

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN



“APLICACIÓN ANDROID PARA LA
INTERPRETACIÓN CULTURAL:
LA RAMBLA DE PUÇA”

TRABAJO FIN DE GRADO

Diciembre – 2025

AUTOR: Marcos Conejeros Campo

DIRECTOR: Prof. Pablo Corral González

RESUMEN

El presente proyecto se centra en el desarrollo de una aplicación nativa para Android, utilizando Android Studio y Kotlin, con el objetivo de enriquecer la experiencia de los visitantes en espacios naturales y rutas de senderismo en la zona de Alicante. Esta iniciativa, propuesta por la asociación "Observatorio del Vinalopó" , busca facilitar la interpretación del patrimonio natural y cultural a través de dispositivos móviles.

Los objetivos principales de la aplicación son:

- Capacitar a entidades locales como el "Observatorio del Vinalopó" y el ayuntamiento para difundir información sobre rutas, puntos de interés, miradores y otros servicios.
- Proporcionar acceso a recursos multimedia como audios, fotos y videos que complementen la visita.
- Recopilar datos sobre la afluencia de visitantes, inicialmente a través del registro de descargas.

La aplicación se desarrolla con tecnologías como Jetpack Compose para la interfaz de usuario, Firebase Firestore para la gestión de datos en tiempo real y el SDK de Google Maps para la visualización geográfica y la funcionalidad de geofencing.

Este proyecto contextualiza la aplicación en el patrimonio de la Rambla de Puça, un entorno de gran valor histórico, paisajístico y medioambiental en Petrer destacando que la tecnología móvil es una herramienta eficiente para la interpretación del patrimonio, superando las limitaciones de los paneles informativos tradicionales.

El proyecto fusiona las Tecnologías de la Información y la Comunicación (TIC) con la educación ambiental, buscando no solo mejorar la experiencia del visitante, sino también promover activamente la conservación de los espacios naturales.

ABSTRACT

This project focuses on developing a native Android application using Android Studio and Kotlin to enhance the visitor experience in natural areas and hiking trails in the Alicante region. This initiative, proposed by the "Observatorio del Vinalopó" association, aims to facilitate the interpretation of natural and cultural heritage through mobile devices.

The main objectives of the application are:

- To enable local entities like "Observatorio del Vinalopó" and the city council to disseminate information about trails, points of interest, viewpoints, and other services.
- To provide access to multimedia resources such as audio clips, photos, and videos that complement the visit.
- To collect data on visitor numbers, initially through download registration.

The application is being developed with technologies such as Jetpack Compose for the user interface, Firebase Firestore for real-time data management, and the Google Maps SDK for geographical visualization and geofencing functionality.

The document contextualizes the application within the heritage of the Rambla de Puça, an environment of great historical, scenic, and environmental value in Petrer. It highlights that mobile technology is an efficient tool for heritage interpretation, overcoming the limitations of traditional information panels.

The project merges Information and Communication Technologies (ICT) with environmental education, seeking not only to improve the visitor's experience but also to actively promote the conservation of natural spaces.

ÍNDICE

RESUMEN	3
ABSTRACT	4
1. INTRODUCCIÓN	8
1.1. OBJETIVOS Y ESTRUCTURA	8
1.2. ESTADO DEL ARTE.....	10
1.2.1. RUTA DE PUÇA	10
1.2.2. INTERPRETACIÓN DEL PATRIMONIO NATURAL Y CULTURAL.....	12
1.2.3. APLICACIONES MÓVILES PARA LA INTERPRETACIÓN DEL PATRIMONIO	13
2. MATERIAL Y MÉTODOS	18
2.1. HARDWARE.....	18
2.2. SOFTWARE	18
2.2.1. SISTEMA OPERATIVO ANDROID.....	18
2.3. LENGUAJES DE PROGRAMACIÓN	22
2.3.1. JAVA.....	22
2.3.2. KOTLIN.....	22
2.4. HERRAMIENTAS UTILIZADAS.....	22
2.4.1. JDK	22
2.4.2. ANDROID STUDIO.....	23
2.4.3. ANDROID SDK	25
2.4.4. FIREBASE	25
2.4.5. GOOGLE MAPS	26
2.4.6. JETPACK COMPOSE.....	26

3. RESULTADOS Y DISCUSIÓN	28
3.1. REQUISITOS DEL PROYECTO	28
3.2. ESTRUCTURA DEL PROYECTO.....	28
3.3. PERMISOS NECESARIOS.....	30
3.4. SERVICIOS OFRECIDOS	32
3.5. DIAGRAMA DE FLUJO	33
3.6. INTERFAZ DE USUARIO	35
3.6.1. LOGIN	35
3.6.2. PANTALLA PRINCIPAL	36
3.6.3. RUTAS.....	39
3.6.4. LISTA DE LOCALIZACIONES	40
3.6.5. MAPA	43
3.6.6. DETALLES DE LOCALIZACIÓN.....	44
3.6.7. OBSERVATORIO	44
3.6.8. GALERÍA	45
4. CONCLUSIONES Y LÍNEAS FUTURAS	47
5. BIBLIOGRAFÍA	49

Índice de tablas

Tabla 1: Especificaciones PC	18
Tabla 2: Especificaciones Móvil.....	18

Índice Ilustraciones

Ilustración 1: Situación de la Rambla de Puça o Pusa	10
Ilustración 2: Arquitectura del sistema operativo Android.....	19
Ilustración 3: Consola de Firebase.....	26
Ilustración 4: Estructura del proyecto	29
Ilustración 5: Permisos de ubicación	31
Ilustración 6: Permisos de trabajo en segundo plano.....	31
Ilustración 7: Diagrama de flujo	34
Ilustración 8: Pantalla Login.....	35
Ilustración 9: Pantalla con acceso de usuario registrado.....	36
Ilustración 10: Pantalla de Android solicitando permisos de ubicación	37
Ilustración 11: Pantalla de Android solicitando permiso de notificaciones.....	38
Ilustración 12: Acceso a la galería bloqueado para usuarios no identificados.....	39
Ilustración 13: Pantalla para la selección de ruta.....	39
Ilustración 14: Dialogo para añadir una ruta nueva, sólo visible para acceso admin	40
Ilustración 15: Pantalla con lista de ubicaciones	41
Ilustración 16: Dialogo para añadir nueva ubicación a la ruta, sólo visible para admin	42
Ilustración 17: Ventana de Google Maps con las ubicaciones de la ruta marcadas.....	43
Ilustración 18: Detalles de la ubicación La Canal de Ferro	44
Ilustración 19: Pantalla de información sobre el observatorio.....	45
Ilustración 20: Pantalla de la galería.....	46

1. Introducción

1.1 OBJETIVOS Y ESTRUCTURA

El presente proyecto se centra en el desarrollo de una aplicación nativa para Android diseñada para enriquecer la experiencia de los visitantes de espacios naturales y rutas de senderismo en la zona de Alicante. Esta iniciativa surge como respuesta a la necesidad de proporcionar información accesible y relevante sobre el entorno, tal como lo planteó la asociación ecologista "Observatorio del Vinalopó".

La aplicación, desarrollada con Android Studio y el lenguaje de programación Kotlin, tiene como objetivo principal facilitar la interpretación del patrimonio natural y cultural a través de dispositivos móviles.

En esencia, la aplicación busca:

- Capacitar a la asociación ecologista "Observatorio del Vinalopó" y/o al ayuntamiento local para que puedan difundir de manera eficiente información valiosa a los visitantes. Esto incluye detalles sobre las rutas homologadas y senderos disponibles, la ubicación y descripción de elementos destacados del patrimonio natural, material e inmaterial, así como la señalización de zonas de descanso, miradores y otros servicios de interés.
- Facilitar el acceso a una variedad de recursos multimedia que complementen la visita y contribuyan a una mejor interpretación del patrimonio. Entre estos recursos se contemplan audios con explicaciones sobre puntos de interés a lo largo de las rutas, material visual como fotos y videos, y la posibilidad de recibir alertas informativas desde los centros de interpretación sobre posibles riesgos o incidencias que puedan surgir.

La aplicación se desarrollará utilizando Android Studio y Kotlin, aprovechando las funcionalidades del sistema operativo Android para ofrecer una experiencia de usuario óptima. Se prestará especial atención al diseño de una interfaz intuitiva y accesible, que permita a los visitantes acceder fácilmente a la información y los recursos multimedia.

Por tanto, podemos decir que la aplicación tiene como objetivo principal facilitar la interpretación del patrimonio a través de dispositivos móviles, capacitando a entidades locales para difundir detalles sobre rutas homologadas, puntos de interés (naturales, materiales e inmateriales), zonas de descanso y servicios. Asimismo, busca proporcionar acceso a recursos multimedia como audios, fotos y videos que complementen la visita, y la capacidad de emitir alertas informativas sobre posibles riesgos. Un objetivo adicional es la recopilación de datos sobre la afluencia de visitantes, inicialmente a través del registro de descargas.

Desarrollada con Android Studio y Kotlin, la aplicación aprovecha Jetpack Compose para la construcción de su interfaz de usuario, garantizando una experiencia intuitiva y fluida.

La gestión de datos se realiza mediante Firebase Firestore, lo que permite la carga y actualización en tiempo real de las ubicaciones. La visualización geográfica se implementa con el SDK de Google Maps para Android, mostrando los puntos de interés y la ubicación del usuario. La funcionalidad de geofencing, apoyada por los servicios de ubicación de Google Play, es crucial para la activación de alertas contextuales.

En resumen, este proyecto fusiona las Tecnologías de la Información y la Comunicación con la educación ambiental y la interpretación del patrimonio, buscando no solo mejorar la experiencia del visitante, sino también promover activamente la conservación de los espacios naturales.



1.2 ESTADO DEL ARTE

1.2.1 RUTA DE PUÇA

La rambla de Puça o Pusa, también conocida como Rambla de los Molinos, constituye uno de los elementos paisajísticos e históricos más significativos del municipio alicantino de Petrer, se trata de un curso de agua estacional propio del clima mediterráneo semiárido, que desemboca en el curso medio del río Vinalopó, auténtico eje vertebrador de la provincia de Alicante. La rambla de Pusa, también conocida como río de Petrer ha contribuido a modelar el territorio y condicionado tanto los usos agrícolas como el poblamiento a lo largo de los siglos.

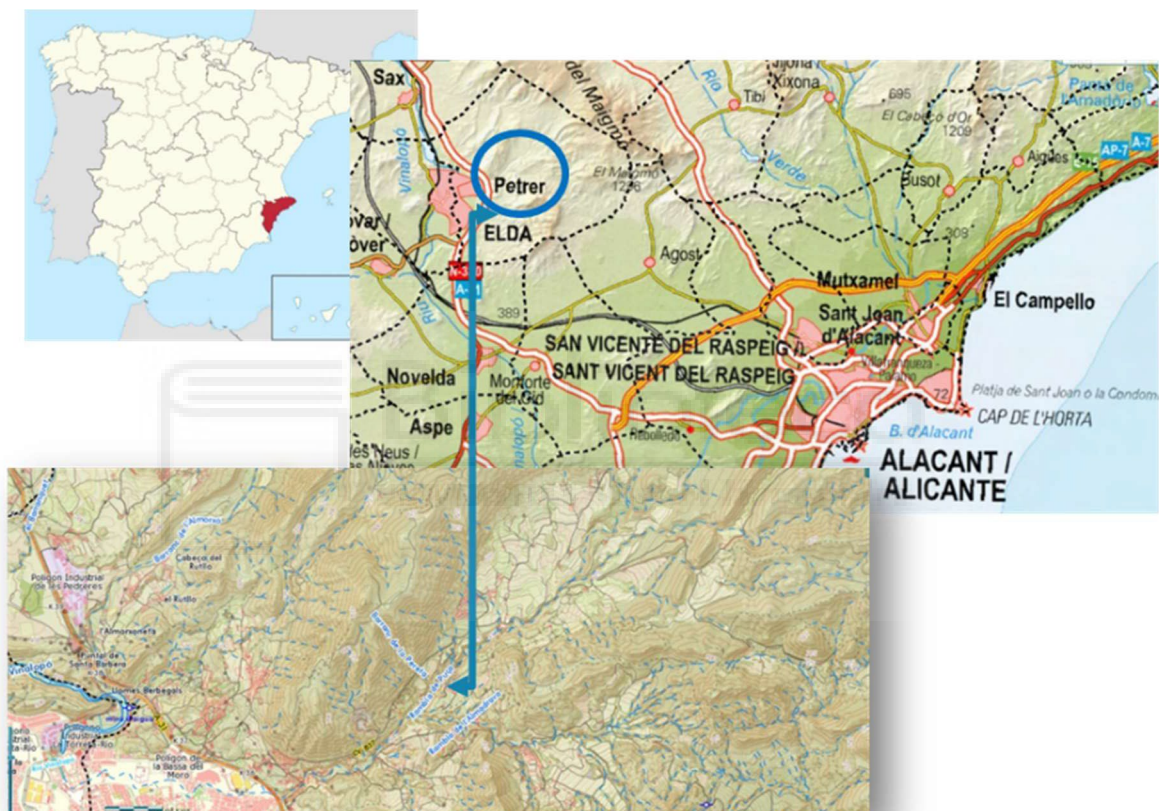


Ilustración 1: Situación de la Rambla de Puça o Pusa

Su importancia radica, en primer lugar, en su función hidráulica en el pasado. A pesar de ser un cauce de régimen irregular, las aguas torrenciales que descenden por la rambla fueron aprovechadas desde época medieval para el riego de huertas y, sobre todo, para el funcionamiento de los molinos harineros que le dieron nombre. A lo largo de su recorrido se asentaron distintos ingenios hidráulicos, testimonio del aprovechamiento tradicional de los recursos hídricos en una zona marcada por la escasez de agua.

Según el Diccionario geográfico-estadístico-histórico de España y sus posesiones de Ultramar [1], hacia mediados del siglo XIX este cauce “no criaba ningún pescado”, pero

sí daba movimiento a tres molinos harineros. No obstante, si observamos referencias más recientes —naturalísticas, senderistas o patrimoniales— se mencionan varios molinos a lo largo del recorrido. Algunos destacan por su relevancia cultural o trayectoria histórica. Entre ellos:

- Molino de L'Assut (antigua vaquería)
- Molí de Mahoma
- Molí del Pinxe
- La Calera (hornos de cal)
- Molí del Turco
- Molino de la Pólvora (dedicado al azufre)
- Molí de El Salt
- Molí de Ponça
- Molí Ganxo o Molí la Reixa (actual mesón-restaurante)

En segundo lugar, la rambla de Puça posee un gran valor patrimonial y etnográfico. Los molinos, las acequias y las infraestructuras asociadas constituyen parte de la memoria colectiva de Petrer, al recordar la importancia de la molienda y la agricultura en la economía local hasta bien entrado el siglo XX. Estos elementos, junto al propio cauce, conforman un corredor cultural que conecta la historia de la ingeniería hidráulica con la vida cotidiana de generaciones de petrerenses.

Por último, la rambla tiene una indudable relevancia medioambiental y paisajística. Además de configurar un corredor natural que conecta distintos hábitats, desde las estribaciones de la Serra del Frare hasta su desembocadura en el río Vinalopó, es un espacio de transición entre la sierra y el llano agrícola, de gran interés para el estudio de la geomorfología y de la adaptación humana al medio.

En conjunto, la rambla de Puça o de los Molinos es un ejemplo singular de cómo el agua, incluso en territorios áridos, ha articulado el paisaje, la economía y la cultura local, mereciendo ser reconocida y preservada como parte del patrimonio natural e histórico de Petrer.

El Observatorio del Vinalopó, asociación ecologista entre cuyos fines se encuentra no sólo pedir la regeneración del que acaso sea el río que mejor caracteriza el sur de la Comunitat Valenciana, el río Vinalopó, sino también de aquellos elementos hídricos que conforman su cuenca hidrográfica. La rambla de Puça constituye el exponente más representativo al conservar una estructura geomorfológica bien conservada propia de las ramblas del semiárido peninsular. Asimismo, socialmente la población local lo ha elevado a exponente de su identidad e historia. Por último, la proyección turística de la localidad es frecuente que incluya el Molí del Salt como punto de interés, además de ser un hito bañarse bajo la cascada especialmente durante los calurosos y largos estiajes petrerenses. Todo ello ha animado a los socios a aprobar la realización, primero, de un estudio de la rambla que contó con la ayuda financiera del Ajuntament de Petrer y, en la actualidad, a

encomendar a un ingeniero la redacción de un proyecto de interpretación del patrimonio etnohistórico de la rambla contando con las nuevas aportaciones de las TIC basadas, principalmente, en el que tal vez sea el aparato más comúnmente disponible y empleado por la ciudadanía: el teléfono inteligente.

La telefonía móvil concilia el disfrute del senderismo por espacios ambiental y patrimonialmente relevantes, como es el caso de la rambla de Puça, con la obtención de información sobre aquellos hitos que los caracterizan (los molinos, en el presente caso) sin necesidad de detener la marcha un tiempo excesivo leyendo los textos, habitualmente densos y extensos, de paneles informativos dispuestos por diferentes administraciones. Asimismo, las inclemencias meteorológicas, el paso del tiempo o el vandalismo, terminan por hacerlos ilegibles o desaparecer. Por el contrario, la información proporcionada desde un teléfono móvil no sólo proporciona una disponibilidad del material interpretativo (explicaciones, mapas, imágenes, etc.) con una excelente calidad, sino que puede adaptarse fácilmente a ciertos requerimientos de accesibilidad para aquellas personas que así lo requieran (aumentar el tamaño de la letra, mensajes acústicos, etc.). A ello cabe sumar que la información queda disponible para el usuario de la aplicación si desea preparar previamente la ruta o hacerlo a posteriori, una ventaja de la que claramente adolecen los habituales paneles informativos.

1.2.2. INTERPRETACIÓN DEL PATRIMONIO NATURAL Y CULTURAL

La interpretación del patrimonio natural y cultural se define como un proceso comunicativo cuyo objetivo es revelar el significado y las relaciones entre los recursos naturales y culturales, fomentando así la comprensión y el aprecio del público. Freeman Tilden, considerado uno de los padres de la interpretación moderna, la describió como una actividad educativa que busca provocar en el visitante el deseo de profundizar en el conocimiento del patrimonio [2].

A lo largo del tiempo, la interpretación ha evolucionado significativamente. Inicialmente, se centraba en métodos tradicionales como las visitas guiadas, los paneles informativos y los folletos impresos. Sin embargo, la creciente disponibilidad y accesibilidad de las Tecnologías de la Información y la Comunicación (TIC) ha abierto nuevas vías para la interpretación, permitiendo experiencias más interactivas, personalizadas y atractivas [3].

La interpretación desempeña un papel crucial en la conservación, educación y disfrute del patrimonio. Al proporcionar a los visitantes información relevante y significativa, se aumenta su conciencia sobre el valor del patrimonio y se fomenta su responsabilidad en su protección. Además, la interpretación puede contribuir al desarrollo sostenible de las comunidades locales, generando beneficios económicos y sociales a través del turismo cultural y natural [4].

Sin embargo, la interpretación del patrimonio también enfrenta diversos retos. Uno de ellos es la necesidad de llegar a un público cada vez más diverso, con diferentes niveles de conocimiento, intereses y habilidades. Otro desafío es la accesibilidad, ya que muchas personas con discapacidades encuentran barreras para acceder a la información y disfrutar de las experiencias interpretativas.

Las TIC ofrecen numerosas oportunidades para superar estos desafíos y mejorar la interpretación del patrimonio. Las aplicaciones móviles, por ejemplo, pueden proporcionar información personalizada y adaptada a los intereses de cada visitante, ofrecer contenido multimedia enriquecido y facilitar la accesibilidad a personas con discapacidades [5]. No obstante, la plena implementación de las TIC exige un esfuerzo, por parte de las administraciones públicas, en la superación de la conocida como “brecha digital”, con objeto de que estas desplieguen la amplia gama de posibilidades que ofrecen en la interpretación del patrimonio.

La progresión tecnológica en este campo apenas acaba de despuntar en lo que a las posibilidades que ofrece en la interpretación del patrimonio se refiere. Un buen ejemplo de ello lo tenemos en algunos proyectos como el “Proyecto Europea”, dedicado a mejorar una plataforma digitalizada de todo el patrimonio cultural en Europa con materiales culturales digitalizados, incluyendo una biblioteca, un museo y un archivo digitales o “VAST Project” para la valorización de la historia narrativa y el paisaje patrimonial y resaltar la reminiscencia de las construcciones militares y los artefactos de la Primera Guerra Mundial, inspeccionados y exhibidos en 3D para producir nuevos materiales de circulación y comunicación (modelos digitales 3D, plataformas WebGIS, conferencias, artículos comunicativos y científicos, visitas virtuales, etc.), entre otros muchos proyectos de esta naturaleza. Ambos proyectos constituyen ejemplos del Internet de las Cosas (IoT), siendo ampliamente empleado en la interpretación del patrimonio cultural dada su capacidad para conectar objetos físico-culturales con TIC y las posibilidades que ofrece para una interpretación innovadora y una conexión con los valores ocultos del patrimonio. Se enfoca en la accesibilidad (guiando a los visitantes) y la digitalización cultural narrativa creativa (diseminando la importancia de los valores intangibles) [6].

1.2.3. APLICACIONES MÓVILES PARA LA INTERPRETACIÓN DEL PATRIMONIO

Las aplicaciones móviles se han convertido en una herramienta cada vez más popular para la interpretación del patrimonio, ofreciendo una amplia gama de funcionalidades y adaptándose a diferentes contextos y necesidades.

Se pueden clasificar las aplicaciones móviles utilizadas en la interpretación del patrimonio en varios tipos, entre los que destacan:

- **Guías de campo digitales:** Estas aplicaciones proporcionan información detallada sobre la flora, fauna, geología y otros aspectos naturales de un área

determinada. Permiten a los usuarios identificar especies, aprender sobre su ecología y acceder a información adicional.

- **Audioguías móviles:** ofrecen recorridos guiados por un lugar, proporcionando narraciones de audio sobre los puntos de interés. Pueden incluir mapas, imágenes y otros recursos multimedia.
- **Aplicaciones de rutas:** proporcionan información sobre rutas de senderismo, ciclismo u otras actividades al aire libre. Pueden incluir mapas, perfiles de elevación, puntos de interés y funciones de seguimiento de la ubicación.
- **Aplicaciones de realidad aumentada:** Estas superponen información digital (imágenes, texto, audio, video) al mundo real, permitiendo a los usuarios interactuar con el patrimonio de una manera innovadora y atractiva. Dentro de este grupo podemos citar algunos ejemplos que abordan un diverso espectro de espacios de relevancia patrimonial:
 - **Interpretación de prácticas del patrimonio inmaterial en cementerios [7]:** A través de una aplicación móvil, que utiliza la realidad aumentada, se superpone información audiovisual en sitios de entierro. El estudio concluye que el diseño de experiencias a través de estas tecnologías móviles puede dar un significado mejorado a lugares a menudo escasamente valorados como es el caso de los cementerios.
 - **Empleo de TIC para la interpretación del patrimonio de Jardines Históricos [8]:** Este artículo destaca la importancia de integrar tecnologías inteligentes como aplicaciones móviles, realidad aumentada y realidad virtual para enriquecer la experiencia de los visitantes en jardines históricos. Utiliza un estudio de caso del Jardín Ram Bagh en Amritsar, India, para demostrar cómo la interpretación limitada puede obstaculizar la comprensión del patrimonio. El objetivo es mejorar la participación, la adquisición de conocimientos, la accesibilidad y la inmersión para lograr resultados positivos en el visitante.
 - **Medios sincrónicos con reconocimiento de ubicación y visualización aumentada para turistas del mundo real (SMART): una aplicación para el sitio patrimonial de Khalifatabad, Bagerhat, Bangladesh [9]:** El documento propone una aplicación móvil para mejorar la experiencia turística en el sitio de patrimonio mundial de Khalifatabad en Bangladés. Argumenta que la falta de acceso a la información histórica y un sistema de guía deficiente impiden a los visitantes una comprensión más profunda del valor cultural del sitio. La aplicación tiene como objetivo proporcionar información auténtica y ayudas visuales para ofrecer una visita más inmersiva y atractiva.
- **Aplicaciones de diseño colaborativo:** Existen numerosas aplicaciones móviles que se utilizan para la interpretación del patrimonio o prácticas deportivas en el

medio natural. Algunas de las más relevantes para el contexto de este proyecto son:

- **Wikiloc:** Una plataforma colaborativa que permite a los usuarios descubrir y compartir rutas al aire libre. Ofrece información detallada sobre las rutas, incluyendo mapas, fotos, perfiles de elevación y comentarios de otros usuarios (Wikiloc.com).
- **AllTrails:** Una aplicación similar a Wikiloc, con una amplia base de datos de rutas de senderismo y ciclismo. Permite a los usuarios buscar rutas por ubicación, dificultad, longitud y otros criterios (AllTrails.com).
- **Aplicaciones de parques nacionales y espacios naturales:** Muchos parques nacionales y espacios naturales protegidos han desarrollado sus propias aplicaciones móviles para proporcionar información a los visitantes. Estas aplicaciones suelen incluir mapas, guías de flora y fauna, información sobre actividades y eventos, y alertas de seguridad.
 - Especialmente avanzada es la **Plataforma de servicios de turismo y navegación para el patrimonio cultural digital móvil (M-DCH)** [10]. Este trabajo presenta una plataforma de acceso abierto llamada DEH para el patrimonio cultural digital móvil (M-DCH). La plataforma organiza el contenido en unidades de Puntos de Interés (POI), Líneas de Interés (LOI), Áreas de Interés (AOI) y Sitios de Interés (SOI), permitiendo a los usuarios descargar y cargar contenido. También introduce una función de agrupación que permite a los creadores de grupos editar el contenido de los miembros.
 - **Creación de una interpretación personal en un sitio patrimonial virtual en 3D para mejorar la participación de los visitantes** [11]: El proyecto propone un nuevo enfoque para los sitios de patrimonio virtual que permite a los visitantes crear sus propias interpretaciones personales para aumentar la participación. Se desarrolló una aplicación móvil de realidad virtual (RV) que permite a los usuarios crear contenido de realidad mixta (RM) con activos virtuales históricos, emocionales y personales. Los hallazgos muestran cómo los visitantes se involucraron con el patrimonio virtual y realizaron interpretaciones ricas a través de esta plataforma.
 - **Proyecto Pórticos de Bolonia: Un repositorio 3D para la nominación de la UNESCO al WHL** [12]: creación de un repositorio 3D para la nominación de los pórticos de Bolonia como Patrimonio de la Humanidad de la UNESCO. El proyecto busca no solo resaltar los pórticos como una obra arquitectónica, sino también su significado social y antropológico. La plataforma permitirá el acceso en línea a una gran cantidad de datos y recursos, y también animará a los ciudadanos a participar subiendo su propio contenido y comentarios a través de las redes sociales.
 - **De la creación de significados a la compartición de significados: interpretación participativa en un sitio patrimonial virtual en 3D** [13]

: el proyecto explora la interpretación participativa en un sitio de patrimonio virtual en 3D, donde los usuarios pueden crear y compartir contenido. Los autores desarrollaron una aplicación de realidad virtual móvil llamada MyInterpret y un estudio de usuario reveló que el contenido generado por los usuarios mejora significativamente la experiencia del usuario. El estudio subraya cómo la interpretación socialmente generada se percibe de manera diferente a la interpretación oficial y cómo las funciones sociales son importantes para interactuar con el contenido generado por el usuario.

- Cabe destacar la aplicación móvil **Discover Québec** [14], que pretende documentar el patrimonio cultural tangible e intangible mediante un enfoque transmedia para documentar el patrimonio cultural tangible e intangible. La aplicación presenta múltiples "mini narrativas" y puntos de interés con diferentes tipos de medios (fotos, grabaciones de sonido y audiovisuales) para proporcionar múltiples puntos de vista y dar vida a los sitios históricos. El enfoque fomenta la participación de los ciudadanos y permite a los visitantes crear su propio itinerario y narrativa coherente.

Especialmente innovadora es la experiencia proporcionada por un proyecto que incluye un laboratorio móvil polivalente con acceso en línea dotado de varias técnicas optoelectrónicas, como escaneo láser 3D e imágenes multispectrales, para la monitorización en tiempo real y las intervenciones de restauración. Se centra en el uso de la ciencia avanzada y la tecnología de la información para mejorar la conservación del patrimonio cultural y la capacitación de los profesionales [15].

El conjunto de proyectos y aplicaciones descritas demuestran el potencial de las tecnologías móviles para mejorar la experiencia de los visitantes en espacios naturales, proporcionando información accesible, interactiva y personalizada. Sin embargo, también presentan algunas limitaciones. Algunas aplicaciones pueden requerir una conexión a Internet para acceder a toda la información, lo que puede ser un problema en áreas con poca cobertura. Otras pueden carecer de funcionalidades específicas, como la accesibilidad para personas con discapacidades. Pero, con carácter general, comparten un tema común: la aplicación de la tecnología para transformar la forma en que las personas interactúan, experimentan y conservan el patrimonio cultural. Desde el uso de aplicaciones móviles y realidad aumentada para mejorar la participación en cementerios, jardines históricos y sitios arqueológicos, hasta la creación de plataformas 3D y repositorios para la conservación y nominación de sitios históricos, la tecnología se presenta como una herramienta vital. Además, varios estudios resaltan la importancia del contenido generado por el usuario y la interpretación participativa, lo que permite a los ciudadanos y visitantes no solo consumir información, sino también contribuir activamente a la narrativa del patrimonio cultural. Esta convergencia de tecnología y patrimonio fomenta una comprensión más profunda y una conexión personal con los activos culturales, tanto tangibles como intangibles.

En el contexto del presente proyecto, donde la aplicación se utilizará en rutas de senderismo con posibles limitaciones de conectividad, la elección de Android nativo y Kotlin se justifica por su capacidad para ofrecer una funcionalidad offline robusta y un acceso completo a los recursos del dispositivo. El desarrollo nativo permite almacenar datos localmente, acceder al GPS para la navegación y optimizar el rendimiento de la aplicación, garantizando una experiencia de usuario satisfactoria incluso en áreas sin cobertura.



2. MATERIAL Y MÉTODOS

2.1. HARDWARE

En este apartado voy a comentar los dispositivos utilizados en este proyecto y sus especificaciones.

PC

Procesador	Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz 3.60 GHz
RAM	32,0 GB
Almacenamiento 1	954 GB SSD
Almacenamiento 2	932 GB HDD
Tarjeta gráfica	NVIDIA GeForce RTX 2060 SUPER (8 GB)
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64

Tabla 1: Especificaciones PC

Teléfono

XIAOMI REDMI 11

Procesador	HELIO G96 Octa-Core Max2.05GHz
RAM	8 GB
Versión Android	11 RP1A200720.011
Almacenamiento interno	128 GB

Tabla 2: Especificaciones Móvil

2.2. SOFTWARE

2.2.1. SISTEMA OPERATIVO ANDROID

Android es el sistema operativo de Google y está basado en una versión modificada del kernel de Linux 2.6, el cual utiliza para controlar servicios del núcleo del sistema como la seguridad, gestión de memoria o gestión de procesos. Actualmente, podemos encontrarlo presente en teléfonos móviles, PDAs, tabletas e incluso en Netbooks. Es una plataforma de código abierto distribuida bajo la licencia Apache 2.0, por lo que su distribución es libre y posibilita el acceso y modificación de su código fuente. Inicialmente fue desarrollado por Google, para más tarde unirse a la Open Handset Alliance (de la cual Google también forma parte), que está integrada por T-Mobile, Intel, Samsung, HTC o Nvidia entre otros. Sin embargo, Google ha sido la compañía que ha publicado la mayor parte del código fuente bajo la licencia Apache. En cuanto al ámbito del desarrollo de software para esta plataforma, a los programadores se les proporciona de manera gratuita el SDK (Software Development Kit) y un entorno de desarrollo, Android Studio.

Arquitectura:

En la ilustración 2 se muestra la arquitectura de Android, la cual está compuesta por 4 capas o niveles relacionados entre sí.

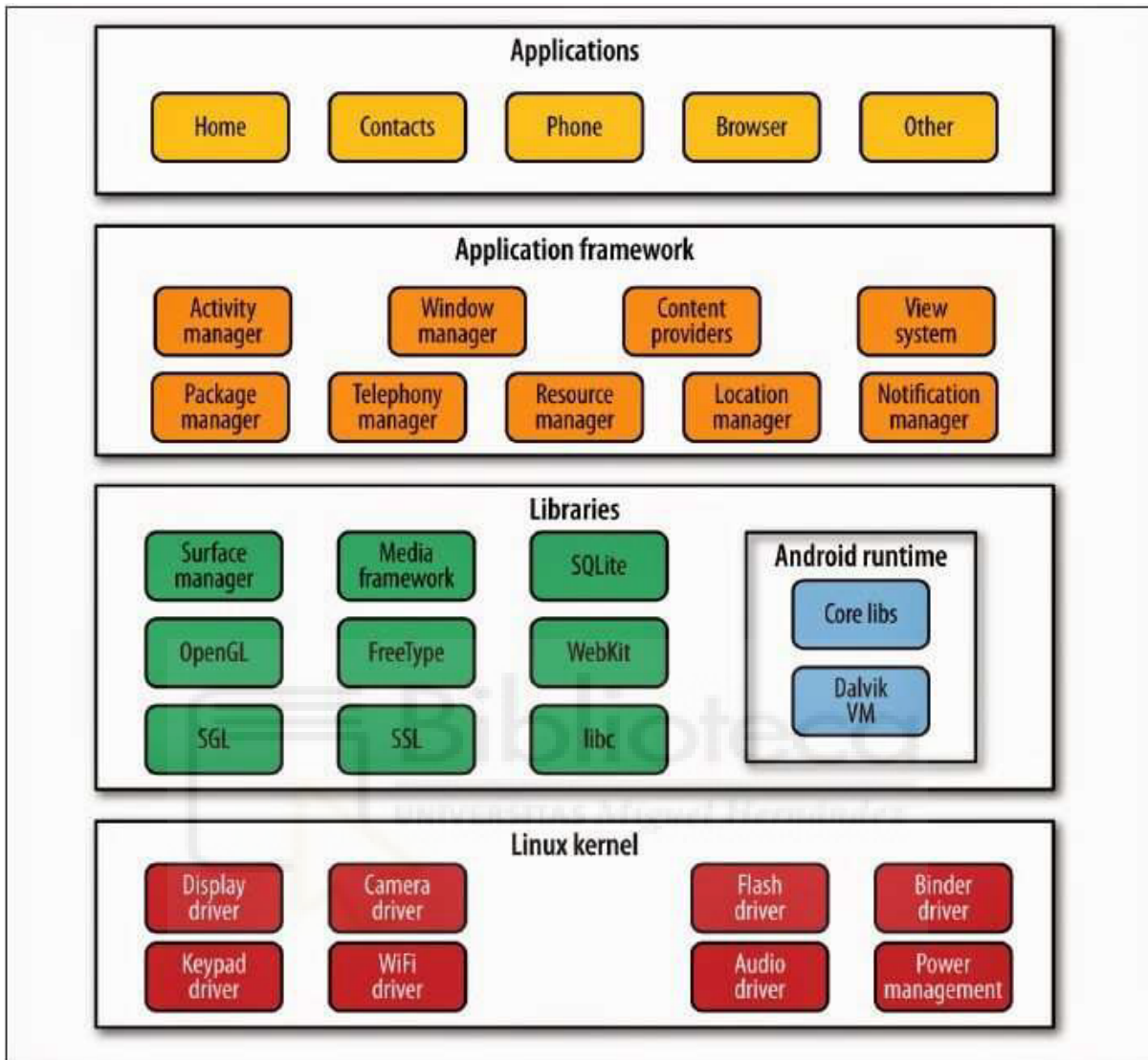


Ilustración 2: Arquitectura del sistema operativo Android.

A continuación, se explican más detalladamente cada una de las secciones de la arquitectura.

- **Aplicaciones:** Android, por defecto, viene con una serie de aplicaciones preinstaladas: agenda de contactos, calendario, gestor SMS, navegador web, etc. El usuario puede instalar nuevas aplicaciones gracias a Google Play u otros programas similares. Debido al diseño abierto, todas las aplicaciones "base" pueden ser sustituidas, excepto el launcher (lanzador de aplicaciones) para evitar la fragmentación.
- **Framework de aplicaciones:** Facilita un marco de desarrollo que estandariza y facilita la programación de aplicaciones Android. Los desarrolladores podrán

acceder a todas las APIs usadas en las aplicaciones base que incorpora Android. La arquitectura de la aplicación está diseñada para reutilizar componentes.

Por debajo de todas las aplicaciones se encuentran un conjunto de servicios y sistemas, incluyendo:

- Un amplio conjunto de vistas (Views) que se pueden usar para construir una aplicación como listas, grids, cajas de texto, botones, etc., e incluso un navegador embebido.
- Content Providers que permiten a las aplicaciones acceder a datos de otras aplicaciones o compartir sus propios datos.
- Resource Manager para poder acceder a recursos como Strings, gráficos y ficheros XML.
- Notification Manager que permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- Activity Manager para manejar el ciclo de vida de las aplicaciones y ofrecer una navegación común.
- **Librerías:** Son un conjunto de librerías escritas en C/C++ usadas en varios componentes del sistema. Están compiladas en el código nativo del compilador.
- **Android Runtime:** Compuesto por las librerías principales y la máquina virtual Dalvik (en versiones más recientes de Android, Dalvik ha sido reemplazado por ART - Android Runtime). Las librerías principales permiten que la funcionalidad disponible en las librerías anteriores esté disponible desde las librerías principales de Java. La máquina virtual Dalvik se ha diseñado de forma que cada aplicación pueda correr en su propio proceso con su propia instancia de la máquina virtual, de manera que un dispositivo móvil pueda correr múltiples máquinas virtuales simultáneamente que ejecutan ficheros en el formato .dex optimizado para minimizar el uso de memoria.
- **Linux Kernel:** Android se basa en una versión de Linux 2.6 para servicios del sistema principales como seguridad, manejo de memoria, manejo de procesos, pila de red o modelo de controladores. Además, actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

La Máquina Virtual Dalvik

Dalvik es una máquina virtual creada por Google para Android. Está basada en registros y optimizada para su uso en dispositivos móviles que suelen tener mayor necesidad de ahorro de energía. Permite a los dispositivos ejecutar varias máquinas virtuales simultáneamente; cada aplicación se ejecuta en su propia máquina virtual, hecho que añade un nivel más de seguridad. La compilación de los programas se realiza en dos fases; primero se llama a un compilador Java y después se convierten los bytecodes de Java a bytecodes de Dalvik generando un ejecutable con extensión .dex.

(Nota: Es importante mencionar que, a partir de Android 5.0 Lollipop, Dalvik fue reemplazado por Android Runtime (ART) como el entorno de ejecución predeterminado. ART compila el código de la aplicación de forma anticipada (AOT - Ahead-Of-Time) en lugar de en tiempo de ejecución (JIT - Just-In-Time) como Dalvik, lo que mejora el rendimiento y la eficiencia energética.)

Componentes

En este apartado hablaremos sobre los componentes o bloques que podemos encontrar en cualquier aplicación de Android. No es obligatorio implementarlos todos, por lo que podríamos decir que las aplicaciones están compuestas por una combinación libre de los bloques o componentes descritos a continuación:

- **Activity (Actividad):** Es una pantalla de la interfaz de usuario. Se pueden definir una o más actividades para manejar diferentes fases del programa. Cada actividad es responsable de guardar su propio estado para que pueda ser reestablecido en el ciclo de vida de la aplicación.
- **Service (Servicio):** Representa una tarea sin interfaz gráfico que se ejecuta de fondo. Otro componente puede arrancarlo, interactuar con él, dejarlo correr o pararlo. Un servicio es implementado como una subclase de Service.
- **ContentProvider (Proveedores de contenido):** Los proveedores de contenido hacen que un determinado grupo de datos estén disponibles para distintas aplicaciones. Extienden la clase ContentProvider para implementar los métodos de la interfaz; las aplicaciones no acceden directamente, sino que lo hacen a través de una clase ContentResolver. Este sistema permite acceder a datos almacenados en el sistema de ficheros, bases de datos SQLite o cualquier otra fuente de datos a través de un sistema unificado.
- **BroadcastReceiver (Receptores broadcast):** Son componentes que responden a los anuncios broadcast originados por el sistema u otras aplicaciones. No tienen interfaz gráfica, pero pueden enviar notificaciones a la barra de estado del sistema. Se implementan como una subclase de BroadcastReceiver.
- **Intent:** Es un mecanismo para describir una acción específica, como por ejemplo enviar un correo. Android diferencia entre dos tipos de intent: explícita e implícita. La explícita se utiliza para iniciar componentes de la propia aplicación y la implícita realiza una petición de acción general permitiendo a componentes de otras aplicaciones manejar la petición.
- **Listeners:** Son muy importantes para el desarrollo de aplicaciones móviles, sobre todo para los que incluyen pantalla táctil. Los Listeners se usan para que cuando pase algo en el código que está descrito (un evento) anteriormente, reaccionen mediante una acción determinada, por ejemplo, si el Listener en cuestión estuviera "escuchando" cuando se pulsa el botón "OK", cuando sea pulsado realizará las funciones que se le haya dicho que ejecute.

2.3. LENGUAJES DE PROGRAMACIÓN

La base utilizada en este proyecto es la plataforma Android, la cual se apoya en pilares fundamentales de la informática actual: JAVA y Kotlin como lenguajes de programación. Cabe destacar que las librerías que incluye el sistema están escritas en C y C++.

2.3.1 JAVA

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems en los primeros años de la década de los 90. Es multiplataforma, es decir, los programas no necesitan ser recompilados para ejecutarse en otra plataforma. Es un lenguaje seguro, robusto, multihilo, etc.

2.3.2 KOTLIN

Kotlin es un lenguaje de programación de propósito general, tipado estáticamente, que se ejecuta en la Máquina Virtual de Java (JVM) y también puede compilarse a JavaScript o a código nativo. Fue desarrollado por JetBrains y se lanzó por primera vez en 2011. En 2019, Google anunció que Kotlin sería el lenguaje preferido para el desarrollo de aplicaciones Android.

Sus principales características incluyen:

- **Interoperabilidad con Java:** Kotlin es 100% interoperable con Java, lo que significa que se puede usar código Java y Kotlin en el mismo proyecto sin problemas.
- **Concisión:** Kotlin reduce la cantidad de código repetitivo, haciendo el código más legible y mantenible.
- **Seguridad de nulls:** El sistema de tipos de Kotlin está diseñado para eliminar NullPointerExceptions, un problema común en Java.
- **Funciones de extensión:** Permite añadir nuevas funciones a una clase sin tener que heredar de ella o usar patrones de diseño como el "Decorator".
- **Corrutinas para programación asíncrona:** Ofrece una forma más sencilla y eficiente de manejar tareas asíncronas y concurrentes.

2.4 HERRAMIENTAS UTILIZADAS

2.4.1 JDK

Java™ Development Kit (JDK) es un software para los desarrolladores de Java. Incluye el intérprete Java, clases Java y herramientas de desarrollo Java (JDT): compilador, depurador, desensamblador, visor de applete, generador de archivos de apéndice y generador de documentación.

El JDK nos permite escribir aplicaciones que se desarrollan una sola vez y se ejecutan en cualquier lugar de cualquier máquina virtual Java. Las aplicaciones Java desarrolladas

con el JDK en un sistema se pueden usar en otro sistema sin tener que cambiar ni recompilar el código. Los archivos de clase Java son portables a cualquier máquina virtual Java estándar.

En nuestro caso estamos utilizando JDK 21.

2.4.2 ANDROID STUDIO

Android Studio es el Entorno de Desarrollo Integrado (IDE) oficial y más completo para el desarrollo de aplicaciones para el sistema operativo Android de Google. Está basado en IntelliJ IDEA de JetBrains y ha sido diseñado específicamente para optimizar la productividad de los desarrolladores Android.

Android Studio proporciona un conjunto integral de herramientas para todo el ciclo de vida del desarrollo de aplicaciones Android, incluyendo:

Diseño de interfaces de usuario: Cuenta con un editor de diseño visual que permite arrastrar y soltar componentes de la interfaz de usuario y previsualizar los diseños en diferentes configuraciones de pantalla.

Escritura de código: Ofrece un editor de código inteligente con sugerencias, refactorización específica de Android y comprobaciones de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad. Soporta lenguajes como Kotlin y Java, además de C++ para el desarrollo nativo con NDK.

Compilación y empaquetado: Utiliza el sistema de compilación basado en Gradle, lo que permite personalizar, configurar y extender el proceso de compilación, así como crear múltiples APKs (paquetes de aplicaciones Android) para una misma app con diferentes características.

Depuración y análisis de rendimiento: Incluye herramientas robustas para depurar el código, como la depuración en línea, y perfiles de rendimiento para monitorear el uso de CPU y memoria, identificar fugas de memoria y optimizar el rendimiento gráfico y de red.

Emulación y pruebas: Dispone de un emulador rápido que permite probar las aplicaciones en una amplia gama de dispositivos virtuales con diferentes configuraciones, así como la posibilidad de probar directamente en dispositivos Android físicos.

Integración con servicios de Google: Soporte integrado con Google Cloud Platform, facilitando la integración con servicios como Google Cloud Messaging y Firebase.

Sistema de compilación basado en Gradle:

Gradle es un sistema de automatización de compilación de código (build automation system) muy popular, especialmente en el ecosistema de Java, Kotlin y, sobre todo,

Android. Su función principal es tomar el código fuente de tu proyecto, junto con sus recursos y dependencias, y transformarlo en un producto final ejecutable, como un archivo .jar para aplicaciones Java o un archivo .apk para aplicaciones Android.

¿Qué hace Gradle?

- Automatización de la compilación (Build Automation):
- Gestiona el proceso de compilar el código fuente.
- Resuelve y descarga automáticamente las dependencias (librerías externas) que tu proyecto necesita.
- Ejecuta pruebas (unitarias, de integración).
- Genera documentación.
- Empaqueta la aplicación en el formato final (ej. .apk para Android).
- Puede desplegar la aplicación en servidores o tiendas de aplicaciones.
- Basado en DSL (Domain-Specific Language).
- Flexible y Extensible.
- Gestión de Dependencias (Dependency Management).
- Rendimiento y Escalabilidad

Gradle en el desarrollo de Android

Gradle es el sistema de compilación oficial para Android Studio y el desarrollo de aplicaciones Android. El Android Gradle Plugin (AGP) es un plugin específico que se ejecuta dentro de Gradle y proporciona todas las funcionalidades necesarias para construir aplicaciones Android, como:

- Compilar código Kotlin/Java a bytecode Dalvik/ART.
- Empaquetar recursos (imágenes, layouts, strings).
- Generar el archivo .apk o .aab (Android App Bundle).
- Gestionar diferentes "sabores" (flavors) y "tipos de compilación" (build types) para generar múltiples variantes de una misma aplicación (ej. una versión gratuita y una de pago, o una versión de depuración y una de lanzamiento).
- Manejar la firma de aplicaciones.

Archivos clave de Gradle en un proyecto Android

- settings.gradle.kts (o settings.gradle): Define los módulos del proyecto.
- build.gradle.kts (o build.gradle) a nivel de proyecto: Contiene la configuración global del proyecto y las versiones del Android Gradle Plugin.
- app/build.gradle.kts (o app/build.gradle): Es el más importante a nivel de módulo. Aquí se definen las dependencias de tu aplicación, las versiones del SDK (minSdk, targetSdk, compileSdk), los tipos de compilación, los sabores de producto, etc.

En resumen, Gradle es una herramienta poderosa y flexible que automatiza el ciclo de vida de desarrollo de software, siendo indispensable para proyectos de Android al proporcionar un control granular y eficiente sobre el proceso de compilación.

2.4.3 ANDROID SDK

Un kit de desarrollo de software (SDK) es un conjunto de herramientas de creación específicas de plataformas para desarrolladores. En este caso para la plataforma Android.

El SDK de Android incluye:

- Herramientas de desarrollo: Programas para compilar, depurar y optimizar el código de tus aplicaciones.
- APIs (Interfaces de Programación de Aplicaciones): Conjuntos de funciones y clases predefinidas que te permiten acceder a las características del dispositivo (cámara, GPS, contactos, etc.) y del sistema operativo Android.
- Emulador de Android: Una herramienta para simular dispositivos Android en tu ordenador, permitiéndote probar tus aplicaciones en diferentes configuraciones de pantalla, versiones de Android, etc., sin necesidad de un dispositivo físico.
- Documentación: Guías y referencias detalladas sobre cómo usar las APIs y las herramientas.
- Ejemplos de código y tutoriales: Recursos para aprender y empezar a desarrollar.

En nuestro caso estamos trabajando con la versión 35, y la versión mínima requerida es la 22.

2.4.4 FIREBASE

Firebase es una plataforma de desarrollo de aplicaciones móviles y web de Google que proporciona un conjunto de herramientas y servicios para ayudarte a construir, mejorar y escalar tus apps de forma rápida y sencilla.

En pocas palabras, Firebase actúa como un Backend-as-a-Service (BaaS), lo que significa que te ofrece funcionalidades de backend sin que tengas que gestionar tus propios servidores. Esto permite a los desarrolladores centrarse en la interfaz de usuario y la lógica del lado del cliente.

En nuestro caso usamos el servicio de base de datos en la nube de Firebase.

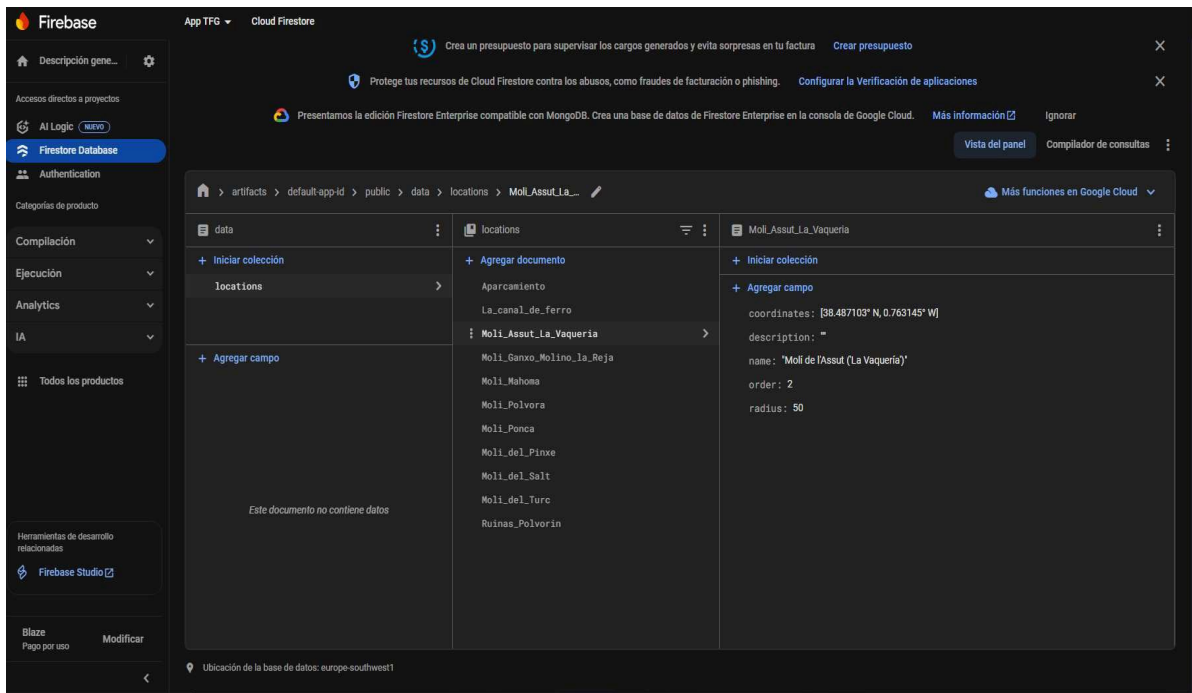


Ilustración 3: Consola de Firebase

2.4.5 GOOGLE MAPS

El SDK de Google Maps para Android es una herramienta esencial que permite integrar mapas interactivos y personalizables directamente en aplicaciones Android. Al aprovechar los datos globales y la robusta infraestructura de Google Maps, este SDK facilita la visualización de ubicaciones, la navegación y la interacción del usuario con el mapa.

Ofrece funcionalidades avanzadas como la adición de marcadores personalizados, el dibujo de líneas y polígonos para destacar rutas o áreas específicas, y el control sobre la vista de la cámara del mapa (zoom, inclinación, rotación). Esto nos permite a los desarrolladores crear experiencias geográficas ricas y adaptadas a las necesidades de cada proyecto, desde simples visualizaciones de puntos de interés hasta aplicaciones complejas con interacción en tiempo real. La implementación requiere una clave API de Google Maps Platform y está sujeta a los términos de uso y al modelo de precios de pago por uso de Google Cloud.

En nuestro caso, utilizamos la versión gratuita, la cual, aunque viene con todas las funcionalidades del SDK tiene grandes limitaciones en cuanto al uso que se puede hacer de ella.

2.4.6 JETPACK COMPOSE

Jetpack Compose es el moderno kit de herramientas declarativo de Android para construir interfaces de usuario (UI) nativas. Es parte de la suite Jetpack de Android y representa un cambio fundamental respecto al enfoque tradicional de vistas imperativas (XML).

En un resumen breve:

- **Declarativo vs. Imperativo:** Con Compose, describes cómo debe verse tu UI en cualquier momento dado (declarativo), en lugar de tener que manipular manualmente las vistas a medida que cambian los datos (imperativo, como en XML). Cuando tus datos cambian, Compose automáticamente "recompone" (redibuja) solo las partes necesarias de tu UI.
- **Basado en Kotlin:** Está completamente escrito en Kotlin y aprovecha sus características para hacer el código de UI más conciso y fácil de leer.
- **Componentes Reutilizables:** Permite construir la UI a partir de funciones "composables" pequeñas e independientes que puedes anidar y reutilizar fácilmente.
- **Acelera el desarrollo:** Reduce significativamente la cantidad de código de UI que necesitas escribir y depurar, lo que puede acelerar el desarrollo.
- **Vista Previa en Vivo:** Android Studio ofrece una funcionalidad de vista previa que te permite ver tus composables sin necesidad de desplegar la aplicación en un emulador o dispositivo.



3. RESULTADOS Y DISCUSIÓN

Este capítulo detalla la implementación de la aplicación, describiendo su estructura, los permisos y servicios clave que utiliza, el flujo de usuario principal y los elementos de su interfaz.

3.1 REQUISITOS DEL PROYECTO

La aplicación es una app Android escrita en **Kotlin** con **Jetpack Compose** cuyo propósito principal es ofrecer al usuario una experiencia interactiva para explorar rutas, acceder a información adicional y visualizar contenidos multimedia, todo ello respaldado por **Firebase**.

Para ello sus objetivos principales han sido:

- El permitir un acceso registrado mediante el inicio de sesión con correo electrónico y contraseña o cuenta de google, o bien un acceso como invitado.
- Proporcionar un menu principal desde el que poder navegar por las distintas secciones de la aplicación:
 - Listado de rutas disponibles.
 - Ventana de información sobre el observatorio del Vinalopó.
 - Galería de imágenes.
- En la pantalla de selección de rutas, cargar todas las rutas desde firebase, mostrando el nombre, la foto y una pequeña descripción de esta. Al mismo tiempo, si el usuario pulsa en la ruta, la aplicación te llevará a una nueva ventana que mostrará la información completa de la ruta.
 - En esta nueva ventana, se mostrará de la misma forma que antes, una lista cargada de firebase con todas las localizaciones, con sus imagenes, descripciones y localizaciones.
 - Habrá también un botón que nos permita abrir un mapa centrado en el primer punto de la ruta y que nos muestre la localización de todos los puntos así como la nuestra.
- En la pantalla que nos muestra la información del observatorio, dicha información se cargará desde firebase, y constará de una imagen, así como de una descripción de quienes són y a que se dedican.
- Por último, la pantalla de la galería, que sólo será accesible para los usuarios registrados, mostrará una galería de imágenes que constará no sólo de las imágenes proporcionadas por el observatorio, sino que también se podrán observar las imágenes subidas por los usuarios desde dichas localizaciones.

3.2 ESTRUCTURA DEL PROYECTO

La aplicación sigue la estructura típica de un proyecto Android, pero con las particularidades propias del uso de Compose, es decir, sin los layouts XML tradicionales.

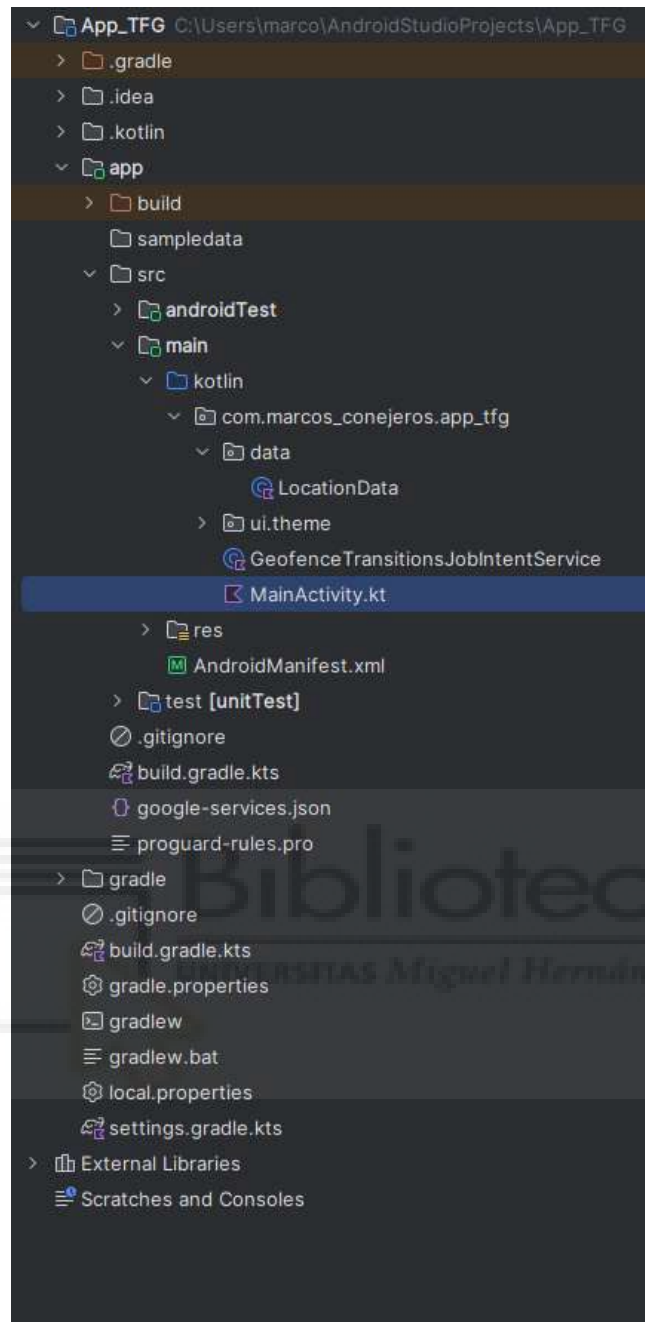


Ilustración 4: Estructura del proyecto

A continuación, se describen los directorios y fichero más importantes.

Directorio Kotlin/

- Es el núcleo de la aplicación, dentro del paquete `com.marcos_conejeros.app_tfg` se encuentran todos los ficheros que describen la lógica de la aplicación.
 - `LoginActivity`: Primera Activity que se lanza al ejecutar la aplicación. Gestiona el acceso el tipo de acceso a la app mediante el inicio de sesión.

- MainActivity: Pantalla principal que gestiona los permisos de localización y notificaciones, también la navegación a diferentes secciones de la aplicación.
- RouteSelectionActivity: Obtiene la lista de rutas de firebase y se las muestra al usuario, permite la creación de nuevas rutas y la navegación a otra activity para mostrar los datos de la ruta seleccionada.
- RouteDetailsActivity: Obtiene la información de la ruta y la muestra. Permite la visualización de la ruta desde un mapa con Google Maps Compose.
- ObservatoryActivity: Muestra la información cargada desde firebase.
- GalleryActivity: Galería de imágenes cargadas desde Firebase Storage.
- Además, contiene otros componentes:
 - Directorio util/: Contiene clases auxiliares necesarias para la lógica de la aplicación, sólo que al necesitar que sean accesibles desde distintas activities se han separado para mantener el código lo más limpio posible.
 - Directorio data/: Contiene modelos de datos que representan información guardada en firebase.

Directorio res/

- Incluye recursos gráficos y de configuración de la aplicación.

Archivo AndroidManifest.xml

- En este archivo se declaran:
 - Permisos de la aplicación.
 - Las activities principales del proyecto.
 - Las claves API necesarias para el acceso a determinados servicios.

Gradle Scripts

- Ficheros de configuración de Gradle que definen las dependencias y plugins utilizados en el proyecto.
 - Entre las más relevantes se encuentran Jetpack Compose, Firebase (Auth, Firestore, Storage), Google Maps Compose y Play Services Location.

3.3. PERMISOS NECESARIOS

La aplicación requiere la concesión de varios permisos por parte del usuario para operar plenamente. Estos permisos son fundamentales para acceder a las funcionalidades del dispositivo y ofrecer la experiencia deseada:

- **android.permission.ACCESS_FINE_LOCATION**: Este es el permiso de ubicación más preciso. Es fundamental para la operación de la aplicación, ya que permite obtener la ubicación exacta del usuario. Esto es esencial para centrar el

mapa en la posición actual del usuario y para el funcionamiento preciso del servicio de geofencing, que detecta la entrada y salida de zonas de interés.

- **android.permission.ACCESS_COARSE_LOCATION:** Aunque a menudo se concede implícitamente junto con ACCESS_FINE_LOCATION o se utiliza como una alternativa menos precisa, este permiso permite el acceso a la ubicación aproximada del usuario (basada en redes Wi-Fi y torres de telefonía móvil). Se incluye para una mayor robustez en la gestión de la ubicación.
- **android.permission.ACCESS_BACKGROUND_LOCATION:** Este permiso es crucial para dispositivos con Android 10 (API 29) o superior. Permite que la aplicación acceda a la ubicación del usuario incluso cuando no está en uso activo (es decir, cuando la aplicación está en segundo plano o el dispositivo está bloqueado). Es indispensable para que las geofences puedan detectar entradas y salidas de zonas de interés de forma continua, asegurando que las alertas se activen en el momento adecuado.

```
<!-- Permisos de Ubicación -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
```

Ilustración 5: Permisos de ubicación

- **android.permission.POST_NOTIFICATIONS:** Introducido en Android 13 (API 33), este permiso es necesario para que la aplicación pueda mostrar notificaciones al usuario en la barra de estado. En nuestro caso, es vital para alertar al usuario cuando entra o sale de una geofence, proporcionándole información relevante sobre la ubicación.
- **android.permission.BIND_JOB_SERVICE:** Este es un permiso de sistema que se declara en el AndroidManifest.xml para el GeofenceTransitionsJobIntentService. No es un permiso que el usuario conceda directamente, sino que es requerido por el sistema Android para permitir que el servicio reciba y procese trabajos en segundo plano de manera eficiente y segura, garantizando la correcta ejecución de las tareas de geofencing.

```
android:permission="android.permission.BIND_JOB_SERVICE"
```

Ilustración 6: Permiso de trabajo en segundo plano

3.4. SERVICIOS OFRECIDOS

La aplicación integra varios servicios de Google y Firebase para ofrecer una experiencia rica y dinámica. La funcionalidad de estos servicios depende directamente de los permisos solicitados al usuario:

- **Servicios de Ubicación (Google Play Services):**
 - **FusedLocationProviderClient:** Este servicio se utiliza para obtener la última ubicación conocida del dispositivo de manera eficiente y con un consumo optimizado de batería. Se apoya en el permiso `ACCESS_FINE_LOCATION` para proporcionar la mejor precisión posible al centrar el mapa en la posición actual del usuario.
 - **GeofencingClient:** Es el componente clave para la gestión de geofences. Permite definir regiones geográficas virtuales alrededor de los puntos de interés. Su correcto funcionamiento depende de `ACCESS_FINE_LOCATION` para la detección precisa de entradas/salidas y de `ACCESS_BACKGROUND_LOCATION` para operar de forma continua en segundo plano. Cuando se detecta una transición de geofence, este cliente activa el `GeofenceTransitionsJobIntentService`.
- **Firestore:**
 - **Firestore:** Actúa como la base de datos en la nube para almacenar y sincronizar las `LocationData`. Las ubicaciones se cargan en tiempo real mediante un `SnapshotListener`, lo que permite que la lista y el mapa de la aplicación se actualicen automáticamente si se modifican los datos en la base de datos. Aunque no requiere permisos de ubicación directos, es fundamental para obtener los datos (coordinates, radius, description) que luego se utilizan para configurar las geofences y el contenido de las notificaciones.
 - **Firestore Authentication:** Gestiona la autenticación de usuarios (ya sea de forma anónima o mediante un token personalizado). Esto es necesario para acceder de forma segura a los datos almacenados en Firestore, aunque no está directamente relacionado con los permisos de ubicación o notificación.
- **Google Maps SDK para Android (Compose):** Permite la integración de mapas interactivos directamente en la aplicación. Este SDK se utiliza para visualizar las ubicaciones de interés como marcadores personalizados y la posición actual del usuario. Se beneficia del permiso `ACCESS_FINE_LOCATION` para mostrar la ubicación del usuario en el mapa, mejorando la experiencia de navegación y orientación.
- **GeofenceTransitionsJobIntentService:** Este servicio en segundo plano es el encargado de procesar los eventos de geofence (entrada/salida de las zonas definidas). Una vez activado por el `GeofencingClient`, consulta Firestore para obtener los detalles de la ubicación (`fetchLocationDetailsAndNotify`) y, si cuenta

con el permiso POST_NOTIFICATIONS, envía una notificación al usuario con información relevante sobre el punto de interés. Su capacidad para operar en segundo plano está respaldada por ACCESS_BACKGROUND_LOCATION y su capacidad para notificar por POST_NOTIFICATIONS.

3.5 DIAGRAMA DE FLUJO

El flujo principal de la aplicación se puede describir de la siguiente manera:

1. **Inicio de la Aplicación:**
 - a. Al iniciar MainActivity, se inicializan los clientes de ubicación (FusedLocationProviderClient, GeofencingClient) y Firebase.
 - b. Se solicitan los permisos de ubicación (fina y en segundo plano) y notificaciones al usuario.
 - c. Se intenta autenticar al usuario con Firebase (token personalizado o anónimo).
2. **Carga y Visualización de Ubicaciones:**
 - a. Una vez autenticado el usuario, se establece un SnapshotListener en la colección de ubicaciones de Firestore.
 - b. Las ubicaciones se cargan y se almacenan en una lista (locations) en el estado de la aplicación.
 - c. La interfaz de usuario principal muestra una LazyColumn con LocationCards para cada ubicación.
3. **Interacción con la Lista de Ubicaciones:**
 - a. Cada LocationCard es clicable y gestiona su propio estado de expansión.
 - b. Al hacer clic, la tarjeta se expande/contrae con una animación suave (animateContentSize), mostrando u ocultando la descripción completa y los detalles de coordenadas/radio.
4. **Gestión de Geofences:**
 - a. Cada vez que la lista de locations se actualiza desde Firestore, se eliminan las geofences antiguas y se añaden nuevas geofences basadas en las coordenadas y radios de las ubicaciones cargadas.
 - b. Las transiciones de geofence (entrada/salida) son manejadas por GeofenceTransitionsJobIntentService, que puede activar notificaciones.
5. **Visualización del Mapa:**
 - a. Al hacer clic en el botón "Abrir Mapa con Ubicaciones" en la pantalla principal, la aplicación intenta obtener la última ubicación conocida del usuario.
 - b. Se cambia el estado showMapScreen a true, lo que activa la visualización del MapScreenContent.
 - c. El mapa se centra en la ubicación actual del usuario (si está disponible) o en la primera ubicación de la ruta, mostrando marcadores para todas las ubicaciones de interés y un marcador azul para la posición del usuario.

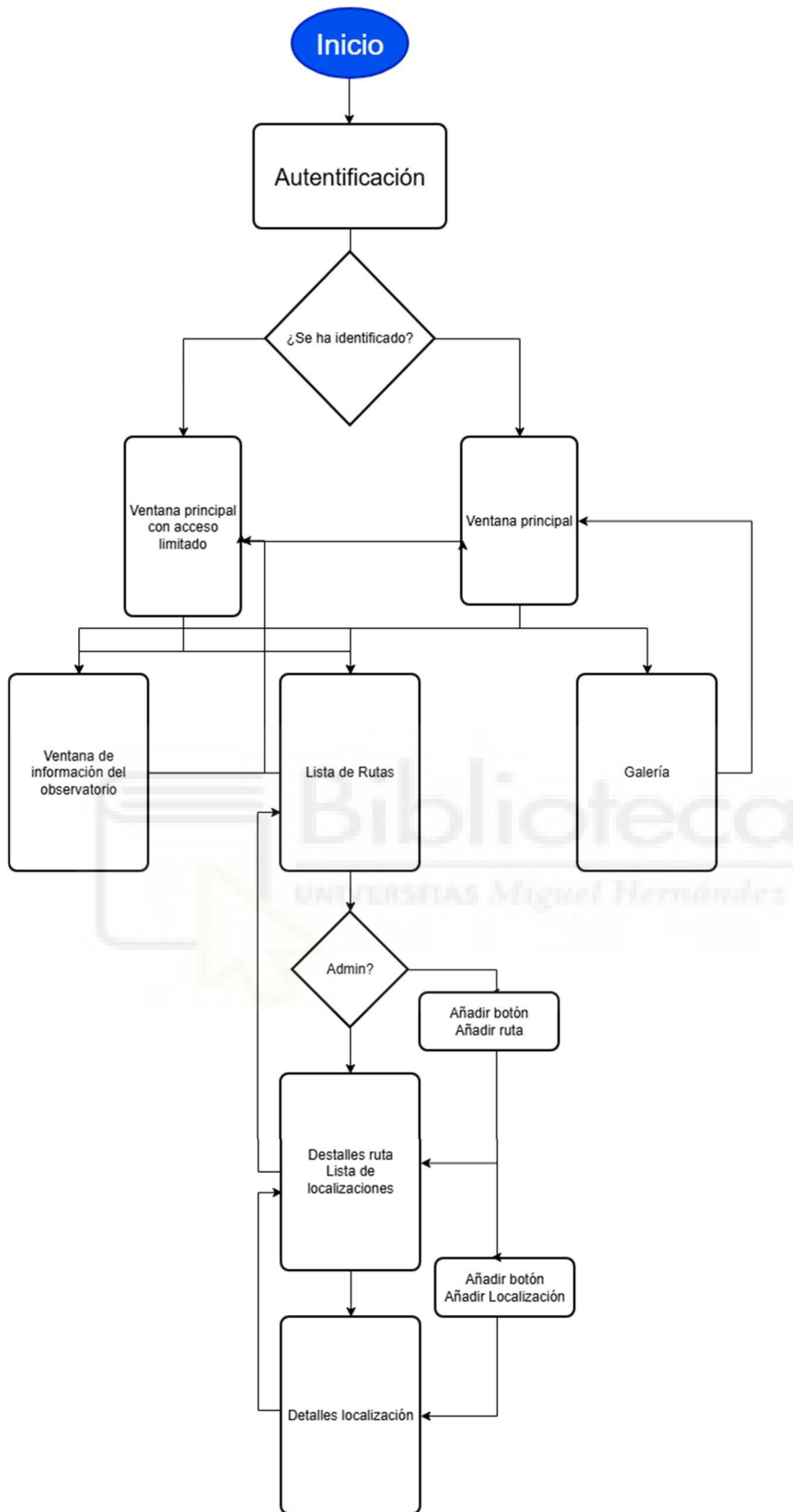


Ilustración 7: Diagrama de flujo del Frontend de la aplicación

3.6 INTERFAZ DE USUARIO

La interfaz de usuario está construida con Jetpack Compose y se compone de una serie de pantallas:

3.6.1. LOGIN

Es la pantalla de inicio de la aplicación, en ella el usuario tiene 3 opciones:

- Entrar con tu cuenta, ya sea mediante el clásico usuario (en este caso email) y contraseña, o mediante el botón de inicio con la cuenta de google.
- Entrar sin usar tu cuenta, con esta opción la aplicación te asigna el rol de invitado.
- Registrarse, la aplicación ejecuta una ventana flotante que le pide al usuario que introduzca un email y una contraseña para registrarse.

Toda la gestión de las cuentas se hace de manera autónoma desde firebase. Desde la aplicación, una vez se está iniciando sesión, cargamos el nivel de acceso que debe de tener el usuario, si entra como invitado, se le asocia un rol de invitado ('guest') y si ha entrado con un email asociado, se lee de firebase su nivel de acceso, usuario ('user') o administrador ('admin').



Ilustración 8: Pantalla Login

3.6.2. PANTALLA PRINCIPAL

En esta pantalla el usuario tiene la opción de elegir que parte de la aplicación quiere explorar.



Ilustración 9: Pantalla principal con acceso de usuario registrado

- Puede navegar hacia donde se encuentra la lógica principal de la aplicación, que sería pulsando el botón de "Ver Rutas".
- Puede ver la información sobre "El observatorio del Vinalopó" pulsando sobre el botón "Observatorio".
- Puede acceder a la galería de fotos de la base de Firebase Storage pulsando en el botón "Galería".

En esta pantalla ocurren varias cosas en segundo plano, que tienen repercusión en lo que el usuario ve en pantalla.

En primer lugar, es en la activity donde se ejecuta esta pantalla donde se lleva a cabo la lógica de los permisos de localización y notificaciones de la aplicación, por lo que, al acceder por primera vez o cuando no tengas los permisos aceptados te aparecerán una serie de mensajes para solicitarlos.

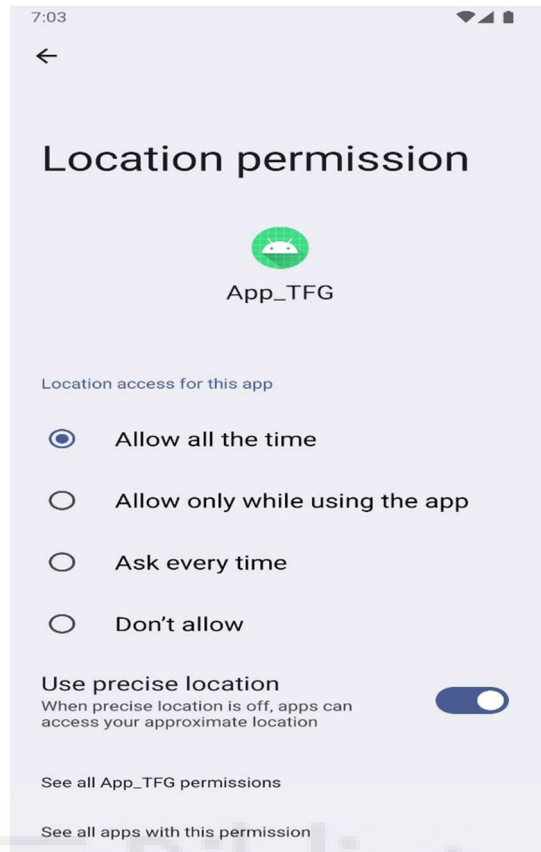


Ilustración 10: Pantalla de Android solicitando permisos de ubicación.

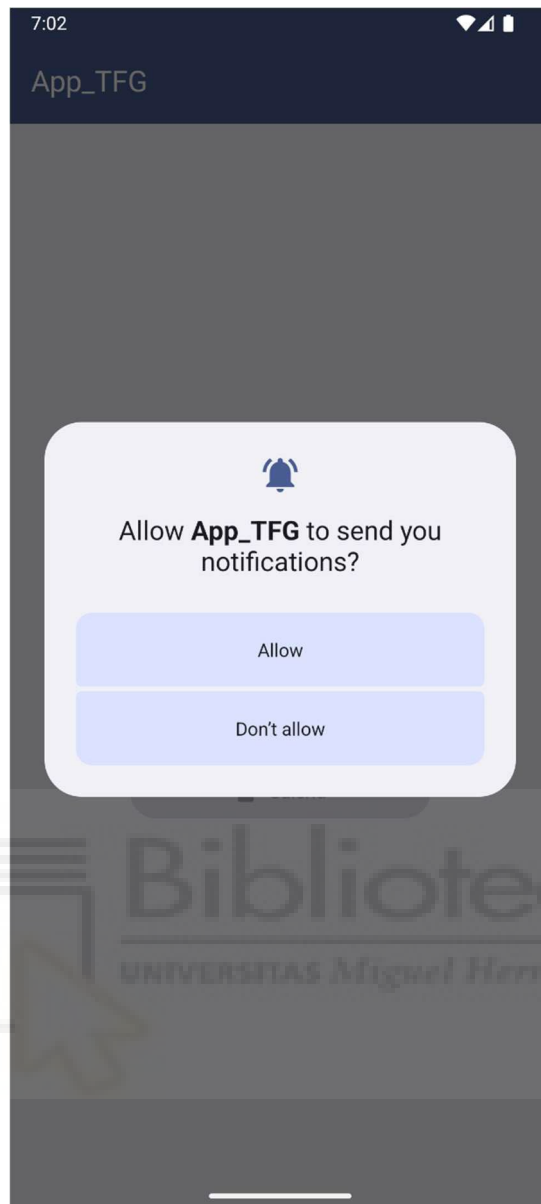


Ilustración 11: Pantalla de Android solicitando permiso de notificaciones.

También en esta pantalla se hace la única diferenciación entre los usuarios registrados y los que no, y es que el botón “Galería” solamente será disponible para los usuarios registrados, los no registrados no tendrán opción de ver la galería de fotos de la aplicación.



Ilustración 12: Acceso a la galería bloqueado para no identificados.

3.6.3. RUTAS

Aquí se le da al usuario la opción de elegir cuál de las rutas quiere seguir. Además, si el usuario está identificado como administrador podrá ver el botón de añadir que se muestra en la imagen, el cual le abrirá una ventana de diálogo para añadir una ruta nueva a la base de datos de la aplicación.

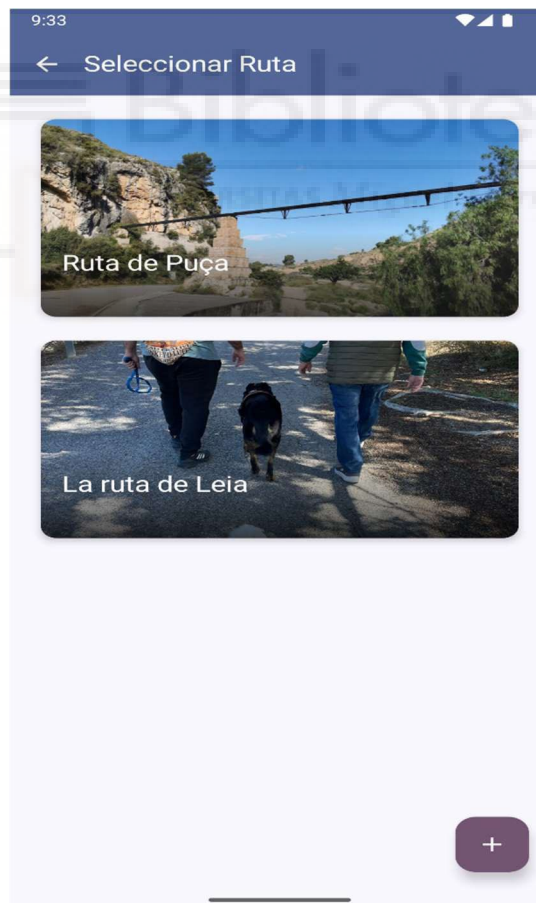


Ilustración 13: Pantalla para la selección de ruta



Ilustración 14: Dialogo para añadir una ruta nueva, sólo visible para acceso admin

3.6.4. LISTA DE LOCALIZACIONES

En esta pantalla el usuario puede ver las distintas localizaciones de la ruta seleccionada, así como una pequeña descripción de estas. También, igual que en la ventana anterior, si el usuario está identificado como administrador de la aplicación le aparecerá un botón que le da la opción de añadir una nueva localización a la ruta.

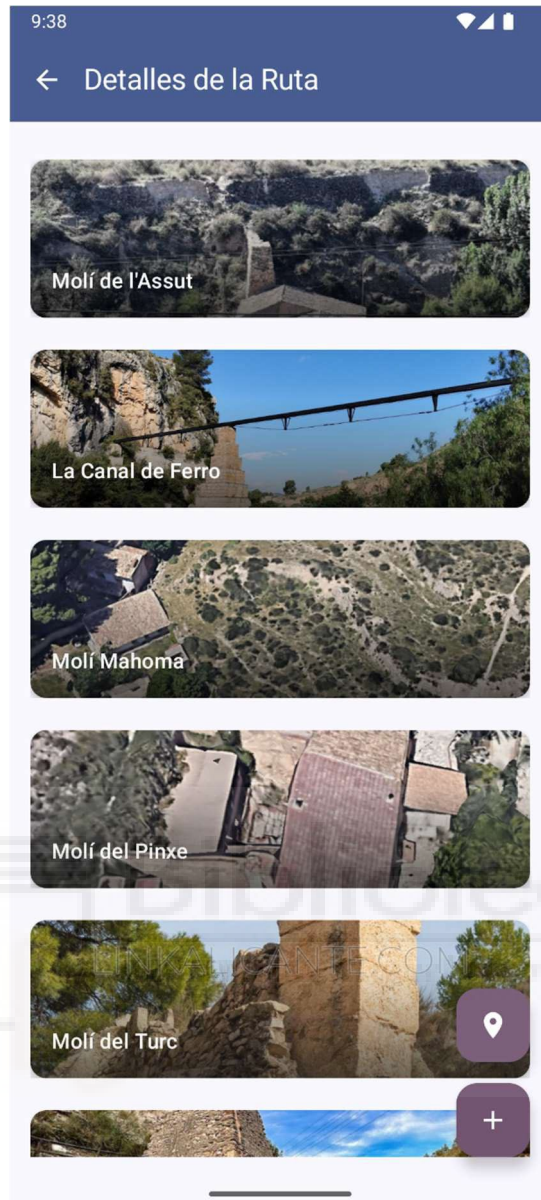


Ilustración 15: Pantalla con lista de ubicaciones

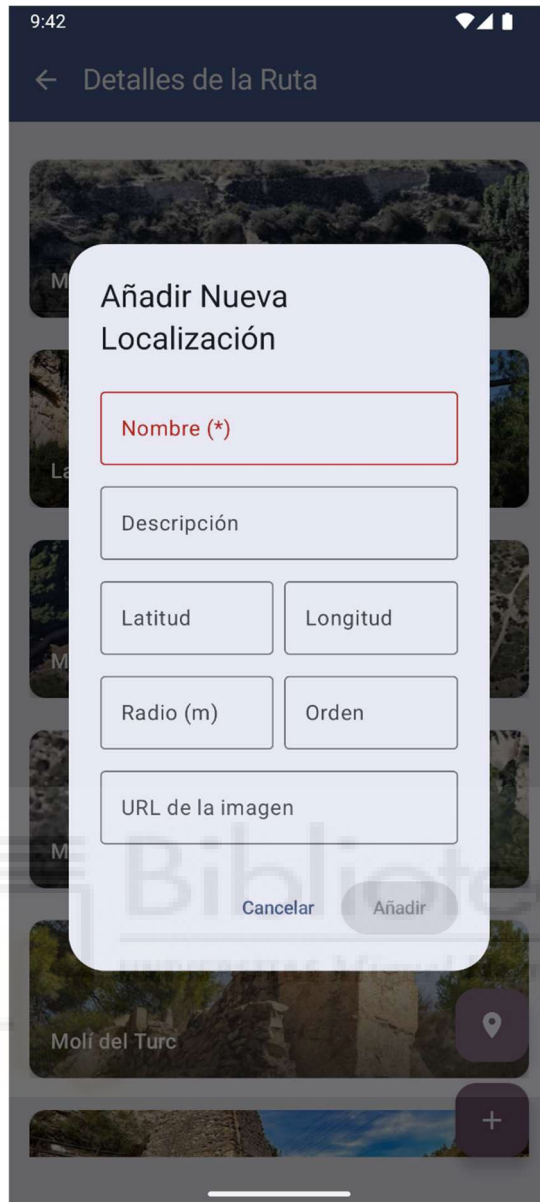


Ilustración 16: Diálogo para añadir nueva localización a la ruta, sólo visible para admin.

3.6.5. MAPA

En esta misma ventana de las localizaciones se puede ver que aparece otro botón flotante, este visible para todos los usuarios, el cual te permite abrir un mapa en el que salen resaltadas todas las localizaciones de la ruta.

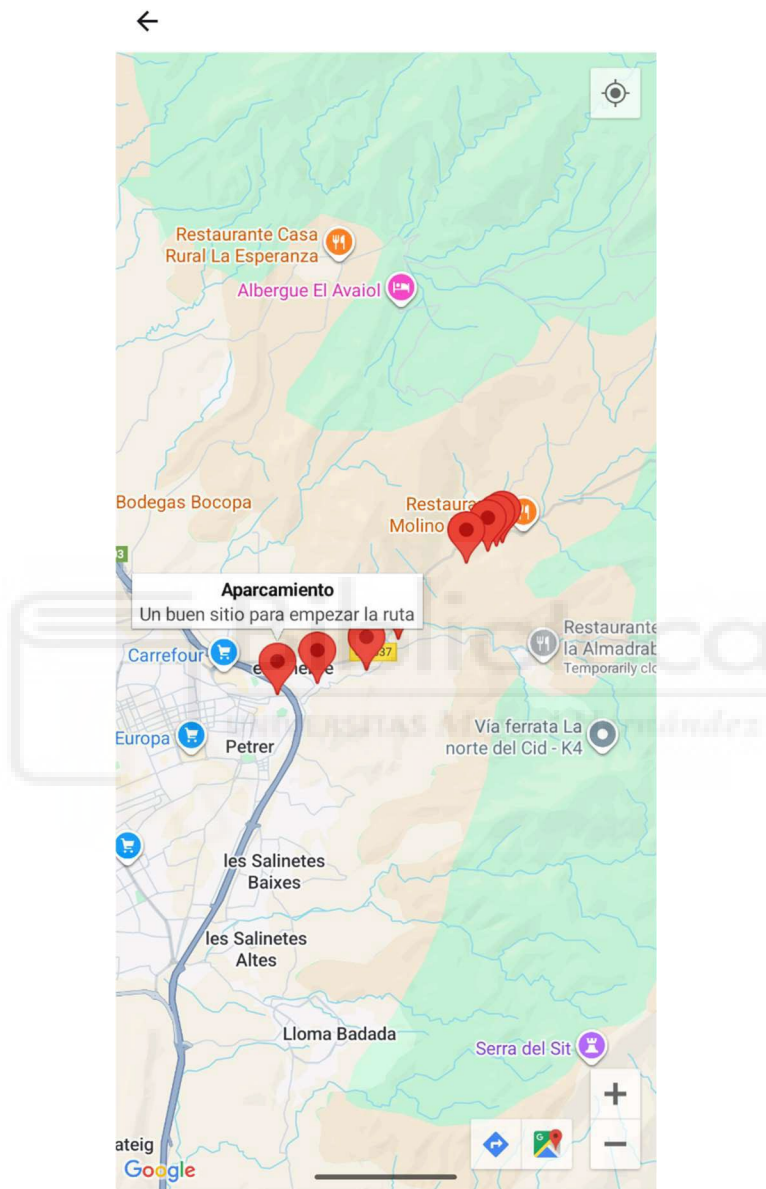


Ilustración 17: Ventana de Google Maps con las ubicaciones de la ruta marcadas.

3.6.6. DETALLES DE LOCALIZACIÓN

En esta pantalla podemos ver la descripción completa de la ubicación así como la imagen ampliada y sus coordenadas.



Ilustración 18: Detalles de la ubicación La Canal de Ferro

3.6.7. OBSERVATORIO

Esta pantalla está destinada a hablar un poco sobre la organización del observatorio del Vinalopó que son los que solicitaron este proyecto, y viene a cumplir la función de ¿Quiénes somos? de todas las páginas web y aplicaciones en las que la organización quiere contar un poco sobre ellos y su labor.



Ilustración 19: Pantalla de información sobre el observatorio.

3.6.8. GALERÍA

Esta pantalla está dedicada exclusivamente a la visualización de imágenes. Su objetivo es permitir que tanto el equipo de la organización como los usuarios registrados compartan fotografías de las rutas y sus entornos. Con ello, buscamos fomentar el sentido de comunidad e incentivar el compromiso colectivo con la conservación.

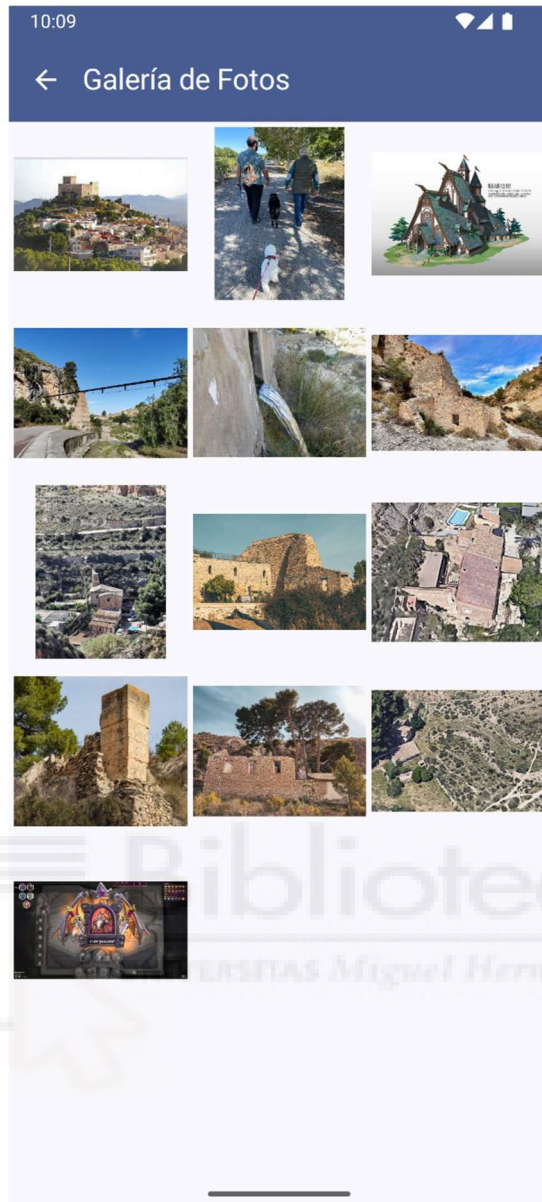


Ilustración 20: Pantalla de la galería

4. CONCLUSIONES Y LÍNEAS FUTURAS

La realización del presente proyecto ha permitido materializar una aplicación móvil nativa para Android que cumple con los objetivos planteados inicialmente: enriquecer la experiencia de los visitantes en espacios naturales y rutas de senderismo en la zona de Alicante, facilitando el acceso a información relevante sobre el patrimonio natural y cultural.

A lo largo de este desarrollo, se ha logrado:

- **Capacitación y Difusión de Información:** La aplicación proporciona una plataforma eficaz para que la asociación ecologista "Observatorio del Vinalopó" y/o el ayuntamiento local puedan difundir información detallada sobre rutas, puntos de interés y servicios, superando las limitaciones de los métodos tradicionales.
- **Acceso a Recursos Multimedia y Alertas:** Se ha implementado la capacidad de integrar diversos recursos multimedia (aunque actualmente se enfoca en texto y coordenadas, la estructura está preparada para audios, fotos y videos) y, crucialmente, la funcionalidad de geofencing permite la emisión de alertas informativas contextuales, mejorando la experiencia del usuario en tiempo real.
- **Recopilación de Datos de Afluencia:** Aunque en una fase inicial centrada en el registro de descargas, la infraestructura sentada sienta las bases para futuras implementaciones de mecanismos más sofisticados de recopilación de datos sobre la afluencia de visitantes, un objetivo clave para la gestión de los espacios naturales.

Desde el punto de vista técnico, el proyecto ha consolidado el conocimiento y la aplicación de tecnologías modernas en el ecosistema Android:

- **Dominio de Kotlin y Jetpack Compose:** La elección y el uso de Kotlin como lenguaje principal y Jetpack Compose para la interfaz de usuario han demostrado ser acertados, permitiendo un desarrollo más conciso, declarativo y eficiente, en línea con las tendencias actuales de Android.
- **Integración de Firebase:** La implementación de Firebase Firestore para la gestión de ubicaciones en tiempo real ha facilitado una sincronización dinámica de datos, lo que es esencial para mantener la información actualizada y accesible. La autenticación anónima o con token ha garantizado un acceso seguro a estos datos.
- **Uso Avanzado del SDK de Google Maps:** La integración del SDK de Google Maps ha permitido una visualización interactiva y personalizada de las rutas y puntos de interés, mejorando la orientación del usuario en el terreno.
- **Gestión de Permisos y Servicios de Ubicación:** Se ha abordado de manera robusta la solicitud y gestión de permisos de ubicación (fina y en segundo plano) y notificaciones, así como la correcta interacción con

FusedLocationProviderClient y GeofencingClient, componentes críticos para la funcionalidad central de la aplicación.

En síntesis, este proyecto no solo ha culminado en una aplicación funcional que fusiona las Tecnologías de la Información y la Comunicación con la interpretación del patrimonio natural, sino que también ha servido como una valiosa experiencia de aprendizaje y aplicación práctica de un conjunto de herramientas y paradigmas de desarrollo actuales. Aunque siempre existen líneas de mejora y expansión, el resultado obtenido representa una base sólida para futuras evoluciones y un paso significativo hacia la promoción de la conservación ambiental a través de la tecnología móvil.



5. BIBLIOGRAFÍA

- [1]. **Madoz, P.** *Diccionario geográfico-estadístico-histórico de España y sus posesiones de Ultramar.* 1845-55.
- [2]. **Tilden, Freeman.** *Interpreting our heritage.* s.l. : ReadHowYouWant.com, 2009.
- [3]. **Falk, J.H y Dierking, L.D.** *Learning from museums: visitors experiences and the making of meaning.* s.l. : Altamira Press, 2000.
- [4]. **Moscardo, G.** *Making visitors mindful: Visitor management and interpretation in heritage sites.* s.l. : Hospitality Press, 1999.
- [5]. *The use of mobile applications for cultural heritage.* **Economou, M.** 3, 2012, Journal of Information Technology & Tourism, Vol. 13, págs. 229-243.
- [6]. *Internet of Things (IoT) systems in future Cultural Heritage.* **Pouryousefzadeh, S. y Akbarzadeh , R.** 2019, 3.^a Conferencia Internacional sobre Internet de las Cosas y Aplicaciones (IdC) , Isfahán, Irán, 2019, págs. 1-5.
- [7]. *Hybrid Cemetery Culture. Making Death Matter in Cultural Heritage Using Smart Mobile Technologies.* **Borrits Sabras, Jacob, Andersen, Hans Jorgen y Rodil, Kasper.** 2015, International Conference on Culture and Computing (Culture Computing), Kyoto, Japan, 2015, págs. 167-174
- [8] *Integrating Smart Technologies in Heritage Interpretation of Historic Gardens for Enhancing Visitors Experiences.* **Sood , M. y Bhandari, H.** 2023, 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2023.
- [9]. *Medios síncronos con reconocimiento de ubicación y visualización aumentada para turistas del mundo real (SMART): Una aplicación para el sitio patrimonial de Khalifatabad, Bagerhat, Bangladesh.* **Rahaman, H., y otros.** 2016, 22.^a Conferencia Internacional sobre Sistemas Virtuales y Multimedia (VSMM) , Kuala Lumpur, Malasia, págs. 1-7.
- [10]. Wikiloc.com. (s.f.). *Wikiloc.* Recuperado de [<https://es.wikiloc.com/wikiloc/about-us.do>]. Visitado por última vez el 2 de diciembre de 2025
- [11] AllTrails.com. (s.f.). *AllTrails.* Recuperado de [<https://www.alltrails.com>]. Visitado por última ves el 2 de diciembre de 2025
- [12]. *Una plataforma de servicios de turismo y navegación para el patrimonio cultural digital móvil (M-DCH).* **Huang, C.M. y Guo, Y.A.** 2018, XV Simposio Internacional sobre Sistemas, Algoritmos y Redes Pervasivos (I-SPAN) , Yichang, China, págs. 185-192.
- [13]. *Creación de una interpretación personal en un sitio patrimonial virtual en 3D para mejorar la participación de los visitantes.* **Kim., Hayun, Bang, Sunyoung y Woo,**

Woontack. San Francisco, CA, EE. UU. : s.n., 2018, 3er Congreso Internacional de Patrimonio Digital (DigitalHERITAGE)celebrado conjuntamente con 2018 24a Conferencia Internacional sobre Sistemas Virtuales y Multimedia (VSMM 2018).

[14]. *Proyecto Pórticos de Bolonia: Un repositorio 3D para la nominación de la UNESCO al WHL.* **Apolonio, F.I., y otros.** 2013, Congreso Internacional de Patrimonio Digital 2013 (DigitalHeritage) , Marsella, Francia, págs. 563-570.

[15] *From Meaning-making to Meaning-sharing: Participatory Interpretation in a 3D Virtual Heritage Site.* **Hayun, K., Gun, L. y Billinghamurst, M. y Woo, W.** 2022. Simposio Internacional IEEE 2022 sobre Realidad Mixta y Aumentada Adjunta (ISMAR-Adjunct).

[16]. *Documentación del patrimonio cultural tangible e intangible mediante un enfoque transmedia: La aplicación móvil Discover Québec.* **Turgeon, L. y Massé, A.** 2013, Congreso Internacional de Patrimonio Digital (DigitalHeritage) , Marsella, Francia, págs. 455-455.

[17]. *Nuevas perspectivas para la conservación del patrimonio cultural, más allá de la ciencia avanzada y las tecnologías de la información.* **Moldovan, A.S., Radvan, R. y Angheluta, L.M. y Ene, D.V.** Dubai, Emiratos Árabes Unidos : s.n., 2011, Developments in E-systems Engineering, págs. 616-620.



