

TorreviejAR

Grado en Ingeniería Multimedia

Trabajo Fin de Grado

Autor:
Pablo Torregrosa Vera

Tutor/es:
José Vicente Berná Martínez

Enero 2019



Universitat d'Alacant
Universidad de Alicante

Resumen

Se trata de una aplicación móvil para dispositivos Android haciendo uso de técnicas de realidad aumentada y geolocalización. Esta aplicación está enfocada en el sector turismo, en concreto, en el turismo de Torrevieja. El objetivo es que los turistas descubran las Lagunas de La Mata de una forma diferente y haciendo uso de la tecnología.

Con este proyecto, lo que conseguimos es que el turista realice una ruta guiada por hasta 54 puntos que rodean la Laguna Salada de La Mata. El motivo de estos puntos es que en esas localizaciones es donde antiguamente se colocaron unas columnas que delimitaban la zona de la explotación salinera. Lo que hemos conseguido es digitalizar eso y mostrar estas columnas a su paso por cada uno de los puntos.

Motivación, justificación y objetivo general

El interés por desarrollar este proyecto viene de la curiosidad que me ha generado desde hace ya un tiempo la realidad aumentada, una tecnología que hace no mucho apenas se sabía de ella y que a día de hoy está muy presente en la vida de mucha gente. Con el auge de Snapchat hace unos años, que se vale de esta tecnología para sus filtros o máscaras, la realidad aumentada llegó a las masas y, actualmente, con Instagram Stories copiando el modelo de Snapchat, las máscaras de perritos o de orejas de conejo están triunfando en la sociedad, sobre todo en la juventud. Por esto anterior y por la afinidad que tengo hacia la mezcla entre lo tecnológico y lo artístico, son las razones principales por las que me motiva la idea de hacer un proyecto con el uso de esta tecnología.

Llegados a la época del uso cotidiano del smartphone en la que nos encontramos, esta tecnología se presta mucho a la labor de ser consumida con smartphone, por lo que podemos intuir que, partiendo de la idea anterior de Snapchat e Instagram Stories, muchísima gente puede estar receptiva e interesada en usar otras aplicaciones de realidad aumentada, ya que la interacción que posee es un atractivo para el usuario, además de la expectación que crea como novedad para el usuario.

Dicho lo anterior, la idea de llevar esta tecnología hacia un sector distinto del ocio únicamente, como es el turismo, me atrae bastante y creo que al usuario también puede gustarle la idea de usar el móvil para sus viajes o excursiones, y, con ayuda de su cámara, poder divisar elementos virtuales que mezclados con los reales pueden hacer de su visita a un determinado lugar una experiencia divertida. También creo que puede despertar el interés de una persona que ha ido a dicho lugar con poco o ningún interés. En ese caso, el uso de una App que interactúe con él y le dé información que desconocía, que le muestre objetos gráficos divertidos y demás información puede hacer que le surja el interés por el lugar visitado. Por no hablar de la comodidad de usar el móvil para documentarte acerca de los lugares visitados, lo cual no olvidemos que sustituye a la mítica guía de viaje, y tiene la gran ventaja de no tener que llevar ese peso encima y poder tenerlo todo en un dispositivo tan ligero como un smartphone. Por todo esto, creo que es útil e interesante.

La realidad aumentada aplicada al turismo no es algo nuevo, pero sí la aplicación de esta tecnología para la creación de una App que ayude a los turistas a descubrir al máximo y mejorar su experiencia en visitas al Parque Natural de las lagunas de La Mata en Torreveja.

Existen ya aplicaciones de este tipo, en las cuales voy a fijarme para la realización de esta, como Euskadi Basque Country Turismo, que hace uso de la aplicación Layar, la cual permite escanear códigos QR o cualquier ítem con código Layar, para mostrar información relacionada con el turismo en Euskadi. Por otra parte, en Galicia también tienen la suya: Turismo Galicia, con la cual puedes localizar puntos de interés turístico que se encuentren cerca de ti usando la cámara del móvil y la realidad aumentada.

Teniendo todo lo anterior en cuenta, me dispongo a crear un proyecto en el que tendré que plantear un diseño de aplicación correcto teniendo en cuenta principios de usabilidad y accesibilidad, interactivo, divertido y sobre todo útil para todos los turistas que quieran usar la tecnología para mejorar su experiencia en la visita al Parque Natural de las lagunas de La Mata, Torrevieja.

Agradecimientos

Quiero agradecer a mi tutor, José Vicente Berná Martínez, por su gran ayuda y el gran trato que me ha dado a lo largo de este proyecto.

Agradecer también a mi familia, amigos y todos aquellos que han contribuido a mi enseñanza y a formarme como persona, en especial a mi abuela.

Índice de figuras

Figura 1. Ejemplo de aplicación de realidad aumentada	9
Figura 2. Aplicación de Ikea con realidad aumentada	10
Figura 3. Cuadro para el análisis Lean Canvas	12
Figura 4. Laguna Salada de La Mata	17
Figura 5. HMD de Ivan Sutherland	18
Figura 6. Aparición de un modelo virtual al enfocar un marcador en AR MApp	20
Figura 7. Interfaz de la App AR MApp	21
Figura 8. Interfaz principal de la App Turismo Benicàssim	22
Figura 9. Mapa en realidad aumentada con puntos clave disponibles para interactuar	22
Figura 10. Menú virtual superpuesto sobre el marcador al que enfocamos con nuestro dispositivo	23
Figura 11. Interfaz principal de Senditur	24
Figura 12. Uso de la Realidad Aumentada en la aplicación Senditur	25
Figura 13. Máscara de perro virtual sobre imagen real	26
Figura 14. Atrapando un Pokémon	27
Figura 15. Interfaz principal de IKEA Place	28
Figura 16. Logo de Unity	29
Figura 17. Logo de Unreal Engine	30
Figura 17b . Comparación SDKs de Realidad Aumentada	32
Figura 18. Logo de Android Studio	33
Figura 19. Logo de React Native	34
Fuente 20. Logo de Ionic	34
Figura 21. Stack tecnológico del proyecto	44
Figura 22. Pantalla principal de la aplicación	48
Figura 23. Prototipo de la funcionalidad Mapa	51
Figura 24. Vista AR usando la cámara del dispositivo	52
Figura 25. Vista del prototipo de la funcionalidad Info	53

Figura 25b. Paleta de colores de la aplicación	54
Figura 26. Planificación de tareas de la iteración 1 en Trello	57
Figura 27. Inicio de un proyecto en Android Studio	58
Figura 28. Datos de los puntos de interés de la ruta	61
Figura 29. Icono azul rectangular en nuestra posición	63
Figura 30. Planificación de tareas de la segunda iteración en Trello	64
Figura 31. Modelo del mojón	65
Figura 32. Modelo 3D en realidad aumentada de La Tierra en nuestra localización	65
Figura 33. Modelo 3D en realidad aumentada del mojón en nuestra localización	66
Figura 34. Vista del apartado AR	67
Figura 35. Interfaz principal	69
Figura 36. Interfaz mapa mostrando marcadores	70
Figura 37. Interfaz mapa mostrando posición actual	71
Figura 38. Interfaz AR	72
Figura 39. Interfaz Info	73

Índice de tablas

Tabla 1. Planificación temporal TFG	15
Tabla 2. Comparación básica de metodologías ágiles	37

1. Introducción

Al hablar de realidad aumentada [1] hablamos de una tecnología usada en dispositivos electrónicos, la cual mezcla la realidad que percibimos los seres humanos con objetos virtuales, ya sean imágenes, letreros, objetos virtuales, etc. En la siguiente imagen podemos ver una aplicación móvil que hace uso de esta tecnología, en la cual, con ayuda de la cámara de la Tablet podemos ver el objeto real (cuerpo humano) junto a contenido virtual (flechas que indican partes del cuerpo).

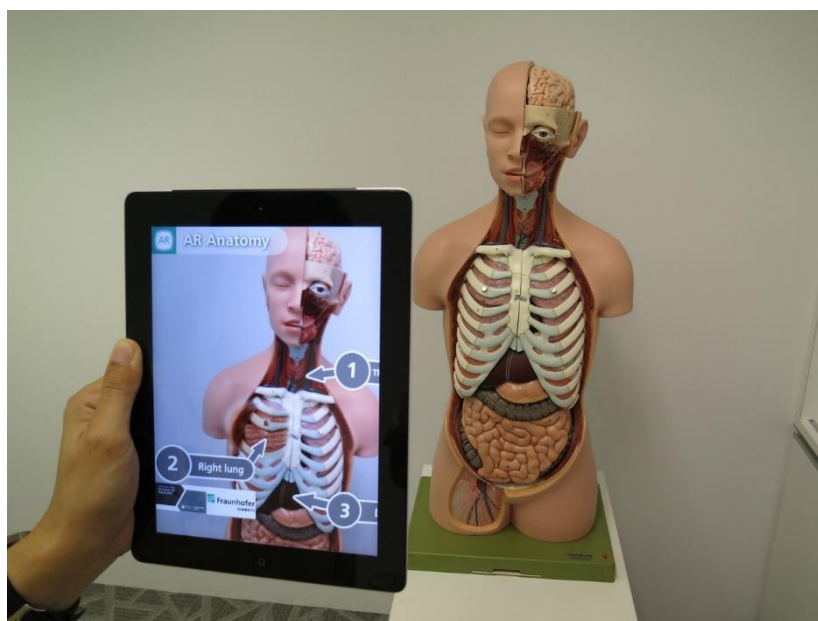


Figura 1. Ejemplo de aplicación de realidad aumentada. (Fuente: www.pixabay.com/es/)

Esta tecnología abarca gran cantidad de sectores. Quizás el más famoso sea el del ocio digital con aplicaciones como Pokemon Go [2], con la que al ir moviéndonos por la calle podremos encontrar y capturar Pokemons. Pero también es usada en publicidad y marketing, como la nueva App de Ikea [3] para visualizar como quedarían los muebles en tu casa (Figura 2), o en medicina, para visualizar órganos superpuestos al cuerpo del paciente, por ejemplo. En el sector deportivo tenemos Fitness AR [4], que permite proyectar un mapa en 3D sobre cualquier superficie para ver tus entrenos y descubrir nuevas rutas. Como podemos ver, el abanico de sectores que usan esta tecnología es amplio y cada vez lo será más, puesto que esta tecnología aún no ha explotado, a pesar de que ya existen bastantes aplicaciones.



Figura 2. Aplicación de Ikea con realidad aumentada. (Fuente: <http://photozou.jp/photo/show/124201/251690038>)

Este proyecto se va a centrar en otro sector que compatibiliza muy bien con el uso del móvil, el turismo, y es que hoy en día casi todo el mundo lleva un teléfono en el bolsillo. Si a eso le sumamos que esta tecnología tiene como dispositivo estándar el Smartphone, mucho mejor. También tenemos que tener en cuenta que los turistas, con la mejora de las cámaras de los smartphones, ya muchos han olvidado las cámaras fotográficas, y esto hace que nuestra aplicación esté muy al alcance del usuario cuando visita lugares, puesto que, si está tomando fotos con su móvil, abrir una app que le aporta información de esos lugares es un plus.

Se pretende que el usuario descubra esos lugares que no conoce, llevándolo a interactuar con objetos virtuales que le hagan esta experiencia amena, ya sea incluyendo modelos 3D divertidos, letreros y/o flechas virtuales que le guíen por el destino turístico, sugerencias sobre lugares cercanos, posibilidad de hacer fotos divertidas del lugar y cualquier detalle interesante que le genere una sensación de experiencia más completa que el hecho de visitar simplemente el lugar.

Otro factor que hay que tener en cuenta es la muestra de objetos en pantalla, ya sean iconos 2D o modelos 3D. En caso de incluir modelos 3D, estos objetos se realizarán con software de modelado 3D como puede ser 3DS Max o nos serviremos de la herramienta Poly [5] de Google, la cual dispone de una galería de modelos 3D disponibles para el usuario.

2. Estudio de viabilidad

En esta sección vamos a realizar un estudio de viabilidad en el que analizaremos un poco si el proyecto en sí mismo, sus pretensiones u objetivos son viables, pertinentes y necesarios. Para esto usaremos una herramienta muy popular en los estudios de viabilidad, el Lean Canvan.

2.1. Lean Canvan

Lean Canvan es una herramientas idónea para analizar un producto que está siendo creado y que tiene carácter innovador, ya sea porque es una solución nueva a un problema ya existente o bien porque es una mejora sobre otras soluciones que ya existen. La idea de Lean Canvan es analizar el proyecto desde diferentes perspectivas de interés como muestra la Figura 3, no solo desde el desarrollo o los costes.

Problema (alternativas)	Solución	Proposición de valor única	Ventaja especial	Segmentos de cliente (usuarios visionarios)
	Métricas clave		Canales	
Estructura de costes			Flujos de ingresos	

*Figura 3. Cuadro para el análisis Lean Canvan
(Fuente propia)*

A continuación analizamos las diferentes secciones del cuadro Lean Canvan:

Segmentos de clientes

Analizamos los posibles clientes de nuestra aplicación:

- Turistas que visiten Torrevieja

- Gente que vive en Torrevieja y alrededores
- Trabajadores del sector turístico de la zona de Torrevieja

Problemas

Los problemas que intentamos solucionar con el proyecto son:

- Fomentar la visita al Parque Natural de Las Lagunas de La Mata
- Guiar a los visitantes durante la visita
- Ahorrar gastos en difusión al ayuntamiento de Torrevieja, ya sea en papel a la hora de hacer mapas, creando flyers, contratando guías u organizando eventos

Riesgos

- Posibilidad de que a la gente no le atraiga el producto
- Necesidad de usar el GPS o conexión a internet en el dispositivo para utilizar la aplicación
- Precisión a la hora de realizar la ruta pasando por las distintas ubicaciones. Hay que tener en cuenta que la ubicación puede aparecer con unos metros de error
- Aplicación que usa tecnología novedosa y que sustituye al clásico mapa de papel
- Desconocimiento de algunas tecnologías para el desarrollo de la aplicación

Proposición de valor única

Se trata de la única aplicación que facilitará la visita al Parque Natural de Las Lagunas de La Mata. Por tanto, no existe competencia y es original. Además, ofrece una alternativa al turismo tradicional.

Solución a los posibles riesgos

Las soluciones propuestas a los posibles riesgos son las siguientes:

- Advertir en la aplicación al usuario sobre el uso de GPS o conexión a internet para su correcto uso
- Realizar un diseño accesible y usable para el usuario de modo que le resulte cómodo su uso
- Correcta planificación y documentación sobre las tecnologías existentes con el fin de elegir las adecuadas para el desarrollo

Canales

La aplicación se dará a conocer por medio de redes sociales y mediante el Ayuntamiento de Torrevieja.

Flujos de ingreso

En principio no tenemos intención de ingresos, ya que se trata de un trabajo académico.

Estructura de costes

Los principales costes serán los costes de desarrollo y el mantenimiento de la aplicación móvil.

3. Planificación

Una buena planificación desde el principio puede ayudar también en la consecución del proyecto. Además permitirá una vez terminado analizar junto a los resultados la adecuación de la planificación al desarrollo completo del proyecto. Esto nos dará experiencia ya que al planificar se realiza una estimación de costes temporales en función de unas capacidades o habilidades supuestas. Después, al analizar los resultados podremos comparar si nos pasamos o nos quedamos cortos con la estimación. La planificación temporal además nos ayudará a seguir unos pasos en el tiempo y no invertir demasiado tiempo en aquello que no debe.

A continuación hemos rellenado una tabla para establecer la planificación. Esta estimación está realizada en función de la experiencia propia obtenida con otros proyectos anteriores.

Tabla 1. Planificación temporal TFG

Contenidos	Tiempo total
Motivación, justificación, objetivo general, Introducción Estado del arte	1 mes
Objetivos Metodología Análisis y especificación Presupuesto, estimaciones, planificación	1 mes
Diseño	1 mes
Implementación	1 mes
Pruebas y validación Resultados Conclusiones y trabajo futuro Referencias, bibliografía y apéndices Agradecimientos, citas, índices	2 semanas

4. Estado del arte.

En este apartado vamos a recoger información acerca del problema para así poder tener un mayor conocimiento y orientación hacia la solución. Para ello debemos abordar distintos temas como: el problema a tratar en cuestión, las aplicaciones relacionadas con la tecnología a utilizar, los lenguajes de programación, frameworks a utilizar en función del lenguaje de programación, el tipo de arquitectura, herramientas a utilizar, etc.

En cuanto a aplicaciones relacionadas debemos analizar aspectos como: funcionalidades que tienen, tecnologías que usan o plataformas en las que se encuentran.

El lenguaje de programación es un tema importante, y para ello debemos encontrar el adecuado para el tipo de aplicación que queremos. Esto lo decidiremos en función de si elegimos un desarrollo nativo o híbrido. Una vez escogido, podemos adentrarnos en la búsqueda de frameworks y librerías que nos faciliten la tarea de programar en ese lenguaje.

Es importante también documentarnos acerca de herramientas que nos facilitan el desarrollo, como las que nos permiten realizar mockups.

4.1. Antecedentes

Basándonos en el libro *Real Salina de La Mata. Auto del deslinde de 1763* [6], vamos a contar un poco de dónde viene nuestro proyecto.

A lo largo del siglo XVIII se llevaron a cabo diferentes deslindes y amojonamientos del territorio circundante a la laguna de La Mata, como forma de garantizar la protección del coto de la Real Salina, perteneciente a la Corona, de cuya explotación obtenía importantes beneficios.

De los cuatro deslindes y amojonamientos que se llevaron a cabo, el que nos incumbe a nosotros es el de la Real Salida de La Mata de 1763, cuyo propósito no era otro que proteger los intereses de la explotación salinera y la idea de progreso para la economía del país.

El amojonamiento es la acción que consiste en delimitar o deslindar, mediante hitos o mojones (postes), para determinar límites de heredades o jurisdicciones. Esta acción generó conflicto porque los vecinos de la zona veían mermadas las posibilidades de expansión de superficies de cultivo.



Figura 4. Laguna Salada de La Mata. (Fuente: Google Maps)

En suma, el objetivo de la demarcación y amojonamiento era restablecer el área circundante de la laguna, mediante mojones o hitos, y garantizar de este modo la conservación del espacio adyacente. Para ello, se colocaron 54 mojones rodeando la Laguna Salada de La Mata.

En la actualidad, la mayoría de esos mojones no son conservados. Nosotros con nuestro trabajo queremos digitalizar todo esto y también recordar el amojonamiento de las Lagunas de La Mata.

4.2. Fundamentos de la realidad aumentada

Al hablar de esta tecnología estamos hablando de mezclar el mundo real y el mundo virtual, es decir, combinar elementos del mundo real con elementos generados por ordenador que se muestran mediante un dispositivo en tiempo real.

El ingeniero Ronald Azuma, quien es un pionero en el campo de la realidad aumentada, define a la misma como una combinación de elementos reales y virtuales, además de ser interactiva, en tiempo real y estar registrada en 3D [1].

El término de realidad aumentada suele ir ligado con el de realidad virtual, pero hay que diferenciarlas. La primera monta una serie de objetos virtual sobre la realidad material del mundo físico, y la segunda, hace que el usuario se sumerja en un entorno completamente virtual aislándose del mundo real. Pero hay otro término que surge de la combinación entre estas dos, que se conoce como realidad mixta, y que permite la interacción de objetos y/o personas tanto reales como virtuales.

El origen de la realidad aumentada [7] surge en el 1962, cuando Morton Heilig, un director de fotografía, patenta un simulador llamado Sensorama el cual proyectaba imágenes en 3D con sonido envolvente, vibración en el asiento e incluso simulaba el viento lanzando aire al usuario. Quizás este simulador nos pueda parecer que se asemeja más a la realidad virtual que a la realidad aumentada, pero es que estos dos términos están muy ligados y no fue hasta más adelante cuando avanzó lo suficiente la tecnología como para poder separar estas dos ramas. En 1973 aún es pronto para diferir estos dos términos, pero es el año en que Ivan Sutherland [7] inventa el casco de realidad virtual, lo que se conoce en la actualidad como HMD (head-mounted display). Se trataba de un dispositivo colgado del techo que tenía unos sensores que permitían medir la orientación hacia la que miramos y así el ordenador recibir esta información y representar en el dispositivo las imágenes 3D.



Figura 5. HMD de Ivan Sutherland (Fuente: https://www.researchgate.net/figure/Ivan-Sutherlands-HMD-31_fig19_216813812)

No fue hasta 1990 cuando Tom Caudell [7] creó el término de realidad aumentada. En esa época surge el que está considerado como el primer sistema de realidad aumentada, de mano de L.B. Rosenberg, que trabajaba para la fuerza aérea de Estados Unidos. Es un dispositivo que da consejos al usuario sobre cómo realizar ciertas tareas a medida que estas se presentan, una especie de guía virtual.

El otro proyecto que dio lugar a la realidad aumentada procedía de la Universidad de Columbia, donde un equipo de científicos inventó un HMD que interactuaba con una impresora. El dispositivo, bautizado como KARMA [7], proyectaba una imagen en 3D para dar instrucciones a su usuario sobre cómo recargar la impresora, en lugar de acudir al manual de uso.

Ya en la actualidad tenemos una mayor complejidad en este campo, y es que también está la realidad mixta [8], que es la combinación de realidad aumentada y realidad virtual, la cual es consumida con gafas, como las Hololens [9] de Microsoft.

La realidad aumentada la podemos separar en dos tipos básicamente:

- **Basada en marcadores:** donde al apuntar a una imagen u objeto concreto (marcador) aparece el objeto virtual.
- **Sin marcadores:** no es necesario apuntar a nada para generarla. Por ejemplo, la geolocalización, como es nuestro proyecto, es un tipo de realidad aumentada sin marcadores, donde en función de la localización se activa la realidad aumentada.

4.3. Aplicaciones AR del sector turístico

A continuación vamos a estudiar aplicaciones que han usado la realidad aumentada para el turismo. Existen diversas aplicaciones parecidas a nuestro proyecto. En este apartado vamos a analizar unas cuantas:

AR MApp [13]

Está enfocada al turismo en Barcelona. Con esta App puedes descubrir atracciones y lugares icónicos de la ciudad condal enfocando un marcador, que suele ser la imagen de un mapa físico de la zona en cuestión que los creadores han diseñado y distribuido por diversos hoteles de la ciudad. Lo único que tienes que hacer es enfocar el mapa y sin necesidad de conexión a internet la aplicación te mostrará los modelos 3D.



Figura 6. Aparición de un modelo virtual al enfocar un marcador en AR MApp (Fuente propia)

Como primera interfaz, tras mostrarnos el logo, nos aparece la elección de idioma entre inglés y español (la elección del inglés como opción además de español es algo que viene muy bien para mejorar la accesibilidad). Ahora nos pregunta si queremos iniciar el tutorial, el cual nos guía y nos sugiere que enfoquemos a un marcador específico. En caso de no disponer de ellos nos da la oportunidad de pulsar un icono que nos lleva a su web donde los podemos descargar. Una vez que continuemos, la aplicación ya accede a la cámara del dispositivo, donde si apuntamos a un marcador nos aparece un modelo 3D junto con un audio que nos cuenta información relevante e histórica acerca del lugar.

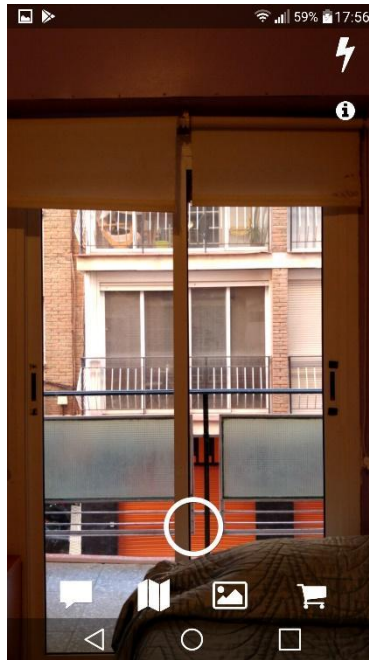


Figura 7. Interfaz de la App AR MApp (Fuente propia)

Como podemos ver en la figura 7, esta es la interfaz que nos aparece después del tutorial y las siguientes veces que iniciemos la aplicación. El rayo de la esquina derecha arriba nos permite encender el flash de la cámara. Justo debajo de él, ese icono nos da la opción de repetir o no el tutorial. Ya en la parte inferior disponemos del botón central circular, que nos permite tomar la captura y automáticamente nos da la posibilidad de compartirlo en otras aplicaciones. Debajo de este, tenemos cuatro botones más: el primero nos permite cambiar el idioma como anteriormente hemos mencionado; el segundo sirve para ubicarnos en el mapa; el tercero para buscar marcadores; y el cuarto nos lleva a su web.

Turismo Benicàssim

Esta aplicación de la Concejalía de Turismo del Ayuntamiento de Benicàssim ofrece información relevante del municipio, de la meteorología, de los eventos culturales, etc.



Figura 8. Interfaz principal de la App Turismo Benicàssim (Fuente propia)

En cuanto a la realidad aumentada, dispone de un botón para activar el modo AR, que nos permite ver un mapa interactivo en 3D de la ciudad si apuntamos con nuestra cámara a uno de los distintos patrones del mapa de Benicàssim que ellos nos facilitan. Nos aporta información de los lugares de interés con letreros virtuales, modelos 3D, rutas de interés turístico y audio guías. Al igual que las anteriores, nos pide como requerimiento el acceso a la cámara de nuestro dispositivo para sacar fotos y grabar video, además del acceso a nuestro contenido multimedia y a la ubicación del dispositivo.

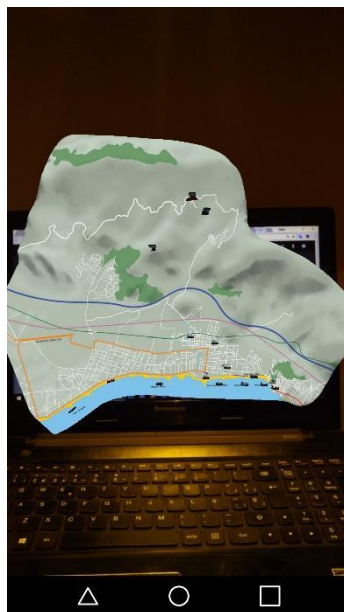


Figura 9. Mapa en realidad aumentada con puntos clave disponibles para interactuar (Fuente propia)

La primera interfaz nos da a elegir entre español o inglés como idioma. Tras esta nos lleva a otra donde en la esquina superior derecha nos dice la temperatura de Benicàssim (suponemos que la actual) y en la parte inferior disponemos de tres botones: el primero es el modo AR, que ya hemos detallado antes, el cual dependiendo del marcador que enfocamos nos puede mostrar un objeto 3D simplemente, un mapa virtual con puntos clave los cuales al ser presionados nos presenta una audio guía del lugar con contenido histórico del mismo (figura 9), o incluso un menú en realidad aumentada como vemos en la figura 10; el segundo nos abre el mapa de Google Maps enfocado en la ciudad de Benicàssim; y el tercero nos muestra categorías, que dentro de ellas tienen subcategorías y al final nos describen un lugar, ya sea con audio guía y texto o texto sólo, y nos llevan a un enlace externo de la web que le corresponde. Estas categorías son: agenda y ocio, playas, lugares de interés, turismo activo, qué hacer en familia y dormir/gastronomía.

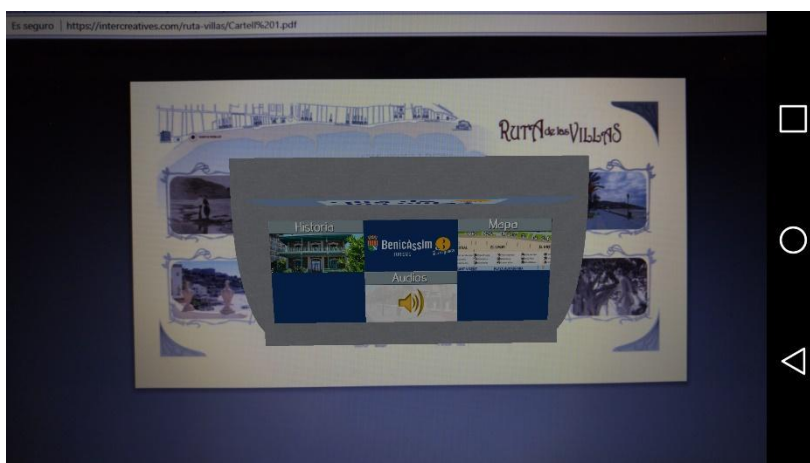


Figura 10. Menú virtual superpuesto sobre el marcador al que enfocamos con nuestro dispositivo (Fuente propia)

Senditur

Con esta App podremos descubrir lugares y rutas de interés cercanos a nuestra posición. Nos localiza con ayuda del GPS de nuestro dispositivo, por tanto las principales funciones no necesitan conexión a internet, y nos permite usar la cámara de nuestro móvil para ver hacia qué dirección se encuentran los puntos de interés más cercanos, que nos saldrán como objetos virtuales interactivos. Al tocarlos nos muestra la distancia a la que nos encontramos, cómo llegar e incluso la altitud a la que se encuentra. Está diseñada para ser consumida en el senderismo, la montaña, viajes y el Camino de Santiago, así como hacer más amena la ruta.

Al abrir la aplicación nos encontramos una interfaz en la que nos aparecerán localizaciones de interés en función de cómo tengamos configurado el apartado de ajustes. Cada una de ellas

nos detalla la distancia que hay hasta llegar a la misma, el cómo llegar, que indica la latitud y longitud además de la provincia a la que corresponde, y que si pulsamos nos lleva a Google Maps con la ruta ya marcada hasta el destino. Por último, la descripción del lugar, que incluye el nombre del lugar, su provincia, comunidad autónoma y la altitud.

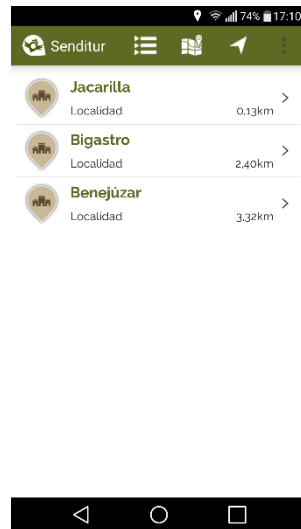


Figura 11. Interfaz principal de Senditur (Fuente propia)

La interfaz, como podemos ver en la figura 11, contiene una barra de navegación con cuatro iconos:

- El primero sirve para consultar el listado de sitios más cercanos a nuestra posición.
- El segundo nos permite visualizar el mapa que hay a nuestro alrededor.
- El tercero inicia la vista de la cámara de nuestro dispositivo para encontrar los lugares haciendo uso de la realidad aumentada, donde aparecen los iconos de los lugares clave y al pulsar sobre ellos nos lleva a la descripción, además en la esquina superior izquierda tenemos una brújula que marca estos lugares clave y en la inferior derecha una ruleta que al pulsarla nos da la opción de filtrar por tipo o por distancia: el tipo nos da la opción de marcar diferentes casillas como localidad, montaña, punto de interés, ruta, servicios o Camino de Santiago; el filtro por distancia nos da la posibilidad de elegir el radio máximo de búsqueda de cada una de las opciones el filtro por tipo y también la distancia mínima a partir de la cual queremos encontrar lugares.
- El cuatro nos permite configurar las dos opciones anteriores de filtro por tipo y por distancia del mismo modo, funcionalidad repetida, y tiene un tercer apartado de información donde nos detalla toda la aplicación y sus funcionalidades.

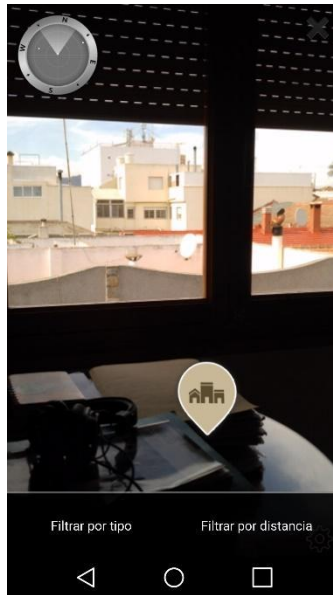


Figura 12. Uso de la Realidad Aumentada en la aplicación Senditur (Fuente propia)

El modo de negocio de esta App se encuentra en la web de la misma. En la página principal de la web aparecen diversos lugares los cuales se sitúan ahí porque el organismo encargado de dicho lugar ha pagado por ello. También existe la opción de que los negocios paguen por promocionar su ruta, de modo que se coloca un banner destacado en la ruta. Por tanto, podemos decir que los creadores quieren acercarse al sector turístico rural e incentivar a que los negocios de dicho sector paguen por promocionarse en este ámbito.

4.4. Aplicaciones AR de otros sectores

Ahora vamos a estudiar las Apps más exitosas y/o interesantes de Realidad Aumentada y que no se centran en el turismo, para poder ver de qué funcionalidades disponen y cuáles nos podrían servir para nuestro proyecto:

Snapchat [10]

Quizás la más famosa. Esta aplicación es usada por millones de usuarios en todo el mundo, sobre todo jóvenes. Se trata de una App de mensajería efímera, ya que el contenido compartido es eliminado automáticamente después de pasar un tiempo determinado. Este contenido puede ser texto, imagen o video, pero lo que destaca de Snapchat es el colorido y el atractivo de sus filtros, los cuales se basan en la tecnología de realidad aumentada. Permite detectar nuestra cara mediante procesado de imagen y aplicarnos una máscara virtual o distintos efectos virtuales sobre la imagen real, como podemos ver en la figura 13. También

hace uso de rótulos personalizados dependiendo la localización y últimamente hasta ha incorporado pequeños juegos combinados con la funcionalidad de fotos.



Figura 13. Máscara de perro virtual sobre imagen real. (Fuente propia)

De acuerdo con analistas, el éxito de la aplicación está en aprovechar el miedo que tienen los jóvenes de perderse los acontecimientos en la vida de los demás [11], ya que una publicación de Snapchat dura 24 horas y sólo puede ser vista en ese período de tiempo. También hay que tener en cuenta que las nuevas generaciones prefieren comunicarse con imágenes antes que con textos.

Con todo lo anterior podemos sacar conclusiones de cara a nuestro proyecto. Hemos visto el funcionamiento de Snapchat y la funcionalidad de compartir imágenes nos podría venir bien para nuestra aplicación.

Pokémon GO [12]

Otro de los motivos de la actual fama de la realidad aumentada es este juego de la saga Pokémon. Se trata de buscar y capturar Pokémon que se encuentran en ubicaciones reales. Además, tendremos que luchar con ellos, sobre todo en puntos físicos establecidos que se denominan gimnasios Pokémon, lo cual hace que se genere interacción social, ya que otros jugadores también es probable que acudan a estos lugares a luchar.



Figura 14. Atrapando un Pokémon. (Fuente: <https://pixabay.com/es/pokemon-ir-pokemon-calle-c%C3%A9sped-1569794/>)

Debido a las constantes actualizaciones, se han ido introduciendo novedades, pero la otra famosa característica son las pokeparadas, que coinciden con lugares interesantes de la geografía real, donde los jugadores pueden acudir en busca de premios.

Por tanto, la realidad aumentada se observa principalmente en el momento en el que se va a capturar un Pokémon, como vemos en la figura 14. En ese instante con la cámara de nuestro dispositivo visualizaremos la realidad junto con modelos 3D de los Pokémon a capturar además del lanzamiento de la Pokeball virtual utilizada para capturarlos. En este momento también dispone de la funcionalidad de tomar una fotografía, esto es algo que podemos incorporar a nuestro proyecto, ya que puede ser de agrado para el usuario tomar una foto cuando aparece el modelo 3d, por ejemplo.

El principal factor de éxito de Pokémon GO es obvio que es la marca Pokémon. Esta saga ha sido un referente tanto en los juegos RPG en los noventa como su serie de televisión.

IKEA Place

La nueva aplicación de IKEA [3] usa las nuevas tecnologías de desarrollo para Realidad Aumentada: ARKit en iOS y ARCore en Android, esto quiere decir que sólo los dispositivos más potentes podrán descargarla, ya que no todos los Smartphones permiten el uso de estas dos. Esta aplicación ha sido diseñada para aquellos clientes que quieren visualizar como quedarían los muebles en su casa sin salir de la misma. Cabe destacar que las medidas y escalas se ajustan a la realidad, para obtener un resultado fiable acerca de los productos, de modo que si colocamos el producto más lejos aparecerá escalado. Una vez probada la aplicación, se puede

decir que las medidas son bastante fiables. Además, la App reconoce los planos horizontales y cuando elegimos un producto virtual para depositarlo en cualquier lugar de nuestra habitación, escanea toda la superficie para obtener una correcta referencia del suelo o lugar donde queremos dejar el objeto, con el fin de que el objeto no quede flotando en el aire. También tenemos la posibilidad de girar el producto para situarlo de la manera que más nos guste.

Al abrir la App nos encontramos con el logo de IKEA y seguidamente pasa automáticamente a la vista de la cámara del dispositivo, en la cual aparece un asistente en forma de letreros que nos guía. Esta vista de cámara dispone de tres botones en la parte inferior:



Figura 15. Interfaz principal de IKEA Place (Fuente propia)

1. El primero permite hacer una fotografía y mediante un marco, elegir la parte de la foto que queremos buscar en la base de datos de productos de IKEA para obtener el más parecido al fotografiado. Esto puede ser útil si queremos un producto parecido a uno que tenemos o vemos en cualquier lugar, o si alguien nos dice que el producto es de IKEA y queremos saber exactamente cuál es de manera rápida.
2. El segundo nos lleva a una interfaz donde categoriza los productos primero por colecciones destacadas, botón de buscador por nombre, una sección "Comenzar" con unos cuantos productos, la siguiente sección ordena los productos por tipos, y la última sección se llama "Todo" donde aparecen unos productos y luego el botón de "Explorar todo", que nos lleva a una sección con todos los productos de la aplicación. De todos estos productos podemos elegir uno para situarlo en nuestra habitación y ver cómo quedaría.

3. El tercer botón nos lleva a la sección “Favoritos”, donde aparecerán los productos que anteriormente hayamos añadido. También aparece un botón que nos lleva a “Ajustes” donde podemos indicar la ubicación del país, permitir el audio o no, y por último la política de privacidad, términos y condiciones e informar de problemas.

Con esta App, IKEA pretende hacer llegar sus productos a un mayor público. El hecho de no tener que ir a la tienda y poder ver cómo los productos quedan en tu espacio vital es el plus que tiene el cliente y algo que le acerca a la compra del producto. Esto es algo similar a nuestro proyecto, donde queremos virtualizar algo que antes se hacía físicamente, pasar de mapa de papel a aplicación móvil virtual que te ayuda a descubrir el lugar.

4.5. Tecnologías valoradas para el desarrollo

En este apartado desglosaremos las distintas tecnologías que vamos a valorar para posteriormente elegir y desarrollar nuestro proyecto. Hay que decir que además de valorar un desarrollo nativo o híbrido de la aplicación, también vamos a valorar las distintas herramientas para el desarrollo de realidad aumentada, ya que debemos integrarla en nuestra aplicación.

Para la parte de realidad aumentada vamos a valorar herramientas como Unity, Unreal Engine, ARKit, ARCore, Vuforia y Wikitude.

Para valorar el desarrollo nativo o híbrido valoraremos herramientas como Android Studio, React Native e Ionic.

No será en este apartado donde escojamos las tecnologías utilizadas finalmente, sino en el apartado 8.1.

4.5.0. Unity



Figura 16. Logo de Unity.(Fuente:

https://commons.wikimedia.org/wiki/File:Unity_Technologies_logo.svg)

Es un motor gráfico muy usado para el desarrollo de videojuegos, pero últimamente han ido incorporando funcionalidades para permitir el desarrollo de experiencias inmersivas como la

realidad virtual y la realidad aumentada. También es usado en ámbitos como la arquitectura para el prototipado de construcciones.

Unity tiene una interfaz fácil de aprender en la que tenemos la vista gráfica del proyecto, el proyecto en sí y todos sus archivos, un panel con las propiedades del objeto seleccionado y un panel con los materiales por defecto del motor además de los añadidos por nosotros para utilizar. También dispone de una consola donde ver el resultado de la compilación de nuestro proyecto, ya que las funcionalidades se programan en lenguaje C# mediante scripts.

4.5.1. Unreal Engine

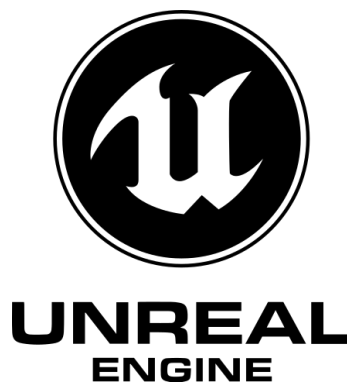


Figura 17. Logo de Unreal Engine. (Fuente:

https://es.m.wikipedia.org/wiki/Archivo:UE_Logo_Black_Centered.svg)

Otro motor gráfico desarrollado por Epic Games. Unreal junto Unity son los motores gráficos más conocidos. Es similar al anterior pero con algunas diferencias. Aquí la programación se realiza en C++ y disponemos de programación con Blueprints, algo más gráfico y se programa uniendo nodos a funciones de programación para obtener un resultado.

4.5.2. ARCore

SDK desarrollado por Google [15] para competir con Apple en torno a la realidad aumentada. Con él podemos crear experiencias para dispositivos Android, sólo en los dispositivos más actuales y de mejores prestaciones, por tanto es un inconveniente para nosotros puesto que el dispositivo del que disponemos no pertenece a los que soportan esta tecnología. ARCore usa tres características principalmente:

- El rastreo de movimiento, para que el dispositivo entienda y rastree tu posición relativa en el mundo.

- Comprensión del ambiente, lo que permite al dispositivo detectar el tamaño y localización de superficies horizontales, verticales y en ángulo.
- Estimación de luz, que permite al teléfono estimar las condiciones de iluminación actuales del entorno.

4.5.3. Vuforia

Vuforia [16] es un kit de desarrollo de software (SDK) de realidad aumentada para dispositivos móviles que permite la creación de aplicaciones de realidad aumentada. Utiliza tecnología de visión por computador para reconocer y rastrear imágenes planas y objetos 3D simples. Es una de las tecnologías más famosas en este campo. Sus características son:

- Detección de objetos, imágenes y textos
- Rastreo de objetivos
- Reconocimiento 2D y 3D
- Escaneo de objetos reales para su reconocimiento
- Botones virtuales
- Mapeo de elementos adicionales vía OpenGL
- Smart Terrain™, la capacidad de reconstruir un terreno en tiempo real, creando un mapa del entorno geométrico en 3D
- Seguimiento extendido, capacidad que muestra continuas experiencias visuales incluso cuando el blanco está fuera de vista
- Ground Plane, que permite colocar contenido digital en superficies horizontales de su entorno
- Licencia gratis y comercial

Podemos decir que es una muy buena herramienta, pero el no disponer de herramientas para facilitar la geolocalización, que es lo que buscamos, es un gran inconveniente.

4.5.4. Wikitude

Otra librería muy usada para el desarrollo de experiencias en realidad aumentada [17]. El punto a favor que tiene en relación a nuestro proyecto es que soporta la geolocalización. Sus características son:

- Reconocimiento de objetos y escena

- Seguimiento instantáneo, que hace posible representar objetos virtuales sin necesidad de marcadores
- Reconocimiento de imagen y de imágenes múltiples
- Servicios basados en localización
- Seguimiento extendido, lo que permite desvincularse del marcador sin perder la experiencia en realidad aumentada
- Servicio de reconocimiento en la nube, permite a los desarrolladores trabajar con miles de imágenes de destino alojadas en la nube
- Cargar y renderizar modelos 3d importados de herramientas como Blender
- Licencia de pago, la gratuita tiene marca de agua que ocupa toda la pantalla

Aquí tenemos una tabla donde comparamos los diferentes kits de desarrollo de realidad aumentada:

SDK Feature Comparison Table

	Wikitude	ARKit	ARcore	Vuforia	MaxST	DeepAR	EasyAR	ARToolKit	Xzimg
Maximum distance capture (m)	2.4 / 5	1.5 / 5	1.0 / 3	1.2 / 3.7	0.5 / 0.9	0.7 / 5	0.9 / 2.7	3 / 3	0.5 / 1
Recognition stability of immovable marker	6	9	9	10	7	8	7	8	4
Recognition stability of movable marker	6	7	6	6	2	7	3	6	3
Minimum angle recognition	10	30	50	30	50	35	35	10	45
Minimum visibility for recognition overlapped marker	100%	50%	75%	20%	50%	10%	10%	100%	25%
2D Recognition	✓	✓	✓	✓	✓	✓	✓	✓	✓
3D Recognition	✓	✓	✓	✓	✓	-	-	-	✓
Geo-Location	✓	✓	✓	-	-	-	-	-	-
Cloud Recognition	✓	✓	✓	✓	-	-	-	-	-
SLAM	✓	✓	✓	✓	✓	-	-	-	-
Total (rating)	8.0	7.5	7.7	7.7	5.2	4.7	4.4	2.8	3.1

Figura 17b . Comparación SDKs de Realidad Aumentada. (Fuente: <https://thinkmobiles.com/blog/best-ar-sdk-review/>)

4.5.5. Android Studio



Figura 18. Logo de Android Studio. (Fuente:

https://commons.wikimedia.org/wiki/File:Android_Studio_icon.svg)

Android Studio [18] es el entorno de desarrollo integrado oficial para la plataforma Android. Con él podemos desarrollar aplicaciones nativas para dispositivos Android. Se desarrolla en lenguaje Java.

Algunas de las características de este entorno son:

- Renderizado en tiempo real
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
- Refactorización específica de Android y arreglos rápidos.
- Un editor de diseño enriquecido que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Soporte para programar aplicaciones para Android Wear.
- Soporte integrado para Google Cloud Platform, que permite la integración con Google Cloud Messaging y App Engine.
- Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones.

Se trata de una muy buena herramienta para desarrollar en Android y con la que la parte de diseño se nos haría muy fácil ya que dispone de un editor con el que podemos diseñar de manera rápida y sin programar código. Además, al ser nativa, la aplicación estaría optimizada

para dispositivos Android obteniendo un mayor rendimiento que usando alguna tecnología híbrida.

4.5.6. React Native

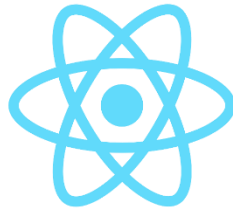


Figura 19. Logo de React Native. (Fuente: <https://commons.wikimedia.org/wiki/File:React-icon.svg>)

React Native [19] es una de las tecnologías más de moda en este momento. Se trata de un framework para desarrollar aplicaciones móviles nativas usando JavaScript.

Hasta hace bien poco, había dos enfoques muy diferentes para desarrollar apps y poder publicarlas en las tiendas oficiales: desarrollos en plataformas nativas (Android-Java, los-ObjectiveC/Swift) y desarrollos en plataforma web incrustada dentro una webview, o Web App (Cordova, Ionic, ...).

Al desarrollar una app en React Native, el código que escribimos es el mismo para las versiones de Android y iOS. Sólo nos tenemos que preocupar de la lógica de negocio y de maquetar una única vez. React Native mantiene los componentes nativos para cada plataforma.

Es una gran ventaja el poder tener en dos plataformas con un mismo desarrollo, pero no todo son buenas noticias. Con esta tecnología obtenemos aplicaciones más pesadas.

4.5.7. Ionic



Fuente 20. Logo de Ionic. (Fuente: <https://commons.wikimedia.org/wiki/File:Logolonic.png>)

Ionic [20] es una herramienta, gratuita y open source, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS. Está construido con Sass y optimizado con AngularJS. Como principales características tenemos:

- Alto rendimiento: hecho para ser rápido gracias a la mínima manipulación del DOM, sin jQuery y con aceleraciones de transiciones por hardware.
- AngularJS y Ionic: Ionic utiliza AngularJS con el fin de crear un marco más adecuado para desarrollar aplicaciones ricas y robustas. Su arquitectura central es robusta y seria para el desarrollo de aplicaciones. Trabaja perfectamente con AngularJS.
- Centro nativo: Ionic se inspira en las SDK de desarrollo móviles nativos más populares, por lo que es fácil de entender para los desarrolladores nativos.
- Un potente CLI: con un sólo comando podrás crear, construir, probar y compilar tus aplicaciones en cualquier plataforma.
- Bonito diseño: Ionic ha sido diseñado para poder trabajar con todos los dispositivos móviles actuales. Con muchos componentes usados en móviles, tipografía, elementos interactivos, etc.

5. Objetivos

Los objetivos definen cuales son las metas de nuestro proyecto. A partir de estos objetivos se dispondrán después recursos, diseños y tomas de decisión. La consecución de los objetivos supondrá la superación de las dificultades o problemas inicialmente planteados.

Estos objetivos queremos que tengan una serie de características. La estrategia que vamos a seguir es que cumplan el principio SMART propuesto por Doran [23], según la cual cada objetivo debe ser:

- Specific (Específico): debemos especificar y concretar al máximo cada uno de nuestros objetivos.
- Measurable (Medible): medir objetivos es algo que puede resultar complicado, pero tiene que quedar muy claro cuáles son nuestras metas. Solo de esa manera podremos cuantificar si alcanzamos los resultados esperados en función de los indicadores que establezcamos para cada uno de ellos.
- Attainable (Alcanzable): nuestro objetivo ha de ser alcanzable dentro del contexto en que nos encontramos.

- Realistic (Realista): el objetivo debe ser realista y estar orientado a resultados objetivos y concretos.
- Timely (Tiempo): todo objetivo tiene que tener un intervalo temporal. Debemos marcarnos un plazo de tiempo para cada objetivo.

Lo que queremos conseguir con nuestra aplicación es facilitar y fomentar el descubrimiento del Parque de Las Lagunas de La Mata ofreciendo a los usuarios una experiencia diferente a través de una visita virtual. Además, queremos que aquellos que no tienen idea de la existencia de este parque o carecen de interés por visitarlo, lleguen a conocerlo a fondo con ayuda de nuestro proyecto, evitando que puedan perderse en el recorrido. Esto queremos realizarlo siguiendo una serie de coordenadas que nos guiarán por el recorrido del parque. Todo ello con el fin de conseguir que el usuario obtenga una experiencia agradable con el uso de esta aplicación.

Por tanto, como objetivos concretos tenemos los siguientes:

- Investigar y analizar proyectos similares
- Investigar el uso y aplicación de la realidad aumentada en aplicaciones móviles y desarrollarla en la aplicación
- Investigar y desarrollar el uso de geolocalización en la aplicación
- Estudiar y seleccionar las tecnologías adecuadas para la realización del proyecto
- Establecer los requisitos mínimos que debe tener la aplicación teniendo en cuenta el diseño, la usabilidad y accesibilidad
- Diseñar y desarrollar la aplicación de acuerdo con los requisitos establecidos
- Conseguir un producto acabado

Por último, como fecha marcada para la consecución de los objetivos tenemos el 28 de enero, día en que la memoria del proyecto debe estar acabada y la aplicación debe ser un producto acabado cumpliendo con los requisitos que establezcamos.

6. Metodología

A lo largo del grado hemos estudiado diversas metodologías para el desarrollo de software. Una metodología representa el camino a seguir para desarrollar software de manera sistemática, describe cómo se organiza un proyecto, establece el orden en el que la mayoría de las actividades tienen que realizarse y los enlaces entre ellas e indica cómo tienen que realizarse las tareas proporcionando las herramientas concretas. Todo esto tiene como objetivo obtener mejores aplicaciones, un mejor proceso de desarrollo y establecer un proceso estándar en una organización.

Existen diversas metodologías que van desde el desarrollo estructurado (análisis, diseño y programación estructurados, modularidad, etc.) hasta las metodologías ágiles (generación de código con ciclos muy cortos de desarrollo, equipos pequeños e involucra al cliente, aceptar cambios en los requisitos, entregas con software funcional frecuentemente) pasando por el desarrollo orientado a objetos (trata funcionalidad y datos de forma conjunta, conjunto de objetos que interactúan entre sí, se centra en el dominio de la aplicación y no tanto en su representación en un lenguaje de programación, etc.).

Hoy en día son las metodologías ágiles las más usadas. Las más conocidas entre ellas son: XP (eXtreme Programming), SCRUM, Kanban y RAD (Rapid Application Development).

A la hora de la elección de una metodología, los principales factores que hay que valorar son: el tamaño y la estructura de la organización y el tipo de aplicación a desarrollar. A continuación vamos a analizar las características de las metodologías ágiles en función de si el proyecto es más prescriptivo o más adaptativo.

Tabla 2. Comparación básica de metodologías ágiles

	Adaptativo	Prescriptivo
Prioridad del negocio	Valor	Cumplimiento
Estabilidad de requisitos	Entorno inestable	Entorno estable
Rigidez del producto	Modificable	Difícil de modificar
Coste del prototipado	Bajo	Alto
Criticidad del sistema	Baja	Alta
Tamaño del equipo	Pequeño	Grande

Queda claro que nuestro proyecto es más adaptativo, por tanto, vamos a buscar una metodología más adaptativa, como puede ser SCRUM o Kanban, en vez de por ejemplo RUP, que es más prescriptiva.

SCRUM tiene como características: método iterativo, equipo autodirigido y autoorganizado, iteraciones de no más de 30 días, incrementos funcionales por cada iteración, equipos multifuncionales. Por tanto, va a ser esta la elegida, ya que en nuestro caso se trata de una sola persona la cual se ocupa del proyecto.

La manera de trabajar con el desarrollo de la aplicación comenzará una vez la memoria del proyecto esté casi terminada, para posteriormente desarrollar la aplicación obteniendo un producto funcional en cada iteración e incrementando estas funcionalidades hasta el final de proyecto.

Se basará en diversos sprints, los cuales tendrán una duración de una semana. En el primer día se definirán las tareas a desarrollar esa semana y se valorarán las del sprint anterior.

Para gestionar el proyecto usaremos la herramienta Trello [21]. La elección está motivada por la facilidad de uso. También se ha valorado Microsoft Project, el cual he descartado por su mayor complejidad y, en mi opinión, su orientación hacia equipos de varias personas. Además, JIRA también ha sido valorada, pero por no ser gratuita ha sido descartada.

7. Análisis y especificación

En esta sección nos centramos en el problema concreto que queremos solucionar y cuál va a ser el alcance de la solución a desarrollar. Para llegar a ello dependemos de muchos factores como por ejemplo los usuarios que lo utilizarán, la población a la que está orientada, etc. Para no dejarnos elementos sin definir es recomendable utilizar alguna herramienta de análisis y especificación, por eso, vamos a usar el estándar IEEE 830, el cual hemos utilizado ya en la carrera.

El IEEE 830 nos facilita una definición sistemática de aquellos aspectos que son necesarios tener en cuenta a la hora de diseñar nuestra solución. Ahora sí que se definirán aspectos relativos a requisitos de rendimiento, de capacidad, funcionales, de interfaz con sistemas de terceros, limitaciones, etc.

El resultado de esta sección será una lista de requerimientos funcionales y no funcionales deseados en forma de tablas. Estos requerimientos deben estar identificados con un identificador único de forma que después, en la fase de diseño, podamos indicar claramente como están siendo resueltos. Los requerimientos funcionales los vamos a identificar como RF1, RF2, etc., mientras que los no funcionales los identificaremos con RNF1, RNF2, etc.

Sabiendo la manera en a que vamos a trabajar los requisitos, vamos a definirlos a continuación:

7.1. Requisitos funcionales

Antes de identificar los requisitos vamos a hacer una descripción de lo que va a ser la aplicación.

Queremos desarrollar una aplicación móvil en la que informemos al turista sobre el Parque Natural de las Lagunas de La Mata de Torrevieja y le permitamos hacer una visita guiada por hasta 54 puntos, que es donde antiguamente se encontraban los mojones, como hemos comentado en el apartado Antecedentes dentro del Estado del arte.

La manera en la que queremos hacer lo citado es utilizando la realidad aumentada, de modo que, en cada uno de los 54 puntos de la ruta, el usuario con la ayuda de la aplicación pueda ver en realidad aumentada un modelo 3D de lo que antiguamente fuera el mojón en cuestión.

Además, queremos también informar sobre diversos aspectos como la historia, la flora, fauna, etc., y mostrar un mapa en el que se vean los 54 puntos además de la ubicación del usuario para evitar que se desoriente.

Los usuarios de la aplicación se espera que sean tanto turistas como ciudadanos de Torreveja y cercanías. Gente de todas las edades.

7.1.1. Requisito funcional 1

Número de requisito	RF1
Nombre de requisito	Menú
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación
Prioridad del requisito	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

En la interfaz principal dispondremos de un menú de tres botones donde cada uno nos llevará a una funcionalidad distinta de la aplicación.

7.1.2. Requisito funcional 2

Número de requisito	RF2
Nombre de requisito	Mapa con las ubicaciones de la ruta
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación
Prioridad del requisito	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Un apartado de la aplicación será en el que se muestre el mapa ampliado de la zona de manera que podamos ver las distintas ubicaciones para realizar la ruta.

7.1.3. Requisito funcional 3

Número de requisito	RF3
Nombre de requisito	Mostrar ubicación del dispositivo en el mapa
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación
Prioridad del requisito	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Se mostrará la ubicación actual del dispositivo para conseguir que el usuario se oriente a la hora de realizar la ruta.

7.1.4. Requisito funcional 4

Número de requisito	RF4
Nombre de requisito	Centrar el mapa hacia donde está el dispositivo
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación
Prioridad del requisito	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Este botón al tocarlo centrará la vista de mapa hacia donde se encuentre el dispositivo, algo que ayuda a orientarnos.

7.1.5. Requisito funcional 5

Número de requisito	RF5
Nombre de requisito	Posibilidad de desplazarnos en el mapa
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación
Prioridad del requisito	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Usando los dos dedos podremos ampliar y disminuir el zoom del mapa. Haciendo el gesto de pellizcar alejaremos el zoom y al contrario acercaremos el mapa. También podremos arrastrar el dedo en cualquier dirección para desplazar el foco del mapa.

7.1.6. Requisito funcional 6

Número de requisito	RF6
Nombre de requisito	Marcadores clasificados
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación
Prioridad del requisito	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja

Al pulsar cada marcador del mapa aparecerá un título que nos dirá que número corresponde a ese marcador, los cuales estarán ordenados.

7.1.7. Requisito funcional 7

Número de requisito	RF7
Nombre de requisito	Visor de Realidad Aumentada
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación

Prioridad del requisito	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
-------------------------	--	--------------------------------	-------------------------------

Otro apartado estará dedicado a la vista de Realidad Aumentada. En él se activará la vista de cámara del dispositivo y cuando nos encontremos cerca del punto de interés, se mostrará el modelo 3D del mojón.

7.1.8. Requisito funcional 8

Número de requisito	RF9
Nombre de requisito	Información general sobre el Parque Natural
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación
Prioridad del requisito	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

En forma de texto y alguna imagen, explicaremos datos de interés acerca del lugar y un poco de su historia.

7.1.9. Requisito funcional 9

Número de requisito	RF10
Nombre de requisito	Navegación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Aplicación
Prioridad del requisito	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja

Posibilidad de navegar por todas las pantallas de la aplicación.

7.2. Requisitos no funcionales

A continuación, vamos a redactar los requisitos no funcionales del mismo modo que hemos hecho anteriormente con los funcionales. Se trata de requisitos no relacionados directamente con la funcionalidad del sistema, sino con la mantenibilidad del mismo, como puede ser su flexibilidad, su capacidad, su modularidad, etc.

7.2.1. Requisito no funcional 1

Número de requisito	RNF1
Nombre de requisito	Aplicación para la plataforma Android
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Aplicación

Prioridad del requisito	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
-------------------------	-------------------------------	--------------------------------	--

El motivo de desarrollar la aplicación nativa y no híbrida es porque obtendremos un mejor rendimiento, y el que sea Android y no iOS es simplemente porque no disponemos de acceso a dispositivos iOS. Además, podríamos desarrollar la aplicación nativa con React Native para ambas plataformas, pero el hecho de que de esta manera la aplicación ocupe más espacio ha sido el motivo para descartar esta opción. Por último, el deseo propio de realizar un desarrollo nativo y aprender este tipo de desarrollo es otro motivo para la toma de esta decisión.

7.2.2. Requisito no funcional 2

Número de requisito	RNF2
Nombre de requisito	Diseño amigable
Tipo	<input checked="" type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Aplicación
Prioridad del requisito	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja

La aplicación se adaptará a los estándares del diseño móvil, para mejorar así la experiencia de usuario. Esto, vamos detallarlo en el apartado de diseño de interfaces haciendo referencia al apartado de experiencia de usuario, que es donde estudiaremos la usabilidad, accesibilidad y todo lo relacionado con la experiencia de usuario.

8. Diseño

El capítulo de diseño es con diferencia el más importante del TFG. Es la etapa previa al desarrollo. En este punto vamos a diseñar la solución de forma que dé respuesta a todos y cada uno de los requerimientos funcionales y no funcionales anteriormente establecidos.

El diseño de soluciones es muy complejo y en él intervienen aspectos a distintos niveles de un proyecto. Por tanto, lo que vamos a hacer es agrupar el diseño en apartados en función de la materia que se está definiendo, concentrando así todo lo relativo a ese contexto en un único apartado.

8.1. Diseño arquitectura tecnológica

En este apartado vamos a describir la arquitectura tecnológica elegida para el desarrollo del proyecto junto con una representación gráfica de la misma.

La plataforma elegida es Android y vamos a realizar un desarrollo nativo utilizando el IDE Android Studio sobre este sistema operativo, en el cual vamos a programar en Java para las funcionalidades, XML para el diseño de estas y JSON para cargar las coordenadas en la vista de realidad aumentada.

Utilizaremos el framework Wikitude Javascript API para el desarrollo de toda la parte de realidad aumentada. Esta herramienta utiliza tecnologías web como son HTML, CSS y JavaScript.

Para mostrar un mapa al usuario con la información de la ruta utilizaremos la API de Google Maps para Android.

Las comunicaciones de la parte cliente con los servicios web deben realizarse mediante conexión 3G/4G o WiFi y para la geolocalización existe la posibilidad de bastarnos con la conexión GPS.

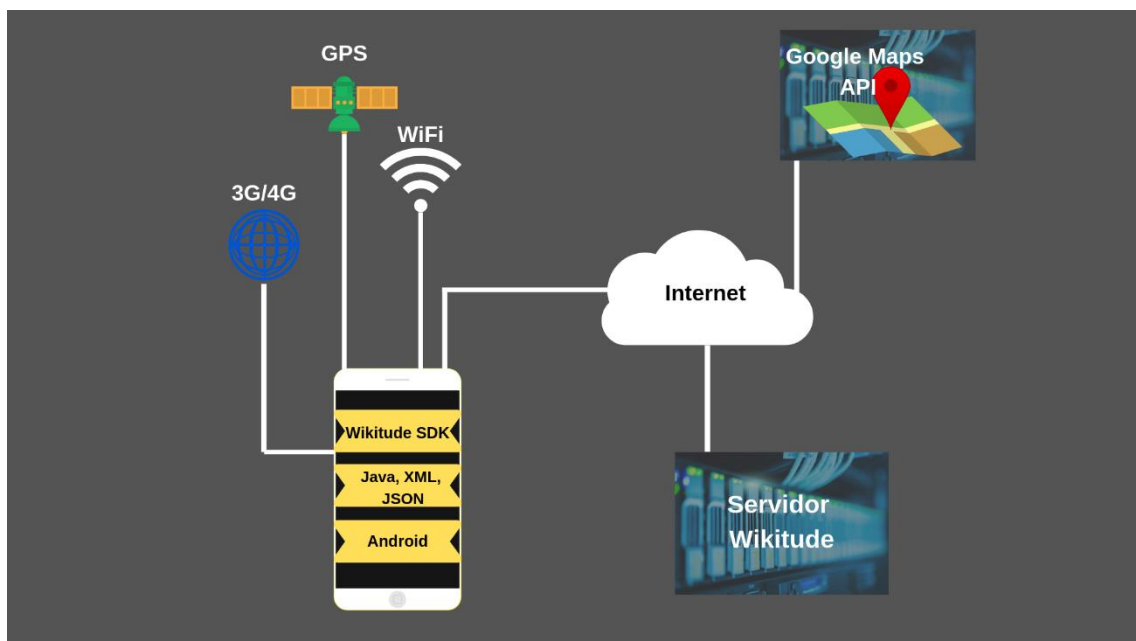


Figura 21. Stack tecnológico del proyecto
(Fuente propia)

8.2. Experiencia de Usuario

La función que tiene realizar un estudio para la experiencia de usuario es conseguir una tecnología amigable, satisfactoria, fácil de usar y, por tanto, realmente útil. En este apartado vamos a estudiar qué aspectos son clave a la hora de hacer un diseño para, en el siguiente

apartado, realizar un correcto diseño de interfaces. Para ello vamos a servirnos del libro de Hassan Montero [25].

Hay que tener en cuenta una serie de conceptos para conseguir una buena experiencia de usuario:

- Usabilidad: se refiere a la facilidad de uso de un producto. Para que esto suceda debe ser fácil de aprender su uso, eficiente a la hora de usarlo, tener la cualidad de ser recordado, ser eficaz a la hora de realizar una tarea y, por último, ser útil y satisfactorio para el usuario.
- Accesibilidad: se refiere a la posibilidad de que pueda ser usado sin problemas por el mayor número de personas posibles. Hay que tener en cuenta las distintas discapacidades, conocimientos, habilidades y experiencias de los distintos tipos de personas. Por tanto, un producto accesible debe ser perceptible, operable, comprensible y robusto.
- Arquitectura de información: una correcta arquitectura de información es aquella que permite al usuario encontrar la información que necesita; que facilita la navegación y comprensión del producto; y que motiva al usuario a explorar los contenidos y funcionalidades.
- Diseño centrado en el usuario: se define conceptualmente el producto en base a la investigación de la audiencia objetiva, se diseña partiendo desde lo más general hasta lo más específico para luego evaluar este diseño, implementarlo y monitorizarlo.
- Interacción: según el modelo propuesto por Norman (1988), podemos considerar la interacción como un proceso iterativo y cíclico, el cual podemos dividir en tres etapas:
 - Detectar qué quiere lograr el usuario
 - Ejecución: formular su intención, especificar la acción y ejecutar la acción
 - Evaluación: comparar lo ocurrido con lo que el usuario quería que ocurriera
- Estilos de interacción: formas en las que el usuario puede interactuar con el producto, ya sean menús de selección, formularios o cualquier otro modo.
- Affordance: lo define como aquellas propiedades perceptibles del objeto que determinan cómo puede ser usado, es decir, que tenga carácter autoexplicativo.
- Modelos mentales: hace referencia a representaciones internas de una realidad externa, representaciones que somos capaces de construir a partir de nuestras experiencias.

- Necesidades y estrategias de búsqueda de información: debemos cuestionarnos el propio comportamiento interactivo de los usuarios al buscar información, qué tipos de necesidades tienen y qué estrategias usan para resolverlas.
- Relación esfuerzo-beneficio: hay que evitar la frustración del usuario. Cuanto mayor sea el beneficio que obtenga el usuario, mayor será su resistencia a la frustración y su tolerancia al esfuerzo de uso.

Además de los conceptos anteriores, debemos tener en cuenta una serie de principios de diseño como son los siguientes:

- Clasificación
- Color
- Eficiencia
- Error humano
- Estética
- Fotografías
- Gestalt
- Iconos
- Inteligencia colectiva
- Jerarquía visual
- Legibilidad e inteligibilidad
- Ley de Fitts
- Mapeo natural
- Ordenación
- Relevancia
- Taxonomías
- Toma de decisiones
- Visibilidad y retroalimentación

Teniendo en cuenta todo esto, vamos a afrontar el diseño de las interfaces de la aplicación en el siguiente apartado.

8.3. Diseño de interfaces

El diseño de interfaces es la parte donde los requisitos funcionales y no funcionales se materializan, es decir, es la parte de la ingeniería donde se desarrollan los mecanismos físicos mediante los cuales los distintos usuarios podrán interactuar con nuestro sistema.

Para realizar el diseño de las interfaces vamos a tener en cuenta las distintas prácticas de usabilidad, accesibilidad y experiencia de usuario que se han estudiado en el apartado anterior.

La definición de las interfaces nos permite la toma de decisiones en etapas de diseño y no en desarrollo, lo cual agilizará esta última fase. Este diseño de interfaces lo podemos realizar con mayor o menor exactitud, lo cual facilitará más o menos el desarrollo. Para conseguir un grado adecuado de fidelidad en los diseños usaremos:

- Wireframes: es la primera representación de nuestra propuesta de solución, normalmente utiliza iconos y símbolos entendibles y está orientado a reflejar de una forma poco refinada las funcionalidades del sistema y dónde encajan más o menos los elementos que lo forman. Con ellos podemos hacernos una idea de cómo serán las funcionalidades sin llegar a concretar todo su funcionamiento. Para realizarlos vamos a usar la herramienta Balsamiq [26], que es una aplicación gráfica para creación de mockups de interfaz de usuario y constructor de estructuras de sitios web y aplicaciones móviles. Le permite al diseñador organizar widgets preconstruidos usando un editor de arrastrar y soltar.
- Mockups: se trata de una vista exacta de cómo será la interfaz y, a diferencia de los wireframes, aquí ya indicamos el estado final de los elementos además de su posición, colores, fuentes, tamaños, etc. En ellos ya se indica el orden de los procesos. Por tanto, son más concretos y cercanos al desarrollo final. Para desarrollar los mockups vamos a usar la herramienta Justinmind [27], que es una herramienta de autor para prototipos de aplicaciones web y móviles y wireframes de sitios web de alta fidelidad.

A continuación, vamos a mostrar los diseños planteados para la creación de nuestra aplicación. Para cada prototipo mostraremos el wireframe y mockup conjuntamente, de modo que veremos un prototipo sencillo, que se realizó en una primera iteración de la fase de diseño, y otro más complejo, que fue realizado en una segunda iteración. Estos diseños están orientados a móvil y, en concreto, de sistema operativo Android (RNF1). Junto a cada imagen daremos

una explicación de su funcionalidad y diremos de qué pantalla procede cada prototipo así como a la que nos lleva realizando según qué acciones. Además, haremos referencia a lo estudiado en el apartado de Experiencia de usuario profundizando en los conceptos del libro de Hassan Montero [25].

En los siguientes diseños debemos cubrir los requisitos definidos anteriormente. Para que lo tengamos claro, cuando estemos cubriendo un requisito lo nombraremos con la nomenclatura definida en el apartado de Análisis y Especificación.

8.3.1. Interfaz principal



Figura 22. Pantalla principal de la aplicación. (Fuente propia)

La figura 22 se trata de la primera pantalla de la aplicación. En ella podemos ver lo siguiente:

Barra de navegación

La cual se encontrará en todas las pantallas de la aplicación. En esta pantalla principal sólo contiene el nombre de la aplicación, pero en las demás contendrá el nombre de la funcionalidad (Mapa, AR o Info), y, a su izquierda, en la esquina superior izquierda, una flecha para volver atrás, que a su vez es un botón, que al pulsarlo nos devuelve a la pantalla anterior.

El poner una flecha como botón es porque la inmensa mayoría de los usuarios ha usado aplicaciones digitales donde con una flecha se vuelve atrás. Esto se ha adoptado como estándar de la funcionalidad “volver atrás” y creemos que es intuitivo. Algo similar ocurre con la barra de navegación, que adoptan muchas aplicaciones móviles. Pero el motivo de la barra de navegación no es sólo ese, sino que aporta información al usuario sobre la parte de la aplicación en la que se encuentra

Imagen

Nos sitúa en el lugar. Las fotografías tienen una gran capacidad para comunicar y evocar emociones, y un impacto directo en la estética del diseño.

Texto breve

Nos ayuda a entender la forma en la que trabaja la aplicación y evita que el usuario se sienta perdido a la hora de usarla.

Los textos deben tener legibilidad e inteligibilidad. La legibilidad de los textos tiene un enorme impacto en la accesibilidad del producto, la inteligibilidad lo tiene en su usabilidad.

La legibilidad la conseguimos con características como el contraste entre colores que mencionamos en el apartado Colores, un buen interlineado, buen tamaño de fuente (suficiente para que alguien con problema visual pueda leer el texto) y una fuente tipográfica que se ve clara.

La inteligibilidad la conseguimos siendo concisos, ya que apenas son dos frases.

Menú

RF1. Tres iconos que a la vez son botones, los cuales hemos clasificado en orientación a la tarea, puesto que son funcionalidades. Aquí estamos cubriendo el principio de diseño de clasificación además de una correcta arquitectura de información, al organizar la información con el objetivo de permitir al usuario encontrar su vía de navegación hacia el conocimiento.

Estos botones al pulsarlos nos llevan a tres distintas funcionalidades. El botón mapa nos lleva al prototipo de la figura 23, el botón AR nos lleva al prototipo de la figura 24 y el botón Info nos lleva al prototipo de la figura 25. El poner iconos y texto en los botones hace que cubramos las necesidades tanto sintácticas como semánticas del usuario, ya que si el usuario no conoce el nombre de la opción (sintaxis), tiene un icono que le ayuda a saber que es visualmente. Además, conseguimos un mapeo natural, que se produce cuando el usuario es capaz de predecir la función de cada elemento interactivo de forma natural.

Este diseño de botones donde al pulsar un icono de una cámara se te abre la cámara o al pulsar el icono de un mapa te lleva a un mapa responde al concepto 'affordance', introducido por Norman [28], quien lo define como aquellas propiedades perceptibles del objeto que determinan cómo puede ser usado. Es decir, aquellas propiedades que le confieren un aspecto autoexplicativo, haciendo obvio y explícito cómo debe ser usado y con qué objetivos.

En cuanto al tamaño, vamos a tener en cuenta la Ley de Fitts. Esta ley dice que cuanto mayor sea la precisión motriz que exijamos al usuario para accionar un elemento interactivo, mayor será el esfuerzo y la tasa de error. Por tanto, hemos elegido un tamaño que no exija una alta precisión y que tampoco ocupe gran superficie de pantalla, lo que nos haría perder UX (experiencia de usuario) por otra parte. Aplicando esto también conseguimos una buena relación esfuerzo-beneficio.

Colores: La elección de estos colores es por el motivo de que esta aplicación está hecha para ser consumida en las Lagunas de La Mata, lugar al aire libre, y a la luz del día. Por tanto, la luz del sol dificulta la visión de las pantallas de los dispositivos. Entonces el poner colores vivos y conseguir altos contrastes entre colores hace que aumente la accesibilidad de este producto. Por otra parte, limitar el número de colores diferentes utilizados a aquellos que el ojo humano puede procesar de un vistazo (alrededor de 5) es una buena práctica de diseño, y nosotros en las interfaces usamos cuatro (blanco, negro, amarillo y azul).

8.3.2. Interfaz Mapa

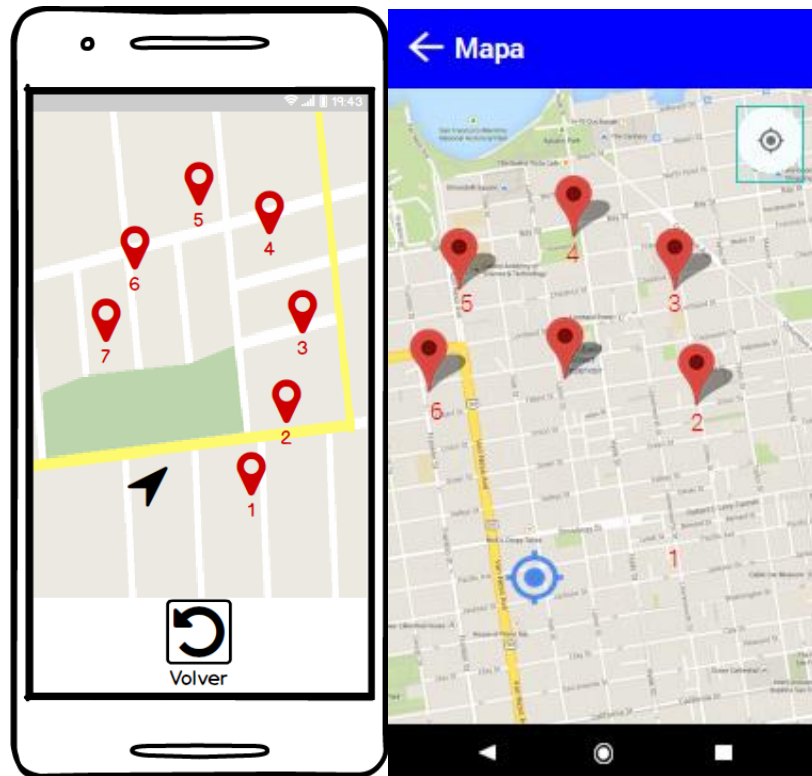


Figura 23. Prototipo de la funcionalidad Mapa. (Fuente propia)

Al tocar el botón mapa de la figura 22 nos lleva a esta pantalla. Se trata de un mapa que nos mostrará nuestra ubicación actual como un círculo azul y los 54 puntos de la ruta en vez de los 6 o 7 del prototipo. Para mostrar el mapa usaremos la API de Google Maps [29]. Al tocar la flecha de arriba a la izquierda en el prototipo de la derecha nos volverá a llevar a la pantalla de inicio (figura 22).

En cada uno de los marcadores, si lo tocamos nos aparece un pequeño rótulo indicando el número de punto que es, ya que hemos seguido el orden de la ruta, que empieza por el punto más cercano a la costa y rodea la laguna.

También podemos hacer más o menos zoom. Con el gesto de pellizcar alejamos y con el inverso acercamos. Esto es considerado un estándar también, al igual que hemos comentado con el botón de volver.

En la esquina superior derecha, justo debajo de la barra de navegación, tenemos un botón que al pulsarlo nos centra el foco del mapa en la ubicación del dispositivo (círculo azul).

En cuanto a la estrategia de diseño, para esta interfaz, no existe una estrategia propia, ya que estamos usando la API de Google Maps [29] y fueron ellos quienes estudiaron y aplicaron un diseño con el objetivo de conseguir una buena experiencia de usuario.

Con esto estamos cubriendo el RF2, RF3, RF4, RF5, RF6 y RF9.

8.3.3. Interfaz AR



Figura 24. Vista AR usando la cámara del dispositivo. (Fuente propia)

Al tocar el botón AR de la pantalla principal (figura 22) nos llevará a esta vista donde se activará la cámara. Al situarnos geográficamente cerca de cada una de las 54 localizaciones de la ruta, siempre y cuando estemos apuntando en la dirección y sentido correctos del punto en cuestión, nos mostrará el modelo 3D del mojón el cual vamos a modelar nosotros mismos. La flecha de arriba a la izquierda, al tocarla, nos llevará de nuevo a la pantalla de inicio (figura 22).

Con este prototipo estamos cubriendo los RF7 y RF9.

8.3.4. Interfaz Info



Figura 25. Vista del prototipo de la funcionalidad Info. (Fuente propia)

A esta vista se llegará pulsando el botón Info de la pantalla principal (figura 22). Aquí mostraremos información acerca del Parque Natural de Las Salinas de La Mata. Esta pantalla contendrá una barra de scroll vertical con la que arrastrando hacia arriba o abajo veremos más información y un botón de volver en la esquina superior izquierda que nos llevará de nueva a la pantalla principal.

Contenido de texto

El contenido que mostraremos está relacionado con las Lagunas. Parte de este contenido será el descrito en el apartado de Antecedentes del Estado del arte.

En cuanto a la legibilidad e inteligibilidad, vamos a seguir la misma tónica de lo que mencionamos en la definición de la interfaz principal.

Imágenes

Las imágenes se situarán entre párrafos de texto de manera ilustrativa, para que el usuario conozca más del lugar y facilita al usuario la tarea de conocer el lugar, ambas características del principio de diseño relativo a las fotografías.

Con todo lo mostrado en esta interfaz cubrimos los requisitos RF8 y RF9.

8.4. Guías de estilos

En este pequeño apartado vamos a hablar de los estilos que incorpora nuestra aplicación. Vamos a definir fuentes, tipos, tamaños y colores de los distintos elementos.

La fuente que utilizamos tanto en la interfaz principal como en la interfaz Info es Lusitana. En la principal a tamaño 16sp y en Info a 20sp. Pero en la barra de acción la fuente es Roboto y tamaño 15px para todas las pantallas.

En cuanto a los iconos, son predeterminados de Android, es decir, que no incluiremos ningún paquete de iconos personalizado.

Los colores que tenemos en la aplicación son los que podemos ver en la siguiente figura:



*Figura 25b. Paleta de colores de la aplicación
(Fuente propia)*

8.5. Diseño de pruebas y validación

Por el tipo de proyecto, es evidente que debemos probar el funcionamiento en las localizaciones en cuestión. Además de esto, vamos a comprobar el correcto funcionamiento de la aplicación en distintos dispositivos.

Para las pruebas de geolocalización, añadiremos varias localizaciones cercanas para ver que la función de realidad aumentada se activa correctamente en cada una de ellas mostrando el modelo 3D del mojón. De este modo luego sólo tendremos que cambiar las coordenadas y añadir los 54 puntos de la ruta. También debemos comprobar que el mapa de la API Google Maps muestra la ubicación del dispositivo correctamente.

Para las pruebas en distintos dispositivos, utilizaremos distintos smartphones con sistema operativo Android para probar que el diseño se muestra correctamente y que las respuestas a botones son las esperadas junto con el rendimiento previsto.

9. Implementación

Tras la fase de diseño, comenzamos con la implementación. Aquí es donde vamos a documentar el proceso de desarrollo. Para ello nos adecuamos a la metodología elegida, en nuestro caso SCRUM, por lo que detallaremos el proceso iterativo característico de esta metodología puesto en práctica en nuestro proyecto. Nuestra planificación consta de dos sprints (iteraciones) de dos semanas cada uno en donde en cada uno de ellos tendremos versiones funcionales de la aplicación, algo que es una característica fundamental de la metodología SCRUM.

Vamos a comenzar con los entornos de desarrollo que hemos elegido y los primeros pasos con ellos:

Android Studio

Al principio, tenemos que preparar el entorno de trabajo. En mi caso, que voy a utilizar Android Studio, ya disponía de esta herramienta. Es el entorno oficial para desarrollo en Android y se utiliza lenguaje Java.

La versión de este software que voy a utilizar es la 3.2. Esta versión dispone del JRE 1.8.0 (Java Runtime Environment), que es el conjunto de utilidades que permite la ejecución de programas Java, a diferencia del JDK, también necesario, que es necesario para desarrollar en Java. De esta última también disponemos de la versión 1.8.0.

Wikitude

El otro entorno de trabajo es Wikitude, el cual hemos elegido por sus funcionalidades en cuanto servicios basados en localización. También se ha valorado Vuforia, herramienta muy usada para desarrollar realidad aumentada, el problema es que no dispone de funcionalidades relacionadas con la localización y está muy orientada a ser usada con marcadores.

A la hora de empezar con Wikitude, hemos tenido que valorar que API de la misma utilizar. El resultado ha sido elegir Wikitude JavaScript API en detrimento de Wikitude Native API para Android. El motivo es básicamente el mismo que hemos mencionado en el párrafo anterior con Vuforia, y es que la API nativa no dispone de las funcionalidades de localización.

9.1. Primera iteración

En esta primera iteración vamos a trabajar para conseguir una serie de funcionalidades de la aplicación. Al final de esta, el objetivo es tener todas las actividades con funcionalidad, aunque en algunas de ellas no sea la versión final.

El objetivo es conseguir la actividad inicial tal y como se ve en el diseño; la actividad Mapa completa, con sus 54 marcadores y la ubicación actual del dispositivo mostrándose sobre un mapa de la API de Google Maps; la actividad AR mostrando un icono en nuestra ubicación actual; y, por último, la actividad Info, con un texto y alguna imagen que ya en la segunda iteración perfeccionaremos.

A continuación podemos ver una imagen de la planificación de tareas realizada en Trello:

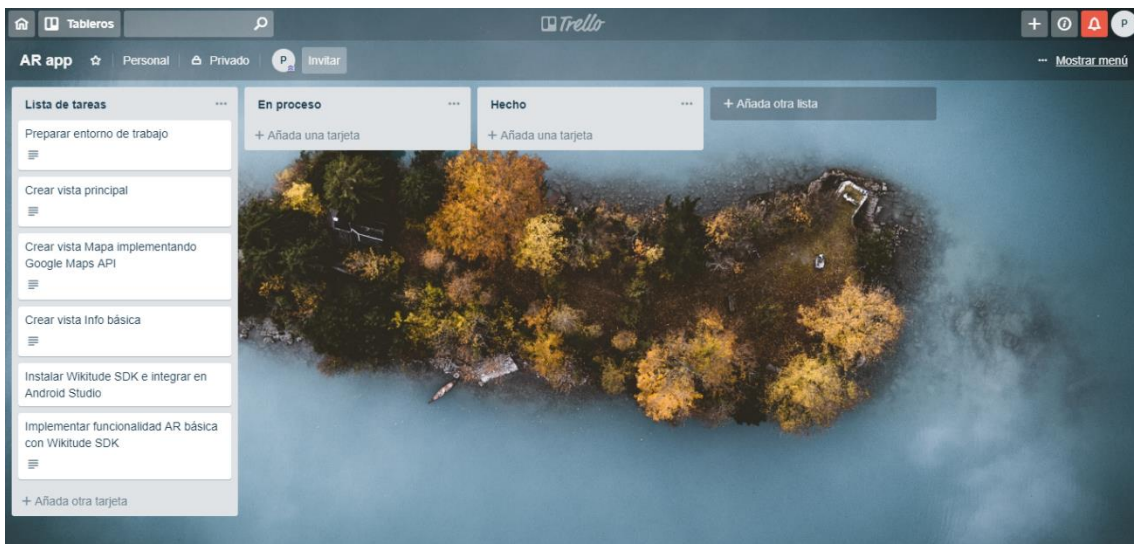


Figura 26. Planificación de tareas de la iteración 1 en Trello. (Fuente propia)

Empezamos el desarrollo

Una vez que iniciamos nuevo proyecto en Android Studio podemos ver la siguiente imagen:

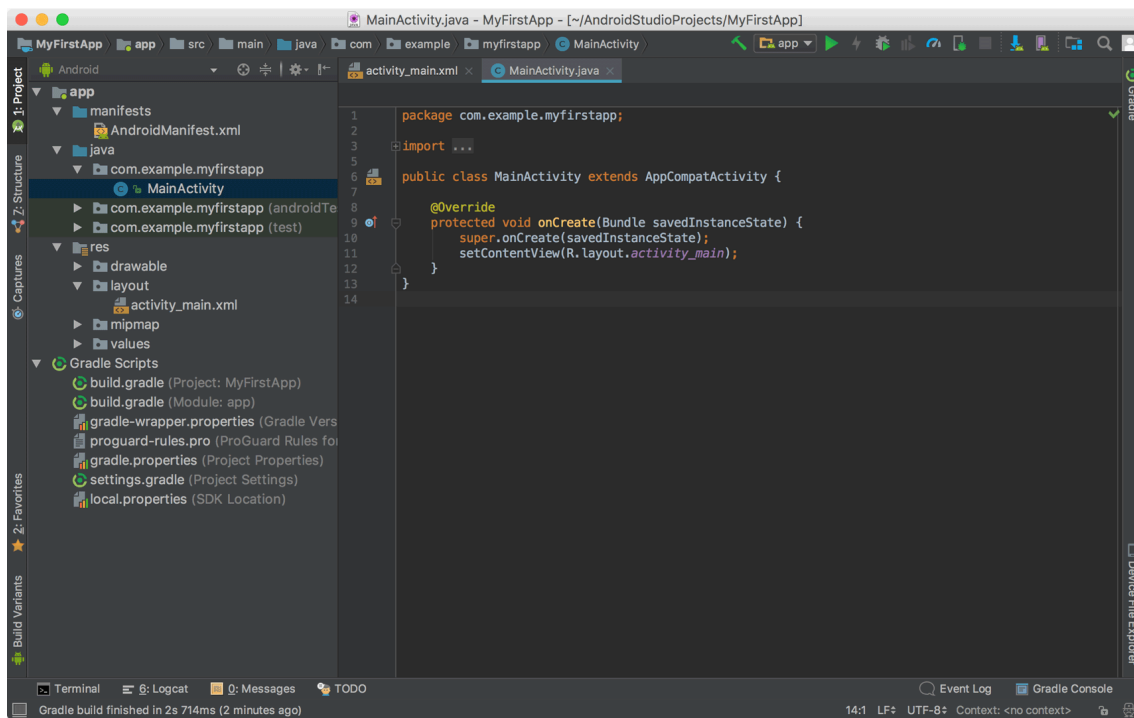


Figura 27. Inicio de un proyecto en Android Studio. (Fuente:

<https://developer.android.com/training/basics/firstapp/creating-project>)

En la figura 27 podemos ver la estructura básica de directorios de un proyecto y los distintos archivos:

- **app > java > com.example.myfirstapp > MainActivity.java**

Esta es la actividad principal (el punto de entrada para nuestra app). Cuando compilamos y ejecutamos la app, el sistema inicia una instancia de esta Activity y carga su diseño. Es en esta carpeta donde se almacenarán todas las clases java de cada una de las actividades.

- **app > res > layout > activity_main.xml**

Este archivo XML define el diseño para la IU (interfaz de usuario) de la actividad. Dentro de esta carpeta layout se encontrarán todos los archivos XML que aportan el diseño de cada una de las actividades.

- **app > manifests > AndroidManifest.xml**

El archivo de manifiesto describe las características fundamentales de la app y define cada uno de sus componentes.

- **Gradle Scripts > build.gradle**

Hay dos archivos con este nombre: uno para el proyecto y otro para el módulo de la app. Cada módulo tiene su propio archivo build.gradle, pero este proyecto por el momento tiene un solo módulo. Trabajaremos principalmente con el archivo build.gradle del módulo para configurar la forma en que las herramientas de Gradle compilan y crean la app.

Estos archivos son los básicos de una app y son los principales con los que vamos a trabajar.

Una vez que hemos descargado el SDK de Wikitude, debemos introducir el archivo wiktudesdk.aar en la carpeta libs de nuestro proyecto. Abrimos el build.gradle de nuestro módulo y añadimos el archivo citado como una dependencia.

Hay que recordar que Wikitude no es open source, pero podemos trabajar con una versión de prueba gratuita que contiene marca de agua. En su web podemos obtener una clave para trabajar con la versión de prueba y eso vamos a hacer.

Primer layout

Un layout es un diseño de una pantalla de la aplicación. Para realizarlos, en Android Studio se usan archivos XML. Además, contamos con Android Studio Layout Editor para facilitarnos esta tarea.

Para crear el layout principal de nuestra aplicación añadiremos: un ImageView, donde introduciremos una imagen, que debemos almacenarla primero en la carpeta drawable, que se sitúa dentro de app/res; un texto con TextView; y tres botones con ImageButton. Al final hemos descartado añadir texto a los botones además del icono ya que se trata de una aplicación con pocos apartados y no creemos que el usuario se pierda aunque no supiera lo que significan los botones. Además, en el texto de la vista principal explicamos de qué va la aplicación.

Este proceso lo repetiremos para las distintas pantallas, variando los elementos a introducir en función del diseño especificado.

Debemos tener en cuenta las distintas pantallas de los dispositivos, por eso haremos los elementos flexibles.

Respuestas a los botones

Aquí trabajamos con los archivos .java y .xml, donde programamos la interacción. En el xml, nos situamos en el editor de diseño y elegiremos el botón que queramos para darle la funcionalidad de que al clickar nos envíe a otra vista. En el java, para que nos lleve a otra actividad al presionar un botón, realizamos un método void con la acción en sí, usando el objeto Intent, que es un objeto que proporciona enlace de tiempo de ejecución entre componentes separados, como dos activity. Esto lo repetiremos con todos los elementos que al pulsarlos nos deben llevar a otra actividad.

Vista mapa

Ahora vamos a desarrollar la vista mapa donde mostraremos los 54 puntos de interés en un mapa estático además de la ubicación del dispositivo.

Para esto vamos a usar la API de Google Maps, por tanto, debemos acceder a Google Cloud Platform y estar registrados. Una vez hecho esto, Google nos proporciona una clave que utilizaremos para el desarrollo. Además, tendremos que instalar Google Play Services SDK dentro del proyecto en Android Studio.

Con todo lo anterior ya empezamos a crear nuestro mapa. En Android Manifest debemos declarar nuestra clave, nuestro número de versión de Google Play Services y los permisos, tanto para la ubicación actual como para las demás funcionalidades del mapa. Ahora, Añadimos un mapView al archivo xml de la actividad y en el archivo java de la misma programamos para que nos muestre un marcador en una localización de prueba.

Una vez que comprobamos que funciona, pasamos a colocar los 54 puntos de interés de la ruta. Como podemos ver, en la figura 28 tenemos las coordenadas de los 54 puntos. Estas son coordenadas UTM, pero Google trabaja con latitudes y longitudes, de modo que tenemos que convertirlas a ese sistema. Para ello usamos una herramienta en la web del Instituto Geográfico Nacional [23] que nos permite convertir las coordenadas. Una vez convertidas, creamos una lista de coordenadas donde las almacenamos y haciendo uso de los métodos de la API de Google Maps añadimos los marcadores y demás funcionalidades.

4.2. Coordenadas de mojones

Mojón	Coordenada	Mojón	Coordenada
1º	X: 706.102 - Y: 4.211.621 *	28º	X: 700.780 - Y: 4.213.371
2º	X: 705.846 - Y: 4.211.539 *	29º	X: 700.528 - Y: 4.213.186
3º	X: 705.458 - Y: 4.211.681	30º	X: 700.117 - Y: 4.212.709
4º	X: 705.200 - Y: 4.211.912	31º	X: 700.247 - Y: 4.212.423
5º	X: 704.875 - Y: 4.212.185	32º	X: 700.648 - Y: 4.211.853
6º	X: 704.645 - Y: 4.212.591	33º	X: 701.300 - Y: 4.211.394
7º	X: 704.708 - Y: 4.212.703	34º	X: 701.653 - Y: 4.211.265
8º	X: 704.485 - Y: 4.212.900	35º	X: 701.995 - Y: 4.211.083
9º	X: 704.514 - Y: 4.213.207	36º	X: 702.298 - Y: 4.211.083
10º	X: 704.605 - Y: 4.213.461	37º	X: 702.527 - Y: 4.210.845
11º	X: 704.746 - Y: 4.213.779	38º	X: 702.995 - Y: 4.210.533
12º	X: 704.614 - Y: 4.213.904	39º	X: 703.290 - Y: 4.210.239
13º	X: 704.356 - Y: 4.214.143	40º	X: 703.548 - Y: 4.210.160
14º	X: 703.968 - Y: 4.213.970	41º	X: 703.704 - Y: 4.209.878
15º	X: 703.909 - Y: 4.213.833	42º	X: 704.026 - Y: 4.209.874
16º	X: 703.688 - Y: 4.213.878	43º	X: 704.244 - Y: 4.209.773
17º	X: 703.520 - Y: 4.213.848	44º	X: 704.459 - Y: 4.209.719
18º	X: 703.382 - Y: 4.213.604	45º	X: 704.518 - Y: 4.209.973
19º	X: 703.214 - Y: 4.213.628	46º	X: 704.647 - Y: 4.209.950
20º	X: 702.866 - Y: 4.213.693	47º	X: 704.814 - Y: 4.210.061
21º	X: 702.329 - Y: 4.213.714	48º	X: 704.922 - Y: 4.210.237
22º	X: 702.237 - Y: 4.213.570	49º	X: 704.672 - Y: 4.210.233
23º	X: 702.032 - Y: 4.213.667	50º	X: 704.620 - Y: 4.210.467
24º	X: 701.798 - Y: 4.213.635	51º	X: 705.057 - Y: 4.210.696
25º	X: 701.610 - Y: 4.213.556	52º	X: 705.459 - Y: 4.210.968
26º	X: 701.366 - Y: 4.213.455	53º	X: 705.778 - Y: 4.211.124 *
27º	X: 701.127 - Y: 4.213.324	54º	X: 705.992 - Y: 4.211.093 *

Figura 28. Datos de los puntos de interés de la ruta. (Fuente propia)

Una vez puestos los marcadores, vamos a añadir el punto azul que marca la ubicación actual del dispositivo. El primer paso es añadir los permisos de localización en tiempo de ejecución.

Vista AR

Esta es la vista de realidad aumentada donde mostraremos el modelo 3D cuando nos encontremos cerca del punto de interés.

Como vamos a usar el framework Wikitude, debemos configurarlo. Para ello, en el archivo build.gradle del módulo de nuestra aplicación introduciremos las dependencias y la clave que Wikitude nos proporciona al registrarnos para poder usar su herramienta. Por otra parte, en el AndroidManifest.xml, añadiremos los permisos para acceder a la cámara, gps y demás elementos que requiere.

Como el SDK de Wikitude no es nativo de Android, lo que hace es desplegar una capa donde se abre la vista de la cámara del dispositivo y manejo de eventos.

Ahora declaramos en el XML de la vista el `architectView`, que es el que manejará la actividad AR. Una vez hecho esto, el primer paso es mostrar la vista de cámara simplemente. Para ello, en la actividad (.java) creamos el objeto `architectView` y una serie de métodos que permitirán a este objeto saber del estado en que se encuentra la aplicación y controlar; un método para lanzar la cámara, donde le damos permiso a la aplicación para abrir la cámara y describimos nuestra clave de Wikitude; otro para que el objeto `architectView` sepa que ha habido un cambio de estado; otro para que el objeto sepa que hemos cerrado; y otro por si hay una interrupción de otra aplicación del móvil, como puede ser una llamada entrante.

Con todo lo anterior hemos conseguido mostrar la vista de cámara. El siguiente paso es mostrar un icono al acercarnos a una ubicación concreta.

Ahora es cuando debemos interactuar con el archivo HTML, que a su vez interactúa con el archivo Javascript (como hemos dicho antes, es un SDK de JavaScript, no nativo), para programar todo lo siguiente acerca de los puntos de interés, control de ubicación, orientación y muestreo de elementos virtuales.

Por otra parte, debemos controlar la localización del dispositivo en todo momento. Esto es algo que Wikitude no controla, por tanto, lo vamos a hacer de forma nativa en la actividad que lanza la experiencia de realidad aumentada. Vamos a usar la clase `LocationProvider` modificada, la que nos recomienda Wikitude en su documentación, ya que la clase nativa de Android nos advierte que puede dar problemas. El motivo es porque nosotros necesitamos un control de localización preciso, ya que queremos mostrar los objetos virtuales cuando nos encontremos en el punto de interés, y como algunos puntos se encuentran cerca de otros podría fallar en estos casos. Con esta clase controlamos la ubicación del dispositivo tanto por GPS como por Internet, de modo que si el GPS no está disponible, por ejemplo, buscaría la posición a través de Internet. Creamos un objeto de dicha clase y en cada uno de los métodos mencionados anteriormente controlamos la localización.

Se supone que con esto debería funcionar el mostrar un icono de prueba en nuestra localización, pero no es así. Resulta que además de los permisos de localización que habíamos dado en el `AndroidManifest.xml`, también debemos dar permisos de localización al crear nuestra actividad, al igual que debemos hacerlo con los permisos de cámara. Una vez hecho esto, la funcionalidad se ejecuta correctamente tal y como podemos ver en la figura 29.



Figura 29. Icono azul rectangular en nuestra posición. (Fuente propia)

Conclusión

En esta primera iteración se han concluido todas las tareas menos la funcionalidad Info. El motivo es porque la funcionalidad de realidad aumentada nos ha llevado algo más de tiempo de lo estimado. Por tanto, esta tarea pasa a la segunda iteración junto con las restantes.

9.2. Segunda iteración

El objetivo aquí es obvio, terminar las funcionalidades que no se acabaron de acuerdo con el diseño realizado previamente. Por tanto, en la vista AR debemos mostrar un modelo 3D (el cual debemos modelar) en las cercanías de cada uno de los 54 puntos y en la vista Info debemos aportar información en forma de texto e imágenes.

A continuación mostramos la planificación de tareas de esta segunda iteración, donde podemos ver etiquetas de distintos colores en las tareas. Las de la iteración 1 en color azul, las de la segunda iteración en color verde y la tarea retrasada de la iteración 1 que pasa a esta iteración con etiqueta naranja.

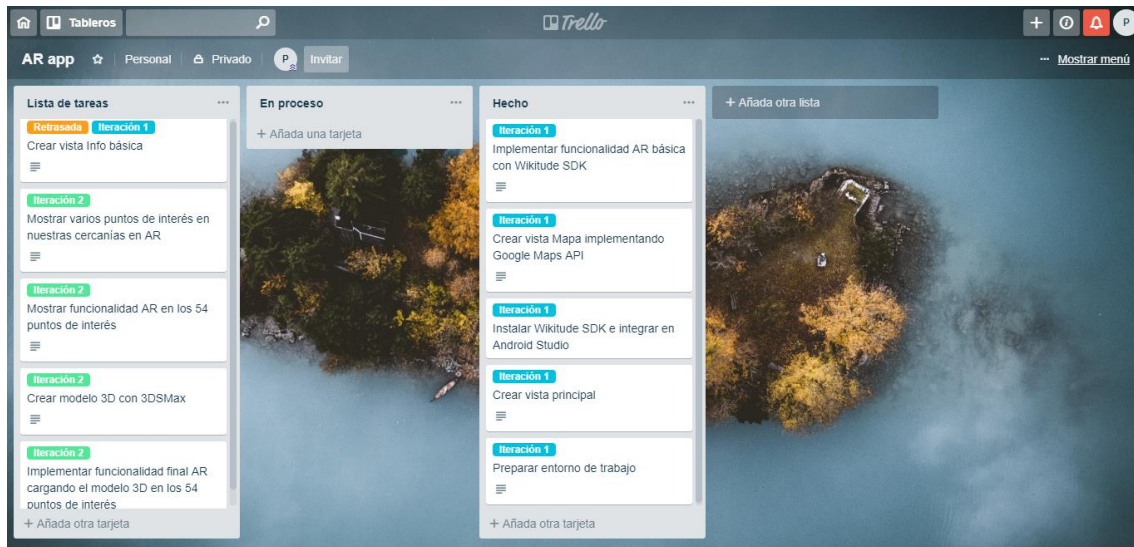


Figura 30. Planificación de tareas de la segunda iteración en Trello. (Fuente propia)

Modelo 3D

El modelo en cuestión lo vamos a realizar con el programa de diseño Autodesk 3DS Max [30]. Lo que queremos es simular un mojón de piedra.

Además del modelado y la texturización debemos tener en cuenta el formato del archivo. Wikitude sólo acepta formato .wt3, al cual podemos convertir nuestro archivo .fbx. Esta conversión la tenemos que hacer con Wikitude 3D Encoder, software que Wikitude pone a disposición de los usuarios en su web junto a los demás paquetes de herramientas.

Se trata de una figura fácil de modelar, pero el problema viene con las texturas y la exportación de formato. Para las texturas, sólo podemos colocar una simple porque el formato wt3 no acepta complejas. Además, hemos tenido que ir probando escalas de texturas, del objeto, colocación, etc.

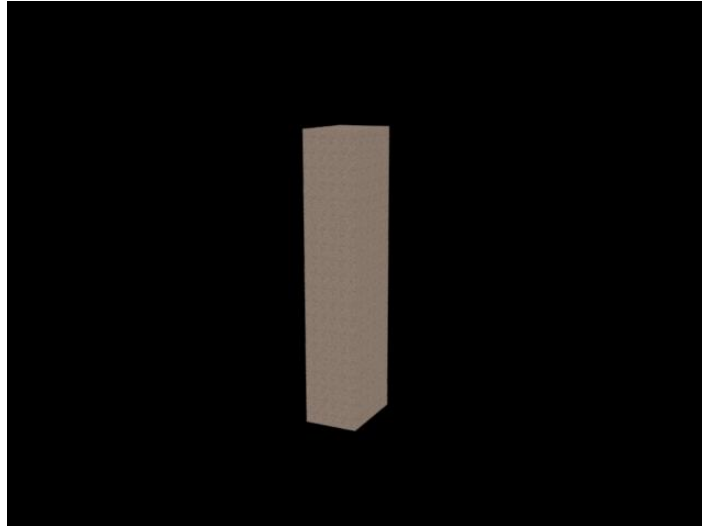


Figura 31. Modelo del mojón. (Fuente propia)

Vista AR

La primera tarea es mostrar en varias localizaciones fijas un icono. Para añadir las coordenadas vamos a convertir un ArrayList de coordenadas a objeto Json usando la librería Gson. Luego recorreremos todas el número de posiciones y a cada una le asignamos las coordenadas.

Ahora vamos a mostrar un modelo de Wikitude en nuestra posición. Para ello la dinámica es la misma que usamos en la primera iteración con la tarea de mostrar un icono, pero debemos cargar un modelo 3D en vez del icono. También creamos una flecha indicatoria hacia la dirección en la que se encuentra el objeto 3D.



Figura 32. Modelo 3D en realidad aumentada de La Tierra en nuestra localización. (Fuente propia)

Tras esto, lo que nos queda es sustituir este modelo de la figura 32 por el de la figura 31. Una vez hecho se verá lo siguiente:



Figura 33. Modelo 3D en realidad aumentada del mojón en nuestra localización. (Fuente propia)

Lo que nos queda por hacer es sustituir cuando se muestra el modelo 3D. Eliminar que se muestre en nuestra ubicación para que se muestre en las distintas localizaciones de la ruta. Esto lo conseguimos combinando las dos acciones anteriores: mostrar el modelo y mostrar en distintas localizaciones.

Una vez hecho esto último, hemos conseguido mostrar el modelo 3D en varias localizaciones cercanas que hemos fijado. Por tanto, con esto estamos cumpliendo la prueba de geolocalización que definimos anteriormente.

Vista Info

De manera similar a como hicimos en el layout principal vamos a realizar este. Introducimos distintos apartados de texto explicando la zona y alguna imagen con información. Tras esto, creamos el controlador de tamaño de letra, la barra de scroll y organizamos el contenido de manera adecuada.

El scroll es una etiqueta XML en la cual tenemos que encerrar nuestro texto e imágenes para poder realizar el scroll correctamente y poder visualizar todo el contenido.

Demo

Este botón ha sido incluido a última hora y el motivo es para poder mostrar la funcionalidad de realidad aumentada como demostración en nuestra ubicación. Se ha añadido dentro de la vista AR y al pulsarlo podemos visualizar el mojón virtual en nuestra posición. Esto es lo mismo que hicimos en la iteración primera pero cambiando el icono por el modelo 3d del mojón. La siguiente imagen muestra dicho botón.

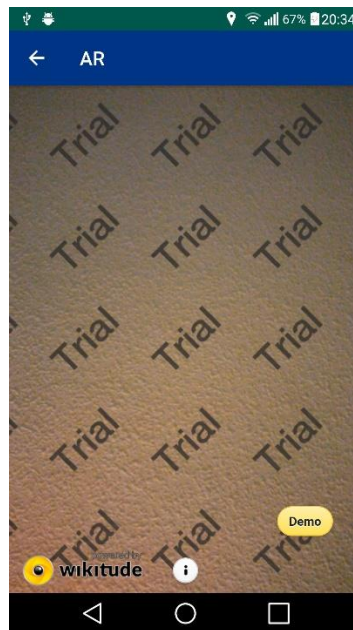


Figura 34. Vista del apartado AR. (Fuente propia)

10. Resultados

El objetivo principal de este proyecto era conseguir una aplicación en la que mostremos realidad aumentada en función de la ubicación del dispositivo y aportar información sobre la zona de Las Lagunas de La Mata, algo que hemos conseguido. Esto en cuanto al producto. Pero, además de esto, teníamos una serie de objetivos que debemos evaluar si hemos alcanzado.

Otro gran objetivo era abordar un proyecto desde principio a fin, pasando por todas las etapas. Comenzar por un estudio del sector sobre el cual íbamos a trabajar, estudiar las tecnologías para escoger las más adecuadas, estudiar el problema a abordar, elegir una metodología de trabajo, especificar el sistema a desarrollar junto con un diseño adecuado para finalmente desarrollar la aplicación.

En cuanto a los costes temporales, por lo general todo ha sido según lo planificado, aunque hay excepciones como el apartado de diseño, para el cual estimamos un mes y fueron tres semanas para su realización.

Con todo esto, hemos desarrollado una aplicación móvil nativa en la que el turista podrá visitar Las Lagunas de La Mata de Torrevieja, realizar una ruta guiada por hasta 54 puntos que rodean la laguna, documentarse sobre el lugar en cuestión y ver lo que antiguamente fueron los mojones que delimitaban la laguna para la explotación salinera, esto último con ayuda de la realidad aumentada.

Para terminar vamos a mostrar unas capturas de las interfaces de la aplicación para poder realizar una comparación con los diseños propuestos y comprobar el cumplimiento de los requisitos establecidos.

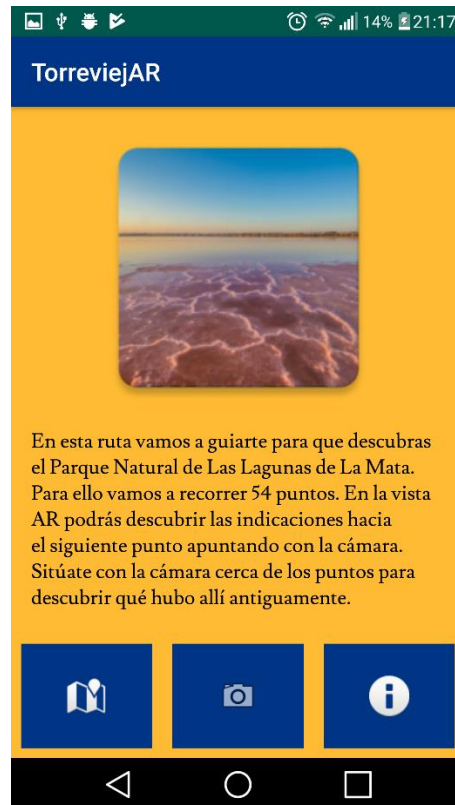


Figura 35. Interfaz principal (Fuente propia)

Esta es la interfaz principal, la que se encontrará el usuario al abrir la aplicación y que le situará en el contexto. Donde dispone de una imagen del lugar en cuestión, un texto explicativo, menú con tres botones que le llevarán a las distintas funcionalidades y la barra de acción que indica el nombre de la aplicación.

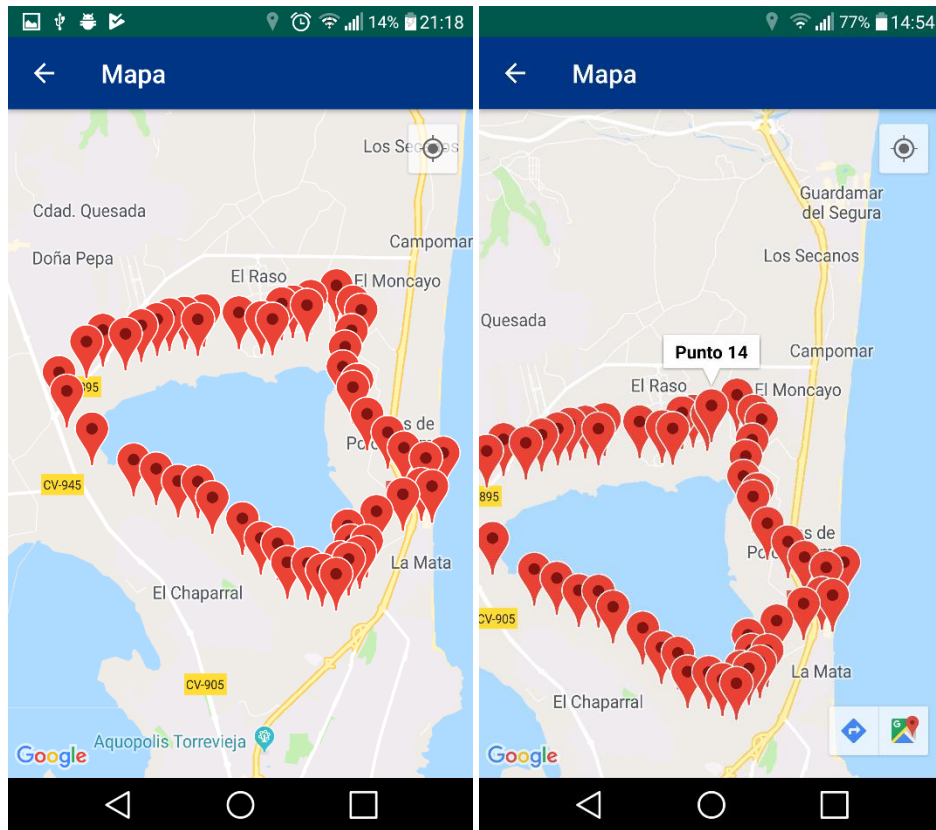


Figura 36. Interfaz mapa mostrando marcadores

Interfaz mapa. Contiene un mapa de la API de Google Maps donde justo al abrir verán los marcadores de los puntos de la ruta y, si pulsan un marcador en concreto, aparecerá un rótulo encima del mismo mostrando que punto es junto con dos botones en la esquina inferior derecha. El botón de la flecha abrirá la aplicación Google Maps creando automáticamente una ruta desde la posición que se encuentre el usuario hacia el punto en cuestión. El botón del mapa simplemente abrirá Google Maps centrando el foco en el punto en cuestión.



Figura 37. Interfaz mapa mostrando posición actual. (Fuente propia)

Si pulsan el botón de la esquina superior derecha les llevará hacia la posición en la que se encuentran y aparecerá un mensaje emergente que dice “Tu posición actual”.

Por último, la barra de acción muestra el nombre de la interfaz junto con un botón de volver.

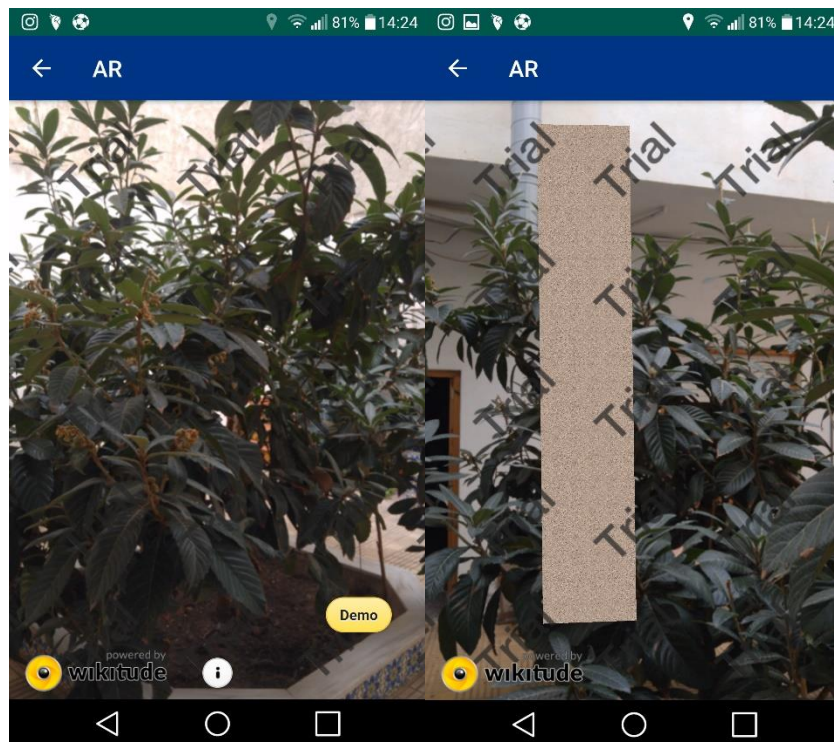


Figura 38. Interfaz AR

La vista de realidad aumentada mostrará lo que ve la cámara en ese momento junto con dos botones: el botón *i* dice si hay localizaciones cargadas o está buscando nuestra localización mediante mensaje emergente; el botón *Demo* nos permite mostrar la realidad aumentada en nuestra ubicación sin tener que desplazarnos a los puntos en cuestión. De no pulsar el botón *Demo*, para mostrar el modelo 3D debemos situarnos en uno de los 54 puntos de la ruta.

La barra de acción muestra el nombre de la interfaz y un botón de volver.



Figura 39. Interfaz Info

Aquí informamos al usuario acerca del lugar en cuestión. Dispone de una barra de scroll vertical para poder desplazarnos hacia el contenido restante. La barra de acción nombra la interfaz en la que nos encontramos y dispone de un botón de volver.

Por último, decir que la aplicación ha sido probado en distintos dispositivos Android y se ha comprobado tanto su correcto funcionamiento como su correcto diseño. Con esto cubrimos las pruebas que diseñamos en el apartado de Diseño de pruebas y validación.

11. Conclusiones y trabajo futuro

TorrevejAR ha sido todo un reto para mí. Suponía adentrarme en el mundo del desarrollo móvil usando nuevas tecnologías para mí. Se ha conseguido desarrollar una aplicación y podemos decir que es algo funcional y que el turista puede usar en su visita a Las Lagunas de La Mata de Torreveja.

A partir de este proyecto podemos seguir trabajando para en un futuro obtener una herramienta potente para el turismo en Torreveja. Como mejoras a esta aplicación se podrían añadir varias funcionalidades: una brújula radar en la vista de realidad aumentada, en la que aparezcan los puntos de la ruta y nos ayude a orientarnos; la posibilidad de tomar una instantánea y compartirla; utilizar una versión premium del framework Wikitude para eliminar la marca de agua; mejorar la vista AR con marcadores personalizados, es decir, que no sea el mismo modelo 3D siempre; incluir una audioguía, etc. Son inmensas las posibilidades, incluso podríamos adaptar esta aplicación al turismo de todo tipo en la zona de Torreveja y no sólo para la visita de Las Lagunas de La Mata.

Pasando a las conclusiones más generales, el objetivo aquí era analizar, diseñar y desarrollar una aplicación para turismo en Las Lagunas de La Mata usando técnicas de realidad aumentada y basándonos en los conocimientos obtenidos en el Grado en Ingeniería Multimedia. Esto, en mi opinión, creo que se ha conseguido, y no sólo eso, sino que he aprendido acerca de tecnologías móviles y a abordar un proyecto desde el inicio hasta el final.

Quiero terminar diciendo que con este proyecto cierro una etapa de mi vida, la académica, en la que he aprendido mucho, tanto académicamente como personalmente, y con este proyecto se cierra un ciclo.

Referencias

1. *Realidad aumentada*. Disponible en: https://es.wikipedia.org/wiki/Realidad_aumentada
2. *Pokemon GO*. Disponible en: <https://www.pokemongo.com/es-es/>
3. *IKEA Place*. Disponible en: <https://itunes.apple.com/us/app/ikea-place/id1279244498?mt=8>
4. *Fitness AR*. Disponible en: <https://itunes.apple.com/us/app/fitness-ar/id1274233318?mt=8>
5. *Poly*. Descubre el mundo del 3D. Disponible en: <https://poly.google.com/>
6. *Real Salina de La Mata*. Auto del deslinde de 1763.
7. *El origen de la realidad aumentada*. Disponible en: <https://blogthinkbig.com/realidad-aumentada-origen>
8. *Realidad mixta*. Disponible en: https://es.wikipedia.org/wiki/Realidad_mixta
9. *Microsoft HoloLens*. Disponible en: <https://www.microsoft.com/es-es/hololens>
10. *Snapchat, ¡la forma más rápida de compartir un momento!* Disponible en: <https://www.snapchat.com/l/es/>
11. *¿Por qué los jóvenes están cambiando Facebook por Snapchat?* Disponible en: <https://www.elespectador.com/tecnologia/por-que-los-jovenes-estan-cambiando-facebook-por-snapchat-articulo-738774>
12. *Pokémon GO*. Disponible en: <https://www.pokemongo.com/es-es/>
13. *AR MApp*. Disponible en: https://play.google.com/store/apps/details?id=com.smartechgroup.bcnmapp&hl=es_419
14. *La nueva aplicación móvil de realidad aumentada 'Turismo de Galicia', una guía para los visitantes*. Disponible en: <http://www.20minutos.es/noticia/2644374/0/nueva-aplicacion-movil-realidad-aumentada-turismo-galicia-guia-para-visitantes/>
15. *Características de ARCore*. Disponible en: <https://developers.google.com/ar/discover/>
16. *Características de Vuforia*. Disponible en: <https://library.vuforia.com/content/vuforia-library/en/features/overview.html>
17. *Características de Wikitude*. Disponible en: <https://www.wikitude.com/products/wikitude-sdk/>

18. *Características de Android Studio.* Disponible en: https://es.wikipedia.org/wiki/Android_Studio
19. *¿Qué es React Native?* Disponible en: <https://clouddistrict.com/blog-dev/que-es-react-native/>
20. *Ionic Framework.* Qué es y como empezar con Ionic Framework. Disponible en: <https://www.phonegapSpain.com/que-es-y-como-empezar-con-ionic-framework/>
21. *Trello.* Trello es la manera gratuita, flexible y visual de organizarlo todo con cualquiera. Disponible en: <https://trello.com/>
22. *Servicio Web de Transformación de Coordenadas. Instituto Geográfico Nacional.* Disponible en: <https://www.ign.es/wcts-app/>
23. Doran, G. T. *There's a S.M.A.R.T. Way to Write Management's Goals and Objectives.* Management Review, Vol. 70 (11), pp. 35-36. 1981
24. *Especificación de Requisitos según el estándar de IEEE 830.* Disponible en: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
25. Hassan Montero, Y. *Experiencia de Usuario: Principios y Métodos.* 2015. Disponible en: http://yusef.es/Experiencia_de_Usuario.pdf
26. Balsamiq. Herramienta para crear wireframes. Disponible en: <https://balsamiq.com/>
27. Justinmind. Herramienta para crear mockups y diseños interactivos. Disponible en <https://www.justinmind.com/>
28. Donald Norman. Disponible en https://es.wikipedia.org/wiki/Donald_Norman
29. Google Maps API. Disponible en: <https://cloud.google.com/maps-platform/?hl=es>
30. 3DS Max. Software de renderización, animación y modelado 3d. Disponible en: <https://www.autodesk.es/products/3ds-max/overview>

