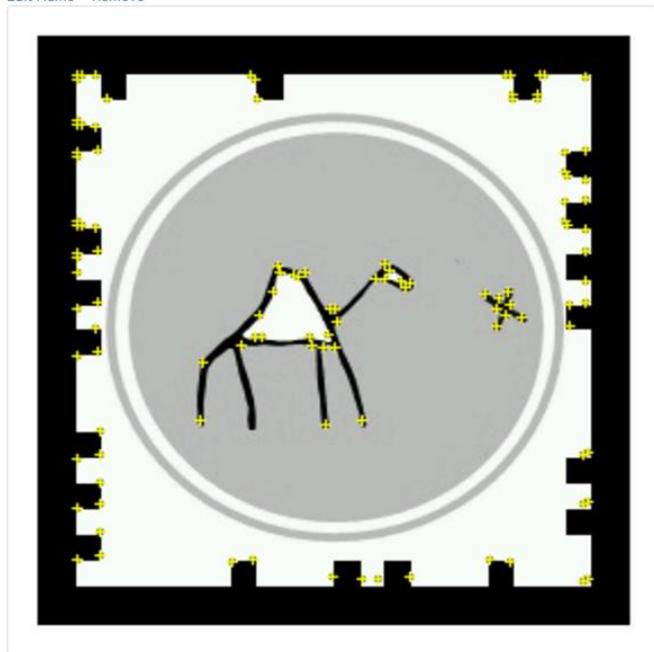
Figura 62 Primera prueba marcador

Por lo que deberíamos añadirle elementos que no afectasen demasiado al diseño del logo. La solución resultó en hacer una mezcla entre unas plantillas que proporcionaba Vuforia (los FrameMarker) y nuestra imagen.

El resultado fue bastante bueno y se decidió usar dichas imágenes como marcadores.

Target1

[Edit Name](#) [Remove](#)



Type: Single Image

Status: Active

Target ID: 8ddb5076f01e451caaacdadb663d253

Augmentable: ★★★★★

Added: Jun 29, 2016 03:46

Modified: Jun 29, 2016 03:46

Figura 63 Segunda prueba marcador.

Con el marcador seleccionado se comenzó a realizar las pruebas aumentadas. Por cada marcador necesitaba mostrar los distintos tipos de contenido aumentado.

- Imágenes.
- Textos con audios.
- Videos.
- Modelo 3D, el cual indicaría en qué otra parte del templo se podían encontrar graffitis del mismo estilo.

Con tanta diversidad de contenido, se debía de buscar la forma de organizarlo para que el usuario pudiera acceder a cada contenido aumentado sin que el resto de elementos le incordiasen. Un menú inicial parecía ser una buena solución.

En dicho menú, se mostrarían una serie de opciones, representadas por burbujas las cuales al ser seleccionadas ocultarían el resto de opciones disponibles y mostrarían el contenido asociado. Así cuando la aplicación reconociera el marcador se mostrarían las siguientes opciones:

- Fotos
- Video
- Texto
- Mapa

El usuario seleccionaría, por ejemplo, la burbuja de “Foto”, la cual ocultaría todas las burbujas anteriormente citadas, mostrando las fotos correspondientes a dicho marcador y una nueva burbuja, que daría la opción de volver a la vista anterior. Para llevar a cabo dicho planteamiento, primero se debería crear el menú, lo cual no resultó muy difícil.

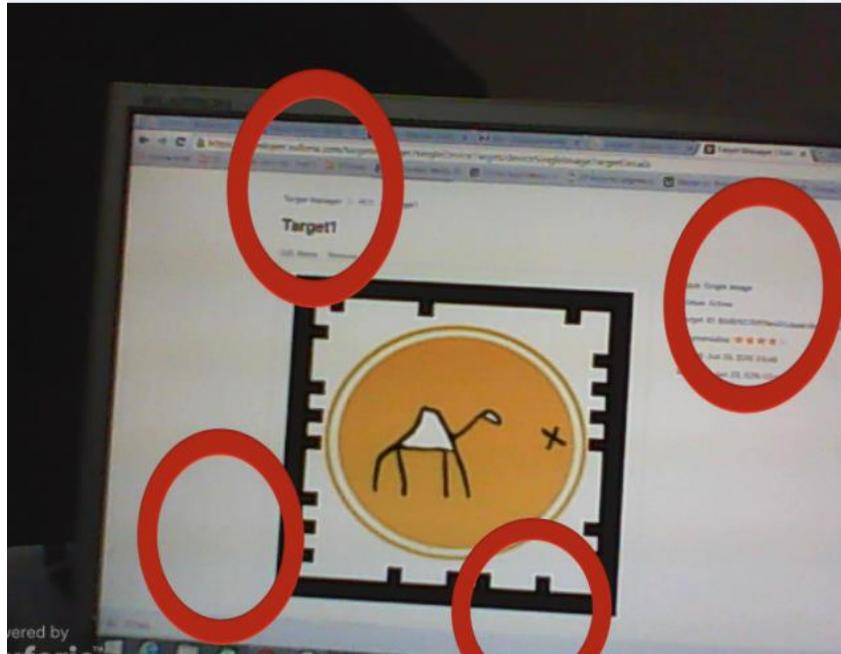


Figura 64 Pruebas AR. Menu.

Una vez conseguida una versión preliminar del menú, se prosiguió con el trabajo y realización de pruebas con los contenidos más complicados; llegando a una conclusión, que hizo replantear la forma de mostrar el contenido final.

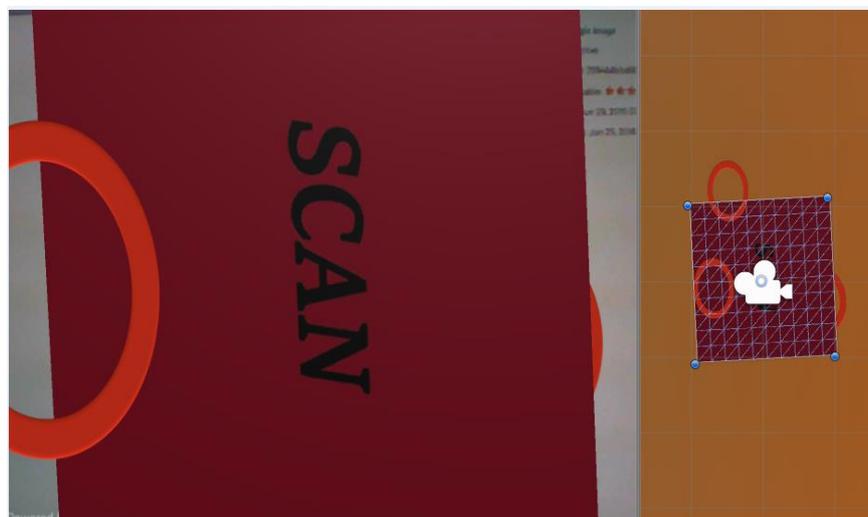


Figura 65 Pruebas AR. Imagen.

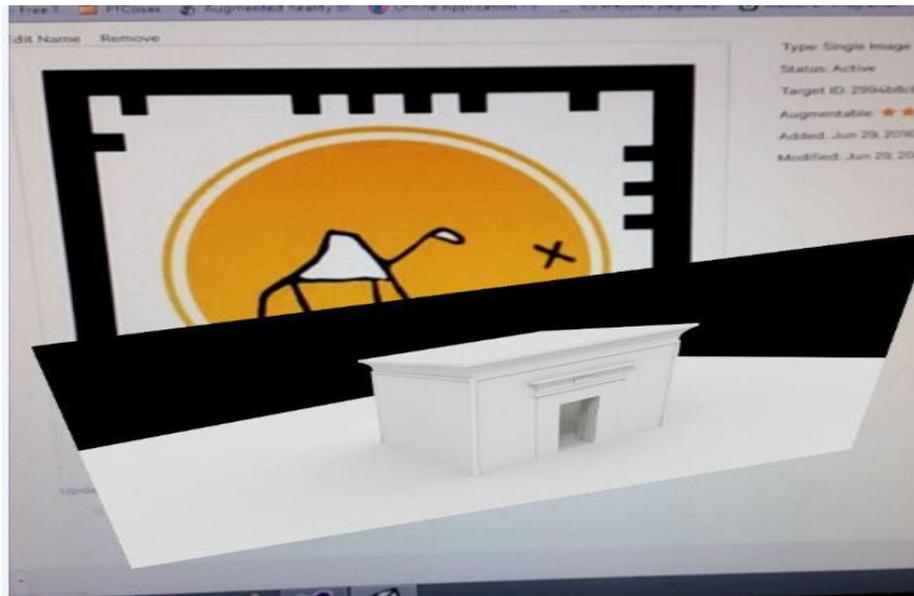


Figura 66 Pruebas AR. Video.

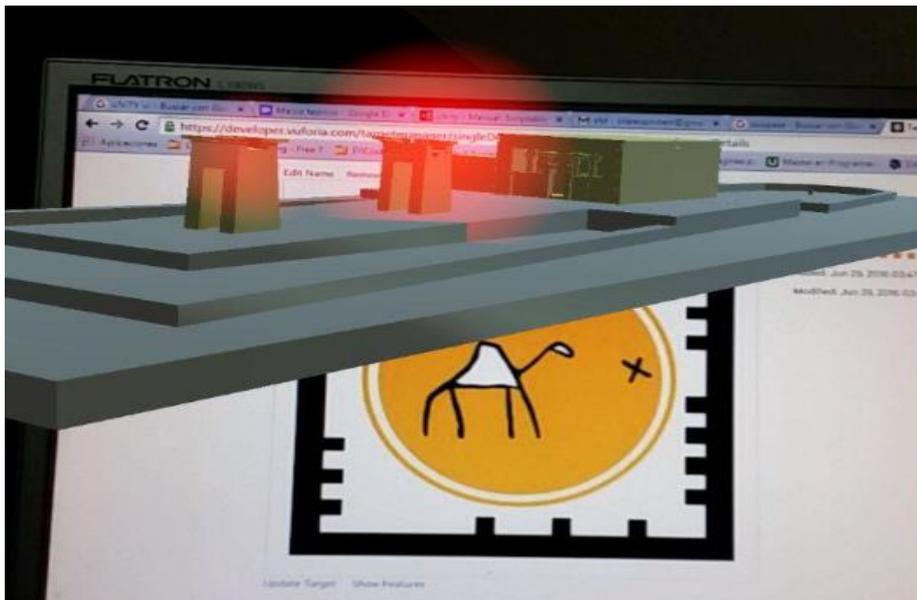


Figura 67 Pruebas AR. Modelo 3D.

Para que un marcador sea reconocido, el usuario ha de estar cerca de él, enfocándolo con el dispositivo en todo momento, ya que en cuanto el marcador salga del campo de visión del objetivo, el contenido desaparecerá. Esto podría tener sentido para una aplicación de uso personal y solitaria, como por ejemplo, en un periódico. Pero no en un

espacio cultural, donde se prevé que varias personas estén compartiendo un mismo espacio, en el mismo periodo de tiempo.

Si se dejase la aplicación con esta forma de mostrar el contenido virtual, se podría generar colas y malestar, entre los posibles visitantes a la hora de acceder a los marcadores y eso, no parecía para nada aceptable.

Por lo tanto, se decidió que el menú virtual previamente mostrado, cargase diferentes escenas individuales sin realidad aumentada, de tal modo que el usuario solo tuviera que enfocar su dispositivo hacia el marcador, para decidir que deseaba ver y una vez seleccionado, pudiera alejarse del marcador, enseñar el resultado a un compañero, dar vueltas por la zona...

Esta solución era mucho menos visual y descargaba a la aplicación de contenido aumentado, dejándola al final como una aplicación móvil con trazas de realidad aumentada; nada que ver con lo que imaginábamos al principio. Pero si queríamos primar la efectividad y lógica frente a la novedad, esta era la mejor opción.

Finalmente tenía terminada la aplicación, o eso era lo que se pensaba. De momento solo se había probado con un marcador, pero cuando me encontraba haciendo el resto de marcadores surgió una duda.

Si se iba a reutilizar las escenas que se cargaban en cada resultado, esto es, la escena de video, imágenes, texto y modelo 3D. ¿Cómo sabría la aplicación que contenido debía cargar en dichas escenas? Ya que dicho contenido, variaría dependiendo del marcador que hubiese sido reconocido.

Esto obligó a indagar dentro del código de Vuforia, buscando como se sabía en cada momento que marcador tenía delante el dispositivo, debía de existir un identificador y una forma de acceder a él. Al final, después de varios días estudiando los códigos de la aplicación, se encontró y logró usar el identificador de los marcadores adecuadamente, pudiendo así saber antes de cargar las diferentes escenas de que marcador se trataba y qué datos debería mostrar la aplicación en la siguiente escena.

Era el momento de probarlo, para saber si la aplicación sería capaz de reconocer y soportar 8 marcadores diferentes.

La primera impresión confirmó uno de los temores que se intuían. Ocho marcadores distintos eran demasiado para una aplicación móvil. La app tardaba mucho en cargar la escena que se encargaba de hacer el reconocimiento, por lo que tras un par de ejemplos, la cámara del móvil colapsaba. Por no mencionar que a cada ejecución que se realizaba calentaba el móvil más y más. Y por si fuera poco, la aplicación confundía los marcadores y mostraba datos que no pertenecían al marcador reconocido.

He de reconocer que en este punto me asusté bastante. La única opción que quedaba era realizar dos escenas distintas, una para escanear marcadores en la parte exterior del templo y otra para escanearlos en la parte interior, pero esto no parecía una buena solución, así que se optó por probar con otro tipo de marcadores, los FrameMarkers.

En lugar de usar los FrameMarkers para crear mis imágenes, los usaría como marcadores, eliminando así la necesidad de cargar ningún tipo de imágenes en la aplicación, puesto que Vuforia ya dispone de dichos marcadores dentro de la misma herramienta. Tan solo había que añadirle de fondo el logotipo del proyecto e indicar, en cada uno de ellos, de que FrameMarker se trataba. Esto en su ID.

Se realizaron las pruebas con los nuevos marcadores y no solo la aplicación final resultó ser mucho menos pesada, sino que además el reconocimiento era perfecto y fluido.

Finalmente se había finalizado con la parte importante del proyecto, solo faltaban algunos retoques gráficos, los cuales se tendrían que hacer con Photoshop (una herramienta que nunca había utilizado) y de paso, añadir algunas opciones más.

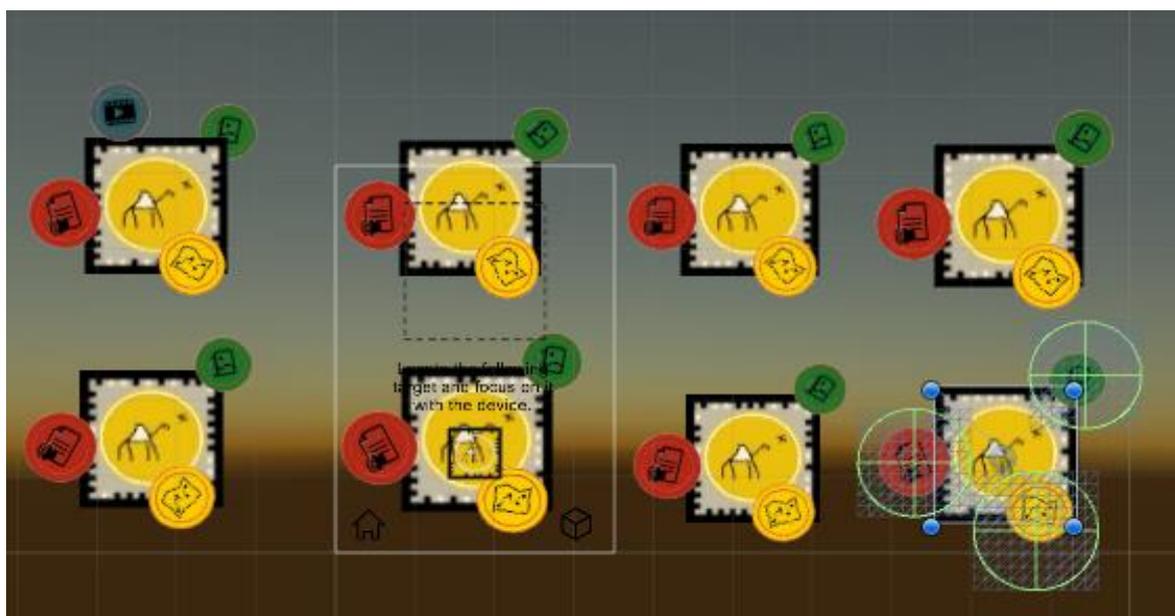


Figura 68 *Marcadores finales.*

6.4.5 Opciones

Desde un principio se pensó en añadir una opción de ayuda, puesto que el usuario final quizás no sabría manejar la aplicación, por lo que una guía explicativa de qué significaba cada botón y hacia donde le dirigía cada opción parecía algo necesario.

Puestos a añadir información adicional, se pensó que quedaría bien incluir una pequeña explicación acerca del proyecto Ta-hut, así como una mención a todos los que habían trabajado en él, tal y como estaba en el tríptico. Por lo que se incluyó junto con la ayuda, creando así, la pantalla de “Opciones”, en lugar de una única pantalla de “Ayuda”.

Para ello se creó un nuevo canvas, el cual permanecería desactivado hasta que el usuario pulsara el botón o icono de “Opciones”. Tras el cual, se mostraría una primera vista en la que se daría la opción de elegir la explicación del proyecto, o la ayuda en sí. Una vez hecha la selección, esa vista pasaría a estar desactivada y la vista (otro nuevo canvas) de la opción escogida se activaría. En esta parte del desarrollo no surgieron muchos problemas, puesto que en gran medida se trataba de activar y desactivar elementos.

6.4.6 Idioma

Desde un principio se sabía que la aplicación final debería estar disponible en Inglés y Español. En un principio se pensó en añadir un campo más en la pestaña de opciones, pero mientras se investigaba cómo realizar el cambio de lenguaje, se encontraron varias opiniones profesionales que instaban a poner el idioma como un valor por defecto, el cual dependiera exclusivamente del idioma en el que estuviera el dispositivo. De tal forma que si un ciudadano Inglés se descargaba la aplicación, no se viera obligado a estar buscando en las opciones la forma de cambiarlo a su idioma nativo.

Siguiendo estas buenas prácticas, se creó un diccionario en Inglés, otro en Español y un pequeño parser, el cual después de comprobar el idioma del dispositivo, cambiaría todo el texto de la aplicación, tanto el texto plano como el perteneciente a botones. Para el resto de textos y audios, es decir, los cargados mediante la ruta del archivo, solo hace falta comprobar el idioma y dependiendo de cuál sea, cargaría un archivo u otro.

Para ello, según las buenas prácticas promovidas por la comunidad de desarrolladores de aplicaciones, todo texto por defecto, se encontraría escrito en inglés. De tal forma que si no existe una traducción para el idioma del dispositivo, se quede el idioma más globalizado, siendo así más probable que el usuario entienda algo de la aplicación.

No obstante, algunos títulos o textos que son en verdad imágenes, no pueden cambiarse mediante esta fórmula, por lo que se deberían crear nuevas imágenes para solventarlo.

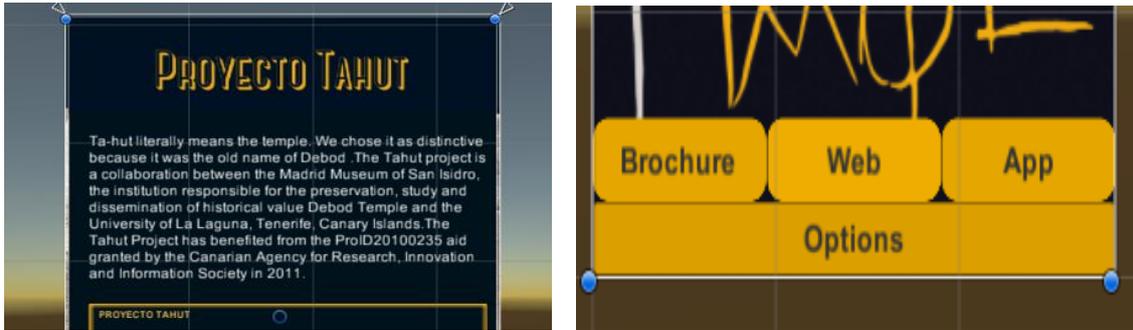


Figura 69 Ejemplos del idioma default.

6.5 Diseño Final

En este último apartado se mostrará la aplicación en su faceta final y se explicará su funcionamiento.

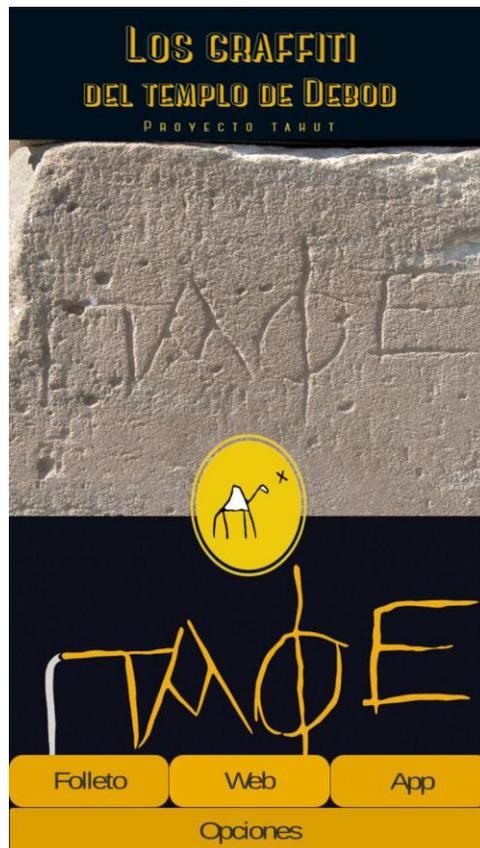


Figura 70 Escena inicial

La pantalla inicial no dista mucho de la mostrada en el prototipo. Consta de cuatro botones de los cuales solo dos redirigen a otras escenas distintas.

- Folleto. Como su nombre indica, carga la escena del folleto, la cual tiene como objetivo suplantar al antiguo tríptico.
- Web. Al igual que el caso anterior, habla por sí solo. Abre el navegador del dispositivo y carga la página web del templo.
- App. Carga la escena que se encarga de manejar la parte de la realidad aumentada.
- Opciones. Muestra la pantalla de opciones.

6.5.1 El Folleto



Figura 71 Escena del folleto.

La funcionalidad de esta pantalla ya se mostró anteriormente; solo se añadieron algunos botones al sistema de audio y se dotó de funcionalidad el botón de opciones.

- Texto. Es “scrollable” es decir, el usuario puede ir moviendo el texto para leerlo completamente, deslizando su dedo a través de la pantalla hacia arriba/abajo.
- Audio. Audio que narra el mismo texto que le precede; donde los controles son los habituales en piezas de audio o video.
- Botones inferiores. Empezando de izquierda a derecha:
 - Home. Redirige a la pantalla principal. Ocurre lo mismo si el usuario presiona la tecla de retroceso de su dispositivo.
 - Flechas. Muestran la página anterior/posterior del folleto. En caso de llegar a la última de todas , el botón de “siguiente” se desactivará, al igual que en la primera página el botón de “anterior” está desactivado.
 - Caja de herramientas. Muestra la pantalla de opciones.

6.5.2 App

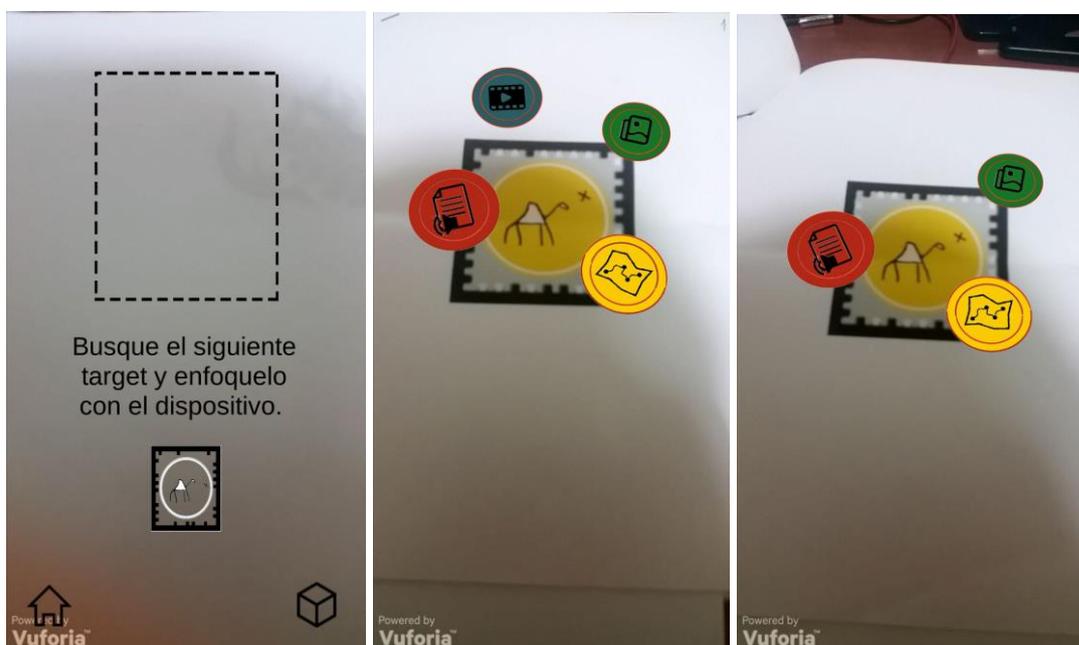


Figura 72 Escena de la AR.

La parte de la realidad aumentada si cambió bastante, respecto a lo imaginado en el prototipo. Como ya explicamos anteriormente, la aplicación mostraría un menú como contenido aumentado, el cual cargaría la escena correspondiente con su contenido específico.

En las dos imágenes de la derecha podemos observar cómo no todos los marcadores tienen activas todas las opciones del menú, de hecho, solo el primer marcador dispone de video. Aunque en las imágenes no se pueda apreciar, las opciones mostradas se mueven por sus respectivas zonas, dando la sensación de que están levitando sobre el marcador.

En la primera imagen, podemos ver como se muestra un pequeño texto, explicando al usuario que es lo que ha de hacer en esta vista, así como la tecla de "Home" y "Caja de herramientas", las cuales, mantienen su funcionalidad.

A continuación mostraremos la escena que se carga en cada una de las opciones mostradas:



Figura 73 Icono imágenes.

Imágenes

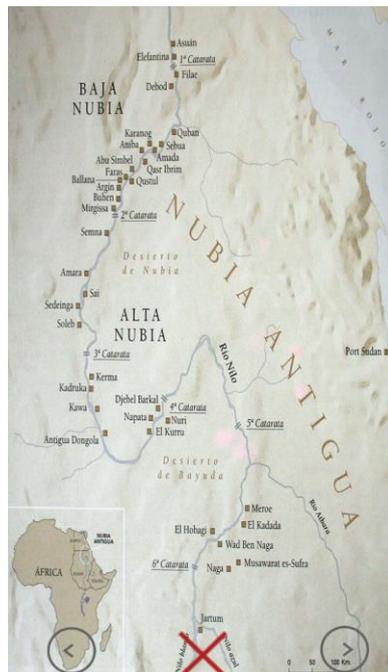


Figura 74 Escena de imágenes.

Esta escena se encargará de mostrar las imágenes disponibles para este graffito. En algunos casos solo hay disponible una única fotografía, en tal situación, las flechas que se muestran en la parte inferior de la pantalla estarán desactivadas.

La X grande y roja cierra la escena y vuelve a la escena anterior, es decir, la encargada de reconocer los marcadores.



Figura 75 *Icono texto.*

Texto

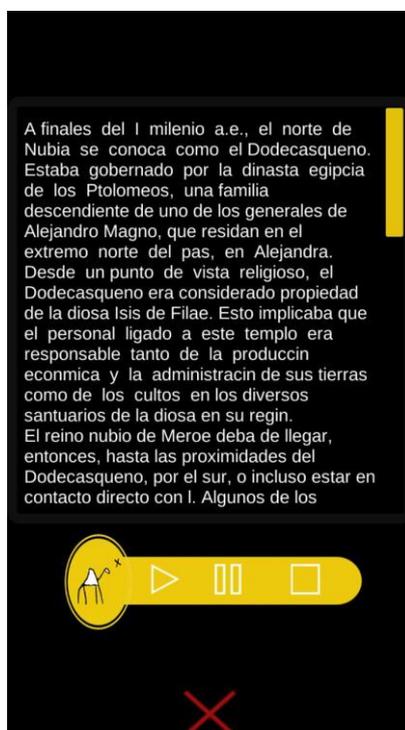


Figura 76 *Escena del texto/audio.*

La escena resultante en este caso, está compuesta por un texto y un audio, los cuales, se manejan exactamente igual, que sus homólogos en el folleto anteriormente explicado.

La X, al igual que en la escena anteriormente explicada y que en el resto de escenas siguientes, vuelve a la escena principal de la parte de AR.

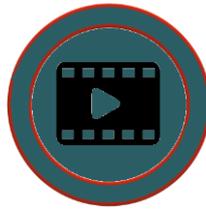


Figura 77 *Icono video.*

Video

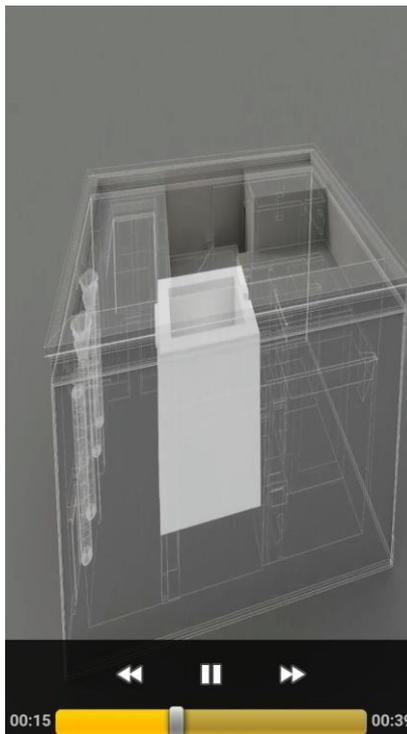


Figura 78 *Escena del video.*

Figura 66:

La escena de video, como su nombre indica, muestra un video que se comienza a reproducir nada más cargar la escena. En la misma, aparte del video, se pueden ver los controles básicos para su resolución.

Para salir de esta escena no hay un botón especial, por el contrario, se deberá pulsar el botón de retroceso del dispositivo.



Figura 79 Icono mapa.

Mapa

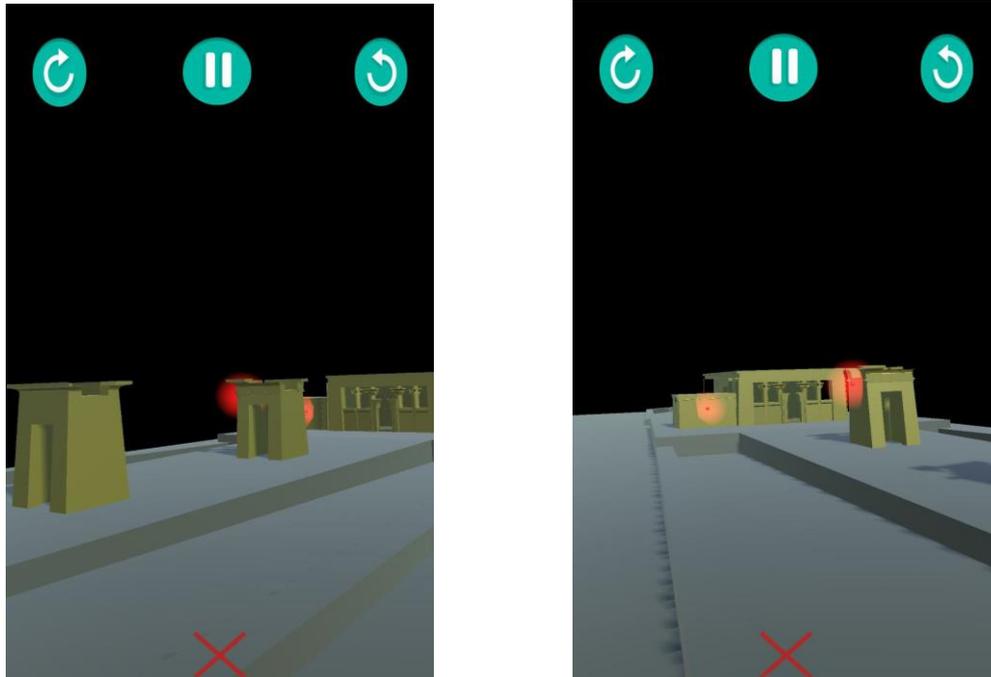


Figura 80 Escena modelo 3D.

Esta opción muestra un modelo 3D, en el que se remarcan con luces rojas, el lugar donde se encuentran otros graffitis del mismo estilo, al que está junto al marcador actual.

Quiero aclarar en este punto, que dichas marcas están puestas de manera totalmente aleatoria, ya que desconozco la localización exacta de cada graffito. De la misma manera, podrían existir más o menos localizaciones.

En cuanto al control de esta pantalla.

- Control táctil. El usuario puede deslizar su dedo de hacia los lados para mover el modelo, al igual que puede “pellizcar” la imagen para hacer zoom.
- Botones de rotación. Situados en la parte superior a ambos extremos de la pantalla, hacen girar el modelo sobre su eje.
- Botón de pausa. Para la rotación.

Para salir y volver a la escena del reconocimiento, se ha de pulsar la X roja, al igual que en sus escenas hermanas.

6.5.3 Opciones

Como ya se ha comentado, se puede acceder a la vista de “Opciones”, desde las tres pantallas principales.

- Mediante el botón de opciones. En la pantalla *Inicial*.
- Mediante el botón “Caja de herramientas”. En las pantallas de *Folleto* y *App*.

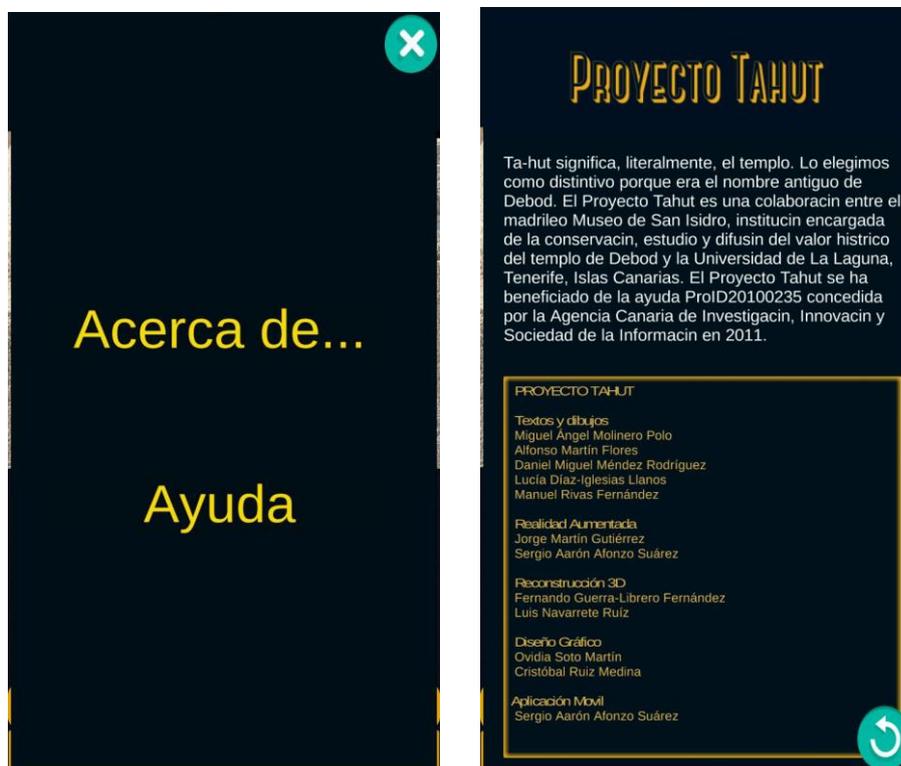


Figura 81 Panel de opciones.

La primera imagen que se vería al pulsar cualquiera de esos botones, sería la del cuadro de opciones (Imagen de la izquierda).

En esta vista, las acciones son limitadas. Para cerrar el cuadro de opciones bastará con pulsar el botón con la X que se encuentra en la esquina superior derecha. Si por el contrario se pulsara la opción de “Acerca de...” se abriría la imagen de la derecha en la cual se muestra un breve resumen del proyecto Ta-hut; así como un recuadro en el que

se nombran a todas las personas que han trabajado en la realización del tríptico y de la aplicación.

Para volver a la vista inicial del cuadro de opciones, bastará con pulsar el botón de la esquina derecha inferior. Si por el contrario se pulsara la opción de “Ayuda”, se abriría la vista del manual de instrucciones.

Esta vista es un poco más compleja que las anteriores. Y se divide también en tres secciones:

- **Botones superiores.** Los cuales nos servirán para decidir que parte de la aplicación queremos que se nos explique. Se encuentra dividido en :
 - Pantalla Inicial.
 - Folleto.
 - App.
- **Imagen y texto.** La imagen superior mostrará con un recuadro rojo, la parte que se está explicando en el texto, mostrado este, en la parte inferior de la misma. Al pulsar las flechas se cambiará de imagen y texto.
- **Botón de “Volver”.** Esquina inferior derecha, al igual que en la vista anterior.



Figura 82 Panel de ayuda de la pantalla inicial y del folleto.

El botón que muestra la ayuda de la parte de realidad aumentada (el último de la derecha), tiene una funcionalidad especial, dado que la ayuda en esta parte, es más compleja al incluir varias escenas distintas con sus propios controles y al pulsar el botón de “App”, se activaría una nueva fila de botones entre la imagen y el texto, los cuales harían referencia a todas las escenas de esta parte del proyecto.

Así, la nueva ristra de botones, nombrados de izquierda a derecha, mostrarían la siguiente ayuda:

- Escena principal de la realidad aumentada (Escaner).
- Escena del modelo 3D (Mapa).
- Escena de las Imágenes (Galería).
- Escena de texto y audio (Explicación).
- Escena de video (Media).



Figura 83 Panel de ayuda de la parte de AR.

En el Anexo V, se muestran todos los marcadores utilizados en la aplicación final.

Capítulo 7

Conclusiones

En este capítulo final, corresponde mencionar aquellos elementos que a modo de conclusiones, hayamos podido extraer de este proyecto de final de carrera que nos atañe, realizando un análisis del trabajo realizado, así como los resultados obtenidos y las posibles mejoras que se le podrían incorporar, pues en este ámbito, antes de acabar del todo un desarrollo, están llegando indicios y la aparición de nuevas herramientas, que con mucha probabilidad, mejorarían el resultado del mismo.

Como último apartado, nos arriesgaremos a dar una previsión de futuro, de hacia dónde nos lleva el uso de la realidad aumentada y sus posibilidades. No dejando de ser simplemente una opinión particular, basada en lo leído y consultado en este tiempo de preparación y elaboración del proyecto.

7.1 Conclusión

A lo largo de este proyecto, se han explicado varios conceptos que posiblemente a priori, el lector medio desconocería, pero que al ir dirigido al tribunal evaluador, parto de la idea, de que sabrán valorar el esfuerzo realizado, dentro de la aparente simplicidad del reto afrontado.

Como hemos podido comprobar, el campo o amplitud de uso de la realidad aumentada, puede llegar a ser considerable en el día a día del usuario medio, pudiendo servir para muchas tareas rutinarias; desde escoger un sitio para comer en función de la cercanía, con opiniones y recomendaciones de los clientes, hasta saber el precio de los productos que tienes en frente, sin necesidad de buscar la etiqueta correspondiente en las estanterías.

Pero especialmente en el ámbito que nos ocupa en el presente, sin esperar siquiera al futuro; pues cada vez nos lo encontramos en alguna salida turística, o en algunas instalaciones de servicios culturales, donde se ofrece la “Realidad Aumentada” como un valor añadido y complementario en el aporte de información a clientes y usuarios, que puedan contar con los medios tecnológicos adecuados (móviles), no solo para captar textos, sino audios, vídeos y otros elementos visuales. Tecnología, que hoy por hoy, está al alcance de muchos y con tendencia a generalizarse, en sustitución de otros soportes convencionales, mucho más costosos económicamente y que nunca llegarían a la actualización casi constante y a la complementariedad y variedad que se puede ofrecer con estas.

Conociendo el objetivo y el contexto; con las tecnología existentes que podrían hacer posible la realización de dicho proyecto, era cuestión de documentarse e ir realizando las pruebas necesarias para poder hacer un correcto uso de las herramientas que se iban encontrando. Hasta conseguir con el objetivo final, que no era otro que concluir con cierto éxito el proyecto, con sensación de utilidad, de poder ser aplicado para aquello para lo que se pensó.

La aplicación que se imaginó inicialmente se iba a apoyar mucho más en la realidad aumentada, ya que el templo contaba con su tríptico informativo y su aplicación previa, la cual seguiría siendo utilizada junto con la aplicación. No obstante, la desaparición de Metaio afectó de manera global al proyecto, ya que al dejar de existir esa plataforma, la aplicación anterior dejaba de funcionar y dado que dicha aplicación se nombraba en la versión final del tríptico, éste quedaba desfasado, con miles de copias impresas, que de utilizarse, darían una gris imagen.

Por nuestra parte, la desaparición de Metaio repercutió en que la mayor baza tecnológica de la que se disponía, la localización y mapeado mediante nube de puntos (SLAM) quedaba fuera de nuestro alcance, por lo que nos vimos obligados a replantear el proyecto, llegando a la conclusión final de que se necesitaría hacer uso de marcadores para poder mostrar el contenido aumentado.

De esta manera se llegó al uso “Vuforia” como alternativa convirtiendo así el proyecto, en una aplicación que haría a la vez de folleto informativo acerca del centro cultural Templo de Debod y ampliaría dicha información con tecnología aumentada basada en marcadores.

No obstante, tras lograr mostrar y seleccionar el contenido aumentado en la aplicación, se decidió que, teniendo en cuenta al público y el objetivo, al cual se enfocaba la aplicación, no era factible hacer una aplicación de realidad aumentada equivalente a la usada sobre libros de textos o folletos, puesto que el tránsito de personas y el espacio reducido en las habitaciones del templo, podría estropear la experiencia.

Por lo que la aplicación se vio forzada a cambiar de planteamiento una última vez, de tal manera que ahora, la única muestra de realidad aumentada era un menú animado, el cual servía para dividir la aplicación, mostrando un contenido distinto según el lugar del templo donde nos encontrásemos.

No obstante, la aplicación resultante cumple con todos los requisitos exigidos por el templo, que no es poco, además de ser robusta, fluida, bastante intuitiva y disponer de medios para garantizar su correcto uso y disfrute.

Finalmente, el proyecto final es resultado de haber trabajado especialmente:

- Unity Con un total de 8 escenas distintas, 30 scripts creados, 7 prefabs, 8 animaciones y 3 librerías añadidas (Contando Vuforia).
- Vuforia En sustitución de la también estudiada Metaio, probando y estudiando todas sus posibilidades y realizando cursos.

- Photoshop Para la creación de algunos botones propios, los cuales fueron creado a partir de iconos de licencia gratuita y la creación/modificación de los marcadores.
- Metodología ágil Para dividir el proyecto en tareas menores y obligarme a realizar pequeños avances semanales.

7.2 Líneas futuras

Cabe mencionar que las posibles mejoras de implantación son, esencialmente en cuanto a contenido; puesto que la aplicación se ha optimizado para su correcto funcionamiento, a lo largo de todas las etapas de su desarrollo y se ha hecho de tal manera, que sea fácilmente escalable. Aun así:

- Se pueden añadir más marcadores.
- Se puede añadir o eliminar opciones de los marcadores.
- Se pueden añadir páginas al folleto informativo.
- Se pueden añadir o eliminar puntos de señalización en el modelo 3D.
- Se pueden añadir idiomas nuevos

No obstante, cabría la posibilidad de mejora de algunos elementos gráfico, claro está, con el apoyo por parte del centro cultural Templo de Debod, puesto que no se disponían de conocimientos en el campo de la edición ni capacidad para el diseño gráfico.

- Se puede mejorar la presentación gráfica de la aplicación.
- Se pueden añadir botones más acordes con la temática del templo.
- Se pueden añadir imágenes con textos traducidos.
- Se puede añadir textura al modelo 3D.

Además, en un futuro próximo, la tecnología SLAM estará disponible en todas las herramientas de realidad aumentada, por lo que se podría abandonar la idea de los marcadores y volver al planteamiento original, es más, con los últimos avances en el campo, se podría hacer uso del aprendizaje del entorno (tecnología usada por Tango), para el reconocimiento a lo largo del templo, creando así una experiencia más rica e integral.

7.3 Aprendizaje

La tecnología de realidad aumentada ha resultado ser más profunda y entretenida de lo que en un principio se había llegado a imaginar, mostrándonos y convenciéndonos, que durante los próximos años se convertirá en un campo de estudio y desarrollo muy demandado dentro de la comunidad.

Días atrás, hablando con un amigo desarrollador, salió a la luz el tema de la realidad aumentada y comentó, que no le parecía una tecnología que tuviese futuro más allá de los videojuegos; al preguntarle la razón de por qué tenía esa idea, comentó que le

parecía absurdo tener que estar haciendo uso de imágenes específicas para mostrar un simple objeto.

Esa reflexión me hizo ver, que si el compañero, el cual se encuentra dentro de la comunidad activa de desarrolladores de nuevas tecnologías, desconocía de las posibilidades y bondades que nos ofrece la realidad aumentada. ¿Qué pensaría el usuario medio, que no dispone de los conocimientos técnicos y que está comenzando a leer en los medios, sobre una nueva tecnología que muestra elementos virtuales a través de nuestros móviles?

Con esta anécdota se pretende dar a entender, que si incluso dentro del mundo de los desarrolladores, la realidad aumentada aún es un misterio para algunos, a pesar de los movimientos y avances por parte de las grandes compañías, dentro del sector. Cuando el conocimiento y el uso de esta tecnología se popularice, el mercado se colapsará y se verá impulsado por la constante competición, en busca de ser el mejor y más cualificado; dando lugar a aplicaciones y usos de la realidad aumentada, que hoy no imaginamos.

Como ejemplo del impacto que puede tener el uso de la realidad aumentada en la sociedad moderna, queremos hacer eco de una aplicación que ha salido al mercado mientras se redactaba esta memoria del proyecto. Quizás algunos que investigan en las redes, tengan noticias de la aplicación a la que vamos a hacer mención.

Hace unos días, la compañía de videojuegos Nintendo, en conjunto con Niantic (los responsables de Ingress), sacaba en el mercado Americano, Australiano, Japonés y Neozelandés, la aplicación móvil disponible para Android e iOS, Pokémon Go.

Dicha aplicación, basada en el famoso universo protagonizado por Pikachu, hace uso de la realidad aumentada, para mostrar al usuario pequeños monstruos de bolsillo (de ahí el nombre, poke = bolsillo, mon = monstruo), en el mundo físico.

Dichos personajes ficticios no aparecen aleatoriamente, sino que por el contrario, están situados en lugares específicos de tu zona, dependiendo del tipo al que pertenezca el pokémon mostrado. En este mundo imaginado por Nintendo, cada pokémon es de un tipo específico (agua, fuego, piedra), el cual le da ventajas o desventajas frente a otros pokémons. Basándose en esto, Niantic pensó que sería más realista, hacer que pokémons del tipo agua aparecieran cerca de lugares con una considerable cantidad de agua, como playas, ríos, etc. Presentando de forma similar al resto de tipos pokémons.

De esta manera obliga al usuario a caminar por las calles, playas y montañas, si quiere lograr el objetivo de reunir a todos los pokémons existentes. Y son muchos.

Esta aplicación ha sido una sorpresa total, hasta tal punto, que en solo una semana y media, las acciones de Nintendo un 60% sobre su valor bursátil, siendo una de las aplicaciones más descargadas y usadas de todos los tiempos, tanto en Android como en iOS. Y todo esto, habiéndose comercializado tan solo cuatro mercados nacionales y en menos de dos semanas desde su salida.

El usuario es libre de buscar los videos, noticias e imágenes, que muestran a cientos de personas en la calle mirando sus dispositivos móviles, inmensas masas de gente rondando la misma zona, gritando el nombre del pokémon que andaban buscando, comisarías de policía desbordadas, emitiendo comunicados que explicaran a los ciudadanos por qué no podían entrar en su patio, para atrapar un pokémon, etc.

Esta aplicación, por inverosímil que parezca, no hace un uso innovador de la tecnología existente, al contrario, se basa en una de las técnicas de realidad aumentada más habituales de todas, el geoposicionamiento; lo que demuestra que no hace falta esperar a que la tecnología avance, para crear una aplicación mundialmente exitosa, sino que con la tecnología que tenemos y una buena idea (además de algunos millones), se puede lograr hacer historia.

Volviendo al proyecto, como punto final concluir que a pesar de las adversidades y los cambios de planes, el trabajo resultante, ha sido muy satisfactorio puesto que ha dado la oportunidad de conocer y aprender a utilizar tecnologías que para y por la propia formación, resultaban muy apetecibles; las cuales seguramente, que serán de las más comentadas y utilizadas en los próximos años.

Además se ha aprendido a tener siempre un plan B y un plan C, como alternativas ante imprevistos, teniendo que adaptar el proyecto a las circunstancias, aunque sin renunciar a la primera a conseguir el objetivo prefijado; puesto que como se ha advertido desde un principio y muchas veces a lo largo de la carrera, este mundo de la tecnología está en constante movimiento, cambiando de manera muy rápida e impredecible. Y por lo tanto, los informáticos, como desarrolladores, nos vemos en la obligación de estar aprendiendo siempre, si no queremos quedarnos desfasados y obsoletos de un año para otro.

Anexos

Anexo I. Buenas prácticas para la selección de targets.

A la hora de decidir qué imágenes queremos usar como marcadores para nuestra aplicación de realidad aumentada, deberíamos tener en cuenta ciertos criterios para saber si nuestra elección de target es aceptable.

1. La imagen debería tener muchos colores, bordes claros y altos contrastes.
2. Son preferibles formatos comunes (rectángulos, cuadrados...)
3. No usar textos o superficies planas y sombreadas
4. El el caso de que el resto de la imagen esté sombreada o sea monocolor, recortar la mejor parte para usarla como patrón.
5. Que la imagen no sea muy oscura o con puntos de reflexión (reflejos)
6. Cuantos más vértices contenga la imagen, mejor.

Además de estos puntos, cada herramienta puede tener otros sistemas de evaluación añadidos. Algunas por ejemplo, ofrecen herramientas que dan soporte a la hora de elegir un marcador correcto.

En el caso de Vuforia, nos valora la calidad de la imagen como target, añadiéndole estrellas (en un rango de 0-5) y nos da la opción de ver cuántos puntos interesante (*Blobs*) contiene nuestra imagen.

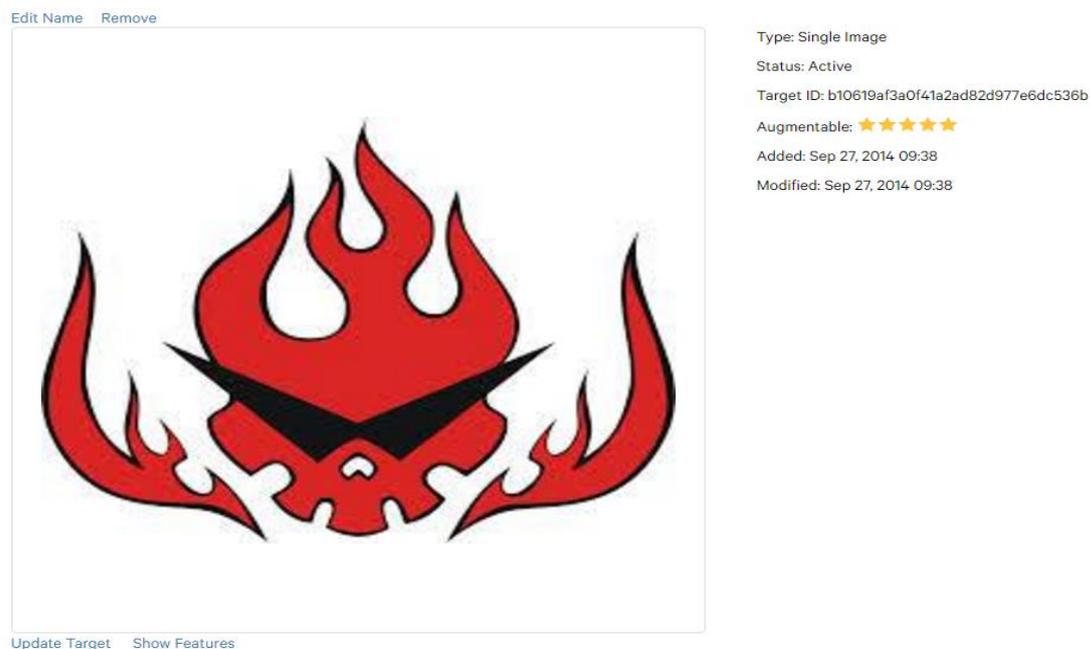


Figura 84 Ejemplo buen target.

Edit Name Remove



Type: Single Image
Status: Active
Target ID: b10619af3a0f41a2ad82d977e6dc536b
Augmentable: ★★★★★
Added: Sep 27, 2014 09:38
Modified: Sep 27, 2014 09:38

Update Target Hide Features

Figura 85 *Ejemplo buen target mostrando blobs.*

Nota: Vuforia no recomienda usar targets que obtengas una puntuación de tres estrellas o menos en su plataforma.

Anexo II. Contenidos ofrecidos por el cliente.

Tríptico



Figura 86 Cara exterior del tríptico.



Figura 87 Cara interior del tríptico.

Textos en español

Imagen 1. Fachada de la capilla de Adikhalamani

Dibujo en realidad aumentada 1: mapa de la Baja Nubia con los emplazamientos mencionados: Tebas, primera catarata, isla de Filae, Debod, wadi Allaqui, Dakka, Maharraqa

Dibujo en realidad aumentada 2: reconstrucción 3D de la capilla de Adikhalamani

A finales del I milenio a.e., el norte de Nubia se conocía como el Dodecasqueno. Estaba gobernado por la dinastía egipcia de los Ptolomeos, una familia descendiente de uno de los generales de Alejandro Magno, que residían en el extremo norte del país, en Alejandría. Desde un punto de vista religioso, el Dodecasqueno era considerado propiedad de la diosa Isis de Filae. Esto implicaba que el personal ligado a este templo era responsable tanto de la producción económica y la administración de sus tierras como de los cultos en los diversos santuarios de la diosa en su región.

El reino nubio de Meroe debía de llegar, entonces, hasta las proximidades del Dodecasqueno, por el sur, o incluso estar en contacto directo con él. Algunos de los monarcas meroítas colaboraron, a fines del s. IV a.e., en actividades constructivas en varios de estos templos, como el de Isis de Filae y el de Thot en Dakka. Se ha supuesto que estas muestras de devoción estarían ligadas al viaje de la estatua de la diosa a los distintos santuarios de su propiedad, un desplazamiento que se realizaba anualmente.

En 206 a.e., una revuelta en la Tebaida provocó la escisión del sur de Egipto frente a Alejandría, la capital ptolemaica. Los reyes meroítas aprovecharon ese conflicto interno egipcio para extender su dominio sobre todo el Dodecasqueno.

Es en ese momento, bajo el rey Adikhalamani (que gobernó, aproximadamente, entre el 200 y el 185 a.e.), cuando se erige en Debod la capilla que lleva su nombre. Ésta es el núcleo central del templo actual.

Tras la reconquista de la región por Egipto, los reyes alejandrinos iniciaron una política constructiva que les permitiera recuperar la fidelidad de sus habitantes. En el caso de Debod, esta actitud significó la ampliación del edificio tal como lo conocemos hoy. En esa intervención ptolemaica, la devoción religiosa se unía al interés político. La zona se había convertido en una vía excepcional de tráfico de mercancías. Alejandría se proveía de productos exóticos africanos a través de la Baja Nubia; ésta permitía también el acceso a las minas de oro del wadi Allaqui, reabiertas por la dinastía.

Con la conquista romana de Egipto la frontera quedó consolidada en Maharraqa durante varios siglos, varias decenas de kilómetros al sur de Debod. El templo se mantuvo, por tanto, dentro del territorio del Imperio romano, en un espacio de actividad económica próspera y de mantenimiento de los ritos de la religión egipcia tradicional.

Imagen 2: Graffito toproskynema

Dibujo en realidad aumentada 3: el mismo graffito

El templo de Debod se construyó para albergar a dos dioses, Isis de Filae y Amón de Debod, lo que es una circunstancia poco habitual.

El ritual de culto diario se desarrollaba en las cámaras internas. Es posible que estas también recibiesen una de las etapas de la procesión anual de Isis por el Dodecasqueno. Otras actividades ceremoniales tendrían su marco en el espacio público exterior.

Alguno de los *graffiti* fueron producidos en el momento en que Debod albergaba todavía el culto a sus dioses originales. En la identificación de estos *graffiti* hay que tener en cuenta una premisa básica: el interior de los templos egipcios sólo estaba abierto a los sacerdotes; estos tenían que haber pasado por un proceso de purificación individual, respetado ciertas normas en su indumentaria y haber cumplido unas reglas estrictas de higiene personal y comportamiento social. En consecuencia, los *graffiti* que se grabaron cuando el culto estaba activo no pueden estar en el interior, solo pueden hallarse en aquellos espacios donde tenía acceso la población.

El *graffito* en que se lee en griego τοπροσκυνημα με, ubicado en la cara oriental de la portada del segundo pilono, cumple esta regla. La primera palabra, *toproskinema*, significa 'veneración'. Siempre iba seguida por el nombre del fiel; en este caso, el devoto grafitero solo llegó a grabar la primera sílaba de su nombre, με, *me*. El primer término indica por su mismo significado que se hizo cuando la deidad residía aún en el interior del santuario. El uso de la lengua griega no debe extrañar. Tras la conquista macedónica, la sociedad egipcia se había ido convirtiendo en bilingüe, con una presencia en aumento continuo del griego; la conquista romana no supuso un cambio significativo en esta tendencia.

Imagen 3: dromedarios de la pared norte

Dibujo en realidad aumentada 4: uno de los dromedarios

A mediados del s. III de nuestra era, se produjo el paulatino establecimiento de tribus nómadas o seminómadas en la Baja Nubia y en los desiertos circundantes. Habían adquirido capacidad de movilidad tras la adopción de los dromedarios como animales de monta y transporte; esta nueva arma había favorecido también su reestructuración social. Los blemmíes, procedentes del desierto oriental, se instalaron en la Baja Nubia, desde donde se atrevieron a atacar el extremo meridional del Imperio romano. Los nóbadas, originarios del desierto occidental, se establecieron entre los blemmíes y la primera catarata, actuando como protectores de los intereses de Roma.

Sin embargo, a pesar de los desembolsos romanos para mantenerlos apaciguados, los nómadas hicieron causa común en varias ocasiones. El ataque a Filae del año 450 es de especial importancia histórica, pues tras la respuesta de Constantinopla se firmó un tratado. De acuerdo a sus términos, los nómadas, fieles aún a los dioses antiguos, recibieron el permiso para participar en las fiestas de Isis en su templo, que se mantenían aún activas, y para llevar la estatua de la diosa a su país, en momentos señalados del año, conservando la tradición de la procesión isíaca por el Dodecasqueno. Este tratado es el que aseguró que Filae se mantuviera como centro de culto pagano a pesar de los decretos de imposición del cristianismo y de prohibición de los dioses tradicionales que pusieron fin a la religión egipcia antigua en el resto de Egipto. Suponemos que la pervivencia del culto pagano en Filae puede extrapolarse también a Debod, aunque las fuentes escritas guardan silencio sobre este aspecto. Sólo los decretos de Justiniano de 535-537 conseguirían hacer cerrar los santuarios de la isla de Isis. Para entonces, el proceso de cristianización de la Baja Nubia ya se había iniciado restando fuerzas a una hipotética contestación.

Varias escenas con dromedarios y, excepcionalmente, otros animales del desierto, pueden corresponder a este momento. No parece extraño que las poblaciones nómadas representasen al animal que se había convertido en el centro de su nuevo modo de vida y de su poder. Una segunda circunstancia, complementaria a la anterior, aboga por esta atribución: todas las escenas con dromedarios están en el exterior del templo. De ambas premisas podríamos deducir que son obra de una población que no entra en el interior del santuario, ya fuera porque estaba todavía vetado a los fieles pues aún se mantenía el culto tradicional, ya fuera por respeto a las antiguas prácticas que se habían desarrollado en él en un pasado muy reciente y de las que los nómadas se consideraban herederos.

La más significativa de las escenas de animales es la que ocupa la pared exterior norte, una veintena de dromedarios y dos humanos que los llevan de las riendas. A una distancia de dos sillares está tallado un cuadrado que parece el destino de animales y hombres. La escena debe de representar una caravana que se dirige hacia su lugar de descanso, aunque también podríamos relacionarla con la peregrinación de los nómadas a Filae en días festivos.

Imagen 4: cruz patada copta

Dibujo en realidad aumentada 5: el mismo *graffito*

Foto en realidad aumentada 1: *dipinti* de cruces patadas

Foto en realidad aumentada 2: rostro de una divinidad con testimonio de *damnatio*

Una inscripción del rey nóbada Silko, esculpida en torno al 545 sobre la fachada del templo de Kalabsha, muy cerca de Debod, afirma que Dios le dio la victoria sobre los blemmés. El texto señala también el triunfo del cristianismo en la Baja Nubia y la consolidación del reino de Nobatia. Seguramente el proceso de cristianización se estaba desarrollando desde varias generaciones antes. En él no estuvieron ausentes las influencias de monjes coptos, mercaderes y viajeros de diversos intereses procedentes del Egipto ya cristianizado. La consecuencia es que el cierre de Debod a los cultos paganos fue definitivo.

La Nubia cristiana, unificada, se convertiría en una de las potencias de la época. El “Gran Rey” de Nobatia tenía a más de una docena de “pequeños reyes” bajo su dependencia. La región inició un proceso de intensa urbanización. Numerosas iglesias atestiguan también el desarrollo de un estilo arquitectónico y decorativo propio.

Resulta difícil llegar a determinar si Debod fue convertido en iglesia cristiana tras el abandono del culto pagano. No hay textos medievales que lo afirmen o permitan suponerlo. Los primeros viajeros europeos tampoco mencionan la incorporación de elementos arquitectónicos o culturales que hubieran facilitado el desarrollo de rituales del nuevo credo. Además, las salas parecen pequeñas para congregar a los fieles en una oración común.

Sin embargo, hay una presencia muy llamativa de cruces patadas en el templo. Las hay grabadas y las hubo también pintadas –estas últimas hoy perdidas por la inmersión del templo durante décadas en las aguas de la primera presa de Asuán–. Las cruces patadas son las características del cristianismo egipcio. Se trata de cruces con brazos de igual longitud –como las griegas– que terminan en dos prolongaciones abiertas en ángulo. Casi todas las de Debod se encuentran en el eje principal del edificio, es decir, en el trayecto desde el exterior a la capilla central, lo que parece indicar un cierto itinerario hacia ese espacio. La degradación actual de los rostros de la mayoría de los dioses de la capilla central puede ser otro testimonio del valor otorgado a ese espacio; son numerosos los templos egipcios donde una destrucción idéntica puede fecharse en época cristiana para la adaptación del edificio antiguo a nuevos usos religiosos. La presencia de las cruces y la condenación de los rostros de los dioses antiguos abre la pregunta de si la capilla de Adikhalamani o la capilla central pudieron haber sido usado como lugar de culto en época cristiana.

Imagen 5: cúpulas

Foto en realidad aumentada 3: una cúpula diferente a la del tríptico

Varios grupos de *graffiti* son de una cronología incierta y difícil de atribuir a alguna de las etapas concretas de la historia de Nubia. El que presenta un mayor número de ejemplares es el de las “cúpulas”. Su importancia no está sólo en su cantidad y en su visibilidad sino también en la información que proporcionan desde el punto de vista cronológico.

Se trata de grupos de depresiones fusiformes, alargadas, realizadas por frotación de un instrumento punzante sobre la superficie de la piedra con la finalidad de obtener cierta cantidad de arena, que se llevaban consigo los visitantes por considerarla sagrada. Se localizan únicamente en los portales de los pilonos. Su elevado número, unos quince conjuntos, y su profundidad, muestran la frecuencia de la visita y la veneración de la que fue objeto el edificio. Sobre uno de estos conjuntos se han superpuesto *graffiti* tanto cristianos como islámicos, lo que marca un término cronológico y muestra que este ejemplar fue anterior a esas creencias.

Así, aunque algunos autores testimonian que la práctica de obtener arena de lugares sacros ha llegado hasta nuestros días, al menos una parte de estos “arañazos” en el templo de Debod debe ser fechada en momentos anteriores a la irrupción del cristianismo o en los primeros momentos de éste.

Imagen 6: *graffito* islámico

Dibujo en realidad aumentada 6: el mismo *graffito*

En 1315, el rey nubio Kerenbes era hecho prisionero por los mamelucos de Egipto y reemplazado por Abdalla ibn Sanbu. Se ponía así fin a cerca de ocho siglos de Nubia cristiana. La islamización, que se había iniciado ya antes de ese cambio dinástico, fue un proceso muy rápido, aunque una minoría cristiana se mantuvo tolerada. La situación económica internacional había relegado a Nubia a una situación periférica. Los nuevos gobernantes no pretendieron tener ninguna presencia en la política africana.

A comienzos del s. XVI, la Baja Nubia fue conquistada por el Imperio otomano tras la toma de Egipto por el sultán Selim I. Una guarnición bosnia en Qasr Ibrim aseguró el control turco. Durante siglos quedó abandonada; su única función consistía en servir de paso hacia las regiones más meridionales del Valle del Nilo y hacia África central. El agotamiento fue casi total. Los arqueólogos que trabajaron en la Campaña de Salvamento de la Unesco, en la década de 1960, apenas han encontrado restos de ocupación humana de estos siglos, salvo en varios puntos estratégicos del Valle que habían sido reforzados como puestos de control.

La paulatina recuperación se produjo tras el paso de las tropas egipcias de Mohammed Ali, que entre 1820 y 1825, devolvieron el interés del Imperio otomano por el Sudán actual. Estas campañas abrieron la vía a comerciantes y la Baja Nubia fue convertida de nuevo en ruta de tránsito de marfil y esclavos hacia el Mediterráneo.

Una veintena de inscripciones en árabe son el testimonio de este largo periodo bajo control islámico. Se trata fundamentalmente de invocaciones a Alá y a Mahoma y proclamaciones de fe. El primer portal acumula buena parte de estos epígrafes. Como en el caso de las cruces cristianas, estas inscripciones debían de tener una finalidad de execración de la sacralidad pagana del lugar y de protección a los fieles de la nueva religión dominante. La imagen presenta una superposición de al menos cinco declaraciones de fe. En ellas puede leerse en árabe “la illah illa Allah”, lo que significa “No hay más dios que Dios”. Cada una parece escrita por una mano diferente, con signos de tamaño distinto unos encima de otros hasta el punto de que los más recientes cubren a los más antiguos.

Imagen 7: Nombre y fecha de un viajero europeo

Dibujo en realidad aumentada 7: el mismo *graffito*

Otro conjunto numeroso de *graffiti* está compuesto por las firmas de viajeros occidentales. En la Edad Media y en la Moderna, algunos misioneros europeos se desplazaban por el Nilo hacia el Sur; los más antiguos buscaban el reino casi mítico del Preste Juan; tanto ellos como los más recientes pretendían, sobre todo, hacer conversos al catolicismo. No se ha reconocido en Debod testimonios de su paso.

Desde comienzos del s. XIX, aparecen viajeros que atraviesan la región por el placer del “descubrimiento”: Ludwig Burckhardt recorre la Baja Nubia en 1813. Antes incluso de que publicara la memoria de su viaje, otros aventureros occidentales le emulan, ya fuera para conocer mejor la región, ya fuera para obtener informaciones sobre las antigüedades o para llevárselas. En Debod se conserva alrededor de treinta firmas de viajeros occidentales, la mayoría inscritas en las paredes de la cámara de los *naoi*. Suelen incluir el año en que llegaron al templo: la mayoría durante la primera mitad del s. XIX. Así, estos *graffiti* se inscribieron como un acto de reafirmación personal, una forma de dejar constancia del valor y de la determinación que había permitido a sus autores llegar hasta una región considerada, por entonces, inhóspita.

“N.P. 1819” fue grabado por Nathaniel Pearce, un aventurero británico que estuvo enrolado en la marina, desertó, se convirtió temporalmente al Islam y actuó como guía de viajeros por Egipto a comienzos del siglo XIX. Acompañó al cónsul Henry Salt durante su viaje a Etiopía en 1805. Allí se casó con una joven local y permaneció hasta 1816. En 1819, año en que el *graffito* fue grabado en Debod, había regresado a El Cairo y empezado a actuar como agente de Salt, claramente en la búsqueda de antigüedades. El año siguiente moría, preparando en Alejandría un envío de piezas egipcias a Inglaterra. Sus memorias fueron publicadas póstumamente, en 1831 y son una fuente muy rica de información sobre Etiopía pero Egipto apenas es mencionado.

Imagen 8: dioses egipcios de la pared trasera

Dibujo en realidad aumentada 8: hilera de cúpulas y pies de los dioses

La pared trasera de los templos egipcios fue adquiriendo gradualmente una importancia excepcional. Se trata de la misma pared que en su lado interior cierra la capilla donde se guardaban las estatuas de los dioses o, si se expresara en términos egipcios, “donde los dioses residían”. En consecuencia, es el punto más cercano a la divinidad al que podía acercarse un egipcio que no tuviera acceso al edificio pues, como ya se ha mencionado, sólo quienes estaban en estado de pureza ritual podían acceder al interior. La parte posterior de los templos se convirtió, así, en un espacio para el contacto de los fieles con la divinidad. En templos tardíos solía cubrirse con la representación de un conjunto de dioses, relacionados entre sí en términos míticos. Frente a ellos se esculpía una efigie del monarca haciéndoles una ofrenda. Con frecuencia las autoridades acondicionaban de modo especial el entorno de esta pared trasera para favorecer las prácticas de religiosidad individual.

En el caso de Debod, es evidente que se había planificado la representación de una hilera de dioses frente a un rey oferente. No podemos saber a qué momento de la construcción del edificio corresponde, pero posiblemente a una de las últimas fases de la decoración, por lo que podría ser del s. I. Pero la escena quedó incompleta, como tantas otras decoraciones del exterior del edificio, y sólo se esculpieron las dos primeras deidades: Amón de Debod, con cabeza de carnero, y Mahesa, con cabeza de león.

A sus pies se extienden unas hileras de cúpulas semiesféricas que parecen ser el resultado de extracciones puntuales de la arenisca. Posiblemente, como en el caso de las cúpulas identificadas en las portadas de los pilonos, se realizarían para prácticas de religiosidad popular.

Lo que para nosotros resulta más sorprendente es que no solo alcanzan los pies de los dos dioses esculpidos, sino que se extienden siguiendo la hilada de sillares hasta el centro de la pared. Su longitud, a una buena distancia desde los dioses que hoy pueden verse, permite plantear la sugerencia de que la procesión de dioses que iba a decorar este muro estuviera ya presente de alguna manera. Sabemos que el proceso de esculpir un relieve pasaba por varias fases de las que la inicial era el dibujo de la escena. Primero se realizaba el boceto con tinta negra y después, con tinta roja, la versión definitiva. Cabe la posibilidad de que al menos la mitad del muro presentara el dibujo de la escena cuando se interrumpió el trabajo y que los fieles realizaran las cúpulas tanto sobre los dioses ya esculpidos como sobre los que solo estaba en la fase inicial de dibujo.

Textos en Inglés

Image 1: Façade of the sanctuary of Adikhalamani

Drawing in augmented reality 1: map of Lower Nubia with the mentioned locations:

Thebes, first cataract, island of Philae, Debod, wadi Allaqui, Dakka, Maharraqa

Drawing in augmented reality 2: 3D reconstruction of the sanctuary of Adikhalamani

At the end of the first millennium *BCE* (before common era), the North of Nubia was known as the *Dodecaschoinos*. It was governed by the Egyptian Ptolemaic dynasty, a family descending from one of Alexander the Great's generals, who resided in the North of the country, in Alexandria. From a religious point of view, the *Dodecaschoinos* was considered to be property of the goddess Isis from Philae. This implied that the personnel attached to this temple was responsible for the economic production and administration of its fields as well as for the worship in the different sanctuaries of the goddess in her region.

The Nubian Kingdom of Meroe could have expanded, then, until the South of the *Dodecaschoinos*, if it was not in direct contact with it. Some of the Meroitic monarchs collaborated, at the end of 4th c. *BCE* in building activities in some of these temples, such as the one of Isis in Philae and of Thoth in Dakka. It has been assumed that these signs of devotion would be linked to the travel of the statue of the goddess to the different sanctuaries of her property, a festive procession carried out annually.

In 206 *BCE*, a revolt in the Thebaid provoked the split of Southern Egypt against Alexandria, the Ptolemaic capital in the North. Meroitic kings took advantage of the internal Egyptian conflict to extend their control over the whole *Dodecaschoinos*.

It is in this moment, under the reign of King Adikhalamani (who governed, approximately, between 200 and 185 *BCE*), when in Debod the sanctuary of the same name was built. This is the heart of the actual temple.

After the reconquest of the region by Egypt, Alexandrian kings started a building program that would allow them to recover its inhabitants' fidelity. In the case of Debod, this policy entailed the extension of the building as it is known today. In that Ptolemaic intervention, religious devotion joined political interest. The area had become an exceptional route for the traffic of goods. Alexandria was provided with African exotic products through Lower Nubia; this region also allowed access to the gold mines of Wadi Allaqui, reopened by the dynasty.

With the Roman conquest of Egypt, the frontier was consolidated in Maharraqa for many centuries, several tens of kilometers to the South of Debod. Therefore, the temple remained inside Roman Empire's territory, in an area of prosperous economic activity and of preservation of the traditional Egyptian religious rites.

Image 2: *Graffito* troposkynema

Drawing in augmented reality 3: the same *graffito*

The temple of Debod was built to accommodate two Gods –Isis of Philae and Amon of Debod– which is an unusual circumstance.

Daily worship ritual was put into practice in the inner chambers. It is possible that these were also used as a station in the annual also received one of the journeys of the annual procession of Isis through the *Dodecaschoinos*. Other ceremonial activities would take place at the public outdoor area.

Some of the *graffiti* were produced at the moment in which Debod still housed the cult to its original gods. In the identification of these *graffiti* a basic premise must be taken into account: the interior of Egyptian temples was only opened to priests. These had to go through a process of individual purification, respecting certain rules in their clothing and fulfilling some strict rules of personal hygiene and social behaviour. Consequently, the *graffiti* that were engraved when the worship was active cannot be placed in the interior. They can only be found in those areas accessible to wider population.

The *graffito* in which it can be read in Greek τοπροσκυνημαμε [toproskynema me], located in the eastern side of the façade of the second pylon, fulfils this rule. The first word –*toproskinema*– means ‘veneration’. It was always followed by the name of the worshipper. In this case, he could only engrave the first syllable of his name, *me*. The first term indicates, by its same meaning, that it was made when the deity still resided inside the sanctuary. The use of the Greek language should come as no surprise. After the Macedonian conquest, the Egyptian society increasingly became bilingual, with growing presence of the Greek language. The Roman conquest did not mean a significant change in this tendency.

Image 3: dromedaries of the north wall
Drawing in augmented reality 4: one of the dromedaries

By the middle of the third century of our era, the gradual establishment of nomadic and semi-nomadic tribes in Lower Nubia and in the surrounding deserts took place. They had acquired mobility capacity after the adoption of dromedaries as riding and transport animals. This new weapon had also favoured their social restructuring. The Blemmyes, from the eastern desert, settled in Lower Nubia, from where they dared to attack the southern frontier of the Roman Empire. The Nobatae, from the western desert, settled between the Blemmyes and the first cataract, acting as protectors of Rome's interests.

However, in spite of Roman disbursements to keep them appeased, nomads made common cause against the Empire several times. Philae's attack in 450 has a special historical importance, since after Constantinople's response a treaty was signed. In accordance with its stipulations, nomads –still faithful to their ancient Gods– received the permission to participate in Isis' celebrations at her temple, which were still active, and to carry the Goddess' statue to her country, at particular moments of the year, preserving the tradition of the Isiac procession through the *Dodecaschoinos*. This treaty ensured that Philae remained as the centre for pagan cult in spite of the decrees imposing Christianity and the prohibition of the traditional Gods which put an end to the ancient Egyptian religion in the rest of Egypt. It may be supposed that the survival of pagan worship in Philae could also be extrapolated to Debod, although the written sources remain silent about this aspect. Only Justinian's decrees from 535-537 succeeded in closing close the sanctuaries of the island of Isis. By then, the process of Christianization in Lower Nubia had already begun diminishing pagan forces to a hypothetical response.

Several scenes with dromedaries and, exceptionally, other desert animals, may belong to this moment. It is not surprising that nomad populations represented the animal that had become the focus of their new way of life and of their power. A second circumstance, complementing the previous one, advocates for this attribution: all scenes with dromedaries are in the outer walls of the temple. From both premises we could deduce that they are the work of a population that did not go inside the sanctuary. Access might be barred, either because it was still prohibited for the devotees since traditional worship still remained, or because of the respect to the old practices that had been developed inside it in the very recent past, of which nomads considered themselves their heirs.

The most significant animal scene is the one which occupies the north outer wall, twenty dromedaries and two humans who lead them by the reins. From a distance of two ashlar's a square which looks like the destiny of animals and humans is carved. The scene must represent a caravan heading to its resting place, although it could also be related to the nomads' pilgrimage to Philae in holidays.

Image 4: Coptic Cross Pattée

Drawing in augmented reality 5: the same *graffito*

Picture in augmented reality 1: Dipinti of pattée crosses

Picture in augmented reality 2: face of a deity with *damnatio*'s testimony

An inscription by the Nobatae king Silko, sculpted around 545 on the façade of the Temple of Kalabsha, very close to Debod, asserts that God gave him victory over the Blemmyes. The text also points out the triumph of Christianity in Lower Nubia and the consolidation of the Kingdom of Nobatia. Surely, the process of Christianization was under development for several generations before that. In it, the influences of Coptic monks, merchants and travellers with diverse interests from the already Christianized Egypt were present. The consequence was that pagan cults ceased to be celebrated in Debod

The unified Christian Nubia would become one of the powers at the time. The “Great King” of Nobatia had more than a dozen “small kings” under his authority. The region started a process of strong urbanisation. Numerous churches also attest to the development of a typical architectonic and decorative style of their own.

It is complicated to determine if Debod was turned into a Christian church after the abandon of pagan cult. There are not medieval texts that state it or allow us to presume it. Neither do first European travellers mention the presence inside the building of architectonic or cultural elements that would have facilitated the development of the new credo. Furthermore, the rooms look small to congregate devotees in a common prayer.

Nevertheless, there is a significant presence of Crosses Pattée in this temple. There are engraved ones and there were also painted ones (the latter were lost due to the immersion of the temple in the waters of the first Aswan Dam during decades). Crosses Pattée are characteristic of Egyptian Christianity. They are crosses with arms of same length (as the Greek ones) that end in two extensions opened in angle. Nearly all Debod's crosses are in the main axis of the building, a circumstance which seems to indicate a kind of route to that area. The actual deterioration of the faces of most gods in the central sanctuary can be another proof of the value attached to that space; there are numerous Egyptian temples where a similar destruction can be dated in Christian time for the adaptation of old buildings to new religious purposes. The presence of the crosses and damnation of the ancient Gods' faces raise questions about whether the sanctuary of Adikhalamani or the central sanctuary could have been used as a place for worship during Christian time.

Image 5:Gouges

Drawing in augmented reality 3: a different gouge from that of the leaflet

Some groups of *graffiti* have an uncertain chronology and they are difficult to fit in one of the specific periods of the Nubian history. Gouges present the largest number. Their importance derives not only from their quantity and visibility but also, from the information that they provide from a chronological point of view.

These are groups of long spindle-shaped depressions made by rubbing a sharp instrument on the surface of the stone with the aim of obtaining certain quantity of sand that visitors took because it was considered sacred. They are only located on the gateways of the pylons. Their high number, about fifteen groups, and their depth, show the frequency of visits and the veneration the building had been subjected to. Both Christian and Islamic *graffiti* have been superimposed on one of these groups, and this palimpsest is important for chronological deductions, since it shows that this gouge was previous to those beliefs.

Thus, although some authors testify that the practice of taking sand from sacred places has reached our days, at least part of these “scratches” on the temple of Debod should be dated before the incursion of Christianity or during the first moments of it.

Image 6: Islamic *graffito*
Drawing in augmented reality 6: the same *graffito*

In 1315, Kerenbes, the Nubian King, was captured by Egyptian mamluks and was replaced by Abdallah ibn Sanbu. Thus, an end was put to almost eight centuries of Christian Nubia. Islamization, which had already started before that dynastic change, was a very fast process, although a Christian minority remained tolerated. The international economic situation had relegated Nubia to a peripheral situation. The new governors did not pretend to have any presence in African politics.

At the beginning of the 16th century, Lower Nubia was conquered by the Ottoman Empire after the occupation of Egypt by sultan Selim I. A Bosnian garrison in Qasr Ibrim guaranteed Turk control. The region was abandoned for centuries; its only function consisted in being a crossing to the southernmost area of the Nile valley and to Central Africa. Exhaustion was almost total. Archaeologists who worked on the Campaign of Nubia launched by the Unesco, in the 1960s, have barely found remains of human occupation during these centuries, except for some strategic locations around the Valley that had been reinforced as checkpoints.

The gradual recovery started after the passage of Muhammad Ali's Egyptian troops that returned the interest of the Ottoman Empire for the current Sudan between 1820 and 1825. These campaigns opened the route to traders, and Lower Nubia was turned into a transit route for ivory and slaves to the Mediterranean.

Twenty inscriptions in Arabic bear witness to this long period under Islamic control. These are mainly invocations of Allah and Muhammad, and proclamations of faith. The first gateway accumulates a large proportion of these epigraphs. As it is the case with Christian crosses, these inscriptions could have been intended to execrate the Pagan sacredness of the place and to create protection for the devotees of the new dominant religion. The image presents a superposition of at least five declarations of faith. On them, it is possible to read "la illah illa Allah" in Arabic, which means "There is no deity but God". Each of them seems to be written by a different hand, with signs of different size, one over the other to the extent that the latest ones cover the oldest ones.

Image 7: Name and date of an European traveller
Drawing in augmented reality 7: the same *graffito*

Another large group of *graffiti* is formed by the signatures of western travellers. In the Middle Ages and Modern Age, some European missionaries travelled along the Nile to the south. The earliest ones looked for the almost mythic kingdom of Prester John. They and specially the most recent ones mainly sought to gain converts to Catholicism. There are no testimonies in Debod about their stay.

Since the beginning of the 19th century, travellers who cross the region for the pleasure of “discovery” are attested: Ludwig Burckhardt travels around Lower Nubia in 1813. Even before he published his travel report, other western adventurers emulate him, either to know better the region, to obtain information about antiques or to take them. In Debod, about thirty signatures of western travellers are preserved, and most of them are engraved in the walls of *naoí's* chamber. They usually include the year when they arrived at the temple: most of them during the first half of the 19th century. Thus, these *graffiti* were engraved as an act of personal awareness, a way of recording the value and determination that had allowed their authors to arrive at a region considered inhospitable by that time.

“N.P. 1819” was engraved by Nathaniel Pearce, a British adventurer who was enrolled in the navy, deserted, temporarily converted to Islam and acted as a guide for travellers through Egypt at the beginning of the 19th century. He accompanied the Consul Henry Salt during his voyage to Ethiopia in 1805. There he married a young local woman and stayed until 1806. In 1819, the year when the *graffito* was engraved in Debod, he had returned to Cairo and had begun to act as Salt’s agent, clearly searching for antiques. He died the following year, preparing in Alexandria a shipment of Egyptian pieces to England. His memories were posthumously published –in 1831– and they are a very rich source of information about Ethiopia, but Egypt is barely mentioned.

Image 8: Egyptian gods on the rear wall

The rear wall of Egyptian temples was gradually acquiring an exceptional importance. It is the same wall that closes in its interior side the sanctuary where the statues of gods were kept or, in Egyptian terms, “where gods resided”. Consequently, it is the closest point to divinity an Egyptian who did not have access to the building could reach, because only the ones that were in state of ritual purity could enter inside. Thus, the rear part of the temples became a space for the contact of devotees with the divinity. In later temples, it was usually covered with a representation of a group of gods, related to each other in mythical terms. In front of them, an effigy of the monarch making an offering was carved. Frequently, authorities prepared the surroundings of this rear wall in a special way to favour personal religious expressions.

In the case of Debod, it is clear that the representation of a line of gods in front of an offering king had been planned. It cannot be known to which moment of the construction of the building it corresponds, but probably to one of the last phases of the decoration, therefore it could be from the first century. But the scene was incomplete, as many other of the exterior of the building, and only the first two deities were carved: Amun from Debod, ram-headed, and Mahes, lion-headed.

Rows of semi-spherical domes which seem to be the result of punctual extractions of sandstone were extended at their feet. Perhaps, as in the case of the identified domes in the pylon’s covers, they would be made as personal religious expressions.

What is more surprising is that not only they reach the sculpted feet of the two Gods, but that they extended following the row of ashlar to the middle of the wall. Its length, at a very good distance from the Gods that can be seen today, allows considering the suggestion that the God’s procession which was going to decorate this wall was already present somehow. We know that the process of carving a relief underwent several phases of which the first one was the drawing of the scene. Firstly, the sketch was made with black ink, and then the final version with red ink. It is possible that at least half of the wall presented the drawing of the scene when the work was interrupted, and that the devotees made the domes both over the Gods already carved and over those in the initial phase of the drawing.

Imágenes

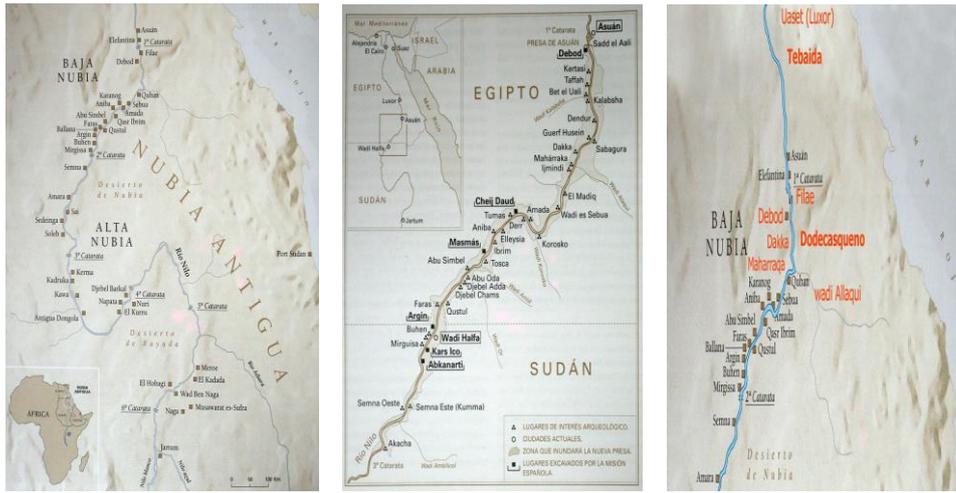


Figura 88 Mapas

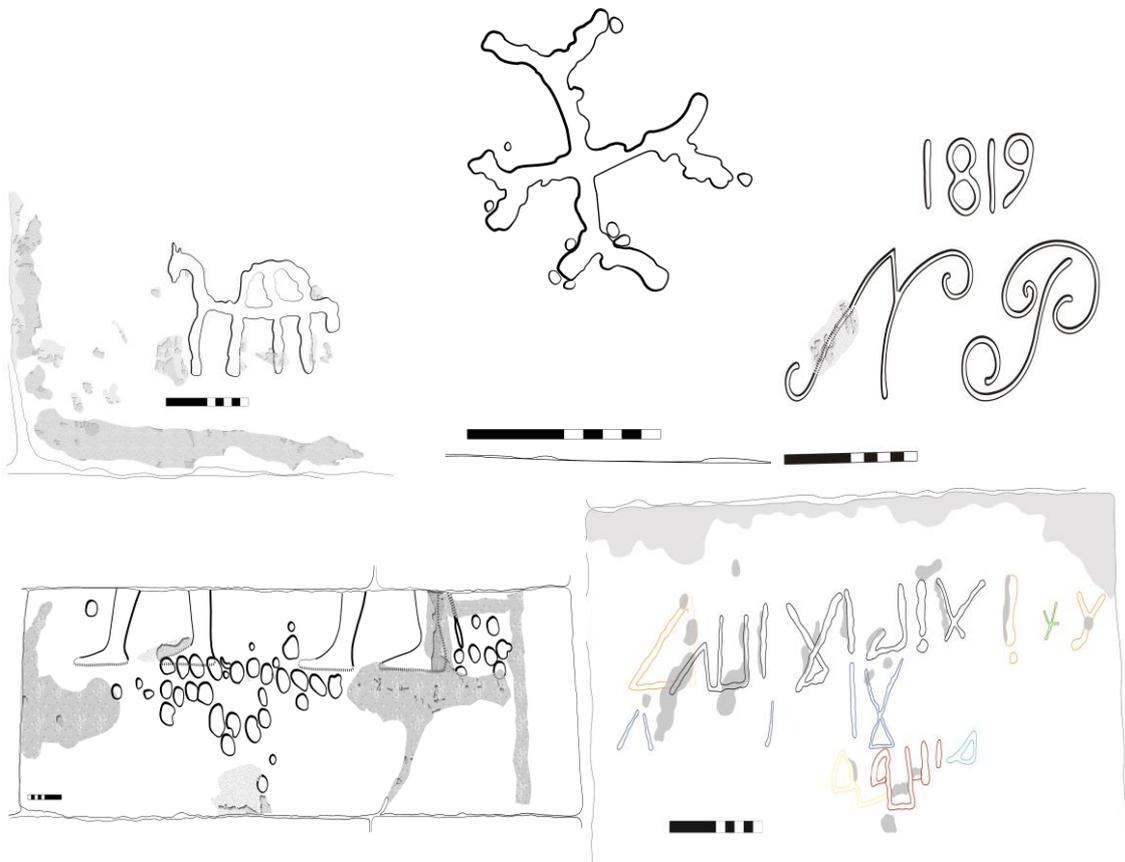


Figura 89 Dibujos de graffitis.



Figura 90 Imágenes de graffitis 1



Figura 91 Imágenes de graffitis 2



Figura 92 Imágenes de portada.

Otros contenidos

Lista de contenidos que por su naturaleza no pueden ser incluidos en un archivo de texto de forma visual.

Videos

Fig_1_VideoDebod.mov

Audios

audio_1_bueno.mp3
audio_2_bueno.mp3
audio_3_bueno.mp3
audio_4_bueno.mp3
audio_5_bueno.mp3
audio_6_bueno.mp3
audio_7_bueno.mp3
audio_8_bueno.mp3

audio 1_GB.mp3
audio 2_GB.mp3
audio 3_GB.mp3
audio 4_GB.mp3
audio 5_GB.mp3
audio 6_GB.mp3
audio 7_GB.mp3
audio 8_GB.mp3

Modelo 3D

temploDebod 1.blend

Anexo III. Prototipo.

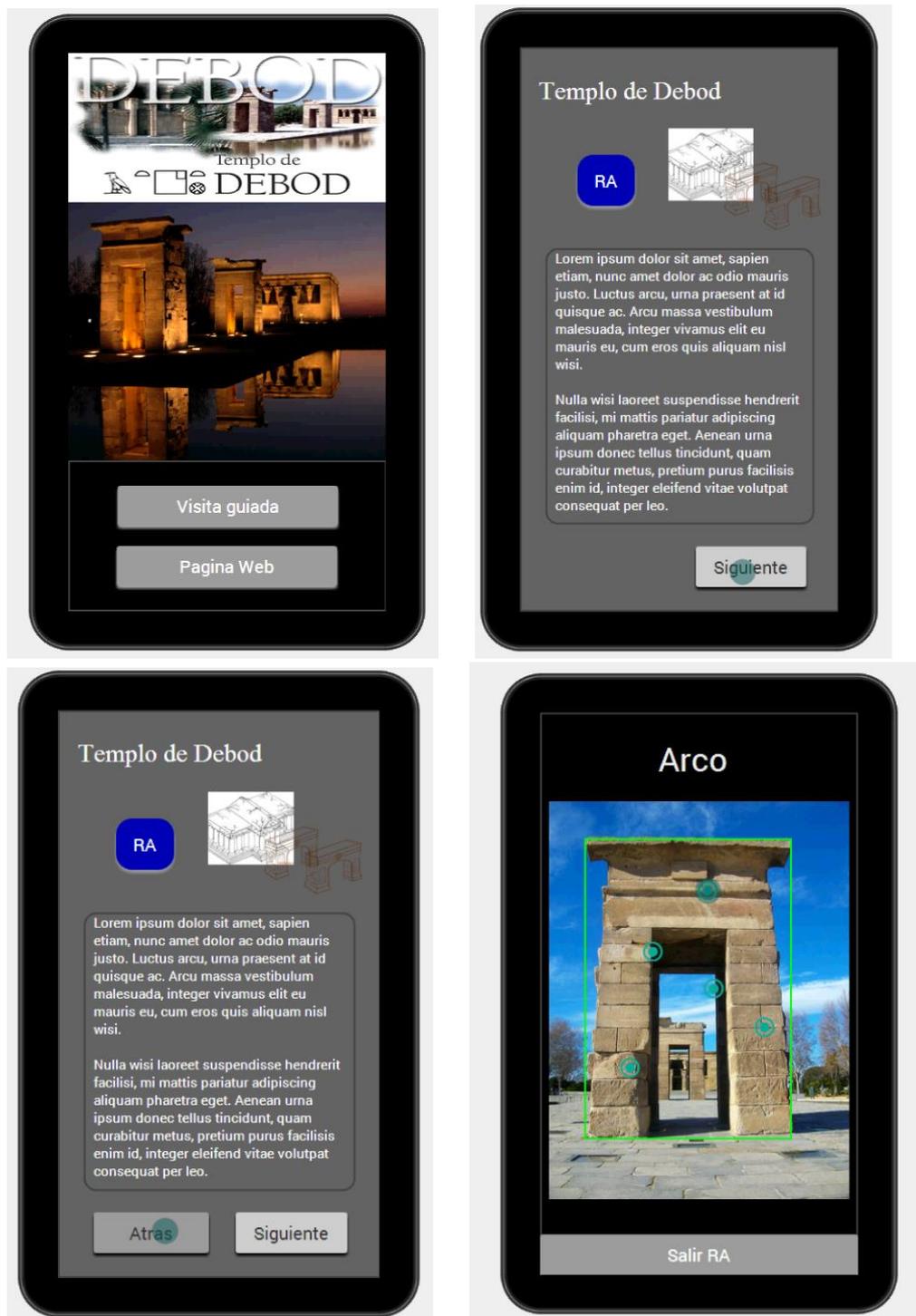


Figura 93 Pantallas del prototipo A



Figura 94 Pantallas del prototipo B

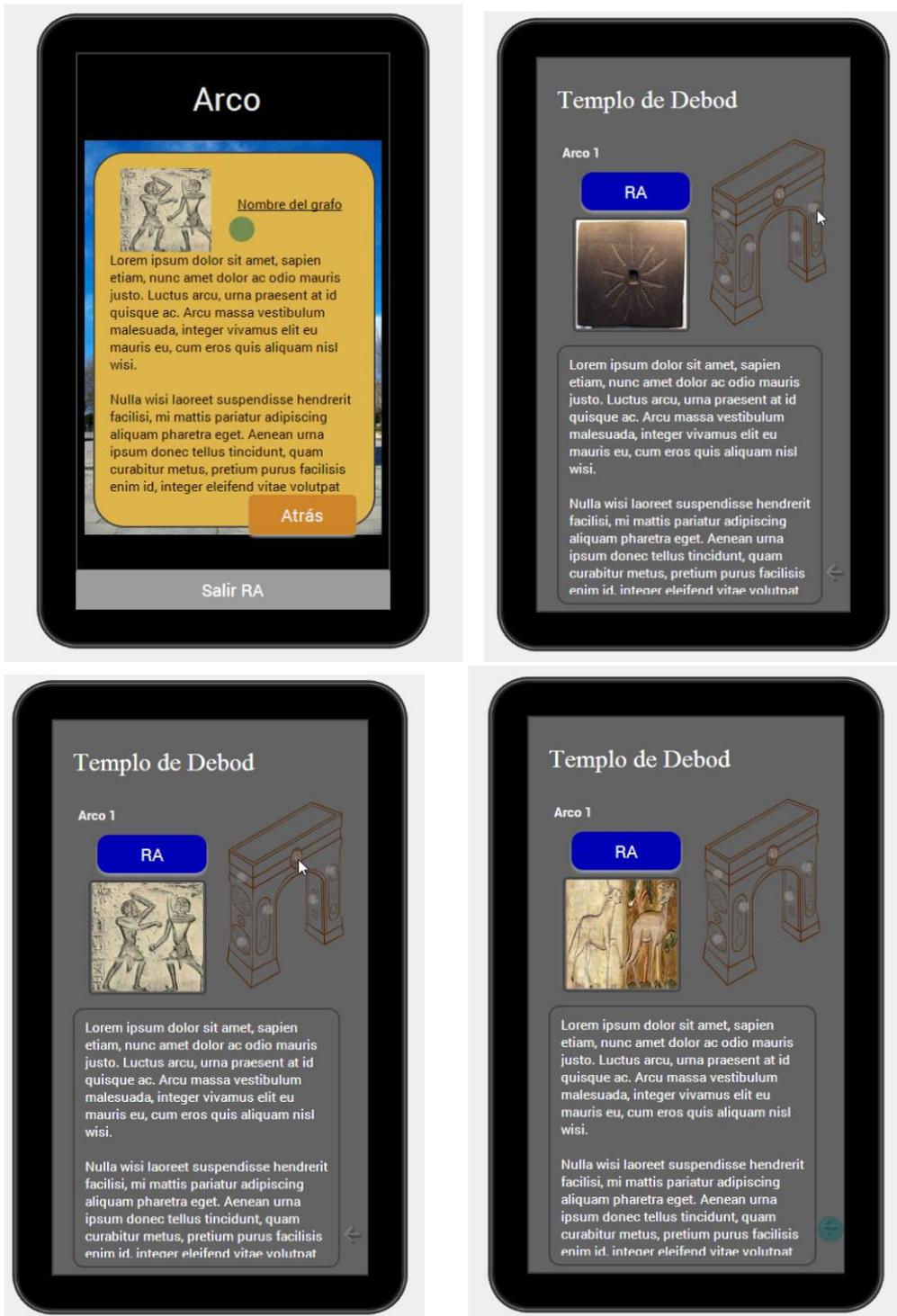


Figura 95 Pantallas del prototipo C

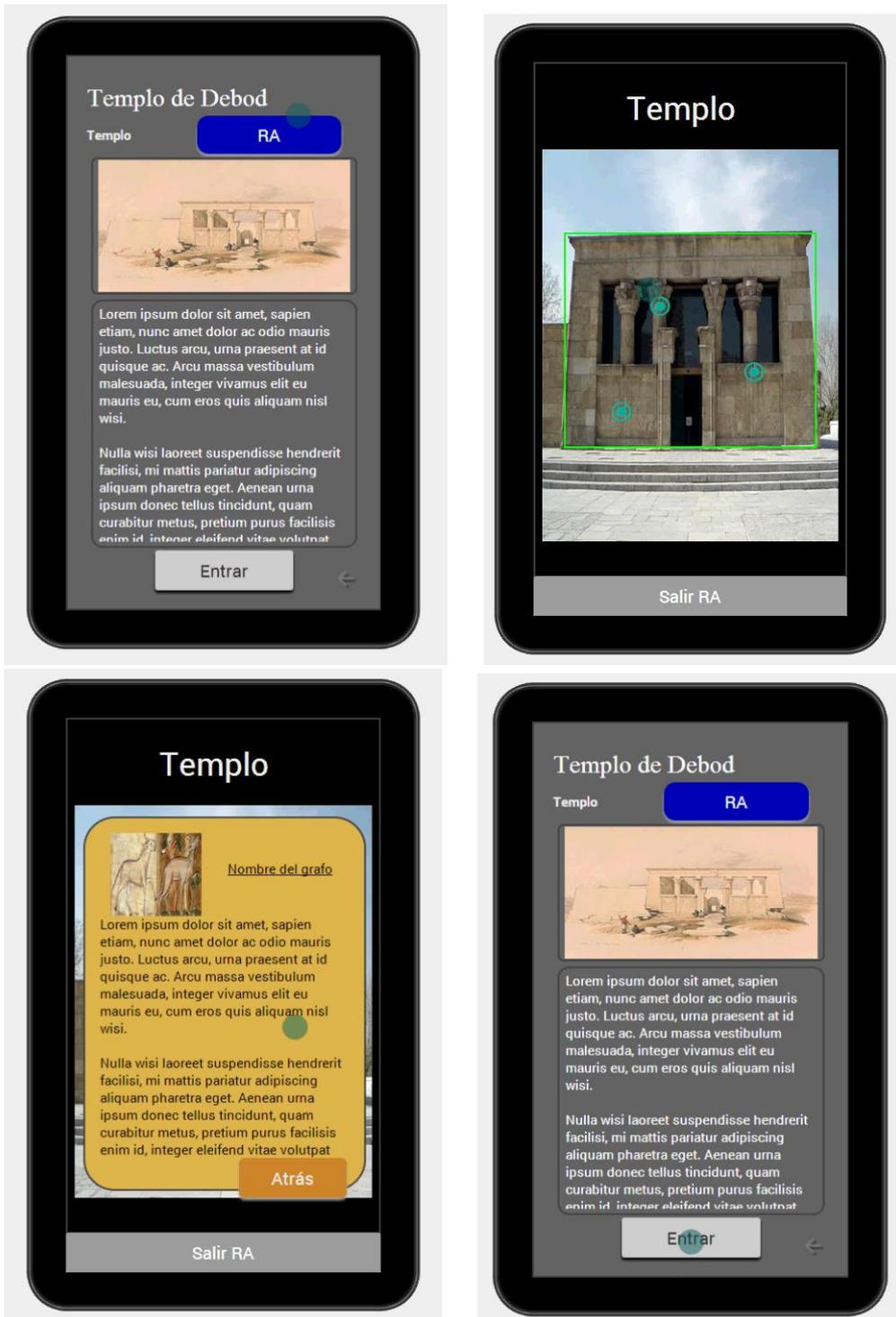


Figura 96 Pantallas del prototipo D



Figura 97 Pantallas del prototipo E

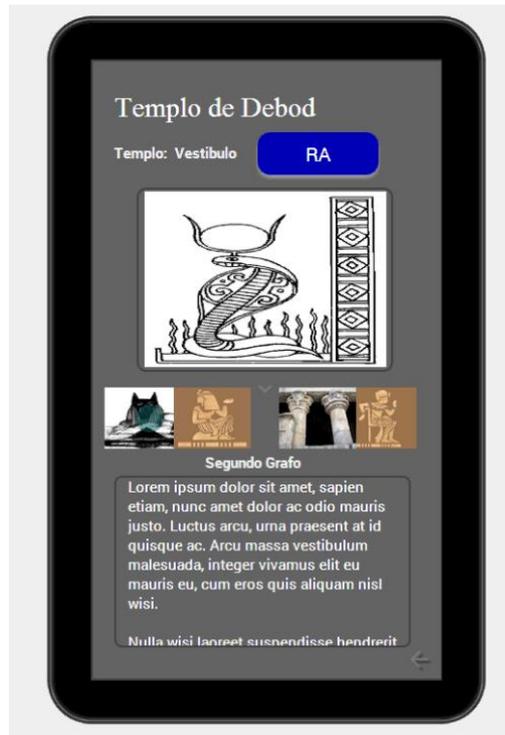


Figura 98 Pantalla del prototipo F

Anexo IV. Scripts.

Muestra de algunos de los scripts más importantes realizados para el correcto funcionamiento de la aplicación.

Usados en el folleto

Estructura ScripTableObject

```
[CreateAssetMenu(fileName = "NewDataStructure", menuName = "DataStructure")]
// This is the representation of our Structure, inside of Asset/Create
public class DataStructure : ScriptableObject
{
    public string itemName = "New Item";
    // What the item will be called
    public Sprite imgItem;
    // This reference to de UI Image
    //public TextAsset textItem;
    // This reference to de UI Text
    //public AudioClip audioItem;
    // This is where the audio will be allocated
}
}
```

Carga de texto, imágenes y audio

```
public class Lectura : MonoBehaviour
{
    //[SerializeField] private List<DataStructure> dataList;
    [SerializeField] Text txt;
    [SerializeField] GameObject img;

    private int MaxElement = 8; // This
    value must be changed if new screen is added.

    private TextAsset TextObject;
    private AudioClip aud;
    private SpriteRenderer myImage;

    private TextAsset dictionaryTextFile;
    private string theWholeFileAsOneLongString;
    private int actualItem = 0;
    private string language;

    [SerializeField] private Button next;
    [SerializeField] private Button prev;

    [SerializeField] private AudioSource audioSource;

    void Start()
    {
        ChangeData(0);
    }

    public void ChangeData(int id)
```

```

{
    string TextPath;
    string AudioPath;
    string ImagePath;

    language = Application.systemLanguage.ToString();
    actualItem = id;
    switch (id)
    {
        case 0:
            TextPath = "Texts/" + language + "/Texto1";
            AudioPath = "Audio/" + language + "/audio_1";
            ImagePath = "Images/Imagen1";
            break;
        case 1:
            TextPath = "Texts/" + language + "/Texto2";
            AudioPath = "Audio/" + language + "/audio_2";
            ImagePath = "Images/Imagen2";
            break;
        case 2:
            TextPath = "Texts/" + language + "/Texto3";
            AudioPath = "Audio/" + language + "/audio_3";
            ImagePath = "Images/Imagen3";
            break;
        case 3:
            TextPath = "Texts/" + language + "/Texto4";
            AudioPath = "Audio/" + language + "/audio_4";
            ImagePath = "Images/Imagen4";
            break;
        case 4:
            TextPath = "Texts/" + language + "/Texto5";
            AudioPath = "Audio/" + language + "/audio_5";
            ImagePath = "Images/Imagen5";
            break;
        case 5:
            TextPath = "Texts/" + language + "/Texto6";
            AudioPath = "Audio/" + language + "/audio_6";
            ImagePath = "Images/Imagen6";
            break;
        case 6:
            TextPath = "Texts/" + language + "/Texto7";
            AudioPath = "Audio/" + language + "/audio_7";
            ImagePath = "Images/Imagen7";
            break;
        case 7:
            TextPath = "Texts/" + language + "/Texto8";
            AudioPath = "Audio/" + language + "/audio_8";
            ImagePath = "Images/Imagen8";
            break;
        default:
            TextPath = "Texts/" + language + "/Fachada1";
            AudioPath = "Audio/" + language + "/audio_1";
            ImagePath = "Images/Imagen2";
            break;
    }
}

```

```

TextObject = (TextAsset)Resources.Load(TextPath);
txt.text = TextObject.text;
aud = (AudioClip)Resources.Load(AudioPath);
audioSource.clip = aud;

```

```

        //myImage = myImage.AddComponent<SpriteRenderer>();
        // = (Sprite)Resources.Load("Images/Imagen2");
        img.GetComponent<Image>().sprite = Resources.Load<Sprite>(ImagePath);
//img.GetComponent<Image>().sprite = dataList[id].imgItem;

        /*txt.text = dataList[id].textItem.text;
        img.GetComponent<Image>().sprite = dataList[id].imgItem;
        audioSource.clip = dataList[id].audioItem;
        actualItem = id;*/

        if (actualItem >= MaxElement - 1)
        {
            next.enabled = false;
            prev.enabled = true;

        }
        else if (actualItem <= 0)
        {
            next.enabled = true;
            prev.enabled = false;
        }
        else
        {
            next.enabled = true;
            prev.enabled = true;
        }
    }

    public void Siguiete()
    {
        ChangeData(actualItem + 1);
    }

    public void Anterior()
    {
        ChangeData(actualItem - 1);
    }

    public void Home()
    {
        SceneManager.LoadScene(1);
    }

}

```

Usados en la realidad aumentada

Objeto global, usado para mantener valores globales a todas las escenas.

```
public class GlobalObjectHandler : MonoBehaviour {

    public static GlobalObjectHandler Instance;
    public object currentObject;

    // List of target names, we use that for manage the content sowed.
    public List<String> TargetsName;
    public string thisTargetName;

    // This variable will be used to display the hel view in te AR mode
    public bool Tracked;
    public GameObject introCanvas;

    public bool inicial, folleto, app;

    void Awake()
    {
        if (Instance == null)
        {
            DontDestroyOnLoad(gameObject);
            Instance = this;
        }
        else if (Instance != this)
        {
            Destroy(gameObject);
        }
    }

    // Use this for initialization
    void Start()
    {
        GlobalObjectHandler.Instance.Tracked = true;
        //ChangeCanvas(true);
    }

    // Update is called once per frame
    void Update()
    {
        ChangeCanvas(Tracked);
        StateManager sm = TrackerManager.Instance.GetStateManager();
        IEnumerable<TrackableBehaviour> tbs = sm.GetActiveTrackableBehaviours();

        //ChangeCanvas(Tracked);

        foreach (TrackableBehaviour tb in tbs)
        {
            //thisTargetName = tb.TrackableName;
            Debug.Log("TargetName -> " + thisTargetName);
        }
        SavePlayer();
    }

    public void SavePlayer()
    {
        //GlobalObjectHandler.Instance.thisTargetName = thisTargetName;
    }
}
```

```

        GlobalObjectHandler.Instance.TargetsName = TargetsName;
    }

    public void ChangeCanvas(bool status)    // When any target is detected, the
help canvas dissappear
    {
        GlobalObjectHandler.Instance.Tracked = status;
        introCanvas.SetActive(Tracked);
        Debug.Log("Intro canvas -----> " + introCanvas.activeSelf);
    }
}

```

Controlador del menu aumentado

```

public class MenuController : MonoBehaviour {

    public GameObject menuButton;
    public int name;

    GameObject thisBtn;

    SphereCollider InvisibleButton;
    Vector3 ButtonPosition;
    float ButtonRadius;

    bool mode;

    bool hideState;

    // Use this for initialization
    void Start () {

        thisBtn = this.gameObject;

        InvisibleButton = this.GetComponent<SphereCollider>();
        ButtonPosition = InvisibleButton.center;
        ButtonRadius = InvisibleButton.radius;

        mode = true;
        hideState = true;
    }

    void Update () {

        if(mode==true && hideState==true)
        {
            //Keeps the button size/position as initially created
            InvisibleButton.center = ButtonPosition;
            InvisibleButton.radius = ButtonRadius;

            menuButton.SetActive(true);

            thisBtn.SetActive(true);
        }
    }
}

```

```

    }

    if(mode==false && hideState==true)
    {

        menuButton.SetActive(false);

        Application.LoadLevel(name);

        thisBtn.SetActive(true);

    }

    if(mode==true && hideState==false)
    {

        menuButton.SetActive(false);

        thisBtn.SetActive(false);

    }

}
}

```

Script que cambia la imagen según el targuet detectado.

```

public class ChangeImage : MonoBehaviour {

    [SerializeField] private List<ImageStructure> dataList;
    [SerializeField] GameObject img;

    private int actualItem = 0;
    public string TargetName;
    private int Indice = 0;
    bool lookFor = false;

    [SerializeField] private Button next;
    [SerializeField] private Button prev;

    // Use this for initialization
    void Start () {
        TargetName = GlobalObjectHandler.Instance.thisTargetName;
        while (!lookFor)
        {
            if (TargetName != GlobalObjectHandler.Instance.TargetsName[Indice])
// Found the actual target name
            {
                Indice++;
            }
            else
            {
                lookFor = true;
            }
        }
        ChangeData(Indice, 0);
    }

    public void ChangeData(int id, int id2)
    {
        img.GetComponent<Image>().sprite = dataList[id].imgItem[id2];
        actualItem = id2;
    }
}

```

```

        if (actualItem >= dataList[id].imgItem.Count - 1)
        {
            next.enabled = false;
            prev.enabled = true;
        }
        else if (actualItem <= 0)
        {
            next.enabled = true;
            prev.enabled = false;
        }
        else
        {
            next.enabled = true;
            prev.enabled = true;
        }
    }

    public void Siguiente()
    {
        ChangeData(Indice, actualItem + 1);
    }

    public void Anterior()
    {
        ChangeData(Indice, actualItem - 1);
    }

    public void Home()
    {
        SceneManager.LoadScene(3);
    }
}
}

```

Script que cambia el texto y el audio según el tarquet detectado.

```

public class ChangeText : MonoBehaviour {

    [SerializeField] private List<TextStructure> dataList;
    [SerializeField] Text txt;
    [SerializeField] private AudioSource audioSource;
    public int ID;
    private TextAsset TextObject;
    private AudioClip aud;
    public string TargetName;
    int search = 0;
    bool lookFor = false;
    private string language;

    // Use this for initialization
    void Start () {
        TargetName = GlobalObjectHandler.Instance.thisTargetName;

        while (!lookFor)
        {
            if (TargetName != GlobalObjectHandler.Instance.TargetsName[search])
            {
                search++;
            }
            else

```

```

        {
            lookFor = true;
        }
    }
    ChangeData();
}

public void ChangeData()
{
    string TextPath;
    string AudioPath;

    language = Application.systemLanguage.ToString();

    switch (search)
    {
        case 0:
            TextPath = "Texts/" + language + "/Texto1";
            AudioPath = "Audio/" + language + "/audio_1";
            break;
        case 1:
            TextPath = "Texts/" + language + "/Texto2";
            AudioPath = "Audio/" + language + "/audio_2";
            break;
        case 2:
            TextPath = "Texts/" + language + "/Texto3";
            AudioPath = "Audio/" + language + "/audio_3";
            break;
        case 3:
            TextPath = "Texts/" + language + "/Texto4";
            AudioPath = "Audio/" + language + "/audio_4";
            break;
        case 4:
            TextPath = "Texts/" + language + "/Texto5";
            AudioPath = "Audio/" + language + "/audio_5";
            break;
        case 5:
            TextPath = "Texts/" + language + "/Texto6";
            AudioPath = "Audio/" + language + "/audio_6";
            break;
        case 6:
            TextPath = "Texts/" + language + "/Texto7";
            AudioPath = "Audio/" + language + "/audio_7";
            break;
        case 7:
            TextPath = "Texts/" + language + "/Texto8";
            AudioPath = "Audio/" + language + "/audio_8";
            break;
        default:
            TextPath = "Texts/" + language + "/Fachada1";
            AudioPath = "Audio/" + language + "/audio_1";
            break;
    }
    TextObject = (TextAsset)Resources.Load(TextPath);
    txt.text = TextObject.text;
    aud = (AudioClip)Resources.Load(AudioPath);
    audioSource.clip = aud;
}

public void Home()
{
    SceneManager.LoadScene(3);
}

```

```
}  
}
```

Script usado para manejar la rotación del modelo 3D, así como la activación de las posiciones.

```
public class Rotation : MonoBehaviour  
{  
  
    public float rotationRate = 10.0f;  
    private bool isRotating, rotateRight, rotateLeft;  
    // [SerializeField] private List<LightStructure> myLights;  
    // public List<GameObject> myLights;  
    public GameObject templo;  
    private GameObject myLights;  
    public string TargetName;  
    int search = 0;  
    bool lookFor = false;  
  
    // Use this for initialization  
    void Start()  
    {  
        TargetName = GlobalObjectHandler.Instance.thisTargetName;  
  
        while (!lookFor)  
        {  
            if (TargetName != GlobalObjectHandler.Instance.TargetsName[search])  
            {  
                search++;  
            }  
            else  
            {  
                lookFor = true;  
            }  
        }  
        ActivateLights(search);  
        isRotating = false;  
        rotateRight = false;  
        rotateLeft = false;  
    }  
  
    // Update is called once per frame  
    void Update()  
    {  
        if (isRotating)  
        {  
            if (rotateRight)  
            {  
                transform.Rotate(-Vector3.up * rotationRate * Time.deltaTime);  
            }  
            else if (rotateLeft)  
            {  
                transform.Rotate(Vector3.up * rotationRate * Time.deltaTime);  
            }  
        }  
    }  
  
    public void rigt()  
    {
```

```

        rotateRight = true;
        rotateLeft = false;
        isRotating = true;
    }

    public void left()
    {
        rotateLeft = true;
        rotateRight = false;
        isRotating = true;

    }

    public void stop()
    {
        isRotating = false;
        rotateRight = false;
        rotateLeft = false;
    }

    public void ActivateLights(int search)
    {
        switch (search)
        {
            case 0:
                myLights = templo.transform.Find("First_Positions").gameObject;
                myLights.SetActive(true);
                break;
            case 1:
                myLights = templo.transform.Find("Second_Positions").gameObject;
                myLights.SetActive(true);
                break;
            case 2:
                myLights = templo.transform.Find("Third_Positions").gameObject;
                myLights.SetActive(true);
                break;
            case 3:
                myLights = templo.transform.Find("Fourth_Positions").gameObject;
                myLights.SetActive(true);
                break;
            case 4:
                myLights = templo.transform.Find("Fifth_Positions").gameObject;
                myLights.SetActive(true);
                break;
            case 5:
                myLights = templo.transform.Find("Sixth_Positions").gameObject;
                myLights.SetActive(true);
                break;
            case 6:
                myLights = templo.transform.Find("Seventh_Positions").gameObject;
                myLights.SetActive(true);
                break;
            case 7:
                myLights = templo.transform.Find("Eighth_Positions").gameObject;
                myLights.SetActive(true);
                break;
        }
        /*int i = 0;
        while (i < myLights.Count)
        {
            if (i != search)

```

```

        {
            for(int j=0; j < myLights[i].LightImet.Count -1; j++)
                myLights[i].LightImet[j].SetActive(true);
            i++;
        }
    }*/
}

public void DesactivateLights(int search)
{
    //myLights.SetActive(false);

    switch (search)
    {
        case 0:
            myLights = templo.transform.Find("First_Positions").gameObject;
            myLights.SetActive(false);
            break;
        case 1:
            myLights = templo.transform.Find("Second_Positions").gameObject;
            myLights.SetActive(false);
            break;
        case 2:
            myLights = templo.transform.Find("Third_Positions").gameObject;
            myLights.SetActive(false);
            break;
        case 3:
            myLights = templo.transform.Find("Fourth_Positions").gameObject;
            myLights.SetActive(false);
            break;
        case 4:
            myLights = templo.transform.Find("Fifth_Positions").gameObject;
            myLights.SetActive(false);
            break;
        case 5:
            myLights = templo.transform.Find("Sixth_Positions").gameObject;
            myLights.SetActive(false);
            break;
        case 6:
            myLights = templo.transform.Find("Seventh_Positions").gameObject;
            myLights.SetActive(false);
            break;
        case 7:
            myLights = templo.transform.Find("Eighth_Positions").gameObject;
            myLights.SetActive(false);
            break;
    }
    /*int i = 0;
    while (i < myLights.Count)
    {
        if (i != search)
        {
            for (int j = 0; j < myLights[i].LightImet.Count - 1; j++)
                myLights[i].LightImet[j].SetActive(false);
            i++;
        }
    }*/
}

public void Home()
{
    DesactivateLights(search);
}

```

```

        SceneManager.LoadScene(3);
    }
}

```

Parser para controlar el lenguaje.

```

public class ControllerLanguages : Singleton<ControllerLanguages> {

    [SerializeField] private bool debug = false;

    private TextAsset localizedText;
    private string fullString;
    private List<string> rowList;
    [SerializeField] private Dictionary<string, string> dictionary;

    void Start() {
        Initialize();
        if(debug) PrintDictionaryOnConsole();
    }

    void Initialize() {
        // Get the localization TextAsset for the device language (If null
        -> get English by default)
        localizedText =
        (Resources.Load("Localization/"+Application.systemLanguage) ??
        Resources.Load("Localization/English")) as TextAsset;

        // Put the TextAsset into a long string
        fullString = localizedText.text;

        // Fill the list with all the lines
        rowList = new List<string>();
        rowList.AddRange(fullString.Split("\n"[0]));

        // Separate each row and fill the dictionary
        dictionary = new Dictionary<string, string>();
        for (int i=0; i<rowList.Count; i++) {
            string[] lineSplitted = rowList[i].Split(new string[] {" =
"}, System.StringSplitOptions.None);
            dictionary.Add(lineSplitted[0], lineSplitted[1]);
        }
    }

    public static string LocalizeText(string text) {
        return Instance.dictionary[text];
    }

    public static void PrintDictionaryOnConsole() {
        string output = "ControllerLanguages - Language Dictionary:" +
        "\n";
        output += "\n" + "Application.systemLanguage = " +
        Application.systemLanguage;
        foreach(string key in Instance.dictionary.Keys) {
            output += "\n" + key + " = " + LocalizeText(key);
        }
        output += "\n";
        Debug.Log(output);
    }
}

```

Anexo V. Marcadores finales.

Primer Marcador



Figura 99 *FirstTarget*

Segundo Marcador



Figura 100 *SecondTarget*

Tercer Marcador



Figura 101 *ThirdTarget*

Cuarto Marcador



Figura 102 *FourthTarget*

Quinto Marcador



Figura 103 *FifthTarget*

Sexto Marcador



Figura 104 *SixthTarget*

Septimo Marcador



Figura 105 *SeventhTarget*

Octavo Marcador



Figura 106 *EighthTarget*

Bibliografía

Índice Bibliográfico

Android Developer. Web Oficial. Disponible en <https://developer.android.com/develop/index.html>

ARToolKit Documentation. Web Oficial. Disponible en <https://www.hitl.washington.edu/artoolkit/documentation/>

Apple Developer. Web Oficial. Disponible en <https://developer.apple.com/develop/>

Ayuntamiento de Madrid. Templo de Debod. Web oficial. Disponible en <http://www.madrid.es/portales/munimadrid/es/Inicio/templodebod/?vgnextoid=46caa0d03aa8b010VgnVCM100000d90ca8c0RCRD&vgnnextchannel=0c369e242ab26010VgnVCM100000dc0ca8c0RCRD&rmColeccion=30fe658f83a5d010VgnVCM100000b205a0aRCRD>

Carlos González Morcillo, David Vallejo Fernández, Javier A. Albusac Jiménez y José Jesús Castro Sánchez. Realidad Aumentada. Un Enfoque Práctico con ARToolKit y Blender. Disponible en <http://www.librorealidadaumentada.com/>

David Ramos. Realidad aumentada para un turismo 'inteligente'. Disponible en <http://www.silicon.es/realidad-aumentada-turismo-inteligente-2290438>

Diputación de Málaga. La Diputación de Málaga y Orange impulsan la primera aplicación de turismo con realidad aumentada y geolocalización de una provincia en España. Disponible en http://www.malaga.es/tecnologia/1325/com1_md3_cd-11865/diputacion-malaga-orange-impulsan-primer-a-aplicacion-turismo-realidad-aumentada-geolocalizacion-provincia-espana

Emiliusvgs. Tutorial básico de Realidad Aumentada con Unity. Disponible en <http://emiliusvgs.com/2014/09/06/tutorial-basico-realidad-aumentada-unity/>

Engadget. The sights and scents of the Sensorama Simulator. Disponible en <https://www.engadget.com/2014/02/16/morton-heiligs-sensorama-simulator/>

FS Gamer: Sistemas Operativos de Smartphones. Disponible en <http://www.fsgamer.com/sistemas-operativos-de-smartphones-152515.html>

Formación Online Grátis. Tutorial Vuforia | Crear aplicación de realidad aumentada. Disponible en <http://www.formaciononlinegratis.net/aplicacion-de-realidad-aumentada-android-vuforia/>

Google Glaas Developer. Web Oficial. Disponible en <https://developers.google.com/glass/design/principles>

IDC Spain. Web Oficial. Disponible en <http://www.idcspain.com/>

Imagina formacion. Realidad Aumentada Multiplataforma con Vuforia. Disponible en <http://www.imaginaformacion.com/cursos/>

Javier "Simón" Cuello y José Vittone. Diseñando apps para móviles. Disponible en <http://appdesignbook.com/es/contenidos/las-aplicaciones/>

JustInMind Examples. Web Oficial. Disponible en <http://www.justinmind.com/examples>

LanceTalent. Los 3 tipos de aplicaciones móviles: ventajas e inconvenientes. Disponible en <https://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/>

Layar Developer. Web Oficial. Disponible en <https://www.layar.com/features/developers/>

Metaio Documentation. Web Oficial. Disponible en <https://my.metaio.com/dev/creator/getting-started/the-first-startup/>

MOMA. "WeARinMoMA". Web Oficial. Disponible en <http://www.sndrv.nl/moma/>

Mtip. Web Oficial. Disponible en <http://www.mtrip.com/es/>

Museo Allard Pierson. Using Augmented Reality In The Museum. Web oficial. Disponible en <http://mesch-project.eu/using-augmented-reality-in-the-museum/>

Niantic Inc. Web Oficial. Disponible en <https://www.nianticlabs.com/blog/>

Tango Developer. Web Oficial. Disponible en <https://developers.google.com/tango/>

Total immersion Web Oficial. Disponible en <http://www.t-immersion.com/products/dfusion-suite/dfusion-studio>

Turismo de Segovia. Presentación Guía turística de Segovia en realidad aumentada. Disponible en <http://www.turismodesegovia.com/es/noticias/noticias-actuales/237-institucionales/343-presentacion-guia-turistica-de-segovia-en-realidad-aumentada>

Unity Documentation. Web Oficial. Disponible en <http://docs.unity3d.com/es/current/Manual/index.html>

Vuforia Developer. Web Oficial. Disponible en <https://developer.vuforia.com/library/>

Wikipedia, la enciclopedia libre. Realidad aumentada. Disponible en https://es.wikipedia.org/wiki/Realidad_aumentada

Wikipedia, la enciclopedia libre. Aplicación móvil. Disponible en https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil

Windows Phone Developer. Web Oficial. Disponible en <https://developer.microsoft.com/es-es/windows>

X. Basogain, M. Olabe, K. Espinosa, C. Rouèche y J.C. Olabe. Realidad Aumentada en la Educación: una tecnología emergente. Disponible en http://www.anobium.es/docs/gc_fichas/doc/6CFJNSalrt.pdf

Xataka. Videoplace, el abuelo artista de Kinect. Disponible en <http://www.xataka.com/historia-tecnologica/videoplace-el-abuelo-artista-de-kinect>

